# ATME COLLEGE OF ENGINEERING

**13th KM Stone, Bannur Road, Mysore - 560 028**

## DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

## (DATA SCIENCE)

## (ACADEMIC YEAR 2023-24)

# LABORATORY MANUAL

**SUBJECT: COMPUTER NETWORK LABORATORY**

**SUB CODE: 21CS52**

**SEMESTER: V-2021 CBCS Scheme**

| Composed by | Verified by | Approved by |
|---|---|---|
| Mrs. ASHA L | Mrs. MADHU NAGARAJU | Dr. J V GORABAL |
| PROGRAMMER | FACULTY CO-ORDINATOR | HOD, CSE-DS |

# INSTITUTIONAL MISSION AND VISION

## Objectives

 To provide quality education and groom top-notch professionals, entrepreneurs and leaders for different fields of engineering, technology and management.

 To open a Training-R & D-Design-Consultancy cell in each department, gradually introduce doctoral and postdoctoral programs, encourage basic & applied research in areas of social relevance, and develop the institute as a center of excellence.

 To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels

 To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.

 To cultivate strong community relationships and involve the students and the staff in local community service.

 To constantly enhance the value of the educational inputs with the participation of students, faculty, parents and industry.

## Vision

 Development of academically excellent, culturally vibrant, socially responsible and globally competent human resources.

## Mission

- To keep pace with advancements in knowledge and make the students competitive and capable at the global level.
- To create an environment for the students to acquire the right physical, intellectual, emotional and moral foundations and shine as torch bearers of tomorrow’s society.
- To strive to attain ever-higher benchmarks of educational excellence.

## DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND ENGINEERING

## (DATA SCIENCE &ENGINEERING)

## Vision of The Department

- To impart technical education in the field of data science of excellent quality with a high level of professional competence, social responsibility, and global awareness among the students

## Mission

- To impart technical education that is up to date, relevant and makes students competitive and employable at global level
- To provide technical education with a high sense of discipline, social relevance in an intellectually, ethically and socially challenging environment for better tomorrow
- Educate to the global standards with a benchmark of excellence and to kindle the spirit of innovation.

## Program Outcomes(PO)

- **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

- **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

- **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

- **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

- **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

- **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

- **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

- **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

- **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

- **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

- **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## Program Specific Outcomes (PSOs)

- PSO1: Develop relevant programming skills to become a successful data scientist
- PSO2: Apply data science concepts and algorithms to solve real world problems of the society
- PSO3: Apply data science techniques in the various domains like agriculture, education healthcare for better society

## Program Educational Objectives (PEOs):

- **PEO1**: Develop cutting-edge skills in data science and its related technologies, such as machine learning, predictive analytic, and data engineering.
- **PEO2**: Design and develop data-driven solutions to real-world problems in a business, research, or social environment.
- **PEO3**: Apply data engineering and data visualization techniques to discover, investigate, and interpret data.
- **PEO4:** Demonstrate ethical and responsible data practices in problem solving
- **PEO5**: Integrate fields within computer science, optimization, and statistics to develop better solutions

## CONTENT LIST

# Introduction

## Introduction to NS-2

NS-2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks. Network Simulator-2 (NS2) is a popular open source network simulator for carrying out network experimentation. Way back when it was being designed, its primary usage was to analyze the performance of congestion control algorithms implemented in Transmission control protocol (TCP).

Even today, it remains the most widely

used network simulator for TCP research. Over the period of time, it gained wide acceptance in industry, and now supports simulation of latest wired as well as wireless networking protocols (e.g., routing algorithms, TCP, User Data Protocol (UDP) ) and paradigms such as Mobile Ad hoc Networks (MANETs) Vehicular Ad hoc Network (VANETs), etc.

Another simulator called ns-3 has gained a lot of popularity in the recent past. It is not a sequel of NS-2. NS-3 APIs are not compatible with those of NS-2 API. Both are completely different tools.

**Features of NS-2:**

- It is a discrete event simulator for networkingresearch.
- It provides substantial support to simulate protocols like TCP, FTP, UDP &DSR.
- It simulates wired and wireless network.
- It is primarily UNIXbased.
- Uses TCL as its scriptinglanguage.
- Otcl: Object oriented support Tcl
- TclCL: Tcl with Classes and OTcllinkage
- Discrete event scheduler

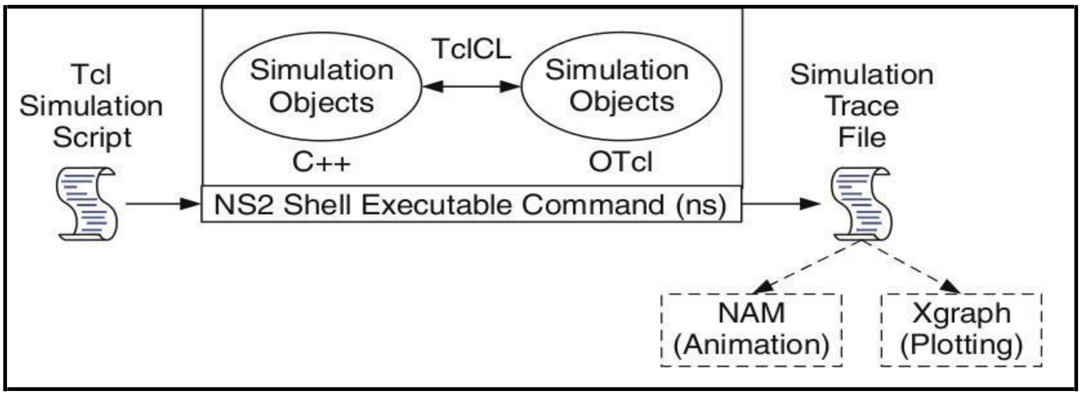


**Fig. 1 Basic Architecture of Network Simulator**

## Why two languages? (TCL and C++)

- NS2 consists of two key languages: C++ and Object-oriented Tool Comman Language(OTcl).
- The C++ defines the internal mechanism (i.e.,a backend) of the simulationobjects.
- The OTcl sets up simulation by configuring the objects as well as scheduling discrete events (i.e., afrontend).
- The C++ and the OTcl are linked together usingTclCL.
- NS2 uses OTcl to create and configure a network, and uses C++ to runsimulation.
- C++ is fast to run but slow tochange.
- OTcl, on the other hand, is slow to run but fast tochange.
- We write a Tcl simulation script and feeditas an input argument to NS2 when running simulation (e.g., executing “nsmyfirst_ns.tcl”).
- Here, “ns” is a C++ executable file obtained from thecompilation.
- myfirst_ns.tcl is an input configuration file specifying system parameters and configuration such as nodes, link, and how they areconnected.
- C++ is used for the creation of objects because of speed andefficiency.
- OTcl is used as a front-end to setup the simulator, configure objects and schedule event because of its ease ofuse.

## Tcl scripting

- Tcl is a general purpose scripting language.[Interpreter]
- Tcl runs on most of the platforms such as Unix, Windows, andMac.
- The strength of Tcl is itssimplicity.
- It is not necessary to declare a data type for variable prior to theusage

## Structure of NS-2 Program:

- Creating a SimulatorObject
- Setting up files for trace &NAM
- Tracing files using theircommands
- Closing trace file and startingNAM
- Creating LINK & NODE topology &Orientation oflinks

## Working ofNS-2

- NS2 provides users with executable command ns which takes an input argument, the name of a Tcl simulation scriptingfile.
- Users are feeding the name of a Tcl simulation script (which sets up a simulation) as an input argument of an NS2 executable command ns.
- In most cases, a simulation trace file is created, and is used to plot graph and/or to createanimation.

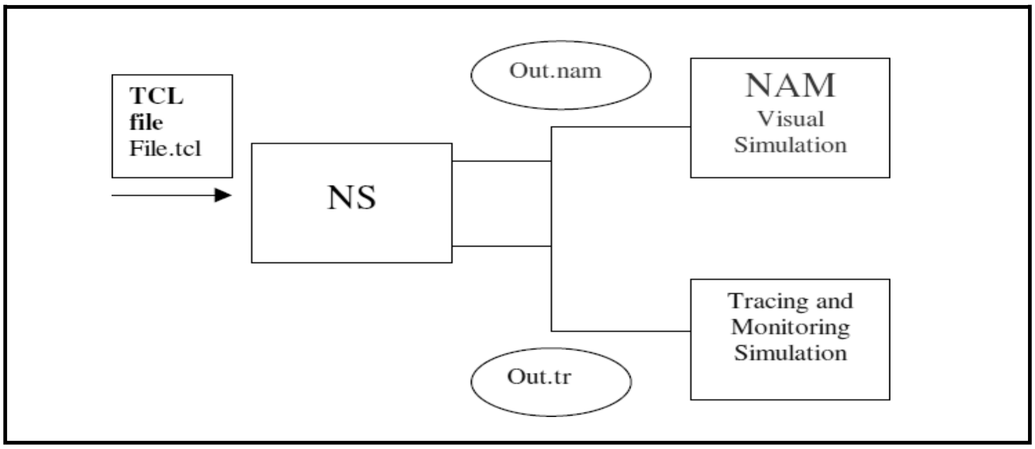


**Fig 2. Working of Network Simulator 2**

### Trace file and NamTrace file:

- Once the simulation is complete, we can see two files: “trace.tr”, and“nam.out”.
- The trace file (trace.tr) is a standard format used byns2.
- In ns2, each time a packet moves from one node to another, or onto a link, or into a buffer, etc., it gets recorded in this trace file.
- Each row represents one of these events and each column has its ownmeaning.
- Start nam with the command 'nam <nam-file>' where '<nam-file>' is the name of a nam trace file that was generated byns.

## Network Animator Window

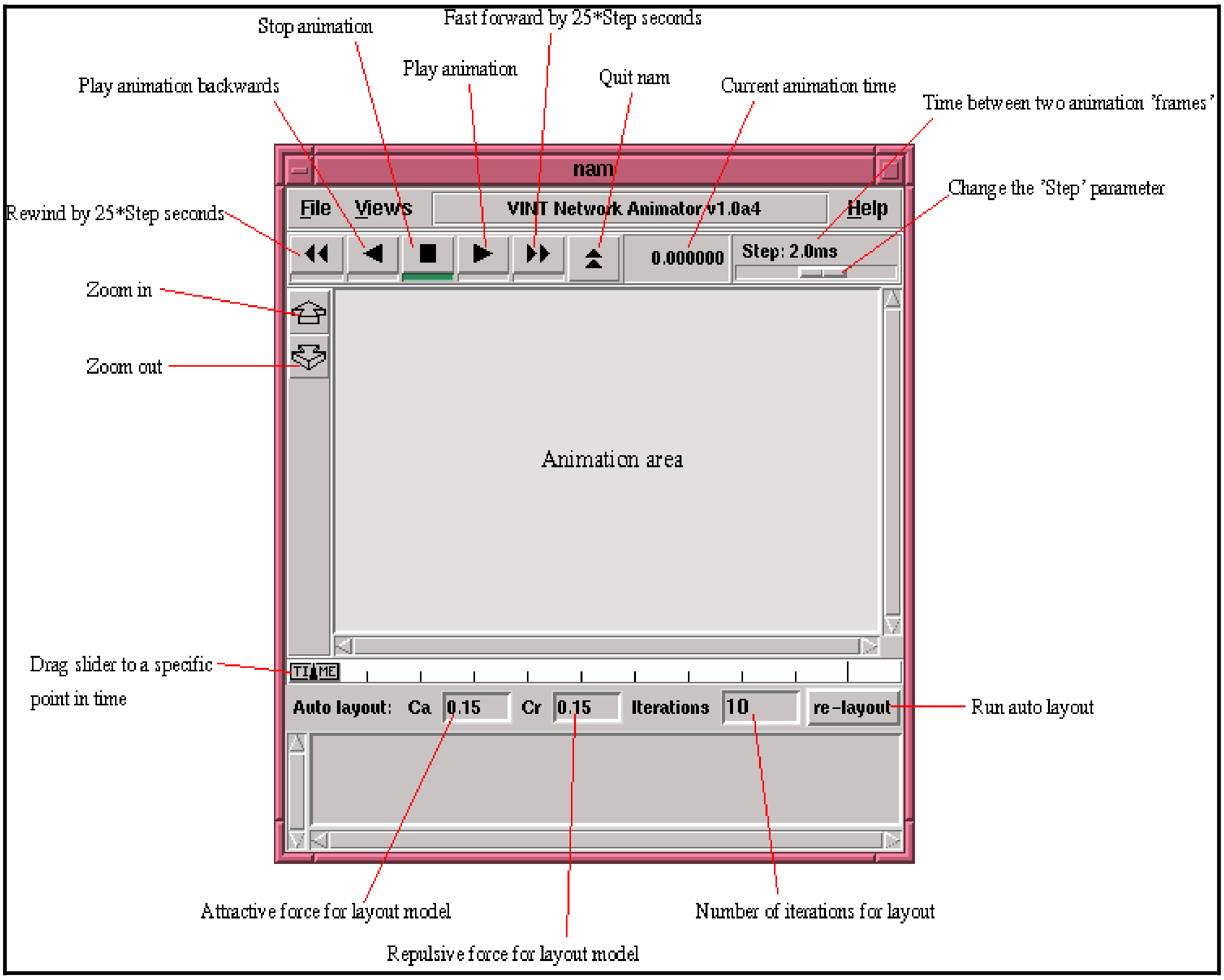

Stop animation
Fast forward by 25*Step seconds
Play animation
Quit nam
Current animation time
Time between two animation 'frames'
Play animation backwards
Change the 'Step' parameter
Rewind by 25*Step seconds
Zoom in
Zoom out
nam
File
Views
VINT Network Animator v1.0a4
Help
0.000000
Step: 2.0ms
Animation area
Drag slider to a specific point in time
TIME
Auto layout:
Ca
0.15
Cr
0.15
Iterations
10
re-layout
Run auto layout
Attractive force for layout model
Number of iterations for layout
Repulsive force for layout model

**Fig. 3 Details of NAM Window**

### Advantages and Disadvantages of NS2:

### Advantages

1. OpenSource
2. Complex scenarios can be easilytested.
3. Results can be quickly obtained – more ideas can be tested in a smaller timeframe.
4. Supportedprotocols
5. Supportedplatforms
6. Modularity

### Disadvantages

1.Limitation in designing large scalesystems
2.May be slow compared to real time network and computationallyexpensive
3.Does not reflect reality in large and complexnetworks.
4. Statistical uncertainty inresults

**Steps to create a scenario file:**

**Step1**: Declare Simulator and setting output file

**Step2**: Setting Node and Link

**Step3**: Setting Agent

**Step4**: Setting Application

**Step5**: Setting Simulation time and schedules

**Step6:** Declare finish.

### Step 1: Declare Simulator and setting

| | |
|---|---|
| $ns [new Simulator] | #first line of tcl script. Creates ns object |
| get file [open out.tr w]<br>$ns trace-all $file | #open the trace file |
| get namfile [open out.nam w]<br>$ns namtrace-all $namfile | #open the nam file |

### Step 2: Setting Node and Link

| | |
|---|---|
| $ n0 [$ns node] | setting a node |
| ns duplex-link $n0 $n2 3Mb 5ms DropTail | #bidirectional link between n0 and n2 is declared bandwidth 3Mbps and delay 5ms. DropTail is a waiting queue type. |
| $ns duplex-link-op $n0 $n2 orient right-down | #Sets positions of node and link for Nam. It does not affect to the result of simulation |
| $ns queue-limit $n2 $n3 20 | #The length of queue on the link from n2 to n3 is 20[packets]. |
| $ns duplex-link-op $n2 $n3 queuePos 0.5 | #The position of queue is set for Nam, 0.5 is the angle between link and queue, it equals to (0.5_). |

### Step 3: Setting Agent

**UDP Agent :**To use UDP in simulation, the sender sets the Agent as UDP Agent while the receiver sets to Null Agent. Null Agents do nothing except receiving the packets.

| | |
|---|---|
| set udp [newAgent/UDP]<br>$ns attach-agent $n0 $udp<br>set null [newAgent/Null]<br>$ns attach-agent $n3 $null | #udp and null Agent are set for n0 and n3, respectively |
| $ns connect $udp $null | Declares the transmission between udp and null. |
| $udp set fid_ 0 | Sets the number for data flow of udp. This number will be recorded to all packets which are sent from udp |
| $ns color 0 blue | Mark the color to discrete packet for showing result on NAM. |

**TCP Agent:** To use TCP in simulation, the sender sets the Agent as TCP Agent while the receiver sets to TCPSink Agent. When receiving a packet, TCPSink Agent will reply an acknowledgment packet (ACK). Setting Agent for TCP is similar to UDP.

| | |
|---|---|
| set tcp [new Agent/TCP]<br>$ns attach-agent $n1 $tcp<br>set sink [new Agent/TCPSink]<br>$ns attach-agent $n3 $sink | # tcp and sink Agent are set for n1 and n3, respectively |
| $ns connect $tcp $sink | #declares the transmission between tcp and sink. |
| $tcp set fid_ 1 | #sets the number for data flow of tcp. This number will be recorded to all packet which are sent from tcp |
| $ns color 1 red | #mark the color to discrete packet for showing result on Nam. |

**Step 4: Setting Application**

In general, UDP Agent uses CBR Application while TCP Agent uses FTP Application.

| |
|---|
| set cbr [new Application/Traffic/CBR]<br>$cbr attach-agent $udp |
| set ftp [new Application/FTP]<br>$ftp attach-agent $tcp |

**Step 5: Setting time schedule for simulation**

Time schedule of a simulation is set as below:

| | |
|---|---|
| $ns at 1.0 "$cbrstart"<br>$ns at 3.5 "$cbrstop" | cbr transmits data from 1.0[sec] to 3.5[sec] |
| $ns at 1.5 "$ftpstart"<br>$ns at 3.0 "$ftpstop" | ftp transmits data from 1.5[sec] to 3.0[sec]. |

**Step 6: Declare finish**

After finish setting, declaration of finish is written at the end of file.

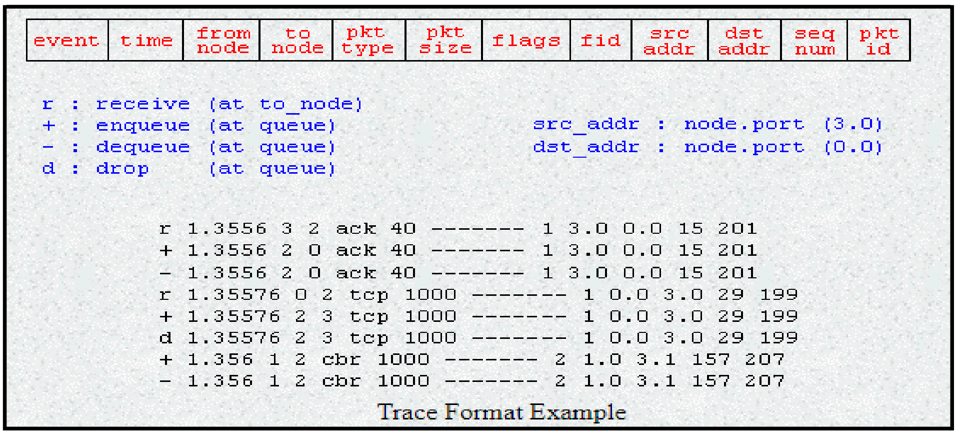
| | |
|---|---|
| $ns at 4.0 "finish" | The finish function is used to output data file at the end of simulation. |
| proc finish {} {<br>global ns file namfile tcpfile<br>$ns flush-trace<br>close $file<br>close $namfile<br>close $tcpfile<br>exit 0<br>} | |

## Execute Simulation and start Nam

By executing below command line, simulation will be started and shows the animation of simulation

```
ns sample.tcl
nam out.nam
```

## View trace file (out.tr)



| event | time | from node | to node | pkt type | pkt size | flags | fid | src addr | dst addr | seq num | pkt id |
|---|---|---|---|---|---|---|---|---|---|---|---|

```
r : receive (at to_node)
+ : enqueue (at queue)              src_addr : node.port (3.0)
- : dequeue (at queue)              dst_addr : node.port (0.0)
d : drop    (at queue)

        r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
        + 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
        - 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
        r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
        + 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
        d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
        + 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
        - 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

Trace Format Example

**Figure 4 . Details of Trace Window**

1. The first field is the event type. It is given by one of four possible symbols r, +, -, d which correspond respectively to receive (at the output of the link), enqueued, dequeued anddropped.
2. The second field gives the time at which the eventoccurs.
3. Gives the input node of the link at which the eventoccurs.
4. Gives the output node of the link at which the eventoccurs.
5. Gives the packet type (eg CBR orTCP)
6. Gives the packet size
7. Someflags
8. This is the flow id (fid) of IPv6 that a user can set for each flow at the input OTcl script one can further use this field for analysis purposes; it is also used when specifying stream color for the NAMdisplay.
9. This is the source address given in the form of"node.port".
10. This is the destination address, given in the same form.
11. This is the network layer protocol's packet sequence number. Even though UDP implementations in a real network do not use sequence number, ns keeps track of UDP packet sequence number for analysispurposes
12. The last field shows the Unique id of thepacket.

## AWK file:

- The basic function of awk is to search files for lines (or other units of text) that contain certain patterns. When a line matches one of the patterns, awk performs specified actions on that line. awk keeps processing input lines in this way until the end of the input files are reached.
- Programs in awk are different from programs in most other languages, because awk programs are **data-driven;** that is, we describe the data to work with, and then what to do when we find it. Most other

languages are **procedural.** When working with procedural languages, it is usually much harder to clearly describe the data of our program will process.

- For this reason, awk programs are often refreshingly easy to both write and read. When we run awk, we can specify an awk **program** that tells awk what to do. The program consists of a series of **rules**. (It may also contain **function definitions**, an advanced feature which we will ignore for now.
- Each rule specifies one pattern to search for, and one action to perform when that pattern is found). Syntactically, a rule consists of a pattern followed by an action. The action is enclosed in curly braces to separate it from the pattern. Rules are usually separated by newlines. Therefore, an awk program looks like this:

***pattern {action }*** ***pattern { action}***

- Since we are dealing with column oriented data, AWK is probably the easiest tool we can use to format our data. AWK is a simple scripting language that scans through a file line byline.Itallowstoaccessanycolumninthecurrentlinebyusingspecialvariables$1,$2, $3, etc. for the first, second and third columns. The definition of each column of the trace file is shown above, so we can use the AWK script to check the value of each column and collect the data weneed.
- The BEGIN and END sections are only executed once (before and after the file has been processed). The middle section is executed for each line of the file. The AWK script keeps three variables to store the throughput in Mb/s for flow 1, flow 2, and the total. For each line of the trace file, it checks to see if a TCP packet ($5 == “tcp”) is received ($1 == “r”) at node 3 ($4 == “3”), which is our destination node. If so, it increments the count for the appropriate flow, using the size of the particular packet (in bytes) from column 6. After each second, it prints the total while converting bytes to Mb.

**BEGIN { print "START" } { print }END { print "STOP" }**

**To run awk script**

**awk -f <filename.awk><input_file><output_file>**

**XGRAPH:**

- Plottingpurposes.
- Comes together with NS2 installationpackage.
- RunningXgraph

**Xgraph <inputfile1>...<inputfilen> -bg <color> -t <graph_title> -x <xtitle> -y <ytitle>**

## NS Simulation Script

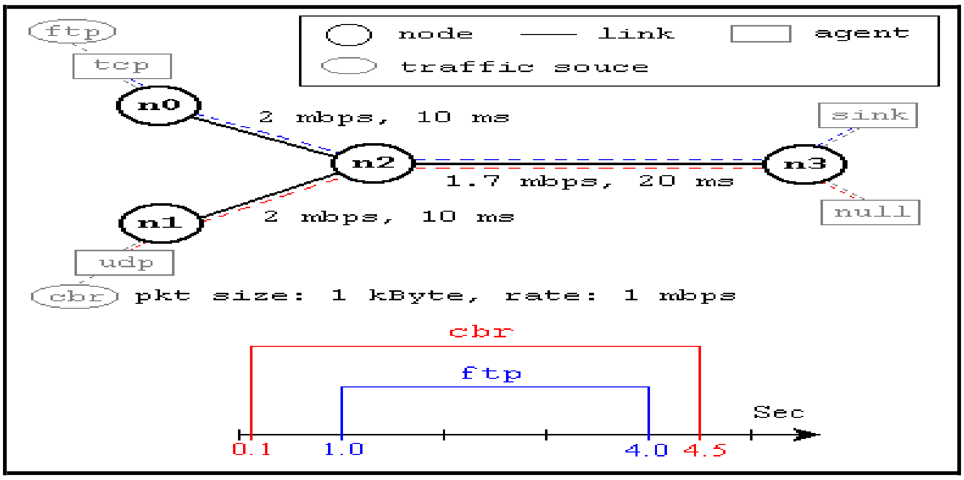


**Fig 5. A Simple Network Topology and Simulation Scenario**

This network consists of 4 nodes (n0, n1, n2, n3) as shown in Fig. 5. 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 1KByte 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.

### An Example Simulation 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 [newAgent/UDP]

$ns attach-agent $n1 $udp set
null [newAgent/Null]

$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2
#Setup a CBR over UDPconnection
set cbr [newApplication/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"
$ns at 1.0 "$ftpstart"
```

```
$ns at 4.0 "$ftpstop"
$ns at 4.5 "$cbrstop"

$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
```

# Basics of Computer Network

**What is a computer network?**

A computer network comprises two or more computers that are connected—either by cables (wired) or WiFi (wireless)—with the purpose of transmitting, exchanging, or sharing data and resources. You build a computer network using hardware (e.g., routers, switches, access points, and cables) and software (e.g., operating systems or business applications).

Geographic location often defines a computer network. For example, a LAN (local area network) connects computers in a defined physical space, like an office building, whereas a WAN (wide area network) can connect computers across continents. The internet is the largest example of WAN, connecting billions of computers worldwide.

You can further define a computer network by the protocols it uses to communicate, the physical arrangement of its components, how it controls traffic, and its purpose.
Computer networks enable communication for every business, entertainment, and research purpose. The internet, online search, email, audio and video sharing, online commerce, live-streaming, and social networks all exist because of computer networks.

**Computer network types**

As networking needs evolved, so did the computer network types that serve those needs. Here are the most common and widely used computer network types:

- **LAN (local area network):** A LAN connects computers over a relatively short distance, allowing them to share data, files, and resources. For example, a LAN may connect all the computers in an office building, school, or hospital. Typically, LANs are privately owned and managed.

- **WLAN (wireless local area network):** A WLAN is just like a LAN but connections between devices on the network are made wirelessly.

- **WAN (wide area network):** As the name implies, a WAN connects computers over a wide area, such as from region to region or even continent to continent. The internet is the largest WAN, connecting billions of computers worldwide. You will typically see collective or distributed ownership models for WAN management.

- **MAN (metropolitan area network):** MANs are typically larger than LANs but smaller than WANs. Cities and government entities typically own and manage MANs.

- **PAN (personal area network):** A PAN serves one person. For example, if you have an iPhone and a Mac, it's very likely you've set up a PAN that shares and syncs content—text messages, emails, photos, and more—across both devices.

- **SAN (storage area network):** A SAN is a specialized network that provides access to block-level storage—shared network or cloud storage that, to the user, looks and works like a storage drive that's physically attached to a computer.

• **CAN (campus area network):** A CAN is also known as a corporate area network. A CAN is larger than a LAN but smaller than a WAN. CANs serve sites such as colleges, universities, and business campuses.

• **VPN (virtual private network):** A VPN is a secure, point-to-point connection between two network end points (see 'Nodes' below). A VPN establishes an encrypted channel that keeps a user's identity and access credentials, as well as any data transferred, inaccessible to hackers.

**Important terms and concepts**

The following are some common terms to know when discussing computer networking:

- **IP address**: An IP address is a unique number assigned to every device connected to a network that uses the Internet Protocol for communication. Each IP address identifies the device's host network and the location of the device on the host network. When one device sends data to another, the data includes a 'header' that includes the IP address of the sending device and the IP address of the destination device.

- **Nodes:** A node is a connection point inside a network that can receive, send, create, or store data. Each node requires you to provide some form of identification to receive access, like an IP address. A few examples of nodes include computers, printers, modems, bridges, and switches. A node is essentially any network device that can recognize, process, and transmit information to any other network node.

- **Routers:**Routers help you to connect with multiple networks. It enables you to share a single internet connection with multiple devices and saves money. This networking component acts as a dispatcher, which allows you to analyze data sent across a network. It automatically selects the best route for data to travel and send it on its way.

•**Switches:** A switch is a device that connects other devices and manages node-to-node communication within a network, ensuring data packets reach their ultimate destination. While a router sends information between networks, a switch sends information between nodes in a single network. When discussing computer networks, 'switching' refers to how data is transferred between devices in a network. The three main types of switching are as follows:

1. **Circuit switching,** which establishes a dedicated communication path between nodes in a network. This dedicated path assures the full bandwidth is available during the transmission, meaning no other traffic can travel along that path**.**
2. **Packet switching** involves breaking down data into independent components called packets which, because of their small size, make fewer demands on the network. The packets travel through the network to their end destination.
3. **Message switching** sends a message in its entirety from the source node, traveling from switch to switch until it reaches its destination node.

- **Ports:** A port identifies a specific connection between network devices. Each port is identified by a number. If you think of an IP address as comparable to the address of a hotel, then ports are the suites or room numbers within that hotel. Computers use port numbers to determine which application, service, or process should receive specific messages.

- **Network cable types:** The most common network cable types are Ethernet twisted pair, coaxial, and fiber optic. The choice of cable type depends on the size of the network, the arrangement of network elements, and the physical distance between devices**.**

**Servers:**Servers are computers that hold shared programs, files, and the network operating system. Servers allow access to network resources to all the users of the network.

**Clients:**Clients are computer devices which access and uses the network as well as shares network resources. They are also users of the network, as they can send and receive requests from the server.

**Transmission Media:**Transmission media is a carrier used to interconnect computers in a network, such as coaxial cable, twisted-pair wire, and optical fiber cable. It is also known as links, channels, or lines.

**Access points:** Access points allow devices to connect to the wireless network without cables. A wireless network allows you to bring new devices and provides flexible support to mobile users.

**Shared Data:**Shared data are data which is shared between the clients such as data files, printer access programs, and email.

**Network Interface Card:**Network Interface card sends, receives data, and controls data flow between the computer and the network.

**Local Operating System:**A local OS which helps personal computers to access files, print to a local printer and uses one or more disk and CD drives which are located on the computer.

**Network Operating System:**The network operating system is a program which runs on computers and servers. It allows the computers to communicate via network.

**Protocol:**A protocol is the set of defined rules that allows two entities to communicate across the network. Some standard protocols used for this purpose are IP, TCP, UDP, FTP, etc.

**Hub:**Hub is a device that splits network connection into multiple computers. It acts a distribution center so whenever a computer requests any information from a computer or from the network it sends the request to the hub through a cable. The hub will receive the request and transmit it to the entire network.

**LAN Cable:**Local Area Network(LAN) cable is also called as Ethernet or data cable. It is used for connecting a device to the internet.

**OSI:**OSI stands for Open Systems Interconnection. It is a reference model which allows you to specify standards for communications.

**Unique Identifiers of Network**
**Below given are some unique network identifiers:**

**Hostname:**Every device of the network is associated with a unique device, which is called hostname.

**IP Address:**IP (Internet Protocol) address is as a unique identifier for each device on the Internet. Length of the IP address is 32-bits. IPv6 address is 128 bits.

**DNS Server:**DNS stands for Domain Name System. It is a server which translates URL or web addresses into their corresponding IP addresses.

**MAC Address:**MAC (Media Access Control Address) is known as a physical address is a unique identifier of each host and is associated with the NIC (Network Interface Card). General length of MAC address is : 12-digit/ 6 bytes/ 48 bits

**Port:**Port is a logical channel which allows network users to send or receive data to an application. Every host can have

### Computer networks and the internet

The internet is actually a network of networks that connects billions of digital devices worldwide. Standard protocols allow communication between these devices. Those protocols include hypertext transfer protocol (the 'http' in front of all website addresses). Internet protocol (or IP addresses) are the unique identifying numbers required of every device that accesses the internet. IP addresses are comparable to your mailing address, providing unique location information so that information can be delivered correctly.

Internet Service Providers (ISPs) and Network Service Providers (NSPs) provide the infrastructure that allows the transmission of packets of data or information over the internet. Every bit of information sent over the internet doesn't go to every device connected to the internet. It's the combination of protocols and infrastructure that tells information exactly where to go.

### How do they work?

Computer networks connect nodes like computers, routers, and switches using cables, fiber optics, or wireless signals. These connections allow devices in a network to communicate and share information and resources.

Networks follow protocols, which define how communications are sent and received. These protocols allow devices to communicate. Each device on a network uses an Internet Protocol or IP address, a string of numbers that uniquely identifies a device and allows other devices to recognize it.

Routers are virtual or physical devices that facilitate communications between different networks. Routers analyze information to determine the best way for data to reach its ultimate destination. Switches connect devices and manage node-to-node communication inside a network, ensuring that bundles of information traveling across the network reach their ultimate destination.

**Architecture:Computer network architecture** defines the physical and logical framework of a computer network. It outlines how computers are organized in the network and what tasks are assigned to those computers. Network architecture components include hardware, software, transmission media (wired or wireless), network topology, and communications protocols.

### Main types of network architecture

There are two types of network architecture: **peer-to-peer (P2P)** and **client/server**.

**In P2P architecture**, two or more computers are connected as "peers," meaning they have equal power and privileges on the network. A P2P network does not require a central server for coordination. Instead, each computer on the network acts as both a client (a computer that needs to access a service) and a server (a computer that serves the needs of the client accessing a service). Each peer makes some of its resources available to the network, sharing storage, memory, bandwidth, and processing power.

**In a client/server network**, a central server or group of servers manage resources and deliver services to client devices in the network. The clients in the network communicate with other clients through the server. Unlike the P2P model, clients in a client/server architecture don’t share their resources. This architecture type is sometimes called a tiered model because it's designed with multiple levels or tiers.

### Network topology

Network topology refers to how the nodes and links in a network are arranged. A network node is a device that can send, receive, store, or forward data. A network link connects nodes and may be either cabled or wireless links.

Understanding topology types provides the basis for building a successful network. There are a number of topologies but the most common are bus, ring, star, and mesh:

A **bus network topology** is when every network node is directly connected to a main cable.

In a **ring topology**, nodes are connected in a loop, so each device has exactly two neighbors. Adjacent pairs are connected directly; non-adjacent pairs are connected indirectly through multiple nodes.

In a **star network topology**, all nodes are connected to a single, central hub and each node is indirectly connected through that hub.

A **mesh topology** is defined by overlapping connections between nodes. You can create a full mesh topology, where every node in the network is connected to every other node. You can also create partial mesh topology in which only some nodes are connected to each other and some are connected to the nodes with which they exchange the most data. Full mesh topology can be expensive and time-consuming to execute, which is why it's often reserved for networks that require high redundancy. Partial mesh provides less redundancy but is more cost effective and simpler to execute.

### Security

Computer network security protects the integrity of information contained by a network and controls who access that information. Network security policies balance the need to provide service to users with the need to control access to information.

There are many entry points to a network. These entry points include the hardware and software that comprise the network itself as well as the devices used to access the network, like computers, smartphones, and tablets. Because of these entry points, network security requires using several defense methods. Defenses may include firewalls—devices that monitor network traffic and prevent access to parts of the network based on security rules.

Processes for authenticating users with user IDs and passwords provide another layer of security. Security includes isolating network data so that proprietary or personal information is harder to access than less critical information. Other network security measures include ensuring hardware and software updates and patches are performed regularly, educating network users about their role in security processes, and staying aware of external threats executed by hackers and other malicious actors. Network threats constantly evolve, which makes network security a never-ending process.

The use of public cloud also requires updates to security procedures to ensure continued safety and access. A secure cloud demands a secure underlying network.

### Advantages of a Computer Network

- Helps you to connect with multiple computers together to send and receive information when accessing the network.
- Helps you to share printers, scanners, and email.
- Helps you to share information at very fast speed

- Electronic communication is more efficient and less expensive than without the network.

### Disadvantages of using Computer Networks

- Investment for hardware and software can be costly for initial set-up
- If you don’t take proper security precautions like file encryption, firewalls then your data will be at risk.
- Some components of the network design may not last for many years, and it will become useless or malfunction and need to be replaced.
- Requires time for constant administration
- Frequent server failure and issues of regular cable faults
-

**Summary:**

- A computer network is a group of two or more interconnected computer systems
- Computer networks help you to connect with multiple computers together to send and receive information
- Switches work as a controller which connects computers, printers, and other hardware devices
- Routers help you to connect with multiple networks. It enables you to share a single internet connection and saves money
- Servers are computers that hold shared programs, files, and the network operating system
- Clients are computer device which accesses and uses the network and shares network resources
- Hub is a device that split a network connection into multiple computers.
- Access points allow devices to connect to the wireless network without cables
- Network Interface card sends, receives data and controls data flow between the computer and the network
- A protocol is the set of defined rules which that allows two entities to communicate across the network
- Hostname, IP Address, DNS Server, and host are important unique indetenfiters of computer networks.
- ARP stands for Address Resolution Protocol
- RAR Reverse Address Resolution Protocol gives an IP address of the device with given a physical address as input.
- Computer network helps you to share expensive software’s and database among network participants
- The biggest drawback of installing computer network is that its initial investment for hardware and software can be costly for initial set-up

### IP Address in Networking-

**In Networking,**

- IP Address is short for **Internet Protocol Address**.
- It is a unique address assigned to each computing device in an IP network.ISP assigns IP Address to all the devices present on its network.
- Computing devices use IP Address to identify and communicate with other devices in the IP network.

### Types Of IP Address

IP Addresses may be of the following two types-

- Static IP Address
- Dynamic IP Address

### Static IP Address

- Static IP Address is an IP Address that once assigned to a network element always remains the same.
- They are configured manually.
- Some ISPs do not provide static IP addresses.
- Static IP Addresses are more costly than dynamic IP Addresses.

### Dynamic IP Address

- Dynamic IP Address is a temporarily assigned IP Address to a network element.
- It can be assigned to a different device if it is not in use.
- DHCP or PPPoE assigns dynamic IP addresses.

An IP address is categorized into two different types based on the number of IP address it contains. These are:

- IPv4 (Internet Protocol version 4)
- IPv6 (Internet Protocol version 6)

### IP Address Format:

- IP Address is a 32 bit binary address written as 4 numbers separated by dots.
- The 4 numbers are called as octets where each octet has 8 bits.
- The octets are divided into 2 components- Net ID and Host ID.

### IPv4

IPv4 is version 4 of IP. It is a current version and the most commonly used IP address. It is a 32-bit address written in four numbers separated by a dot (.), i.e., periods. This address is unique for each device. For example, 66.94.29.13

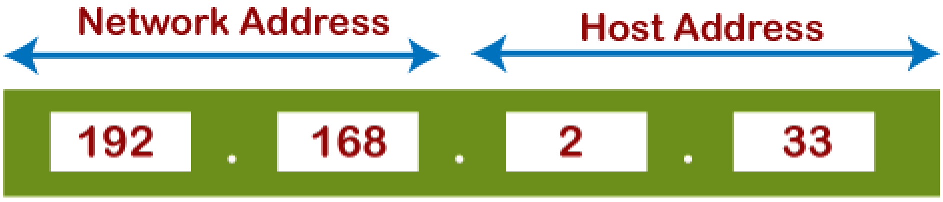
### IPv6

IPv4 produces 4 billion addresses, and the developers think that these addresses are enough, but they were wrong. IPv6 is the next generation of IP addresses. The main difference between IPv4 and IPv6 is the address size of IP addresses. The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit hexadecimal address. IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

### IP Address Format

Originally IP addresses were divided into five different categories called **classes**. These divided IP classes are class A, class B, class C, class D, and class E. Out of these, classes A, B, and C are most important. Each address class defines a different number of bits for its **network prefix (network**

**address)** and **host number (host address)**. The starting address bits decide from which class an address belongs.



**Network Address:** The network address specifies the unique number which is assigned to your network. In the above figure, the network address takes two bytes of IP address.

**Host Address:** A host address is a specific address number assigned to each host machine. With the help of the host address, each machine is identified in your network. The network address will be the same for each host in a network, but they must vary in host address.

### Address Format IPv4

The address format of IPv4 is represented into **4-octets** (32-bit), which is divided into three different classes, namely class A, class B, and class C.

4 octets
192
.
168
.
2
.
33

The above diagram shows the address format of IPv4. An IPv4 is a 32-bit decimal address. It contains four octets or fields separated by 'dot,' and each field is 8-bit in size. The number that each field contains should be in the range of 0-255.

### Class A

**Class A** address uses only first higher order octet (byte) to identify the network prefix, and remaining three octets (bytes) are used to define the individual host addresses. The class A address ranges between 0.0.0.0 to 127.255.255.255. The first bit of the first octet is always set to 0 (zero), and next 7 bits determine network address, and the remaining 24 bits determine host address. So the first octet ranges from 0 to 127 (00000000 to 01111111).

### Class B

**Class B** addresses use the initial two octets (two bytes) to identify the network prefix, and the remaining two octets (two bytes) define host addresses. The class B addresses are range between 128.0.0.0 to 191.255.255.255. The first two bits of the first higher octet is always set to 10 (one and zero bit), and next 14 bits determines the network address and remaining 16 bits determines the host address. So the first octet ranges from 128 to 191 (10000000 to 10111111).

### Class C

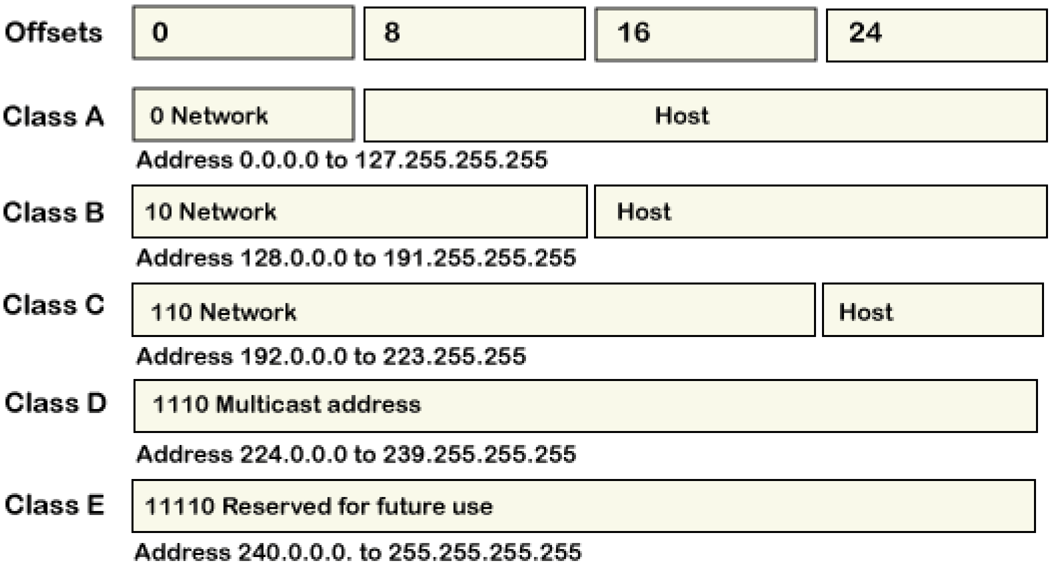
**Class C** addresses use the first three octets (three bytes) to identify the network prefix, and the remaining last octet (one byte) defines the host address. The class C address ranges between 192.0.0.0 to 223.255.255.255. The first three bit of the first octet is always set to 110, and next 21 bits specify network address and remaining 8 bits specify the host address. Its first octet ranges from 192 to 223 (11000000 to 11011111).

### Class D

**Class D** IP address is reserved for multicast addresses. Its first four bits of the first octet are always set to 1110, and the remaining bits determine the host address in any IP address. The first higher octet bits are always set to 1110, and the remaining bits specify the host address. The class D address ranges between 224.0.0.0 to 239.255.255.255. In multicasting, data is not assigned to any particular host machine, so it is not require to find the host address from the IP address, and also, there is no subnet mask present in class D.

### Class E

**Class E** IP address is reserved for experimental purposes and future use. It does not contain any subnet mask in it. The first higher octet bits are always set to 1111, and next remaining bits specify the host address. Class E address ranges between 240.0.0.0 to 255.255.255.255.


Offsets
0
8
16
24
Class A
0 Network
Host
Address 0.0.0.0 to 127.255.255.255
Class B
10 Network
Host
Address 128.0.0.0 to 191.255.255.255
Class C
110 Network
Host
Address 192.0.0.0 to 223.255.255
Class D
1110 Multicast address
Address 224.0.0.0 to 239.255.255.255
Class E
11110 Reserved for future use
Address 240.0.0.0. to 255.255.255.255

In every IP address class, all host-number bits are specified by a power of 2 that indicates the total numbers of the host's address that can create for a particular network address. Class A address can contain the maximum number of 224 (16,777,216) host numbers. Class B addresses contain the maximum number of 216 (65, 536) host numbers. And class C contains a maximum number of 28 (256) host numbers.

### Subnet address of IP address, understand with an example:

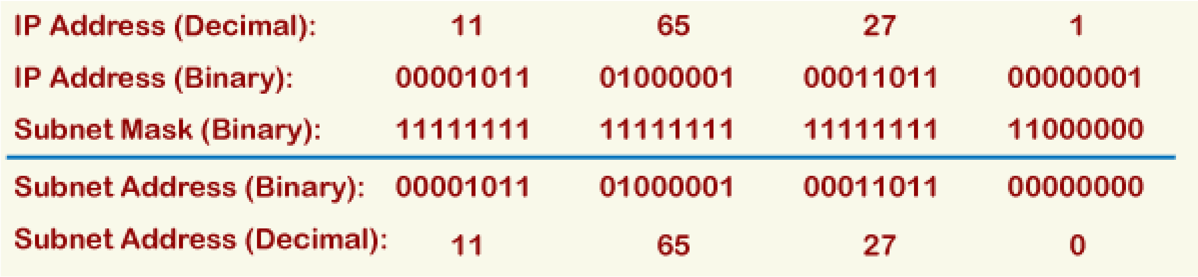
Suppose a class A address is 11.65.27.1, where 11 is a network prefix (address), and 65.27.1 specifies a particular host address on the network. Consider that a network admin wants to use 23 to 6 bits to identify the subnet and the remaining 5 to 0 bits to identify the host address. It can be represented in the *Subnet mask* with all 1 bits from 31 to 6 and the remaining (5 to 0) with 0 bits.

Subnet Mask (binary): 11111111 11111111 11111111 11000000

IP address (binary): 00001011 01000001 00011011 00000001

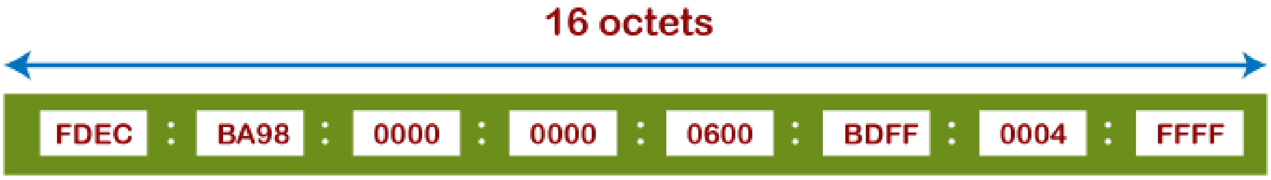
Now, the subnet can be calculated by applying AND operation (1+1=1, 1+0=0, 0+1=0, 0+0=0) between complete IP address and Subnet mask. The result is:

00001011 01000001 00011011 00000000 = 11.65.27.0 subnet address



| | | | | |
|---|---|---|---|---|
| IP Address (Decimal): | 11 | 65 | 27 | 1 |
| IP Address (Binary): | 00001011 | 01000001 | 00011011 | 00000001 |
| Subnet Mask (Binary): | 11111111 | 11111111 | 11111111 | 11000000 |
| Subnet Address (Binary): | 00001011 | 01000001 | 00011011 | 00000000 |
| Subnet Address (Decimal): | 11 | 65 | 27 | 0 |

**IP Address Format IPv6**All IPv6 addresses are 128-bit hexadecimal addresses, written in 8 separate sections having each of them have 16 bits. As the IPv6 addresses are represented in a hexadecimal format, their sections range from 0 to FFFF. Each section is separated by colons (:). It also allows to removes the starting zeros (0) of each 16-bit section. If two or more consecutive sections 16-bit contains all zeros (0 : 0), they can be compressed using double colons (::).



IPv6 addresses are consist of 8 different sections, each section has a 16-bit hexadecimal values separated by colon (:). IPv6 addresses are represented as following format:

xxxx : xxxx : xxxx : xxxx : xxxx : xxxx : xxxx : xxxx

Each "xxxx" group contains a 16-bit hexadecimal value, and each "x" is a 4-bit hexadecimal value. For example:

FDEC : BA98 : 0000 : 0000 : 0600 : BDFF : 0004 : FFFF

You can also remove the starting zeros (0) of each 16-bit section. For example, the above IPv6 can be rewritten by omitting starting zeros (0) as follow:

FDEC : BA98 : 0 : 0 : 600 : BDFF : 4 : FFFF

You can also compress the consecutive sections 16-bit zeros (0 : 0) using double colons (::). But keep in mind that you can do it only once per IP address.

```
FDEC : BA98 : : 600 : BDFF : 4 : FFFF
```

**IP Address Table**

On the basis of ranges, IP addresses are categorized into five address classes which are given below.

| Class | Higher bits | Network address bits | Host address bits | No. of networks | No.of hosts per network | Range |
|---|---|---|---|---|---|---|
| A | 0 | 8 | 24 | $2^{7}$ | $2^{24}$ | 0.0.0.0 to 125.255.255.255 |
| B | 10 | 16 | 16 | $2^{14}$ | $2^{16}$ | 128.0.0.0 to 191.255.255.255 |
| C | 110 | 24 | 8 | $2^{21}$ | $2^{8}$ | 192.0.0.0 to 223.255.255.255 |
| D | 1110 | Not defined and reserved for future | Not defined and reserved for future | Not defined and reserved for future | Not defined and reserved for future | 224.0.0.0 to 239.255.255.255 |
| E | 1110 | Not defined and reserved for future | Not defined and reserved for future | Not defined and reserved for future | Not defined and reserved for future | 240.0.0.0 to 255.255 |

Java is a general-purpose computer programming language that is simple, concurrent, class-based, object-oriented language. The compiled Java code can run on all platforms that support Java without the need for recompilation hence Java is called as "writeonce, run anywhere" (WORA).

The Java compiled intermediate output called “byte-code” that can run on any Java virtual machine (JVM) regardless of computer architecture. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

In Linux operating system Java libraries are preinstalled. It’s very easy and convenient to compile and run Java programs in Linux environment. To compile and run Java Program is a two-step process:

1. Compile Java Program from Command Prompt

```
[root@host ~]# javac Filename.java
```

The Java compiler (Javac) compiles java program and generates a byte-code with the same file name and .class extension.

2. Run Java program from Command Prompt

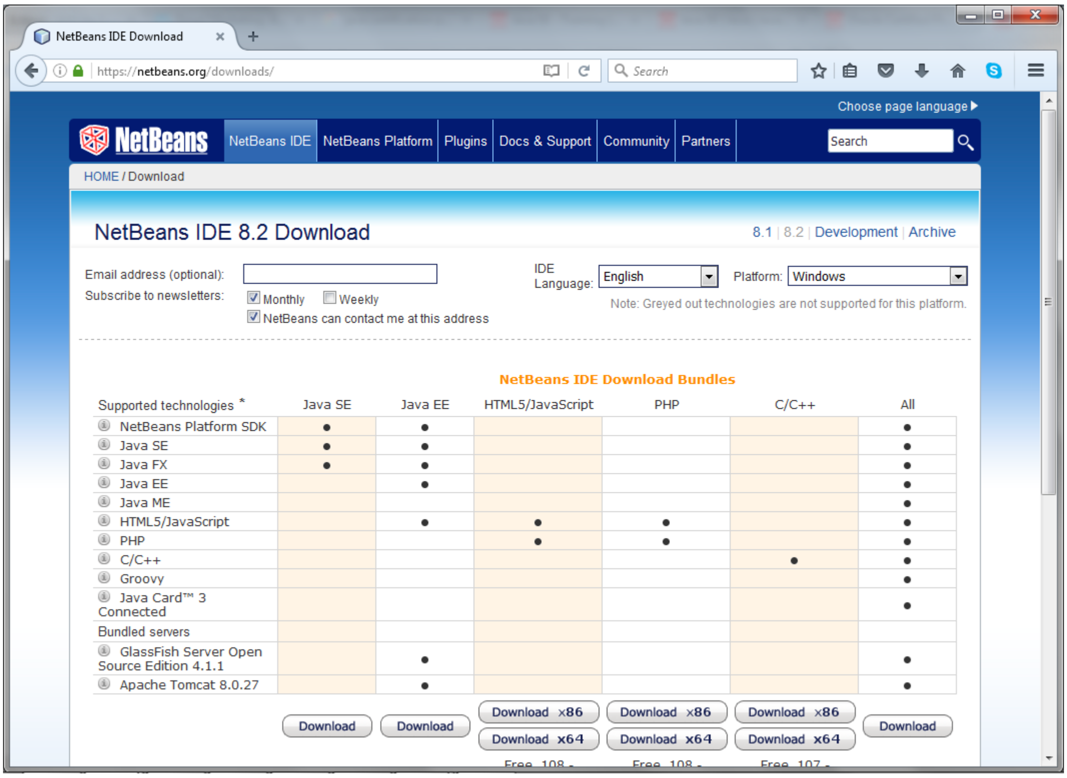
```
[root@host ~]# java Filename
```

The java interpreter (Java) runs the byte-code and gives the respective output. It is important to note that in above command we have omitted the .class suffix of the byte-code (Filename.class).

## How to Download and Install NetBeans IDE on your Computer.

Go to **https://netbeans.org/downloads** to download the latest version of NetBeans IDE.

You will see the following page:



NetBeans IDE 8.2 Download

8.1 | 8.2 | Development | Archive

Email address (optional):

Subscribe to newsletters: ☑ Monthly □ Weekly

☑ NetBeans can contact me at this address

IDE Language: English

Platform: Windows

Note: Greyed out technologies are not supported for this platform.

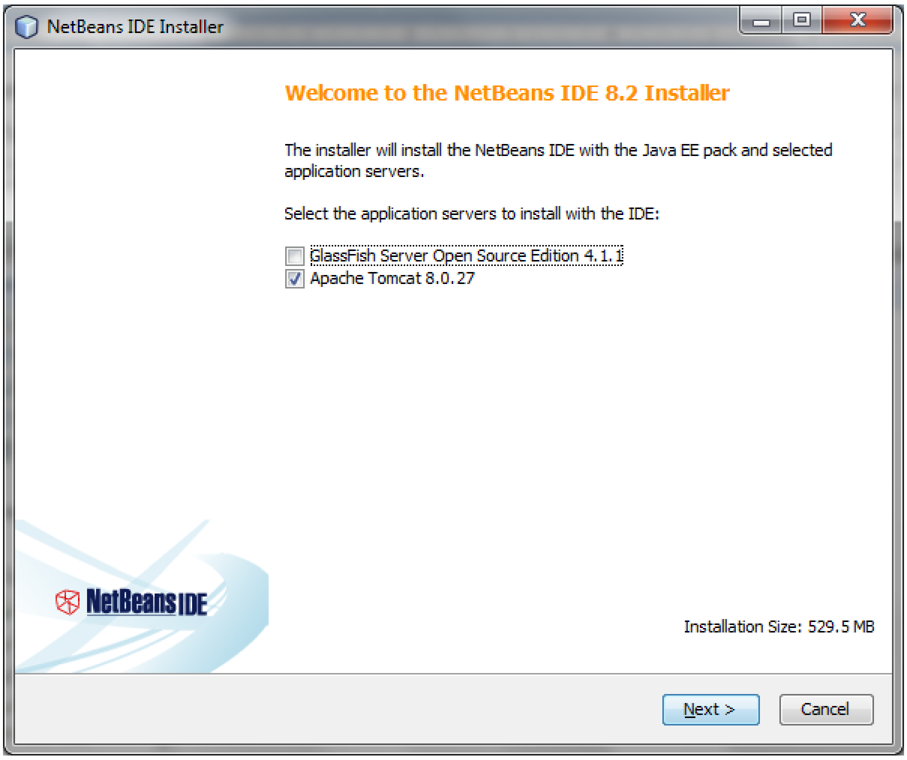
NetBeans IDE Download Bundles

| Supported technologies * | Java SE | Java EE | HTML5/JavaScript | PHP | C/C++ | All |
|---|---|---|---|---|---|---|
| NetBeans Platform SDK | ● | ● | | | | ● |
| Java SE | ● | ● | | | | ● |
| Java FX | ● | ● | | | | ● |
| Java EE | | ● | | | | ● |
| Java ME | | | | | | ● |
| HTML5/JavaScript | | ● | ● | ● | | ● |
| PHP | | | ● | ● | | ● |
| C/C++ | | | | | ● | ● |
| Groovy | | | | | | ● |
| Java Card™ 3 Connected | | | | | | ● |
| Bundled servers | | | | | | |
| GlassFish Server Open Source Edition 4.1.1 | | ● | | | | ● |
| Apache Tomcat 8.0.27 | | ● | | | | ● |
| | Download | Download | Download x86<br>Download x64 | Download x86<br>Download x64 | Download x86<br>Download x64 | Download |
| | | | Free 108 - | Free 108 - | Free 107 - | |

On this download page you see different download bundles. And for Java development only, we can choose either Java SE or Java EE. We'd recommend you to choose Java EE which supports comprehensive Java development (Java EE includes Java SE).

So click the Download button in the column Java EE to download NetBeans installer for Java EE development. The file name of the installer program is something like netbeans-8.2-javaee-windows.exe (on Windows).

Click on the installer file to start installing NetBeans IDE. You will be asked to install GlassFish and Apache Tomcat server



Welcome to the NetBeans IDE 8.2 Installer

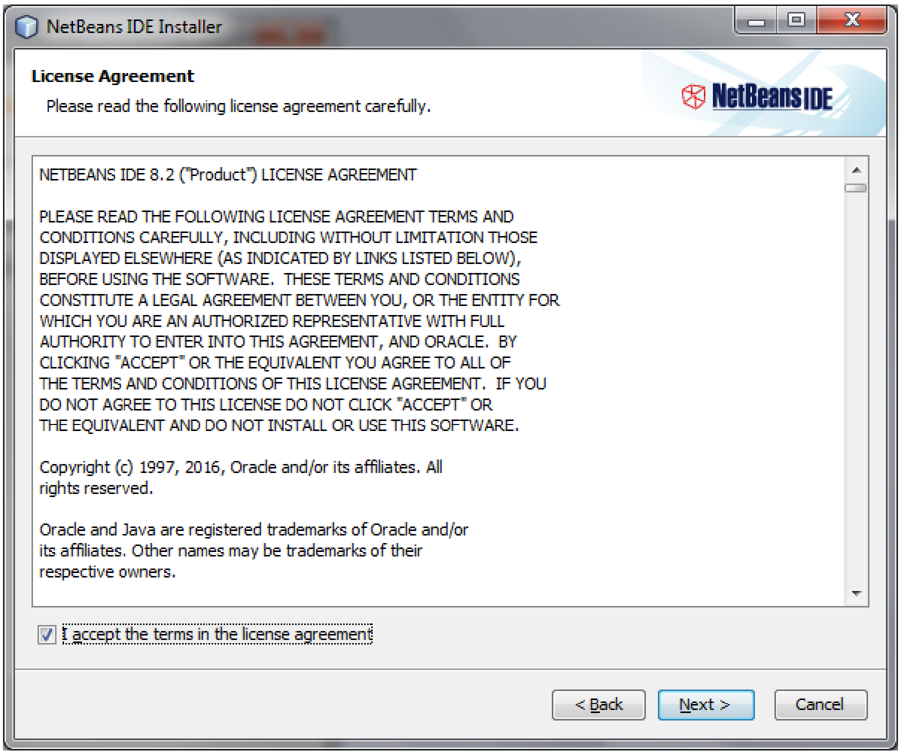
The installer will install the NetBeans IDE with the Java EE pack and selected application servers.

Select the application servers to install with the IDE:

□ GlassFish Server Open Source Edition 4.1.1
☑ Apache Tomcat 8.0.27

Installation Size: 529.5 MB

In this tutorial, you don’t need any server. However you will need them later so let check both, and **click Next.**
In the next screen, **check ‘I accept the terms in the license agreement’:**



License Agreement
Please read the following license agreement carefully.

NETBEANS IDE 8.2 ("Product") LICENSE AGREEMENT

PLEASE READ THE FOLLOWING LICENSE AGREEMENT TERMS AND CONDITIONS CAREFULLY, INCLUDING WITHOUT LIMITATION THOSE DISPLAYED ELSEWHERE (AS INDICATED BY LINKS LISTED BELOW), BEFORE USING THE SOFTWARE. THESE TERMS AND CONDITIONS CONSTITUTE A LEGAL AGREEMENT BETWEEN YOU, OR THE ENTITY FOR WHICH YOU ARE AN AUTHORIZED REPRESENTATIVE WITH FULL AUTHORITY TO ENTER INTO THIS AGREEMENT, AND ORACLE. BY CLICKING "ACCEPT" OR THE EQUIVALENT YOU AGREE TO ALL OF THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT. IF YOU DO NOT AGREE TO THIS LICENSE DO NOT CLICK "ACCEPT" OR THE EQUIVALENT AND DO NOT INSTALL OR USE THIS SOFTWARE.

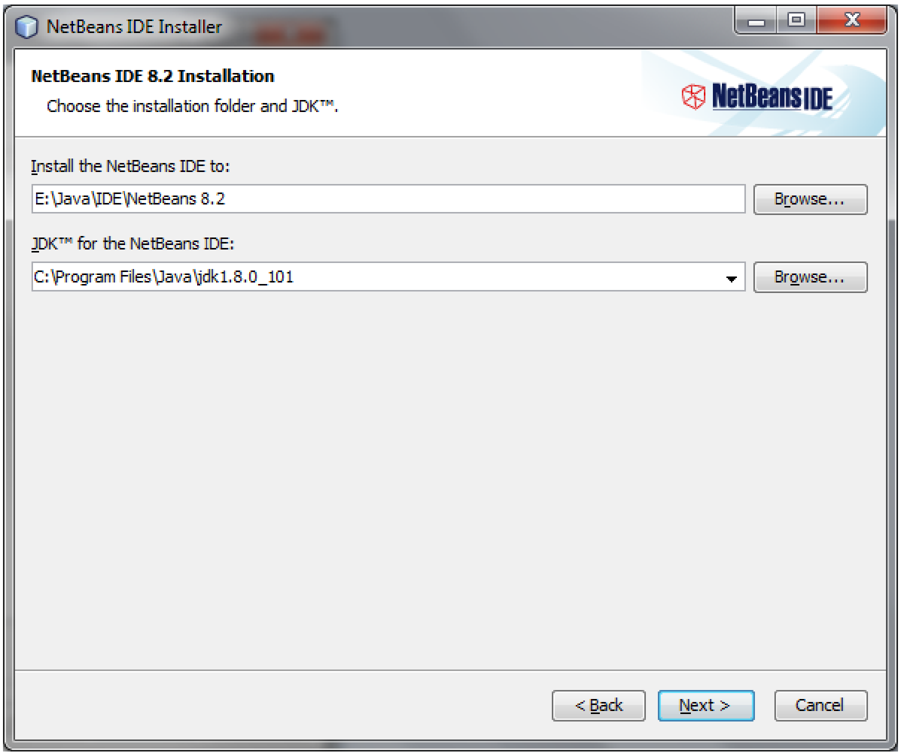
Copyright (c) 1997, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

☑ I accept the terms in the license agreement

**Click Next.**

In the next screen, choose the installation directory and JDK version for the IDE:



NetBeans IDE Installer

**NetBeans IDE 8.2 Installation**

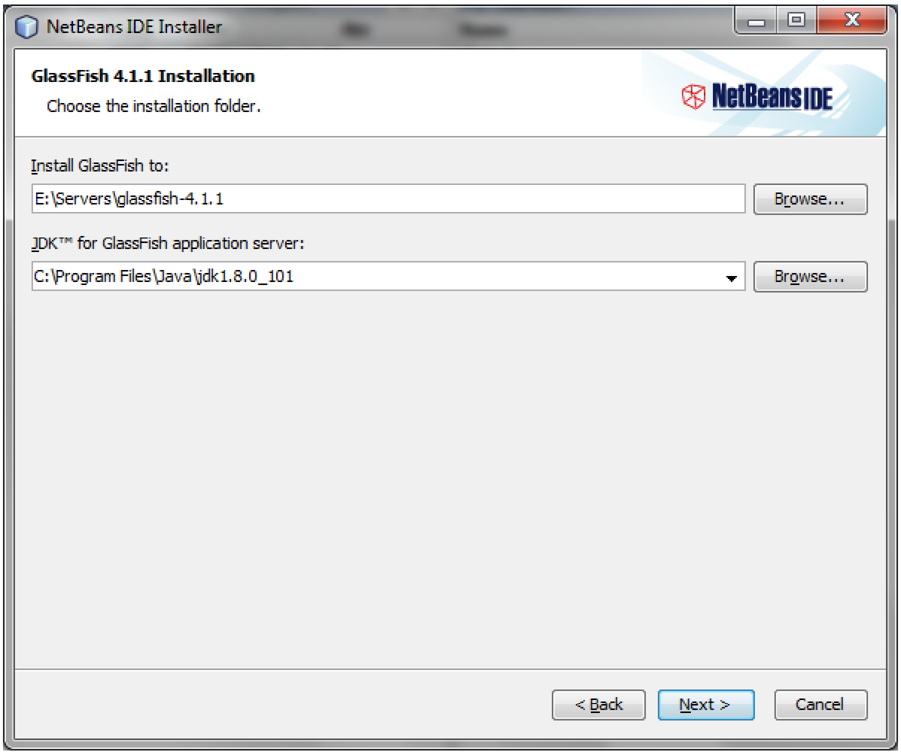
Choose the installation folder and JDK™.

Install the NetBeans IDE to: E:\Java\IDE\NetBeans 8.2

JDK™ for the NetBeans IDE: C:\Program Files\Java\jdk1.8.0_101

You can keep the defaults and click **Next.**

In the next screen, choose installation directory and JDK version for GlassFish server:



NetBeans IDE Installer

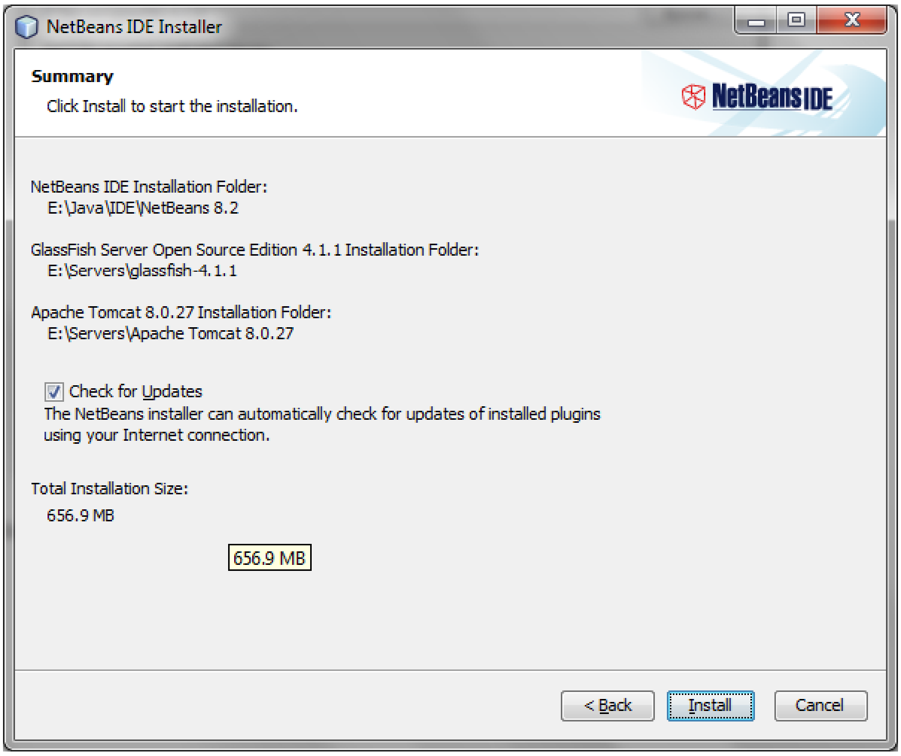
**GlassFish 4.1.1 Installation**

Choose the installation folder.

Install GlassFish to: E:\Servers\glassfish-4.1.1

JDK™ for GlassFish application server: C:\Program Files\Java\jdk1.8.0_101

Click **Next** to see the summary:



**Summary**

Click Install to start the installation.

NetBeans IDE Installation Folder:
E:\Java\IDE\NetBeans 8.2

GlassFish Server Open Source Edition 4.1.1 Installation Folder:
E:\Servers\glassfish-4.1.1

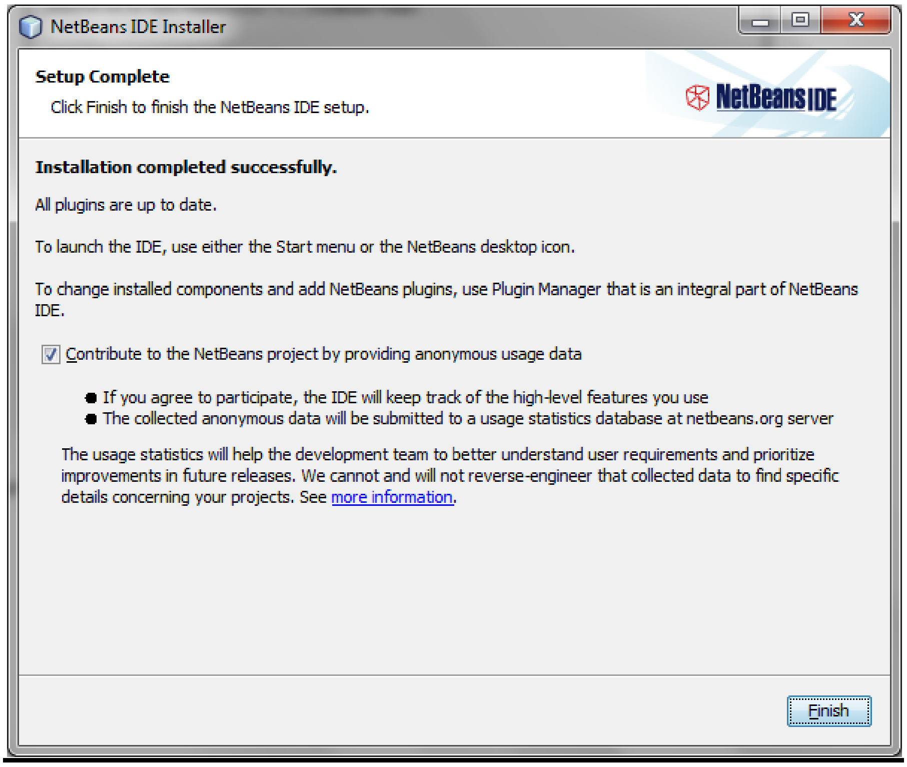
Apache Tomcat 8.0.27 Installation Folder:
E:\Servers\Apache Tomcat 8.0.27

☑ Check for Updates
The NetBeans installer can automatically check for updates of installed plugins using your Internet connection.

Total Installation Size:
656.9 MB

656.9 MB

And click **Install** to start installing NetBeans with GlassFish and Tomcat servers. Wait until the setup complete:



**Setup Complete**

Click Finish to finish the NetBeans IDE setup.

**Installation completed successfully.**

All plugins are up to date.

To launch the IDE, use either the Start menu or the NetBeans desktop icon.

To change installed components and add NetBeans plugins, use Plugin Manager that is an integral part of NetBeans IDE.

☑ Contribute to the NetBeans project by providing anonymous usage data

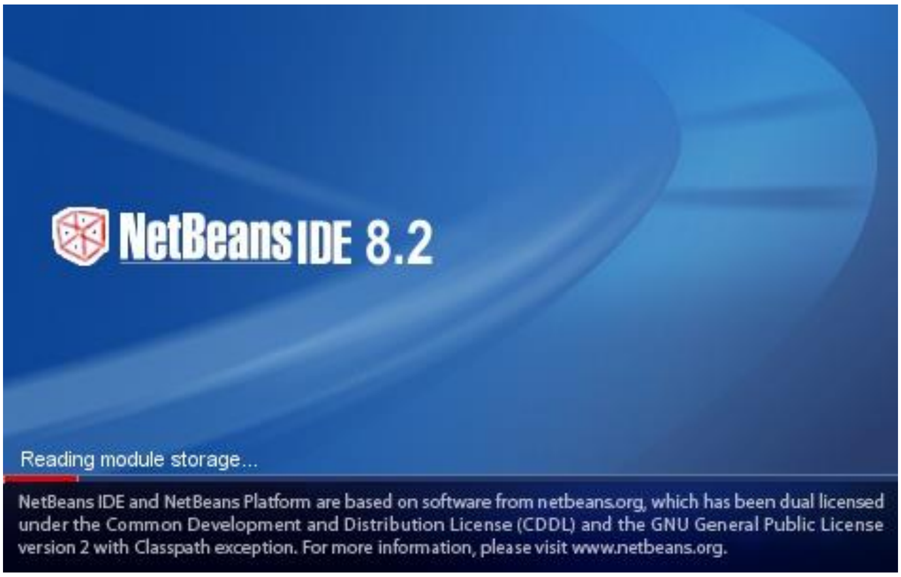
- If you agree to participate, the IDE will keep track of the high-level features you use
- The collected anonymous data will be submitted to a usage statistics database at netbeans.org server

The usage statistics will help the development team to better understand user requirements and prioritize improvements in future releases. We cannot and will not reverse-engineer that collected data to find specific details concerning your projects. See more information.

**Click Finish.**

Now you can start NetBeans IDE from the start menu or Run NetBeans by clicking the NetBeans icon on desktop as shown below

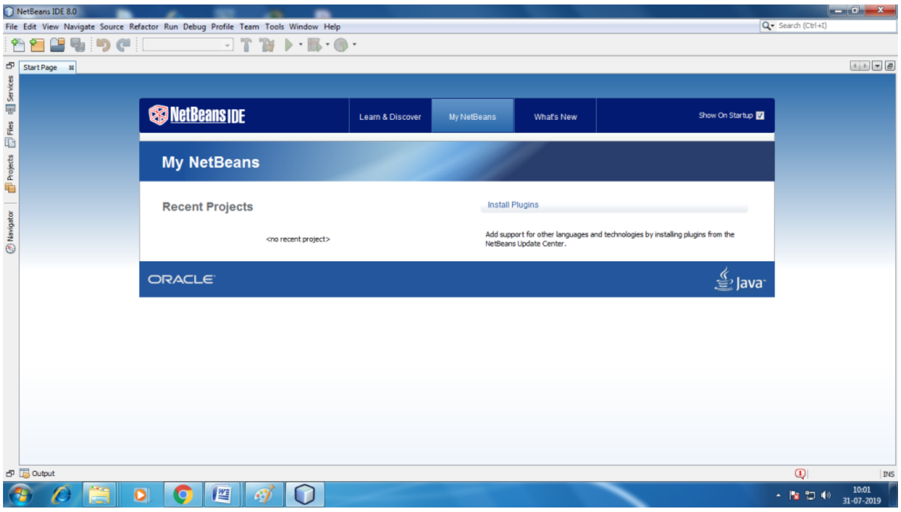
The splash screen appears:



Reading module storage...

NetBeans IDE and NetBeans Platform are based on software from netbeans.org, which has been dual licensed under the Common Development and Distribution License (CDDL) and the GNU General Public License version 2 with Classpath exception. For more information, please visit www.netbeans.org.

And you should see the home screen of NetBeans:



My NetBeans

Recent Projects

<no recent project>

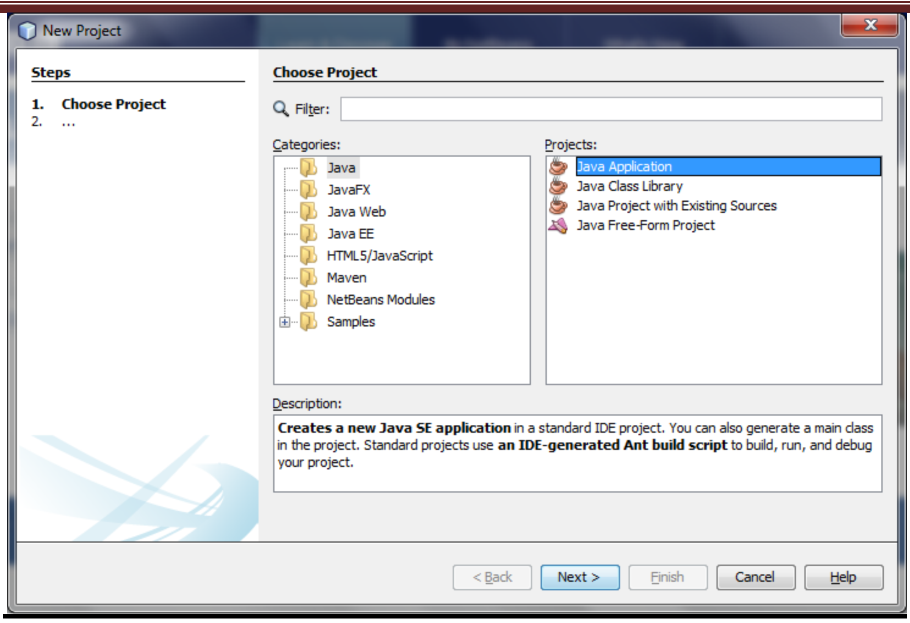
Install Plugins

Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

## How to Create Your First Java Project

Now, let’s create a Java project using NetBeans IDE. Go to menu **File > New Project…**

Under the New Project dialog, choose Java application as shown in the following screenshot:



New Project

Steps

1. Choose Project
2. ...

Choose Project

Filter:

Categories:

- Java
- JavaFX
- Java Web
- Java EE
- HTML5/JavaScript
- Maven
- NetBeans Modules
- Samples

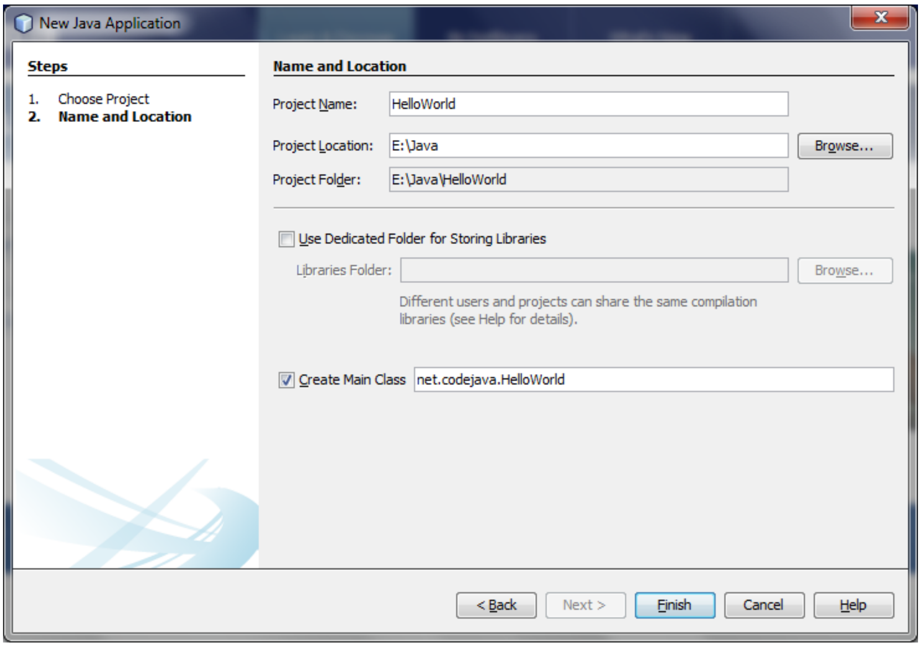
Projects:

- Java Application
- Java Class Library
- Java Project with Existing Sources
- Java Free-Form Project

Description:

**Creates a new Java SE application** in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

Click **Next** to advance to the next step. In the New Java Application screen, type Project Name, specify Project Location and the main class:



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: HelloWorld

Project Location: E:\Java

Project Folder: E:\Java\HelloWorld

□ Use Dedicated Folder for Storing Libraries

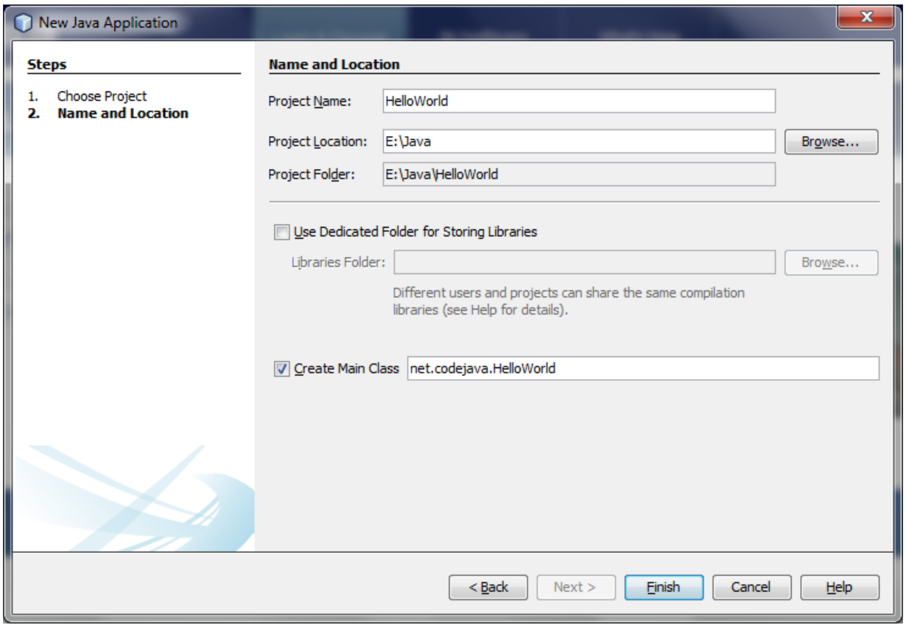
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class net.codejava.HelloWorld

## How To Write Your First Java Program

Click **Next** to advance to the next step. In the New Java Application screen, type Project Name, specify Project Location and the main class:



New Java Application

**Steps**

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name: HelloWorld

Project Location: E:\Java

Project Folder: E:\Java\HelloWorld

□ Use Dedicated Folder for Storing Libraries

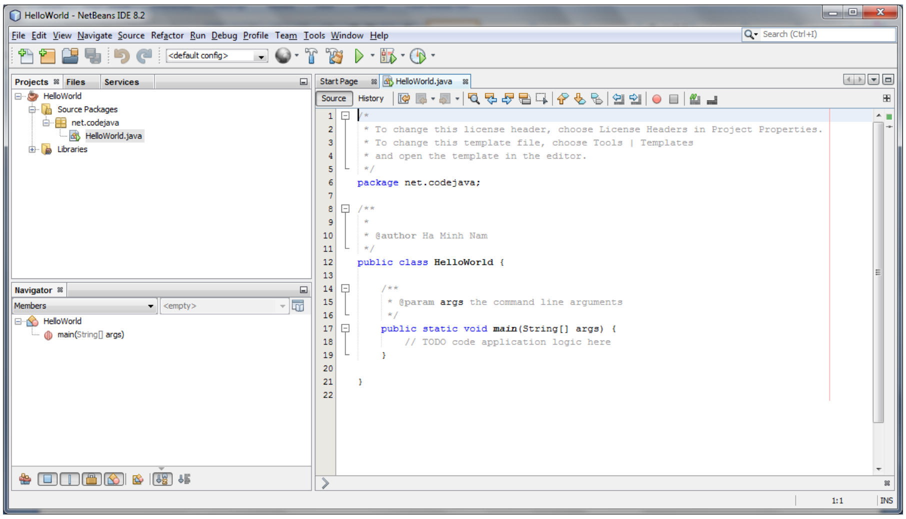
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class net.codejava.HelloWorld

**Note** that we check the option Create Main Class to generate the main class for the application. Here we specify the package name net.codejava before the class name HelloWorld.

Click Finish. NetBeans create the project with a main class very quickly:

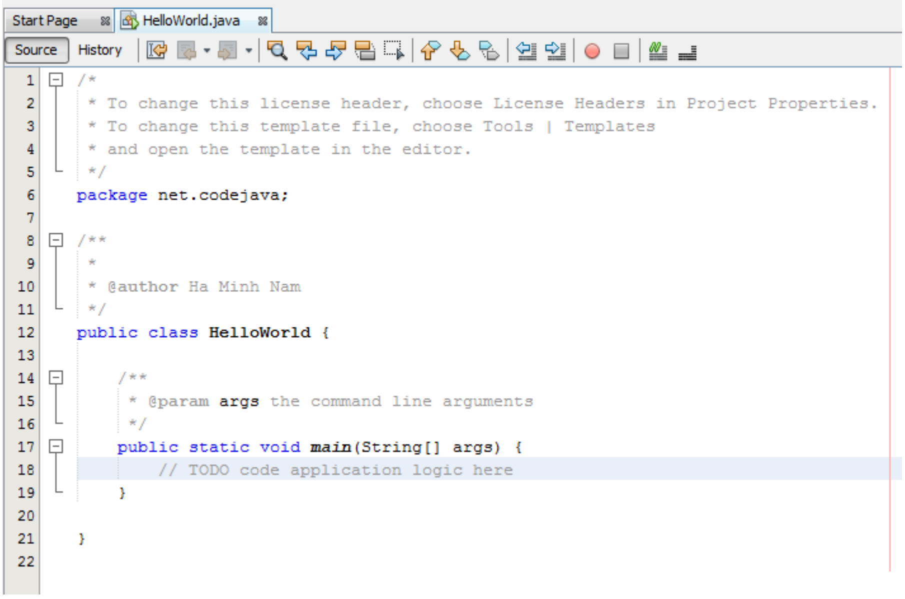


HelloWorld - NetBeans IDE 8.2

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package net.codejava;

/**
 *
 * @author Ha Minh Nam
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

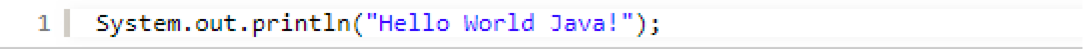
## Write Your First Java Code

You can see a code editor for the HelloWorld.java file as shown in the following screenshot:



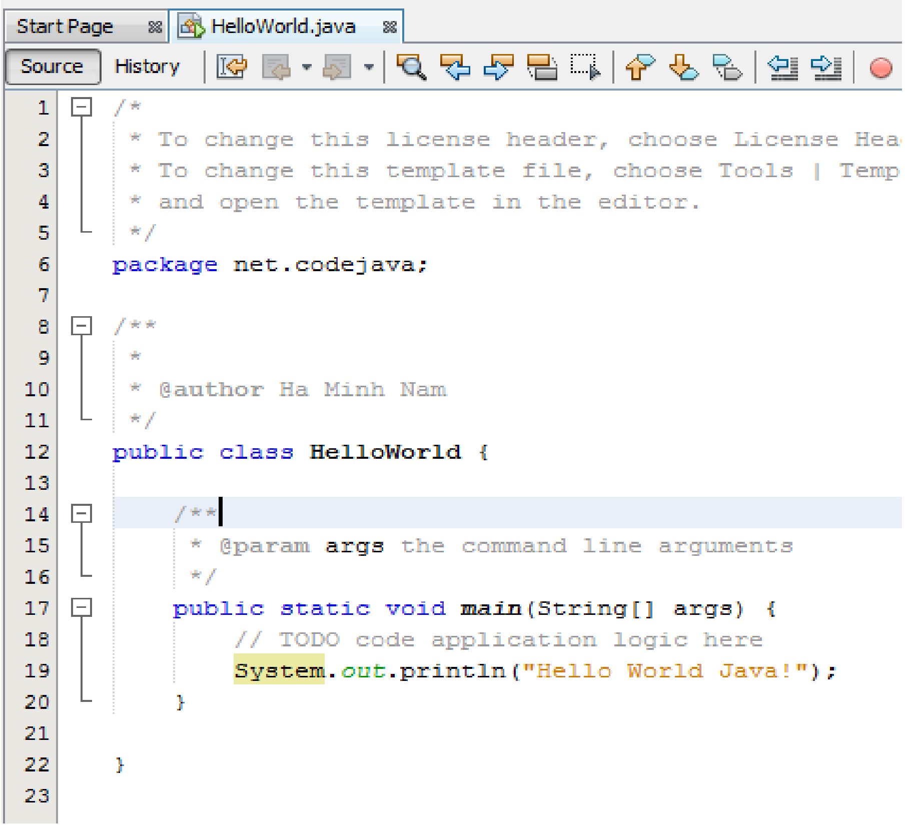
```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package net.codejava;

/**
 *
 * @author Ha Minh Nam
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

The method main() is the main entry to a Java application. All Java programs start from the main()method. Now, let's type some code in this method to print "Hello World Java!" on the screen:



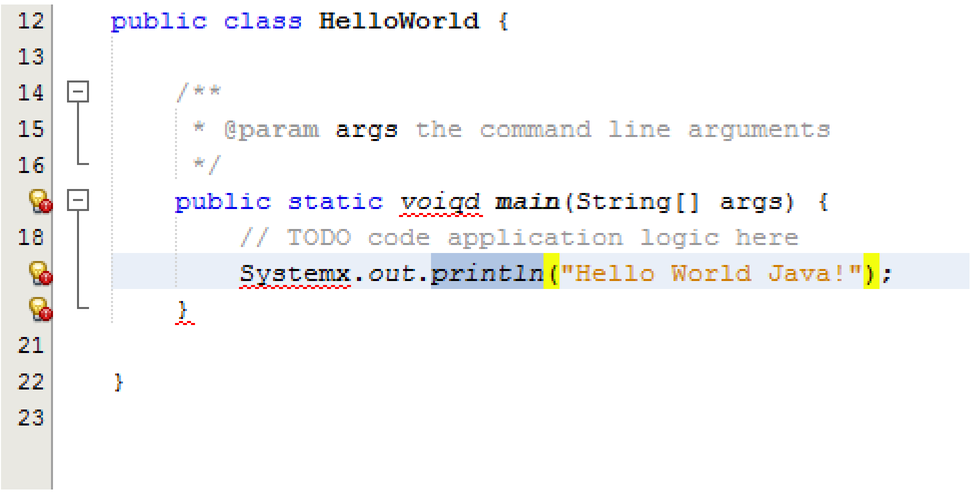
```
System.out.println("Hello World Java!");
```

The whole program should look like this:



```
/*
 * To change this license header, choose License Hea
 * To change this template file, choose Tools | Temp
 * and open the template in the editor.
 */
package net.codejava;

/**
 *
 * @author Ha Minh Nam
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        System.out.println("Hello World Java!");
    }

}
```

NetBeans is very smart, as it compiles the code instantly while you are typing the code. So if there's any error, the IDE will inform you by underling the errors with red color, as shown in the following screenshot:



```
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static voiqd main(String[] args) {
        // TODO code application logic here
        Systemx.out.println("Hello World Java!");
    }

}
```

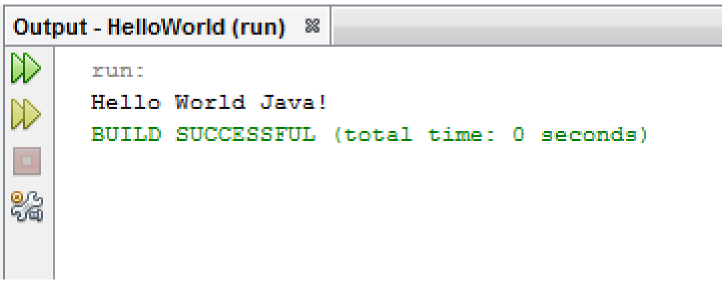
If there's no red marks like this, the code is fine and we're ready to run the program.

## Run Your First JavaProgram

To run the HelloWorld program above, there are several ways:

- Go to menu **Run > Run Project <ProjectName>**
- Click **Run Project** icon in the toolbar.
- Press **F6** key.
- Right click in the code editor, and select **Run File** (or press **Shift + F6**).

You should see the output of this program like this:



```
run:
Hello World Java!
BUILD SUCCESSFUL (total time: 0 seconds)
```

That's it! The HelloWorld program has run and printed the output "Hello World" .

you have successfully created and run your first Java program with NetBeans IDE

# Sample Networking Programs in Java

## UDP SAMPLE PROGRAMS

### UDP SAMPLE PROGRAM 1

**AIM:** This Program Demonstrates Unidirectional Communication Between Udpclient And Udp Server. Message is sent from Udpclient to UdpServer and message is displayed on UdpServer side

### UDP Client  Program

```
import java.io.*;
import java.net.*;
public class UDPClient1
{
   public static void main(String[] args)
   {
  try
            {
        //create client socket
       DatagramSocket  clientsocket=new DatagramSocket();
       // get ipaddress of host machine
       InetAddress IPAddress=InetAddress.getByName("localhost");
      // create buffer for sending data
       byte[] outdata=new byte[1024];
      // read input through keyboard
       BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
       System.out.println("enter some text to send it to UDP server");
       String str=br.readLine();
       // convert string to packet data
       outdata=str.getBytes();
 DatagramPacket sendpacket=new DatagramPacket(outdata,outdata.length,IPAddress,2222);
       clientsocket.send(sendpacket);
       System.out.println("packet sent to server");
       clientsocket.close();
            }
       catch(Exception ex)
       {
          System.out.println(ex.getMessage());
       }
   }
}
```

## UDPServerProgram

```
import java.io.*;
import java.net.*;
public class UDPServer1
{
   public static void main(String[] args)
   {
            try
      {
     // create server object
      DatagramSocket srvrobj=new DatagramSocket(2222);
        // array to store data which comes from client
      byte[] InData=new byte[1024];
      System.out.println("server is running");
      while(true)
      {
         // create datagram encapsulated datagram packet
         DatagramPacket receivepacket=new DatagramPacket(InData,InData.length);
        // server receives packet from client
         srvrobj.receive((receivepacket));
         String s1=new String(receivepacket.getData());
         System.out.println("data received from client is "+s1);

      }
   }
      catch(Exception ex)
      {
         System.out.println(ex.getMessage());
      }
   }
 }
```

**Output**

**Step1**

**UDPServer Side**

First run UDPServer program to start UDPServer

You will get following message that server is started

**server is running**

**Step2**

**UDPClient Side**

then run UDPClient program to send msg from UDPClient TO UDPServer

Below is message when you typed will be sent from UDPClient to UDPServer

**enter some text to send it to UDP server**

**hello atme college of engineering**

**packet sent to server**

**Step3**

Finally message is received from UDPClient program to UDPServer program.Message is displayed on UDPServer Program side as follows

**server is running**

**data received from client is hello atme college of engineering.**

## UDP SAMPLE PROGRAM 2

**AIM:** This Program Demonstrates Unidirectional Communication Between Udpclient And Udp Server. Message is sent from Udpclient to UdpServer and message is displayed on UdpServer side program

### UDPServerProgram

```
import java.net.*;
import java.io.*;
public class UDPServer2
{
    public static void main(String[] args)
    {
     try
        {
        DatagramSocket serverscoket=new DatagramSocket(3000);
        byte[] buffer=new byte[1024];
         System.out.println("server is running");
        DatagramPacket dp=new DatagramPacket(buffer,1024);
        serverscoket.receive(dp);
        String src=new String(buffer,0,buffer.length);
         System.out.println("data from client is: "+src);
         serverscoket.close()
        }
        catch(Exception ex)
        {
            System.out.println(ex.getMessage());
        }
    }
```

## UDPClient Program

```
import java.io.*;
import java.net.*;
public class UDPClient2
{
    public static void main(String[] args)
    {
         try
        {
            DatagramSocket clientsocket=new DatagramSocket();
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            System.out.println("enter some text so that messages is sent from UDPCLIENT to
UDPSERVER");
            String str=br.readLine();
             InetAddress ip=InetAddress.getByName("127.0.0.1");
             DatagramPacket dp=new DatagramPacket(str.getBytes(),str.length(),ip,3000);
             clientsocket.send(dp);
             clientsocket.close();
        }
        catch(Exception ex)
        {
            System.out.println(ex.getMessage());
        }
    }

}
```

**Output**

**Step1**

**UDPServer Side**

First run UDPServer program to start UDPServer

You will get following message that server is started

**server is running**

**Step2**

**UDPClient Side**

then run UDPClient program to send msg from UDPClient program TO UDPServer program

Below is message you have typed which will be sent from UDPClient to UDPServer

**enter some text so that messages is sent from UDPCLIENT to UDPSERVER**

**Academy for Technical and Management Excellence**

**Step3**

The message is received from UDPClient to UDPServer.Message is displayed on UDPServer side as follows

**server is running**

**data from client is: Academy for Technical and Management Excellence**

## UDP SAMPLE PROGRAM 3

**AIM:**This Program Demonstrates Bidirection Communcation Between Udpclient And Udp Server.

### UDPServer Program

```
import java.io.*;
import java.net.*;
public class UDPServer3
{
   public static void main(String[] args)
   {
      try
      {
DatagramSocket serverSocket = new DatagramSocket(9876);
     byte[] receiveData = new byte[1024];
     byte[] sendData;
     System.out.println("server is running");
     while(true)
     {
       DatagramPacket receivePacket =new DatagramPacket(receiveData, receiveData.length);
       serverSocket.receive(receivePacket);
       String strreceived = new String(receivePacket.getData(), 0, receivePacket.getLength());
       InetAddress IPAddress = receivePacket.getAddress();
       int port = receivePacket.getPort();
 String modifiedstring=strreceived.toUpperCase();
       System.out.println("data received from client is"+modifiedstring);
       sendData = modifiedstring.getBytes();
       DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, port);
```

```
        serverSocket.send(sendPacket);
      }
      }
     catch(Exception ex)
     {
        System.out.println(ex.getMessage());
     }
    }
    }
```

## UDPClient Program

```
import java.io.*;
import java.net.*;
public class UDPClient3
{
   public static void main(String[] args)
   {
      try
      {
         System.out.println("enter some text so that message is sent from  UDPCLIENT to 
UDPSERVER and Again from UDPSERVER to UDPClient");
        BufferedReader br = new BufferedReader( new InputStreamReader(System.in));

      DatagramSocket clientSocket = new DatagramSocket();
     InetAddress IPAddress = InetAddress.getByName("localhost");
      byte[] sendData;
      byte[] receiveData = new byte[1024];
      String sentence = br.readLine();
      sendData = sentence.getBytes();
      DatagramPacket sendPacket =new DatagramPacket(sendData, sendData.length,IPAddress, 9876);
      clientSocket.send(sendPacket);
      DatagramPacket receivePacket =new DatagramPacket(receiveData, receiveData.length);
      clientSocket.receive(receivePacket);
      String modifiedSentence = new String(receivePacket.getData(), 0, receivePacket.getLength());
      System.out.println("message FROM UDPSERVER SERVER: " + modifiedSentence);
      clientSocket.close();
       }
      catch(Exception ex)
      {
         System.out.println(ex.getMessage());
      }
   }
```

}

**Output**

**Step1**

**UDPServer Side**

First run UDPServer program to start UDPServer

You will get following message that server is started

```
server is running
```

**Step2**

**UDPClient Side**

Now run UDPClient program to send message from UDPClient TO UDPServer

Below is message you have typed which is sent from UDPClient program to UDPServer program.below is output shown.

```
enter some text so that message is sent from  UDPCLIENT to UDPSERVER and Again Message comes back from UDPSERVER to UDPClient
academy for technical and management excellence
```

**Step3**

**UDP Server Side**

Message is received by UDPServer Program from UDPClient Program .The messa

```
server is running
data received from client isACADEMY FOR TECHNICAL AND MANAGEMENT EXCELLENCE
```

**Step4**

same message is sent back to UDPClient from UDPServer .UDP Client receives string from UDPServer which client had sent same message to UDPServer .Message is displayed as shown below.

```
enter some text so that message is sent from  UDPCLIENT to UDPSERVER and Again from UDPSERVER to UDPClient
academy for technical and management excellencemessage FROM UDPSERVER SERVER: ACADEMY FOR TECHNICAL AND MANAGEMENT EXCELLENCE
```

## Sample TCP Programs

### Sample TCP Program1

**AIM:**This Program Demonstrates Unidirection Communcation Between TCPClient AND TCPServer.Message is Sent from TCPClient program to TCPServer and Message Received by TCPServer  is Displayed on Server side

**TCPServer Program**

```
import java.io.*;

import java.net.*;

public class TCPServer1

 {

 public static void main(String[] args)

    {

         try

        {

        ServerSocket serverobj=new ServerSocket(6600);

         System.out.println("server is started");

        Socket sobj=serverobj.accept();

       System.out.println("client connected to server");

        DataInputStream dis=new DataInputStream(sobj.getInputStream());

        String str=(String)dis.readUTF();

        System.out.println("message from client is: "+str);

        sobj.close();

        serverobj.close();

        }

        catch(Exception ex)

        {

            System.out.println(ex.getMessage());

        }
    }
}
```

**TCPClient Program**

```
import java.io.*;

import java.net.*;

import java.util.*;

public class TCPClient1

{

    public static void main(String[] args)

    {

        try

        {

       Socket sktobj=new Socket("localhost",6600);

       DataOutputStream dos=new DataOutputStream(sktobj.getOutputStream());

        System.out.println("Enter message to be sent to server");

        Scanner sc=new Scanner(System.in);

        String read=sc.nextLine();

        dos.writeUTF(read);

        dos.flush();

        sktobj.close();

        dos.close();

        }

        catch(Exception ex)

        {

            System.out.println(ex.getMessage());

        }

    }

}
```

**Output**

**Step1**

**TCPServer Side**

First run TCPServer Side program to start server

You will get following message that server is started

```
server is started
```

**Step2**

**TCPClient Side**

then run TCPClient program to send msg from TCPClient TO TCPServer

Below is message you have typed on client side which is sent from TCPClient to TCPServer

```
Enter message to be sent to server
Onto the leading edge
```

**Step3**

Finally message is received from TCPClient program to TCPServer program.Message is displayed on TCPServer side as follows

```
server is started
client connected to server
message from client is: onto the leading edge
```

## Sample TCP Program2

**AIM:**This Program Demonstrates Unidirection Communcation Between TCPServerAND TCPClient.Message is Sent from TCPServerto TCPClientand Message Received is Displayed on TCPClientside

### TCPServer Program

```
import java.io.*;

import java.net.*;

import java.util.*;

public class TCPServer2

{
    public static void main(String[] args)
{
      try
      {
         ServerSocket serversocket=new ServerSocket(1234);
          System.out.println("server has started ");
         Socket skt=serversocket.accept();
         System.out.println("server has connnected to client");
         System.out.println("Enter message to be sent to client");
         Scanner sc=new Scanner(System.in);
         String read=sc.nextLine();
         PrintWriter out=new PrintWriter(skt.getOutputStream(),true);
       //  System.out.println("Sending String"+data+""\n");
         out.print(read);
         out.close();
         skt.close();
      }
      catch(Exception e)
      {
         System.out.print(e.getMessage());
      }
    }
}
```

**TCPClient Program**

```
import java.io.*;
import java.net.*;
import java.util.*;
public class TCPClient2 {
   public static void main(String[] args)
   {
 try
      {
         Socket skt=new Socket("localhost",1234);
BufferedReader input=new BufferedReader(new InputStreamReader(skt.getInputStream()));
         //System.out.println("String Received");
         while(!input.ready())
         {
         }
          System.out.println("String Received from server is "+input.readLine());
         System.out.println("\n");
         input.close();
      }
      catch(Exception e)
     {
         System.out.print(e.getMessage());
      }
   }
}
```

**Output**

**Step1**

**TCPServer Side**

First run TCPServer Side program to start server

You will get following message that server is started

**server has started**

**Step2**

TCPClient Side

then run TCPClient program

**Step3**

Finally type the messageto be sent  to TCPClient from TCPServer Program . This message is sent from TCPServer toTCPClient

```
server has started

server has connnected to client

Enter message to be sent to client

mysuru is heritage city
```

**Step4**

HereTCPClient receives message from TCPServer side.Message is displayed on TCPClient side as shown below

```
String Received from server is mysuru is heritage city
```

## Program 6

**AIM:** To write a java program to find the IP address of the system.

```
import java.io.*;

import java.net.*;

public class IPADDRES

{

   public static void main(String[] args) throws UnknownHostException

   {

InetAddress addr = InetAddress.getLocalHost();

     System.out.println("Local HostAddress: "+addr.getHostAddress());

     String hostname = addr.getHostName();

     System.out.println("Local host name: "+hostname);

}

}
```

**OUTPUT**

```
Local HostAddress: 172.16.16.181

Local host name: atme-PC
```

## Program No. 1:Implement three nodes point – to – point network with duplex links between them for different topologies. 1 Set the queue size, vary the bandwidth and find the number of packets dropped for various iterations.

**Program Objective:**

- Understand the Implementation of the Duplex link between the network.

**Theory:**

- Create a simulatorobject.
- We open a file for writing that is going to be used for the tracedata.
- We now attach the agent to thenodes.
- Now we attach the application to run on top of thesenodes
- We now connect the agent and the application for itsworking
- Set the simulationtime
- The next step is to add a 'finish' procedure that closes the trace file and startsnam.

```
set ns [new Simulator]
set ntrace [open prog1.tr w]
$ns trace-all $ntrace
set namfile [open prog1.nam w]
$ns namtrace-all $namfile

proc Finish {} {
global ns ntrace namfile
$ns flush-trace
close $ntrace
close $namfile
exec nam prog1.nam &
puts "The number of packet drops is"
exec grep -c "^d" prog1.tr &
exit 0
}

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

$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 5Mb 10ms DropTail

$ns queue-limit $n0 $n1 10
$ns queue-limit $n1 $n2 05

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

set sink0 [new Agent/TCPSink]
$ns attach-agent $n2 $sink0
```

```
$ns connect $tcp0 $sink0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set type_ CBR
$cbr0 set packetSize_ 100
$cbr0 set rate_ 1Mb
$cbr0 set random_ false
$cbr0 attach-agent $tcp0
$tcp0 set class_ 1
$ns at 1.0 "$cbr0 start"
$ns at 5.0 "Finish"
$ns run
```

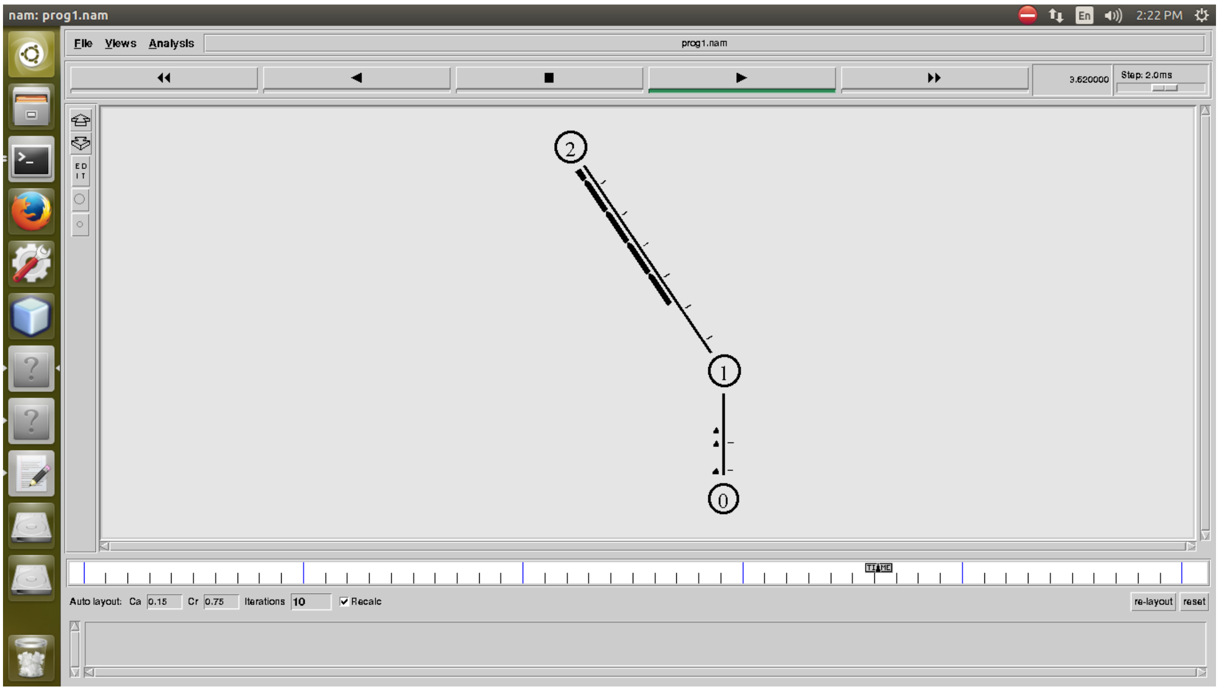
## Output:

### Steps for execution

1. Open gedit and type program. Program name should have the extension " .tcl
   **student@cnpc022:~/naveen$ gedit prog1.tcl**
2. Save the program.
3. Run the simulation program
   **student@cnpc022:~/naveen$ ns prog1.tcl**
   **The number of packet drops is**
   **5**
4. Here "ns" indicates network simulator. We get the topology shown in the snapshot.
5. Now press the play button in the simulation window and the simulation will begin.
6. To see the trace file contents open the file as ,
   **student@cnpc022:~/naveen$  gedit prog1.tr**

## Topology

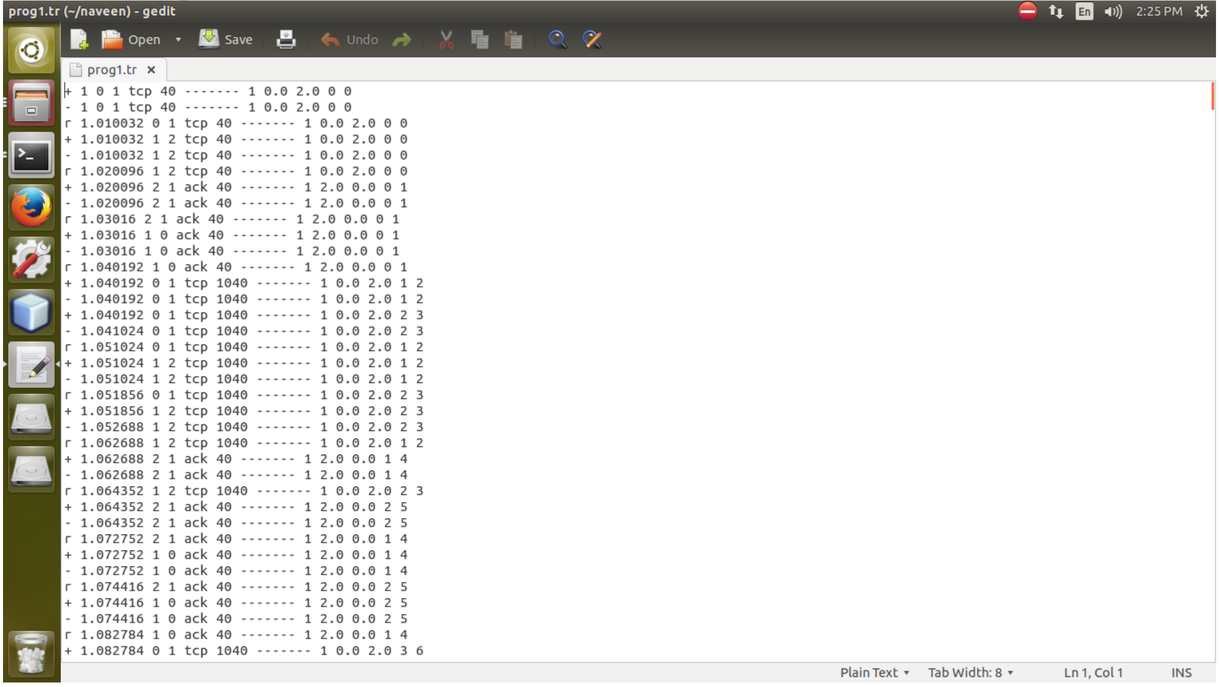
### Snapshot 1:



Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

## Snapshot 2:

### Trace FileSnapshot



```
+ 1 0 1 tcp 40 ------- 1 0.0 2.0 0 0
- 1 0 1 tcp 40 ------- 1 0.0 2.0 0 0
r 1.010032 0 1 tcp 40 ------- 1 0.0 2.0 0 0
+ 1.010032 1 2 tcp 40 ------- 1 0.0 2.0 0 0
- 1.010032 1 2 tcp 40 ------- 1 0.0 2.0 0 0
r 1.020096 1 2 tcp 40 ------- 1 0.0 2.0 0 0
+ 1.020096 2 1 ack 40 ------- 1 2.0 0.0 0 1
- 1.020096 2 1 ack 40 ------- 1 2.0 0.0 0 1
r 1.03016 2 1 ack 40 ------- 1 2.0 0.0 0 1
+ 1.03016 1 0 ack 40 ------- 1 2.0 0.0 0 1
- 1.03016 1 0 ack 40 ------- 1 2.0 0.0 0 1
r 1.040192 1 0 ack 40 ------- 1 2.0 0.0 0 1
+ 1.040192 0 1 tcp 1040 ------- 1 0.0 2.0 1 2
- 1.040192 0 1 tcp 1040 ------- 1 0.0 2.0 1 2
+ 1.040192 0 1 tcp 1040 ------- 1 0.0 2.0 2 3
- 1.041024 0 1 tcp 1040 ------- 1 0.0 2.0 2 3
r 1.051024 0 1 tcp 1040 ------- 1 0.0 2.0 1 2
+ 1.051024 1 2 tcp 1040 ------- 1 0.0 2.0 1 2
- 1.051024 1 2 tcp 1040 ------- 1 0.0 2.0 1 2
r 1.051856 0 1 tcp 1040 ------- 1 0.0 2.0 2 3
+ 1.051856 1 2 tcp 1040 ------- 1 0.0 2.0 2 3
- 1.052688 1 2 tcp 1040 ------- 1 0.0 2.0 2 3
r 1.062688 1 2 tcp 1040 ------- 1 0.0 2.0 1 2
+ 1.062688 2 1 ack 40 ------- 1 2.0 0.0 1 4
- 1.062688 2 1 ack 40 ------- 1 2.0 0.0 1 4
r 1.064352 1 2 tcp 1040 ------- 1 0.0 2.0 2 3
+ 1.064352 2 1 ack 40 ------- 1 2.0 0.0 2 5
- 1.064352 2 1 ack 40 ------- 1 2.0 0.0 2 5
r 1.072752 2 1 ack 40 ------- 1 2.0 0.0 1 4
+ 1.072752 1 0 ack 40 ------- 1 2.0 0.0 1 4
- 1.072752 1 0 ack 40 ------- 1 2.0 0.0 1 4
r 1.074416 2 1 ack 40 ------- 1 2.0 0.0 2 5
+ 1.074416 1 0 ack 40 ------- 1 2.0 0.0 2 5
- 1.074416 1 0 ack 40 ------- 1 2.0 0.0 2 5
r 1.082784 1 0 ack 40 ------- 1 2.0 0.0 1 4
+ 1.082784 0 1 tcp 1040 ------- 1 0.0 2.0 3 6
```

## Program Outcome :

- Implement the Duplex link between the networks.

## Viva Questions:

- Define Duplex link
- Define Bandwidth

## Program No. 2: Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the throughput with respect to transmission of packets.

**Program Objective:**

- Understand the Implementation of the simple ESS in wireless LAN.

```
set ns [new Simulator]

set topo [new Topography]
$topo load_flatgrid 500 500

set nf [open prog4.nam w]
$ns namtrace-all-wireless $nf 500 400
set tf [open prog4.tr w]
$ns trace-all $tf
#Create god object
create-god 3

$ns node-config -adhocRouting AODV \
-llType LL \
-macType Mac/802_11 \
-ifqType Queue/DropTail \
-ifqLen 50 \
-phyType Phy/WirelessPhy \
-channelType Channel/WirelessChannel \
-propType Propagation/TwoRayGround \
-antType Antenna/OmniAntenna \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

$n0 label "tcp0"
$n1 label "sink0/tcp1"
$n2 label "sink1"

$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_ 300
$n2 set Y_ 300
$n2 set Z_ 0
```

```
$ns at 0.1 "$n0 setdest 250 150 15"
$ns at 0.1 "$n1 setdest 200 200 25"
$ns at 0.1 "$n2 setdest 100 100 25"
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0

set sink0 [new Agent/TCPSink]
$ns attach-agent $n1 $sink0
$ns connect $tcp0 $sink0
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n2 $sink1
$ns connect $tcp1 $sink1

$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"
 $ns at 50 "$n1 setdest 350 250 15"
$ns at 80 "$n1 setdest 70 70 15"

proc finish {} {
global ns nf tf
$ns flush-trace
exec nam prog4.nam  &
close $tf
exit 0
}
$ns at 100 "finish"
$ns run
```

**AWK FILE:prog4.awk**

```
BEGIN{
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
time2=$2
}
}

END{
printf("The Throughput from n0 to n1: %f Mbps\n", ((count1*pack1*8)/(time1*1000000)));
printf("The Throughput from n1 to n2: %f Mbps", ((count2*pack2*8)/(time2*1000000)));
}
```

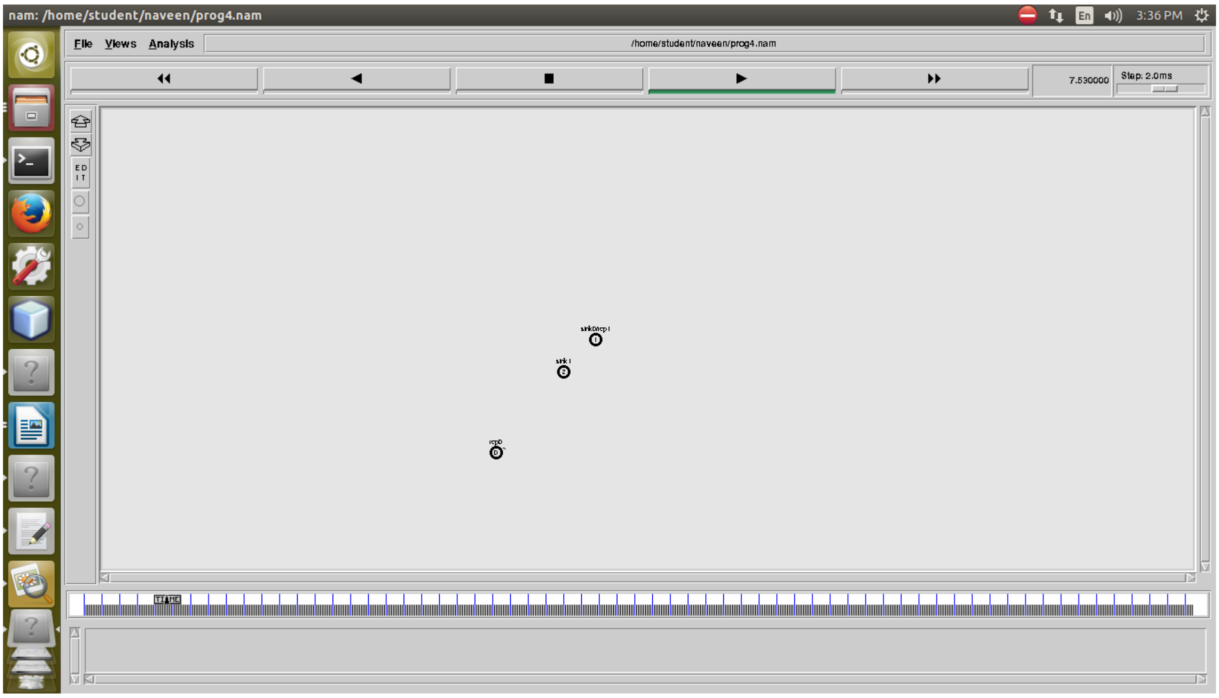
## Output

### Steps for execution

1. Open gedit and type program. Program name should have the extension " .tcl
   **student@cnpc022:~/naveen$ gedit prog4.tcl**
2. Save the program.
3. Open gedit and type program. Program name should have the extension " .awk
   **student@cnpc022:~/naveen$ gedit prog4.awk**
4. Save the program.
5. Run the simulation program
   **student@cnpc022:~/naveen$ ns prog4.tcl**
6. Here "ns" indicates network simulator. We get the topology shown in the snapshot.
7. Now press the play button in the simulation window and the simulation will begins.
8. After simulation is completed run awk file to see the output , **student@cnpc022:~/naveen$ awk –f prog4.awk prog4.tr**
   **The Throughput from n0 to n1: 5444Mbps**
   **The Throughput from n1 to n2: 345Mbps**
9. To see the trace file contents open the file as ,
   **student@cnpc022:~/naveen$ gedit prog4.tr**

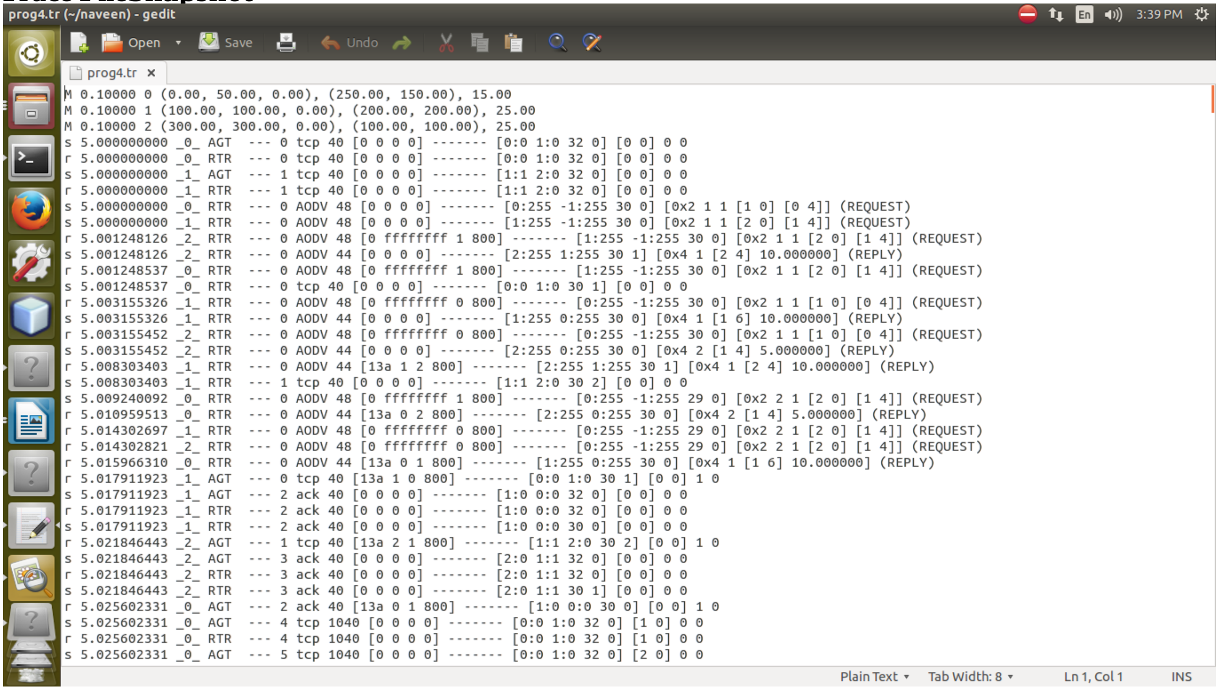
## Topology

### Snapshot 1:



### Snapshot 2:

### Trace FileSnapshot



```
M 0.10000 0 (0.00, 50.00, 0.00), (250.00, 150.00), 15.00
M 0.10000 1 (100.00, 100.00, 0.00), (200.00, 200.00), 25.00
M 0.10000 2 (300.00, 300.00, 0.00), (100.00, 100.00), 25.00
s 5.000000000 _0_ AGT  --- 0 tcp 40 [0 0 0 0] ------- [0:0 1:0 32 0] [0 0] 0 0
r 5.000000000 _0_ RTR  --- 0 tcp 40 [0 0 0 0] ------- [0:0 1:0 32 0] [0 0] 0 0
s 5.000000000 _1_ AGT  --- 1 tcp 40 [0 0 0 0] ------- [1:1 2:0 32 0] [0 0] 0 0
r 5.000000000 _1_ RTR  --- 1 tcp 40 [0 0 0 0] ------- [1:1 2:0 32 0] [0 0] 0 0
s 5.000000000 _0_ RTR  --- 0 AODV 48 [0 0 0 0] ------- [0:255 -1:255 30 0] [0x2 1 1 [1 0] [0 4]] (REQUEST)
s 5.000000000 _1_ RTR  --- 0 AODV 48 [0 0 0 0] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
r 5.001248126 _2_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
s 5.001248126 _2_ RTR  --- 0 AODV 44 [0 0 0 0] ------- [2:255 1:255 30 1] [0x4 1 [2 4] 10.000000] (REPLY)
r 5.001248537 _0_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 4]] (REQUEST)
s 5.001248537 _0_ RTR  --- 0 tcp 40 [0 0 0 0] ------- [0:0 1:0 30 1] [0 0] 0 0
r 5.003155326 _1_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [1 0] [0 4]] (REQUEST)
s 5.003155326 _1_ RTR  --- 0 AODV 44 [0 0 0 0] ------- [1:255 0:255 30 0] [0x4 1 [1 6] 10.000000] (REPLY)
r 5.003155452 _2_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [1 0] [0 4]] (REQUEST)
s 5.003155452 _2_ RTR  --- 0 AODV 44 [0 0 0 0] ------- [2:255 0:255 30 0] [0x4 2 [1 4] 5.000000] (REPLY)
r 5.008303403 _1_ RTR  --- 0 AODV 44 [13a 1 2 800] ------- [2:255 1:255 30 1] [0x4 1 [2 4] 10.000000] (REPLY)
s 5.008303403 _1_ RTR  --- 1 tcp 40 [0 0 0 0] ------- [1:1 2:0 30 2] [0 0] 0 0
s 5.009240092 _0_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [0:255 -1:255 29 0] [0x2 2 1 [2 0] [1 4]] (REQUEST)
r 5.010959513 _0_ RTR  --- 0 AODV 44 [13a 0 2 800] ------- [2:255 0:255 30 0] [0x4 2 [1 4] 5.000000] (REPLY)
r 5.014302697 _1_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 29 0] [0x2 2 1 [2 0] [1 4]] (REQUEST)
r 5.014302821 _2_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 29 0] [0x2 2 1 [2 0] [1 4]] (REQUEST)
r 5.015966310 _0_ RTR  --- 0 AODV 44 [13a 0 1 800] ------- [1:255 0:255 30 0] [0x4 1 [1 6] 10.000000] (REPLY)
r 5.017911923 _1_ AGT  --- 0 tcp 40 [13a 1 0 800] ------- [0:0 1:0 30 1] [0 0] 1 0
s 5.017911923 _1_ AGT  --- 2 ack 40 [0 0 0 0] ------- [1:0 0:0 32 0] [0 0] 0 0
r 5.017911923 _1_ RTR  --- 2 ack 40 [0 0 0 0] ------- [1:0 0:0 32 0] [0 0] 0 0
s 5.017911923 _1_ RTR  --- 2 ack 40 [0 0 0 0] ------- [1:0 0:0 30 0] [0 0] 0 0
r 5.021846443 _2_ AGT  --- 1 tcp 40 [13a 2 1 800] ------- [1:1 2:0 30 2] [0 0] 1 0
s 5.021846443 _2_ AGT  --- 3 ack 40 [0 0 0 0] ------- [2:0 1:1 32 0] [0 0] 0 0
r 5.021846443 _2_ RTR  --- 3 ack 40 [0 0 0 0] ------- [2:0 1:1 32 0] [0 0] 0 0
s 5.021846443 _2_ RTR  --- 3 ack 40 [0 0 0 0] ------- [2:0 1:1 30 1] [0 0] 0 0
r 5.025602331 _0_ AGT  --- 2 ack 40 [13a 0 1 800] ------- [1:0 0:0 30 0] [0 0] 1 0
s 5.025602331 _0_ AGT  --- 4 tcp 1040 [0 0 0 0] ------- [0:0 1:0 32 0] [1 0] 0 0
r 5.025602331 _0_ RTR  --- 4 tcp 1040 [0 0 0 0] ------- [0:0 1:0 32 0] [1 0] 0 0
s 5.025602331 _0_ AGT  --- 5 tcp 1040 [0 0 0 0] ------- [0:0 1:0 32 0] [2 0] 0 0
```

**Program Outcome**

- Simulate and implement a simple ESS in wireless LAN.

**Viva Questions**

What is ESS? Explain in detail
What is simulation?

# Program 3.Write a program for error detecting code using CRC-CCITT (16- bits).

## Program Objective:

- Understand the operation of CRC-CCITT.

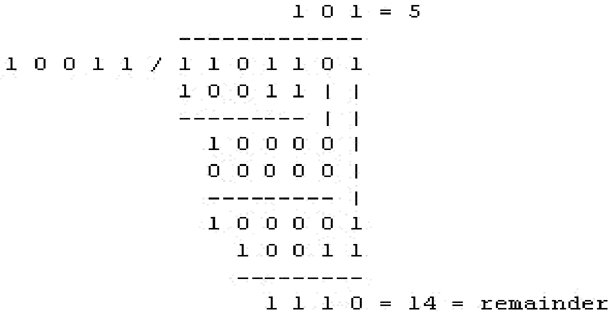
Whenever digital data is stored or interfaced, data corruption might occur. Since the beginning of computer science, developers have been thinking of ways to deal with this type of problem. For serial data they came up with the solution to attach a parity bit to each sent byte. This simple detection mechanism works if an odd number of bits in a byte changes, but an even number of false bits in one byte will not be detected by the parity check.

To overcome this problem developers have searched for mathematical sound mechanisms to detect multiple false bits. The **CRC** calculation or *cyclic redundancy check* was the result of this. Nowadays CRC calculations are used in all types of communications. All packets sent over a network connection are checked with a CRC. Also each data block on your hard disk has a CRC value attached to it.

Modern computer world cannot do without these CRC calculations. So let's see why they are so widely used. The answer is simple; they are powerful, detect many types of errors and are extremely fast to calculate especially when dedicated hardware chips are used.

The idea behind CRC calculation is to look at the data as one large binary number. This number is divided by a certain value and the remainder of the calculation is called the CRC. Dividing in the CRC calculation at first looks to cost a lot of computing power, but it can be performed very quickly if we use a method similar to the one learned at school. We will as an example calculate the remainder for the character 'm'—which is 1101101 in binary notation— by dividing it by 19 or 10011. Please note that 19 is an odd number.

This is necessary as we will see further on. Please refer to your schoolbooks as the binary calculation method here is not very different from the decimal method you learned when you were young. It might only look a little bit strange. Also notations differ between countries, but the method is similar.



```
                    1 0 1 = 5
            -------------
1 0 0 1 1 / 1 1 0 1 1 0 1
            1 0 0 1 1 | |
            --------- | |
              1 0 0 0 0 |
              0 0 0 0 0 |
              --------- |
              1 0 0 0 0 1
                1 0 0 1 1
                ---------
                  1 1 1 0 = 14 = remainder
```

With decimal calculations you can quickly check that 109 divided by 19 gives a quotient of 5 with 14 as the remainder. But what we also see in the scheme is that every bit extra to check only costs one binary comparison and in 50% of the cases one binary subtraction.

You can easily increase the number of bits of the test data string—for example to 56 bits if we use our example value "*Lammert*"—and the result can be calculated with 56 binary comparisons and an average of 28 binary subtractions. This can be implemented in hardware directly with only very few transistors involved. Also software algorithms can be very efficient.

All of the CRC formulas you will encounter are simply checksum algorithms based on modulo-2 binary division where we ignore carry bits and in effect the subtraction will be equal to an *exclusive or* operation. Though some differences exist in the specifics across different CRC formulas, the basic mathematical process is always the same:

- The message bits are appended with *c* zero bits; this *augmented message* is the dividend
- A predetermined *c+1*-bit binary sequence, called the *generator polynomial*, is the divisor
- The checksum is the *c*-bit remainder that results from the division operation

Table 1 lists some of the most commonly used generator polynomials for 16- and 32-bit CRCs. Remember that the width of the divisor is always one bit wider than the remainder. So, for example, you'd use a 17-bit generator polynomial whenever a 16-bit checksum is required.

| | **CRC-CCITT** | **CRC-16** | **CRC-32** |
|---|---|---|---|
| Checksum Width | 16 bits | 16 bits | 32 bits |
| Generator Polynomial | 10001000000100001 | 11000000000000101 | 100000100110000010001110110110111 |

International Standard CRC Polynomials

<u>Algorithm:-</u>

1. Given a bit string, append $0^S$to the end of it (the number of $0^s$is the same as the degree of the generator polynomial) let B(x) be the polynomial corresponding to B.

2. Divide B(x) by some agreed on polynomial G(x) (generator polynomial) and determine the remainder R(x). This division is to be done using Modulo 2 Division.

3. Define T(x) = B(x)–R(x)
   (T(x)/G(x) => remainder 0)

4. Transmit T, the bit string corresponding toT(x).
   Let T' represent the bit stream the receiver gets and T'(x) the associated polynomial. The receiver divides T1(x) by G(x). If there is a 0 remainder, the receiver concludes T = T' and no error occurred otherwise, the receiver concludes an error occurred and requires a retransmission

```
/* CRC */
import java.util.*;
```

```
public class Crc
{
   void div(int a[],int k)
{
int gp[]={1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1};
int count=0;
for(int i=0;i<k;i++)
{
if(a[i]==gp[0])
{
for(int j=i;j<17+i;j++)
{
a[j]=a[j]^gp[count++];
}
count=0;
}
}
}
public static void main(String args[])
{
int a[]=new int[100];
int b[]=new int[100];
int len,k;
Crc ob=new Crc();
System.out.println("Enter the length of Data Frame:");
Scanner sc=new Scanner(System.in);
len=sc.nextInt();
int flag=0;
System.out.println("Enter the Message:");
for(int i=0;i<len;i++)
{
a[i]=sc.nextInt();
}
for(int i=0;i<16;i++)
{
a[len++]=0;
}
k=len-16;
for(int i=0;i<len;i++)
{
b[i]=a[i];
}
ob.div(a,k);
for(int i=0;i<len;i++)
a[i]=a[i]^b[i];
System.out.println("Data to be transmitted: ");
for(int i=0;i<len;i++)
{
   System.out.print(a[i]+" ");
}
System.out.println();
System.out.println("Enter the Reveived Data: ");
for(int i=0;i<len;i++)
{
a[i]=sc.nextInt();
}
ob.div(a, k);
for(int i=0;i<len;i++)
{
if(a[i]!=0)
{
flag=1;
break;
}
}
if(flag==1)
System.out.println("error in data");
else
System.out.println("no error");
}
}
```

**Output1**

```
Enter the length of Data Frame:
5
Enter the Message:
1 1 1 0 1
Data to be transmitted:
1 1 1 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 1 0 0
Enter the Reveived Data:
1 1 1 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 1 0 0
no error
```

**Output2**

```
Enter the length of Data Frame:
5
Enter the Message:
1 1 1 0 1
Data to be transmitted:
1 1 1 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 1 0 0
Enter the Reveived Data:
1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 0 1 1 1 0 0
error in data
```

## Program Outcome

- Identify and apply the operation of CRC-CCITT.

## Viva Questions:

- Explain the features of JAVA?
- What is CRC-CCITT(16bits)?
- How CRC will detect error in a program?

## Program No. 4: Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion in the network.

**Program Objective:**

- Understand the Implementation of the network topology consisting of n nodes.

```
set ns [new Simulator]
set ntrace [open prog2.tr w]
$ns trace-all $ntrace
set namfile [open prog2.nam w]
$ns namtrace-all $namfile

proc finish {} {
global ns ntrace namfile
$ns flush-trace
close $ntrace
close $namfile
exec nam prog2.nam &
puts "the number of ping packets dropped are"
exec grep -c "^d" prog2.tr &
exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]

$ns duplex-link $n1 $n0 1Mb 120ms DropTail
$ns duplex-link $n2 $n0 1Mb 10ms DropTail
$ns duplex-link $n3 $n0 1Mb 10ms DropTail
$ns duplex-link $n4 $n0 1Mb 10ms DropTail
$ns duplex-link $n5 $n0 1Mb 10ms DropTail
$ns duplex-link $n6 $n0 1Mb 11ms DropTail

Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts "node [$node_ id] received ping answer from $from round-trip-time $rtt ms"
}
set p1 [new Agent/Ping]
set p2 [new Agent/Ping]
set p3 [new Agent/Ping]
set p4 [new Agent/Ping]
set p5 [new Agent/Ping]
set p6 [new Agent/Ping]
```

```
$ns attach-agent $n1 $p1

$ns attach-agent $n2 $p2
$ns attach-agent $n3 $p3
$ns attach-agent $n4 $p4
$ns attach-agent $n5 $p5
$ns attach-agent $n6 $p6

$ns queue-limit $n0 $n4 3
$ns queue-limit $n0 $n5 1
$ns queue-limit $n0 $n6 1
$ns connect $p1 $p4
$ns connect $p2 $p5
$ns connect $p3 $p6

$ns at 0.1 "$p1 send"
$ns at 0.3 "$p2 send"
$ns at 0.5 "$p3 send"
$ns at 1.0 "$p4 send"
$ns at 1.2 "$p5 send"
$ns at 1.4 "$p6 send"
$ns at 2.0 "finish"
$ns run
```

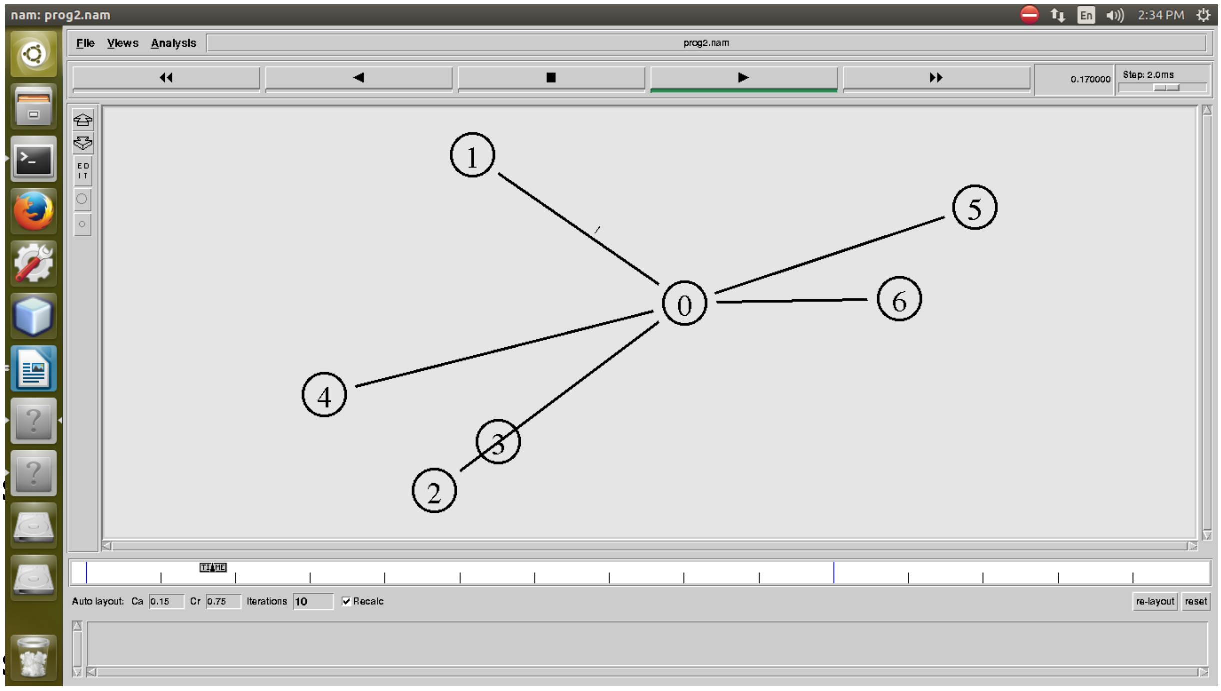
**<u>Output:</u>**

**Steps for execution**

1. Open gedit and type program. Program name should have the extension “ .tcl
   **student@cnpc022:~/naveen$ gedit prog2.tcl**
2. Save the program.
3. Run the simulation program
   ```
   student@cnpc022:~/naveen$ ns prog2.tcl
   node 1 received ping answer from 4 round-trip-time 262.0 ms
   node 4 received ping answer from 1 round-trip-time 262.0 ms
   the number of ping packets dropped are
   4
   ```
4. Here “ns” indicates network simulator. We get the topology shown in the snapshot.
5. Now press the play button in the simulation window and the simulation will begin.
6. To see the trace file contents open the file as ,
   **student@cnpc022:~/naveen$ gedit prog2.tr**

# Topology

## Snapshot 1:



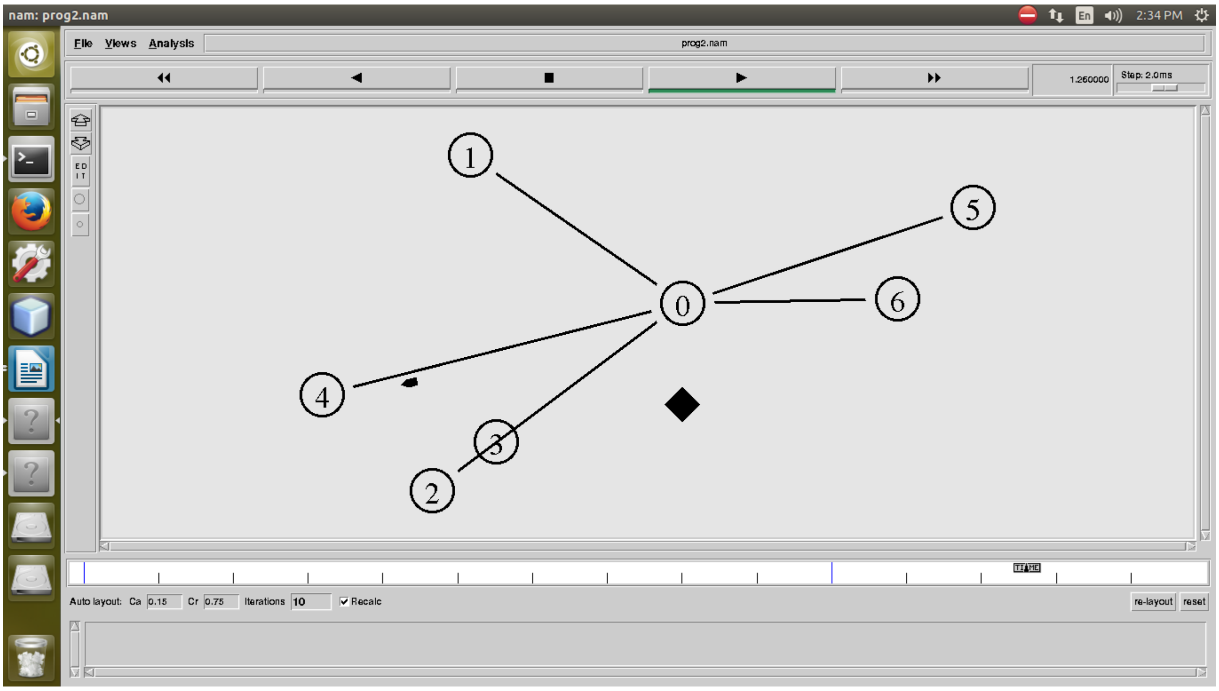
nam: prog2.nam

prog2.nam

0.170000 Step: 2.0ms

Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

## Snapshot 2:



nam: prog2.nam

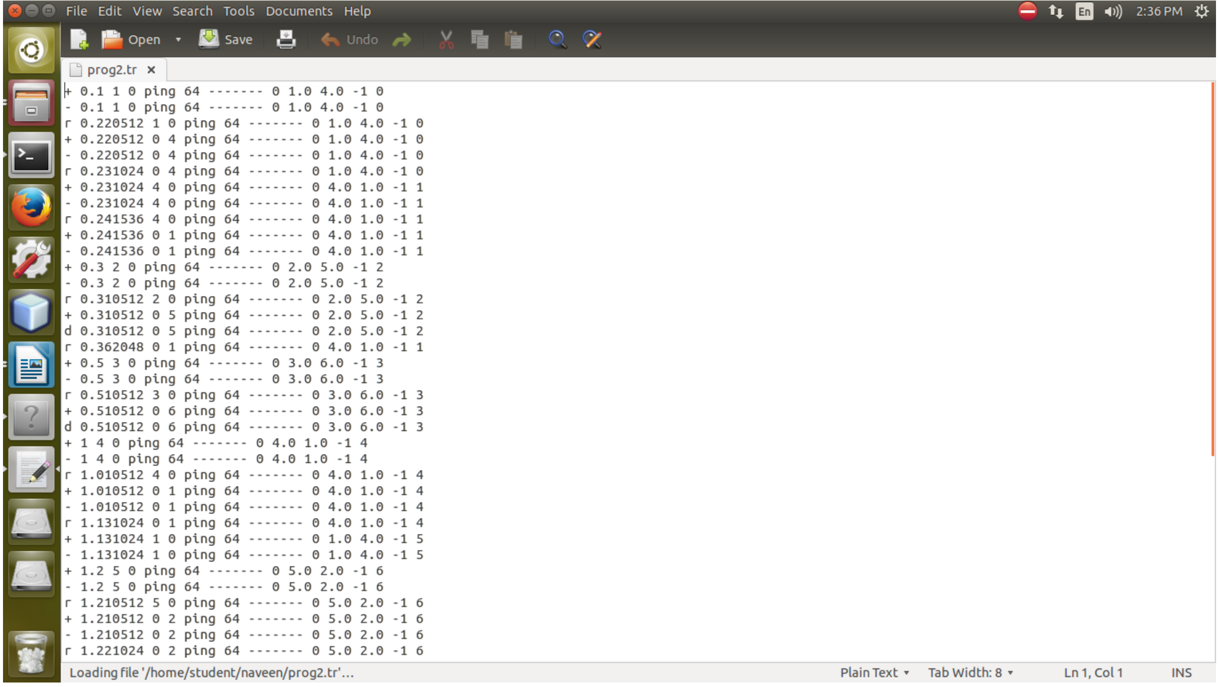
prog2.nam

1.260000 Step: 2.0ms

Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

## Snapshot 3:

### Trace FileSnapshot



```
+ 0.1 1 0 ping 64 ------- 0 1.0 4.0 -1 0
- 0.1 1 0 ping 64 ------- 0 1.0 4.0 -1 0
r 0.220512 1 0 ping 64 ------- 0 1.0 4.0 -1 0
+ 0.220512 0 4 ping 64 ------- 0 1.0 4.0 -1 0
- 0.220512 0 4 ping 64 ------- 0 1.0 4.0 -1 0
r 0.231024 0 4 ping 64 ------- 0 1.0 4.0 -1 0
+ 0.231024 4 0 ping 64 ------- 0 4.0 1.0 -1 1
- 0.231024 4 0 ping 64 ------- 0 4.0 1.0 -1 1
r 0.241536 4 0 ping 64 ------- 0 4.0 1.0 -1 1
+ 0.241536 0 1 ping 64 ------- 0 4.0 1.0 -1 1
- 0.241536 0 1 ping 64 ------- 0 4.0 1.0 -1 1
+ 0.3 2 0 ping 64 ------- 0 2.0 5.0 -1 2
- 0.3 2 0 ping 64 ------- 0 2.0 5.0 -1 2
r 0.310512 2 0 ping 64 ------- 0 2.0 5.0 -1 2
+ 0.310512 0 5 ping 64 ------- 0 2.0 5.0 -1 2
d 0.310512 0 5 ping 64 ------- 0 2.0 5.0 -1 2
r 0.362048 0 1 ping 64 ------- 0 4.0 1.0 -1 1
+ 0.5 3 0 ping 64 ------- 0 3.0 6.0 -1 3
- 0.5 3 0 ping 64 ------- 0 3.0 6.0 -1 3
r 0.510512 3 0 ping 64 ------- 0 3.0 6.0 -1 3
+ 0.510512 0 6 ping 64 ------- 0 3.0 6.0 -1 3
d 0.510512 0 6 ping 64 ------- 0 3.0 6.0 -1 3
+ 1 4 0 ping 64 ------- 0 4.0 1.0 -1 4
- 1 4 0 ping 64 ------- 0 4.0 1.0 -1 4
r 1.010512 4 0 ping 64 ------- 0 4.0 1.0 -1 4
+ 1.010512 0 1 ping 64 ------- 0 4.0 1.0 -1 4
- 1.010512 0 1 ping 64 ------- 0 4.0 1.0 -1 4
r 1.131024 0 1 ping 64 ------- 0 4.0 1.0 -1 4
+ 1.131024 1 0 ping 64 ------- 0 1.0 4.0 -1 5
- 1.131024 1 0 ping 64 ------- 0 1.0 4.0 -1 5
+ 1.2 5 0 ping 64 ------- 0 5.0 2.0 -1 6
- 1.2 5 0 ping 64 ------- 0 5.0 2.0 -1 6
r 1.210512 5 0 ping 64 ------- 0 5.0 2.0 -1 6
+ 1.210512 0 2 ping 64 ------- 0 5.0 2.0 -1 6
- 1.210512 0 2 ping 64 ------- 0 5.0 2.0 -1 6
r 1.221024 0 2 ping 64 ------- 0 5.0 2.0 -1 6
```

Loading file '/home/student/naveen/prog2.tr'...

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 INS

## Program Outcome :

- Implement network topology consisting of n nodes.

## Viva Questions:

- What is network?
- Define Topology
- Explain different types of Topology
- What is congestion?

## Program 5:Write a program to find the shortest path between vertices using bellman-ford algorithm.

**Program Objective:**

- Understand the Implementation of the shortest path for bellman-ford algorithm.

Distance Vector Algorithm is a decentralized routing algorithm that requires that each router simply inform its neighbors of its routing table. For each network path, the receiving routers pick the neighbor advertising the lowest cost, then add this entry into its routing table for re-advertisement. To find the shortest path, Distance Vector Algorithm is based on one of two basic algorithms: the Bellman-Ford and the Dijkstra algorithms.

Routers that use this algorithm have to maintain the distance tables (which is a one-dimension array -- "a vector"), which tell the distances and shortest path to sending packets to each node in the network. The information in the distance table is always up date by exchanging information with the neighboring nodes. The number of data in the table equals to that of all nodes in networks (excluded itself).

The columns of table represent the directly attached neighbors whereas the rows represent all destinations in the network. Each data contains the path for sending packets to each destination in the network and distance/or time to transmit on that path (we call this as "cost"). The measurements in this algorithm are the number of hops, latency, the number of outgoing packets, etc.

The Bellman–Ford algorithm is an algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph. It is slower than Dijkstra'salgorithm for the same problem, but more versatile, as it is capable of handling graphs in which some of the edge weights are negative numbers. Negative edge weights are found in various applications of graphs, hence the usefulness of this algorithm.

If a graph contains a "negative cycle" (i.e. a cycle whose edges sum to a negative value) that is reachable from the source, then there is no cheapest path: any path that has a point on the negative cycle can be made cheaper by one more walk around the negative cycle. In such a case, the Bellman–Ford algorithm can detect negative cycles and report their existence

**Implementation Algorithm:**

1. send my routing table to all my neighbors whenever my link tablechanges
2. when I get a routing table from a neighbor on port P with link metricM:
   a. add L to each of the neighbor'smetrics
   b. for each entry (D, P', M') in the updated neighbor'stable:
      i. if I do not have an entry for D, add (D, P, M') to my routingtable
      ii. if I have an entry for D with metric M", add (D, P, M') to my routing table if M' <M"
3. if my routing table has changed, send all the new entries to all myneighbor

**/* Bellman-Ford */**

```
import java.util.*;
public class Belmanford
 {
   private  int D[];
   private  int n;
   public static final int max_value=999;
   public Belmanford(int n)
   {
      this.n=n;
      D=new int[n+1];
   }
   public void shortest(int s,int a[][])
   {
      for(int i=1;i<=n;i++)
      {
         D[i]=max_value;
      }
      D[s]=0;
      for(int k=1;k<=n-1;k++)
      {
         for(int i=1;i<=n;i++)
         {
            for(int j=1;j<=n;j++)
            {
               if(a[i][j]!=max_value)
               {
                  if(D[j]>D[i]+a[i][j])
                     D[j]=D[i]+a[i][j];
               }
            }
         }
      }
      for (int i=1;i<=n;i++)
      {
         for (int j=1;j<=n;j++)
         {
          if(a[i][j]!=max_value)
          {
              if(D[j]>D[i]+a[i][j])
              {
                 System.out.println("the graph contains -ve edge cycle");
                 return;
              }
          }
         }
      }
      for (int i=1;i<=n;i++)
      {
         System.out.println("distance of source"+s+"to"+i+"is"+D[i]);
      }
   }
```

```
    public static void main(String[] args)
{
        int n=0,s;
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the no.of values");
        n=sc.nextInt();
        int a[][]=new int [n+1][n+1];
        System.out.println("enter the weighted matrix:");
        for (int i=1;i<=n;i++)
        {
            for (int j=1;j<=n;j++)
            {
                a[i][j]=sc.nextInt();
                if(i==j)
                {
                    a[i][j]=0;
                    continue;
                }
                if(a[i][j]==0)
                    a[i][j]=max_value;
            }
        }
      System.out.println("enter the source vertex:");
      s=sc.nextInt();
      Belmanford b=new Belmanford(n);
      b.shortest(s,a);
      sc.close();
      }

}
```

**Output1**

```
enter the no.of values
4
enter the weighted matrix:
0 999 999 999
5 0 3 4
999 999 0 2
999 999 999 0
enter the source vertex:
2
distance of source 2 to1 is 5
distance of source2  to 2 is 0
distance of source 2 to 3 is 3
distance of source2 to 4 is 4
```

**Output2:**

```
enter the no.of values
4
enter the weighted matrix:
```

```
0 4 999 5
999 0 999 999

999 -10 0 999
999 999 3 0
enter the source vertex:
1
distance of source 1to 1 is 0
distance of source1 to 2 is-2
distance of source1 to 3 is 8
distance of source 1 to 4 is 5
```

**Output3**

```
enter the no.of values
4
enter the weighted matrix:
0 4 5 999
999 0 999 7
999 7 0 999
999 999 -15 0
enter the source vertex:
1
the graph contains -ve edge cycle
```

## Program Outcome

- Implement the shortest path for bellman-ford algorithm.

## Viva Questions:

- What is bellman ford algorithm?
- What are the advantages and applications of bell man ford algorithm?

## Program No. 6:Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.

**Program Objective:**

- Understand the Implementation of the Ethernet LAN using n nodes.

```
#Create Simulator
set ns [new Simulator]
#Use colors to differentiate the traffics
$ns color 1 Blue
$ns color 2 Red
set ntrace [open prog5.tr w]
$ns trace-all $ntrace
set namfile [open prog5.nam w]
$ns namtrace-all $namfile
#Use some flat file to create congestion graph windows
set File0 [open WinFile0 w]
set File1 [open WinFile1 w]

#Finish Procedure
proc Finish {} {
global ns ntrace namfile
$ns flush-trace
close $ntrace
close $namfile
exec nam prog5.nam &
#Plot the Congestion Window graph using xgraph
exec xgraph WinFile0 WinFile1 &
exit 0
}

#Plot Window Procedure
proc PlotWindow {tcpSource file} {
global ns
set time 10.0
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "PlotWindow $tcpSource $file"
}
#Create 6 nodes
for {set i 0} {$i<6} {incr i} {
set n($i) [$ns node]
}
#Create duplex links between the nodes
$ns duplex-link $n(0) $n(2) 2Mb 10ms DropTail
$ns duplex-link $n(1) $n(2) 2Mb 10ms DropTail
$ns duplex-link $n(2) $n(3) 0.6Mb 100ms DropTail
#Nodes n(3) , n(4) and n(5) are considered in a LAN
```

```
#Creates a Lan from a set of nodes, Bandwidth, delay characteristics along with link layer,
#interface queue, Mac Layer and channel type for the LAN are defined

set lan [$ns newLan "$n(3) $n(4) $n(5)" 0.5Mb 40ms LL Queue/DropTail MAC/802_3 Channel]

#Orientation to the nodes
$ns duplex-link-op $n(0) $n(2) orient right-down
$ns duplex-link-op $n(1) $n(2) orient right-up
$ns duplex-link-op $n(2) $n(3) orient right

#Setup queue between n(2) and n(3) and monitor the queue
$ns queue-limit $n(2) $n(3) 20

#Set error model on link n(2) to n(3)
set loss_module [new ErrorModel]
$loss_module ranvar [new RandomVariable/Uniform]
$loss_module drop-target [new Agent/Null]
$ns lossmodel $loss_module $n(2) $n(3)

#Set up the TCP connection between n(0) and n(4)
set tcp0 [new Agent/TCP/Newreno]
$tcp0 set fid_ 1
$tcp0 set window_ 8000
$tcp0 set packetSize_ 552
$ns attach-agent $n(0) $tcp0

set sink0 [new Agent/TCPSink/DelAck]
$ns attach-agent $n(4) $sink0
$ns connect $tcp0 $sink0

#Apply FTP Application over TCP
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ftp0 set type_ FTP

#Set up another TCP connection between n(5) and n(1)

set tcp1 [new Agent/TCP/Newreno]
$tcp1 set fid_ 2
$tcp1 set window_ 8000
$tcp1 set packetSize_ 552
$ns attach-agent $n(5) $tcp1

set sink1 [new Agent/TCPSink/DelAck]
$ns attach-agent $n(1) $sink1
$ns connect $tcp1 $sink1

#Apply FTP application over TCP
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ftp1 set type_ FTP
#Schedule Events
```

```
$ns at 0.1 "$ftp0 start"
$ns at 0.1 "PlotWindow $tcp0 $File0"
$ns at 0.5 "$ftp1 start"
$ns at 0.5 "PlotWindow $tcp1 $File1"
$ns at 25.0 "$ftp0 stop"
$ns at 25.1 "$ftp1 stop"
$ns at 25.2 "Finish"
#Run the simulation
$ns run
```

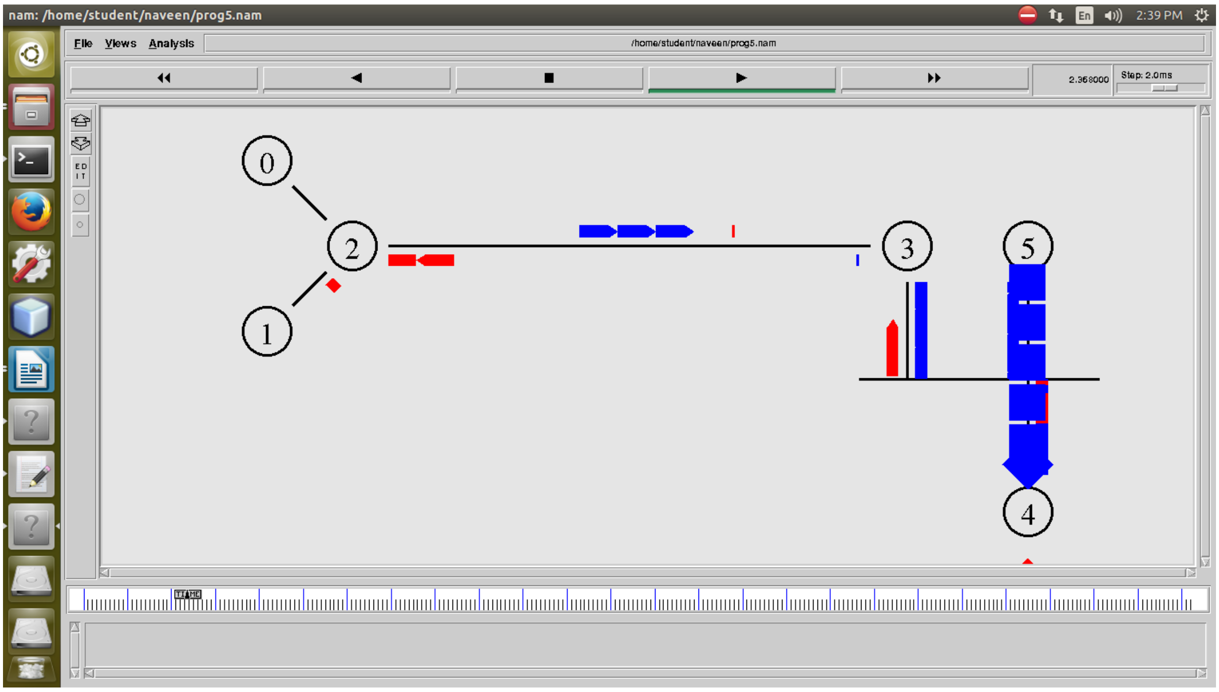
## Output

## Steps for execution

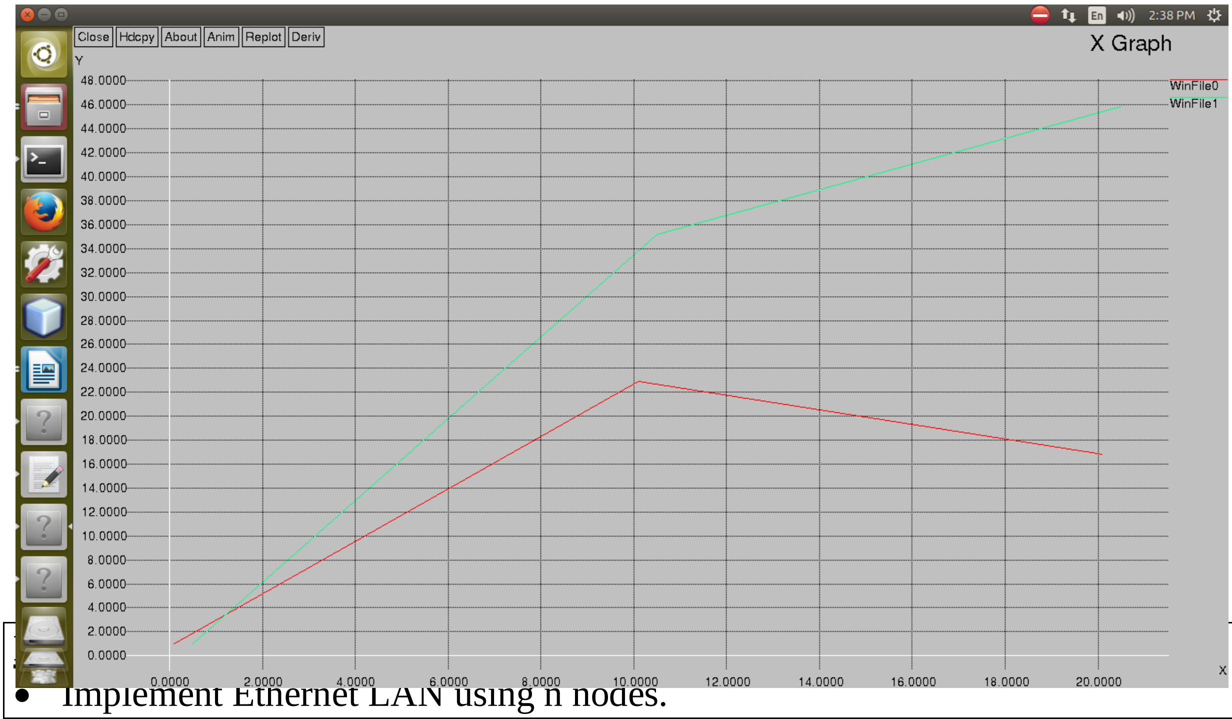
1. Open gedit and type program. Program name should have the extension " .tcl  
   **student@cnpc022:~/naveen$ gedit prog3.tcl**
2. Save the program.
3. Run the simulation program  
   **student@cnpc022:~/naveen$ ns prog3.tcl**
4. Here "ns" indicates network simulator. We get the topology shown in the snapshot.
5. Now press the play button in the simulation window and the simulation will begin.
6. To see the trace file contents open the file as ,  
   **student@cnpc022:~/naveen$ gedit prog3.tr**

## Topology

### Snapshot 1:


nam: /home/student/naveen/prog5.nam
2:39 PM
File Views Analysis
/home/student/naveen/prog5.nam
2.368000
Step: 2.0ms
0
1
2
3
4
5

### Snapshot 2:



- Implement Ethernet LAN using n nodes.

## Viva Questions

- What is LAN? Explain different types of network?
- What do you mean  by congestion window?

is modified and converted to upper case in UDPServer Program and displayed on UDPServer side program.Below is output shown.

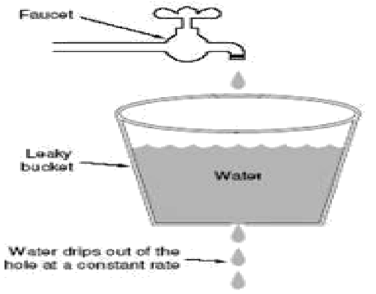
# Program 7:Write a program for congestion control using leaky bucket algorithm.

**Program Objective:**

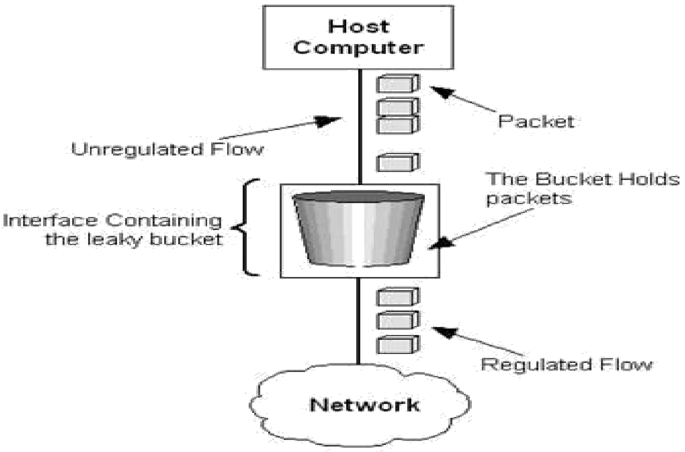
- Understand the operation of congestion control using leaky bucket algorithm.

The main concept of the leaky bucket algorithm is that the output data flow remains constant despite the variant input traffic, such as the water flow in a bucket with a small hole at the bottom. In case the bucket contains water (or packets) then the output flow follows a constant rate, while if the bucket is full any additional load will be lost because of spillover. In a similar way if the bucket is empty the output will be zero.

From network perspective, leaky bucket consists of a finite queue (bucket) where all the incoming packets are stored in case there is space in the queue, otherwise the packets are discarded. In order to regulate the output flow, leaky bucket transmits one packet from the queue in a fixed time (e.g. at every clock tick). In the following figure we can notice the main rationale of leaky bucket algorithm, for both the two approaches (e.g. leaky bucket with water (a) and with packets (b)).



(a) A leaky bucket with water



(b) A leaky bucket with packets

While leaky bucket eliminates completely bursty traffic by regulating the incoming data flow its main drawback is that it drops packets if the bucket is full. Also, it doesn't take into account the idle process of the sender which means that if the host doesn't transmit data forsome time the bucket becomes empty without permitting the transmission of any packet.

**The leaky-bucket algorithm:**

The algorithm can be conceptually understood as follows:

- Consider a bucket with a hole in thebottom.
- The empty space of the bucket represents an amount of credit available measured in bytes.

- The size of the bucket is *b* bytes. This means that if the bucket is empty, *b* bytes of credit is available.
- If a packet arrives and its size is less than the available credit, the packet can be forwarded. Otherwise, it is discarded or queued depending on theapplication.
- The bucket leaks through the hole in its bottom at a constant rate of *r* bytes per second, this indicates creditaccumulation.

## /* Leaky Bucket */

```
public class LeakyBucket
{
  static int min(int x,int y)
{
if(x<y)
return x;
else
return y;
}
public static void main(String[] args)
{
 int drop=0,mini,nsec,cap,count=0,i,process;
int inp[]=new int[25];
Scanner sc=new Scanner(System.in);
System.out.println("Enter The Bucket Size\n");
cap= sc.nextInt();
System.out.println("Enter The Operation Rate\n");
process= sc.nextInt();
System.out.println("Enter The No. Of Seconds You Want To Stimulate\n");
nsec=sc.nextInt();
for(i=0;i<nsec;i++)
{
System.out.print("Enter The Size Of The Packet Entering At "+ i+1+"sec");
inp[i] = sc.nextInt();
}
System.out.println("\nSecond | Packet Recieved | Packet Sent | Packet Left | Packet Dropped|\n");
//System.out.println("--------------------------------------------------------------------\n");
for(i=0;i<nsec;i++)
{
count+=inp[i];
if(count>cap)
{
drop=count-cap;
 count=cap;
 }
System.out.print(i+1);
System.out.print("\t\t"+inp[i]);
mini=min(count,process);
System.out.print("\t\t"+mini);
count=count-mini;
System.out.print("\t\t"+count);
System.out.print("\t\t"+drop);
drop=0;
System.out.println();
}
for(;count!=0;i++)
{
if(count>cap)
{
drop=count-cap;
count=cap;
}
System.out.print(i+1);
System.out.print("\t\t0");
mini=min(count,process);
System.out.print("\t\t"+mini);
count=count-mini;
System.out.print("\t\t"+count);
System.out.print("\t\t"+drop);
System.out.println();
}
}
}
```

**Output1**

```
Enter The Bucket Size
6
Enter The output Rate
2
Enter The No. of Seconds You Want To Stimulate
2
Enter The Size of Packet entering at 01sec
8
Enter The Size of Packet entering at 11sec
6
Second | Packet Recieved | Packet Sent | Packet Left | Packet Dropped|
1               8               2               4               2
2               6               2               4               4
3               0               2               2               0
4               0               2               0               0
```

**Output2**

```
Enter The Bucket Size
5
Enter The output Rate
2
Enter The No. of Seconds You Want To Stimulate
3
Enter The Size of Packet entering at 01sec
5
Enter The Size of Packet entering at 11sec
4
Enter The Size of Packet entering at 21sec
```

```
3

Second | Packet Recieved | Packet Sent | Packet Left | Packet Dropped|
1               5               2               3               0
2               4               2               3               2
3               3               2               3               1
4               0               2               1               0
5               0               1               0               0
```

## Program Outcome

- Identify and apply the operation of congestion control using leaky bucket algorithm.

## Viva Questions:

- What is congestion control?
- Explain leaky bucket algorithm

# VIVA QUESTIONS AND ANSWERS

**1)What is a Link?**

A link refers to the connectivity between two devices. It includes the type of cables and protocols used in order for one device to be able to communicate with the other.

**2)What are the layers of the OSI reference model?**

There are 7 OSI layers: Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer and Application Layer.

**3)What is backbone network?**

A backbone network is a centralized infrastructure that is designed to distribute different routes and data to various networks. It also handles management of bandwidth and various channels.

**4)What is a LAN?**

LAN is short for Local Area Network. It refers to the connection between computers and other network devices that are located within a small physical location.

**5)What is a node?**

A node refers to a point or joint where a connection takes place. It can be computer or device that is part of a network. Two or more nodes are needed in order to form a network connection.

**6)What are routers?**

Routers can connect two or more network segments. These are intelligent network devices that store information in its routing table such as paths, hops and bottlenecks. With this info, they are able to determine the best path for data transfer. Routers operate at the OSI Network Layer.

**7)What is point to point link?**

It refers to a direct connection between two computers on a network. A point to point connection does not need any other network devices other than connecting a cable to the NIC cards of both computers.

**8)What is anonymous FTP?**

Anonymous FTP is a way of granting user access to files in public servers. Users that are allowed access to data in these servers do not need to identify themselves, but instead log in as an anonymous guest.

**9)What is subnet mask?**

A subnet mask is combined with an IP address in order to identify two parts: the extended network address and the host address. Like an IP address, a subnet mask is made up of 32 bits.

**10)What is the maximum length allowed for a UTP cable?**

A single segment of UTP cable has an allowable length of 90 to 100 meters. This limitation can be overcome by using repeaters and switches.

**11)What is data encapsulation?**

Data encapsulation is the process of breaking down information into smaller manageable chunks before it is transmitted across the network. It is also in this process that the source and destination addresses are attached into the headers, along with parity checks.

**12)Describe Network Topology**

Network Topology refers to the layout of a computer network. It shows how devices and cables are physically laid out, as well as how they connect to one another.

**13)What is VPN?**

VPN means Virtual Private Network, a technology that allows a secure tunnel to be created across a network such as the Internet. For example, VPNs allow you to establish a secure dialup connection to a remote server.

**14)Briefly describe NAT.**

NAT is Network Address Translation. This is a protocol that provides a way for multiple computers on a common network to share single connection to the Internet.

**15)What is the job of the Network Layer under the OSI reference model?**

The Network layer is responsible for data routing, packet switching and control of network congestion. Routers operate under this layer.

**16)How does a network topology affect your decision in setting up a network?**

Network topology dictates what media you must use to interconnect devices. It also serves as basis on what materials, connector and terminations that is applicable for the setup.

**17)What is RIP?**

RIP, short for Routing Information Protocol is used by routers to send data from one network to another. It efficiently manages routing data by broadcasting its routing table to all other routers within the network. It determines the network distance in units of hops.

**18)What are different ways of securing a computer network?**

There are several ways to do this. Install reliable and updated anti-virus program on all computers. Make sure firewalls are setup and configured properly. User authentication will also help a lot. All of these combined would make a highly secured network.

**19)What is NIC?**

NIC is short for Network Interface Card. This is a peripheral card that is attached to a PC in order to connect to a network. Every NIC has its own MAC address that identifies the PC on the network.

**20)What is WAN?**

WAN stands for Wide Area Network. It is an interconnection of computers and devices that are geographically dispersed. It connects networks that are located in different regions and countries.

**21)What is the importance of the OSI Physical Layer?**

The physical layer does the conversion from data bits to electrical signal, and vice versa. This is where network devices and cable types are considered and setup.

**22)How many layers are there under TCP/IP?**

There are four layers: the Network Layer, Internet Layer, Transport Layer and Application Layer.

**23)What are proxy servers and how do they protect computer networks?**

Proxy servers primarily prevent external users who identifying the IP addresses of an internal network. Without knowledge of the correct IP address, even the physical location of the network cannot be identified. Proxy servers can make a network virtually invisible to external users.

**24)What is the function of the OSI Session Layer?**

This layer provides the protocols and means for two devices on the network to communicate with each other by holding a session. This includes setting up the session, managing information exchange during the session, and tear-down process upon termination of the session.

**25)What is the importance of implementing a Fault Tolerance System? Are there limitations?**

A fault tolerance system ensures continuous data availability. This is done by eliminating a single point of failure. However, this type of system would not be able to protect data in some cases, such as in accidental deletions.

**26)What does 10Base-T mean?**

The 10 refers to the data transfer rate, in this case is 10Mbps. The word Base refers to base band, as oppose to broad band. T means twisted pair, which is the cable used for that network.

**27)What is a private IP address?**

Private IP addresses are assigned for use on intranets. These addresses are used for internal networks and are not routable on external public networks. These ensures that no conflicts are present among internal networks while at the same time the same range of private IP addresses are reusable for multiple intranets since they do not "see" each other.

**29)What is DoS?**

DoS, or Denial-of-Service attack, is an attempt to prevent users from being able to access the internet or any other network services. Such attacks may come in different forms and are done by a group of perpetuators. One common method of doing this is to overload the system server so it cannot anymore process legitimate traffic and will be forced to reset.

**30)What is OSI and what role does it play in computer networks?**

OSI (Open Systems Interconnect) serves as a reference model for data communication. It is made up of 7 layers, with each layer defining a particular aspect on how network devices connect and communicate with one another. One layer may deal with the physical media used, while another layer dictates how data is actually transmitted across the network.

**31)What is the purpose of cables being shielded and having twisted pairs?**

The main purpose of this is to prevent crosstalk. Crosstalks are electromagnetic interferences or noise that can affect data being transmitted across cables.

**32)What is the advantage of address sharing?**

By using address translation instead of routing, address sharing provides an inherent security benefit. That's because host PCs on the Internet can only see the public IP address of the external interface on the computer that provides address translation and not the private IP addresses on the internal network.

**33)What are MAC addresses?**

MAC, or Media Access Control, uniquely identifies a device on the network. It is also known as physical address or Ethernet address. A MAC address is made up of 6-byte parts.

**34)What is the equivalent layer or layers of the TCP/IP Application layer in terms of OSI reference model?**

The TCP/IP Application layer actually has three counterparts on the OSI model: the Session layer, Presentation Layer and Application Layer.

**35)How can you identify the IP class of a given IP address?**

By looking at the first octet of any given IP address, you can identify whether it's Class A, B or C. If the first octet begins with a 0 bit, that address is Class A. If it begins with bits 10 then that address is a Class B address. If it begins with 110, then it's a Class C network.

**36)What is the main purpose of OSPF?**

OSPF, or Open Shortest Path First, is a link-state routing protocol that uses routing tables to determine the best possible path for data exchange.

**37)What are firewalls?**

Firewalls serve to protect an internal network from external attacks. These external threats can be hackers who want to steal data or computer viruses that can wipe out data in an instant. It also prevents other users from external networks from gaining access to the private network.

**38)Describe star topology**

Star topology consists of a central hub that connects to nodes. This is one of the easiest to setup and maintain.

**39)What are gateways?**

Gateways provide connectivity between two or more network segments. It is usually a computer that runs the gateway software and provides translation services. This translation is a key in allowing different systems to communicate on the network.

**40)What is the disadvantage of a star topology?**

One major disadvantage of star topology is that once the central hub or switch get damaged, the entire network becomes unusable.

**41)What is SLIP?**

SLIP, or Serial Line Interface Protocol, is actually an old protocol developed during the early UNIX days. This is one of the protocols that are used for remote access.

**42)Give some examples of private network addresses.**

10.0.0.0 with a subnet mask of 255.0.0.0

172.16.0.0 with subnet mask of 255.240.0.0 192.168.0.0 with subnet mask of 255.255.0.0

**43)What is tracert?**

Tracert is a Windows utility program that can used to trace the route taken by data from the router to the destination network. It also shows the number of hops taken during the entire transmission route.

**44)What are the functions of a network administrator?**

A network administrator has many responsibilities that can be summarize into 3 key functions: installation of a network, configuration of network settings, and maintenance/troubleshooting of networks.

**45)Describe at one disadvantage of a peer to peer network.**

When you are accessing the resources that are shared by one of the workstations on the network, that workstation takes a performance hit.

**46)What is Hybrid Network?**

A hybrid network is a network setup that makes use of both client-server and peer-to-peer architecture.

**47)What is DHCP?**

DHCP is short for Dynamic Host Configuration Protocol. Its main task is to automatically assign an IP address to devices across the network. It first checks for the next available address not yet taken by any device, then assigns this to a network device.

**48)What is the main job of the ARP?**

The main task of ARP or Address Resolution Protocol is to map a known IP address to a MAC layer address.

**49)What is TCP/IP?**

TCP/IP is short for Transmission Control Protocol / Internet Protocol. This is a set of protocol layers that is designed to make data exchange possible on different types of computer networks, also known as heterogeneous network.

**50)How can you manage a network using a router?**

Routers have built in console that lets you configure different settings, like security and data logging. You can assign restrictions to computers, such as what resources it is allowed access, or what particular time of the day they can browse the internet. You can even put restrictions on what websites are not viewable across the entire network.

**51)What protocol can be applied when you want to transfer files between different platforms, such between UNIX systems and Windows servers?**

Use FTP (File Transfer Protocol) for file transfers between such different servers. This is possible because FTP is platform independent.

**52)What is the use of a default gateway?**

Default gateways provide means for the local networks to connect to the external network. The default gateway for connecting to the external network is usually the address of theexternal router port.

**53)One way of securing a network is through the use of passwords. What can be considered as good passwords?**

Good passwords are made up of not just letters, but by combining letters and numbers. A password that combines uppercase and lowercase letters is favorable than one that uses all upper case or all lower case letters. Passwords must be not words that can easily be guessed by hackers, such as dates, names, favorites, etc. Longer passwords are also better than short ones.

**54)What is the proper termination rate for UTP cables?**

The proper termination for unshielded twisted pair network cable is 100 ohms.

**55)What is netstat?**

Netstat is a command line utility program. It provides useful information about the current TCP/IP settings of a connection.

**56)What is the number of network IDs in a Class C network?**

For a Class C network, the number of usable Network ID bits is 21. The number of possible network IDs is 2 raised to 21 or 2,097,152. The number of host IDs per network ID is 2 raised to 8 minus 2, or 254.

**57)What happens when you use cables longer than the prescribed length?**

Cables that are too long would result in signal loss. This means that data transmission and reception would be affected, because the signal degrades over length.

**58)What common software problems can lead to network defects?**

Software related problems can be any or a combination of the following:

- client server problems
- application conflicts
- error in configuration
- protocol mismatch
- security issues
- user policy and rights issues

**59)What is ICMP?**

ICMP is Internet Control Message Protocol. It provides messaging and communication for protocols within the TCP/IP stack. This is also the protocol that manages error messages that are used by network tools such as PING.

**60)What is Ping?**

Ping is a utility program that allows you to check connectivity between network devices on the network. You can ping a device by using its IP address or device name, such as a computer name.

**61)What is peer to peer?**

Peer to peer are networks that does not reply on a server. All PCs on this network act as individual workstations.

**62)What is DNS?**

DNS is Domain Name System. The main function of this network service is to provide host names to TCP/IP address resolution.

**63)What advantages does fiber optics have over other media?**

One major advantage of fiber optics is that is it less susceptible to electrical interference. It also supports higher bandwidth, meaning more data can be transmitted and received. Signal degrading is also very minimal over long distances.

**64)What is the difference between a hub and a switch?**

A hub acts as a multiport repeater. However, as more and more devices connect to it, it would not be able to efficiently manage the volume of traffic that passes through it. A switch provides a better alternative that can improve the performance especially when high trafficvolume is expected across all ports.

**65)What are the different network protocols that are supported by Windows RRAS services?**

There are three main network protocols supported: NetBEUI, TCP/IP, and IPX.

**66)What are the maximum networks and hosts in a class A, B and C network?**

For Class A, there are 126 possible networks and 16,777,214 hosts For Class B, there are 16,384 possible networks and 65,534 hosts For Class C, there are 2,097,152 possible networks and 254 hosts

**67)What is the standard color sequence of a straight-through cable?**

orange/white, orange, green/white, blue, blue/white, green, brown/white, brown.

**68)What protocols fall under the Application layer of the TCP/IP stack?**

The following are the protocols under TCP/IP Application layer: FTP, TFTP, Telnet and SMTP.

**69)You need to connect two computers for file sharing. Is it possible to do this without using a hub or router?**

Yes, you can connect two computers together using only one cable. A crossover type cable can be use in this scenario. In this setup, the data transmit pin of one cable is connected to the data receive pin of the other cable, and vice versa.

**70)What is ipconfig?**

Ipconfig is a utility program that is commonly used to identify the addresses information of a computer on a network. It can show the physical address as well as the IP address.

**71)What is the difference between a straight-through and crossover cable?**

A straight-through cable is used to connect computers to a switch, hub or router. A crossover cable is used to connect two similar devices together, such as a PC to PC or Hub to hub.

**72)What is client/server?**

Client/server is a type of network wherein one or more computers act as servers. Servers provide a centralized repository of resources such as printers and files. Clients refers to workstation that access the server.

**73)Describe networking.**

Networking refers to the inter connection between computers and peripherals for data communication. Networking can be done using wired cabling or through wireless link.

**74)When you move the NIC cards from one PC to another PC, does the MAC address gets transferred as well?**

Yes, that's because MAC addresses are hard-wired into the NIC circuitry, not the PC. This also means that a PC can have a different MAC address when the NIC card was replace by another one.

**75)Explain clustering support**

Clustering support refers to the ability of a network operating system to connect multiple servers in a fault-tolerant group. The main purpose of this is the in the event that one server fails, all processing will continue on with the next server in the cluster.

**76)In a network that contains two servers and twenty workstations, where is the best place to install an Anti-virus program?**

An anti-virus program must be installed on all servers and workstations to ensure protection. That's because individual users can access any workstation and introduce a computer virus when plugging in their removable hard drives or flash drives.

**77)Describe Ethernet.**

Ethernet is one of the popular networking technologies used these days. It was developed during the early 1970s and is based on specifications as stated in the IEEE. Ethernet is used in local area networks.

**78)What are some drawbacks of implementing a ring topology?**

In case one workstation on the network suffers a malfunction, it can bring down the entire network. Another drawback is that when there are adjustments and reconfigurations needed to be performed on a particular part of the network, the entire network has to be temporarily brought down as well.

**79)What is the difference between CSMA/CD and CSMA/CA?**

CSMA/CD, or Collision Detect, retransmits data frames whenever a collision occurred. CSMA/CA, or Collision Avoidance, will first broadcast intent to send prior to data transmission.

**80)What is SMTP?**

SMTP is short for Simple Mail Transfer Protocol. This protocol deals with all Internal mail, and provides the necessary mail delivery services on the TCP/IP protocol stack.

**81)What is multicast routing?**

Multicast routing is a targeted form of broadcasting that sends message to a selected group of user, instead of sending it to all users on a subnet.

**82)What is the importance of Encryption on a network?**

Encryption is the process of translating information into a code that is unreadable by the user. It is then translated back or decrypted back to its normal readable format using a secret key or password. Encryption help ensure that information that is intercepted halfway would remain unreadable because the user has to have the correct password or key for it.

**83)How are IP addresses arranged and displayed?**

IP addresses are displayed as a series of four decimal numbers that are separated by period or dots. Another term for this arrangement is the dotted decimal format. An example is 192.168.101.2

**84)Explain the importance of authentication.**

Authentication is the process of verifying a user's credentials before he can log into the network. It is normally performed using a username and password. This provides a secure means of limiting the access from unwanted intruders on the network.

**85)What do mean by tunnel mode?**

This is a mode of data exchange wherein two communicating computers do not use IPSec themselves. Instead, the gateway that is connecting their LANs to the transit network creates a virtual tunnel that uses the IPSec protocol to secure all communication that passes through it.

**86)What are the different technologies involved in establishing WAN links?**

Analog connections - using conventional telephone lines; Digital connections - using digitalgrade telephone lines; switched connections - using multiple sets of links between sender and receiver to move data.

**87)What is one advantage of mesh topology?**

In the event that one link fails, there will always be another available. Mesh topology is actually one of the most fault-tolerant network topology.

**88)When troubleshooting computer network problems, what common hardware-relatedproblems can occur?**

A large percentage of a network is made up of hardware. Problems in these areas can range from malfunctioning hard drives, broken NICs and even hardware startups. Incorrectly hardware configuration is also one of those culprits to look into.

**89)What can be done to fix signal attenuation problems?**

A common way of dealing with such a problem is to use repeaters and hub, because it will help regenerate the signal and therefore prevent signal loss. Checking if cables are properly terminated is also a must.

**90)How does dynamic host configuration protocol aid in network administration?**

Instead of having to visit each client computer to configure a static IP address, the network administrator can apply dynamic host configuration protocol to create a pool of IP addresses known as scopes that can be dynamically assigned to clients.

**91)Explain profile in terms of networking concept?**

Profiles are the configuration settings made for each user. A profile may be created that puts a user in a group, for example.

**92)What is sneakernet?**

Sneakernet is believed to be the earliest form of networking wherein data is physically transported using removable media, such as disk, tapes.

**93)What is the role of IEEE in computer networking?**

IEEE, or the Institute of Electrical and Electronics Engineers, is an organization composed of engineers that issues and manages standards for electrical and electronic devices. This includes networking devices, network interfaces, cablings and connectors.

**94)What protocols fall under the TCP/IP Internet Layer?**

There are 4 protocols that are being managed by this layer. These are ICMP, IGMP, IP and ARP.

**95)When it comes to networking, what are rights?**

Rights refer to the authorized permission to perform specific actions on the network. Each user on the network can be assigned individual rights, depending on what must be allowed for that user.

**96)What is one basic requirement for establishing VLANs?**

A VLAN requires dedicated equipment on each end of the connection that allows messages entering the Internet to be encrypted, as well as for authenticating users.

**97)What is IPv6?**

IPv6 , or Internet Protocol version 6, was developed to replace IPv4. At present, IPv4 is being used to control internet traffic, butis expected to get saturated in the near future. IPv6 was designed to overcome this limitation.

**98)What is RSA algorithm?**

RSA is short for Rivest-Shamir-Adleman algorithm. It is the most commonly used public key encryption algorithm in use today.

**99)What is mesh topology?**

Mesh topology is a setup wherein each device is connected directly to every other device on the network. Consequently, it requires that each device have at least two network connections.

**100)What is ns2?**

ns is an object-oriented,discrete event simulator targeted at networking research. ns provides substantial support for simulation of tcp, routing, andmulticast protocols over wired and wireless (local and satellite) networks. Later ns-2 (version 2) was developed at uc berkeley in c++ and otcl (object-oriented extension of tcl).

**101)What is simulation?**

The process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system.

**102) Explain basic architecture of NS-2.**

NS-2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL.

**103)What is trace file?**

The trace file (trace.tr) is a standard format used by ns2. In ns2, each time a packet moves from one node to another, or onto a link, or into a buffer, etc., it gets recorded in this trace file.

**104)What is nam file?**

A visual aid showing how packets flow along the network.

**105)Why awk file is used?**

The basic function of awk is to search files for lines (or other units of text) that contain certain patterns. When a line matches one of the patterns, awk performs specified actions on that line. awk keeps processing input lines in this way until the end of the input files are reached.

**106)What is xgraph?**

It is used for plotting purpose comes together NS2 installation package.

**107)What different layers of TCP/IP model, does link, node, agent and traffic source represent in NS-2?**

Link represents Network access layer; Node represents Internet layer; Agent represents Transport layer; and Traffic Source represents Application layer of TCP/IP model.