CANARA
ENGINEERING COLLEGE

Course File

Department : **ISE**

Program : **B.E**

Course Name : COMPUTER NETWORKS

Course Code : BCS502

www.canaraengineering.in

CANARA ENGINEERING COLLEGE

Benjanapadavu – 574219, MANGALORE
Affiliated to Visvesvaraya Technological University

DEPARTMENT OF

INFORMATION SCIENCE AND ENGINEERING

COMPUTER NETWORKS LABORATORY (BCS502)

LABORATORY MANUAL

V SEMESTER

Mr. Vasanth Nayak, Assistant Professor ,ISE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CANARA ENGINEERING COLLEGE

COMPUTER NETWORKS LABORATORY (BCS502)

INSTRUCTIONS

1. This laboratory manual is an interim record of the sample experiments conducted on programming concepts during the regular laboratory sessions under the above said subject code.
2. Students should come with through preparation for the programs to be developed.
3. Students will not be permitted to attend the laboratory unless they bring the practical record and observation fully completed in all respects pertaining to the experiment conducted in the previous class.
4. Students are supposed to wear their college ID cards before attending the lab.
5. Students are supposed to occupy the systems allotted to them and are not Supposed to talk or make noise in the lab.
6. The students are expected to execute / debug the code to fulfil the aim of the experiment.
7. The students need to test the written programs with the varieties of inputs other than the sample inputs and display the output in the record book.
8. Practical record and observation book should be neatly maintained.
9. Students should obtain the signature of the staff-in-charge / Co – in-charge in the observation book after completing each program.

HEAD OF THE DEPARTMENT OF ISE
Dr. Manoj Gadiyar

CANARA ENGINEERING COLLEGE

Benjanapadavu, Mangalore – 574219

Department of Information Science & Engineering

VISION

The Department of Information Science and Engineering strives to be a centre of learning in the field of Information Technology to produce globally competent engineers catering to the needs of the industry and society.

MISSION

- Impart technical skills in the field of Information Science & Engineering.
- Train and transform students to become technological thinkers and facilitate a quality venture which meets the industrial and societal needs.
- Encourage students to become well-rounded in their professional competencies.

PROGRAM EDUCATIONAL OBJECTIVES

1. Graduates will succeed in the field of Information Science and Engineering, professional career and higher studies.
2. Graduates will analyze the requirements of the software industries and provide novel engineering designs and efficient solutions with legal and ethical responsibility.
3. Graduates will adapt to emerging technologies, work in multidisciplinary teams with effective communication skills and leadership qualities.

PROGRAM OUTCOMES

Engineering graduates in **Information Science and Engineering** will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **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.
3. **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.
4. **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.

5. **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.
6. **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.
7. **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.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **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.
11. **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.
12. **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

1. An ability to understand, analyze and impart the basic knowledge of Information Science and Engineering.
2. An ability to apply the programming, designing, and problem solving techniques in building/simulating the applications, solving the problems and guiding the innovative career paths to become an IT Engineer.

COMPUTER NETWORKS LAB (BCS502)

For

5th Semester B E Degree,
Information Science and Engineering

2024-2025

STUDENT NAME	
STUDENT USN NUMBER	

CANARA ENGINEERING COLLEGE
BENJANAPADAVU-574219

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

COMPUTER NETWORKS LAB (BCS502)
EVALUATION SHEET

NO	EXPERIMENTS	CO1	CO2	CO3	CO4	CO5
A	**AVERAGE**					
	LAB INTERNAL TEST					
B	**RECORD**					
	TOTAL/25)					
	Faculty Name Signature					

CANARA ENGINEERING COLLEGE – MANGALURU

COURSE INFORMATION (Form – 5A)

PROGRAMME	CSE/ISE/AIML/CSD/CSBS (Common)
DEGREE	Bachelor of Engineering (BE)
COURSE TITLE	Computer Networks
SEMESTER	V
COURSE CODE	BCS502
COURSE TYPE (CONTENT)	IPCC
REGULATION/SCHEME	VTU BE 2022
CREDITS	04
L-T-P-S	3-0-2-0
CONTACT HOURS/WEEK	5
TOTAL CONTACT HOURS	40T + 20P
COURSE CATEGORY	Knowledge Centric
SEE & CIE Marks	50, 50

COURSE SYLLABUS:

MODULE	CONTENTS	HOURS
1	Introduction: Data Communications, Networks, Network Types, Networks Models: Protocol Layering, TCP/IP Protocol suite, The OSI model, Introduction to Physical Layer: Transmission media, Guided Media, Unguided Media: Wireless. Switching: Packet Switching and its types	**8**
2	Data Link Layer: Error Detection and Correction: Introduction, Block Coding, Cyclic Codes. Data link control: DLC Services: Framing, Flow Control, Error Control, Connectionless and Connection Oriented, Data link layer protocols, High Level Data Link Control. Media Access Control: Random Access, Controlled Access. Check Sum and Point to Point Protocol	**8**
3	Network Layer: Network layer Services, Packet Switching, IPv4 Address, IPv4 Datagram, IPv6 Datagram, Introduction to Routing Algorithms, Unicast Routing Protocols: DVR, LSR, PVR, Unicast Routing protocols: RIP, OSPF, BGP, Multicasting Routing-MOSPF	**8**
4	Introduction to Transport Layer: Introduction, Transport-Layer Protocols: Introduction, User Datagram Protocol, Transmission Control Protocol: services, features, segments, TCP connections, flow control, Error control, Congestion control	**8**

5	Introduction to Application Layer: Introduction, Client-Server Programming, Standard Client- Server Protocols: World Wide Web and HTTP, FTP, Electronic Mail, Domain Name System (DNS), TELNET, Secure Shell (SSH)	**8**
	Lab Component	
1	Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth, and find the number of packets dropped	**2**
2	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	**2**
3	Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination	**2**
4	Develop a program for error detecting code using CRC-CCITT (16- bits).	**2**
5	Develop a program to implement a sliding window protocol in the data link layer.	**2**
6	Develop a program to find the shortest path between vertices using the Bellman-Ford and path vector routing algorithm	**2**
7	Using TCP/IP sockets, write a client − server program to make the client send the file name and to make the server send back the contents of the requested file if present	**2**
8	Develop a program on a datagram socket for client/server to display the messages on client side, typed at the server side	**2**
9	Develop a program for a simple RSA algorithm to encrypt and decrypt the data	**2**
10	Develop a program for congestion control using a leaky bucket algorithm	**2**
TOTAL HOURS		**60**

COURSE OBJECTIVES:

This course will enable the students to:	
1	Study the TCP/IP protocol suite, switching criteria and Medium Access Control protocols for reliable and noisy channels.
2	Learn network layer services and IP versions.
3	Discuss transport layer services and understand UDP and TCP protocols.
4	Demonstrate the working of different concepts of networking layers and protocols

COURSE OUTCOMES (COs):

CO	DESCRIPTION OF COURSE OUTCOME
	After completion of the course, the students will be able to:
CO:1	Explain the fundamentals of computer networks
CO:2	Apply the concepts of computer networks to demonstrate the working of various layers and protocols in communication network.

CO:3	Analyze the principles of protocol layering in modern communication systems
CO:4	Demonstrate various Routing protocols and their services using tools such as Cisco packet tracer.
CO:5	Explain the principles of network applications.

INTRODUCTION TO JAVA PROGRAMMING

Anyone who is learning to program must choose a programming environment that makes it possible to create and to run programs. Programming environments can be divided into two very different types: integrated development environments and command-line environments. All programming environments for Java require some text editing capability, a Java compiler, and a way to run applets and stand-alone applications. An integrated development environment, or IDE, is a graphical user interface program that integrates all these aspects of programming and probably others (such as a debugger, a visual interface builder, and project management). A command-line environment is just a collection of commands that can be typed in to edit files, compile source code, and run programs.

Command line environment is preferable for beginning programmers. IDEs can simplify the management of large numbers of files in a complex project, but they are themselves complex programs that add another level of complications to the already difficult task of learning the fundamentals of programming.

JAVA was developed by **Sun Microsystems Inc in the year 1991**, later acquired by Oracle Corporation. It was developed by James Gosling and Patrick Naughton. It is a simple programming language. Java makes writing compiling, and debugging programming easy. It helps to create reusable code and modular programs. Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is like c/c++.

Java Terminology

Before learning Java, one must be familiar with these common terms of Java.

1. **Java Virtual Machine(JVM):** This is generally referred to as JVM. There are three execution phases of a program. They are written, compile and run the program.
    - Writing a program is done by java programmer like you and me.
    - The compilation is done by JAVAC compiler which is a primary Java compiler included in the Java development kit (JDK). It takes Java program as input and generates bytecode as output.
    - In Running phase of a program, JVM executes the bytecode generated by the compiler.

   Now, we understood that the function of Java Virtual Machine is to execute the bytecode produced by the compiler. Every Operating System has different JVM but the output they produce after the execution of bytecode is the same across all the operating systems. Therefore, Java is known as a platform-independent language.

2. **Bytecode in the Development process:** As discussed, Javac compiler of JDK compiles the java source code into bytecode so that it can be executed by JVM. It is saved as .class file by the compiler. To view the bytecode, a disassembler like javap can be used.

3. **Java Development Kit (JDK):** While we were using the term JDK, when we learn about bytecode and JVM. So, as the name suggests, it is a complete java development kit that includes everything including compiler, Java Runtime Environment (JRE), java debuggers, java docs etc. For the program to execute in java, we need to install JDK in our computer to create, compile and run the java program.

4. **Java Runtime Environment (JRE):** JDK includes JRE. JRE installation on our computers allow the java program to run, however, we cannot compile it. JRE includes a browser, JVM, applet supports and plugins. For running the java program, a computer needs JRE.

5. **Garbage Collector:** In Java, programmers can't delete the objects. To delete or recollect that memory JVM has a program called Garbage Collector. Garbage Collector can recollect the of objects that are not referenced. So, Java makes the life of a programmer easy by handling memory management. However, programmers should be careful about their code whether they are using objects that have been used for a long time. Because Garbage cannot recover the memory of objects being referenced.

6. **ClassPath:** The classpath is the file path where the java runtime and Java compiler looks for .class files to load. By default, JDK provides many libraries. If you want to include external libraries, they should be added to the classpath.

Integrated Development Environments

There are sophisticated IDEs for Java programming that are available.

1. **Eclipse IDE** -- An increasingly popular professional development environment that supports Java development, among other things. Eclipse is itself written in Java. It is available from http://www.eclipse.org/.

2. **NetBeans IDE** -- A pure Java IDE that should run on any system with Java 1.7 or later. NetBeans is a free, "open source" program. It is essentially the open source version of the next IDE. It can be downloaded from www.netbeans.org.

3. **BlueJ** -- is a Java IDE written in Java that is meant particularly for educational use. Itis available from http://www.bluej.org/.

4. **JCreator** -- for Windows. It looks like a nice lighter-weight IDE that works on top of Sun's SDK. There is a free version, as well as a shareware version. It is available at http://www.jcreator.com. There are other products like JCreator, for Windows and for other operating systems.

Recommended System/Software Requirements

- System: Desktop PC with minimum of 2.6GHZ or faster processor with at least 256 MB RAM and 40GB free disk space.
- Operating system: Flavor of any WINDOWS.
- Software: jdk-12.0.2_windows-x64_bin.
- Editor: Eclipse or NetBeans.

Simple Java Program

```
public class MyFirstJavaProgram
{

    public static void main(String [ ]args)
    {
        System.out.println("Hello World");
    }
}
```

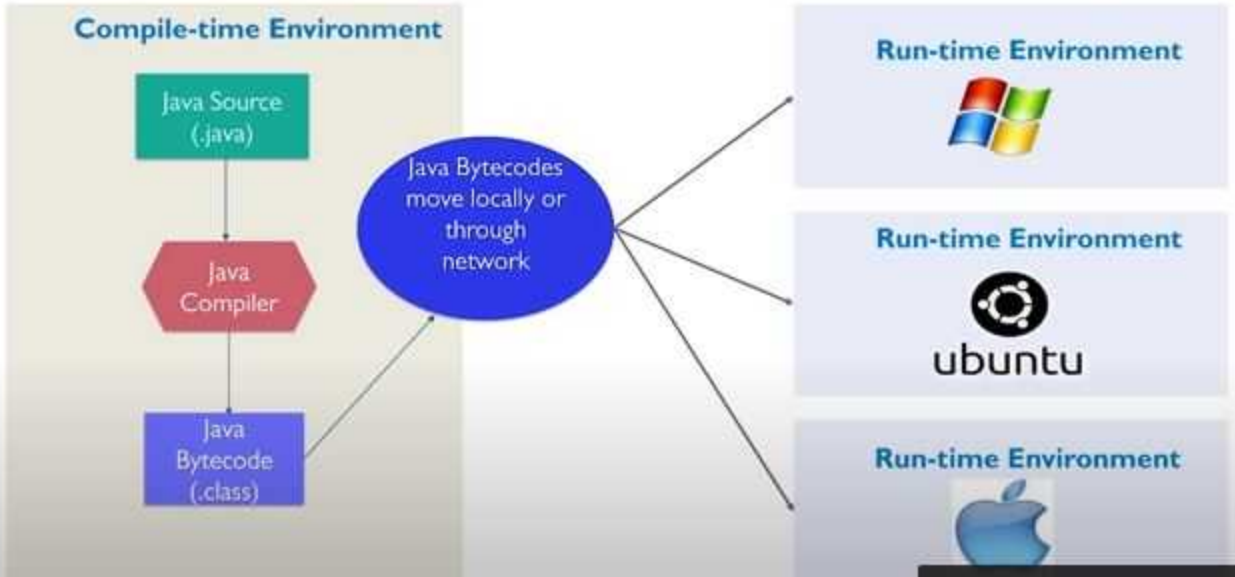
javac

MyFirstJavaProgram.java

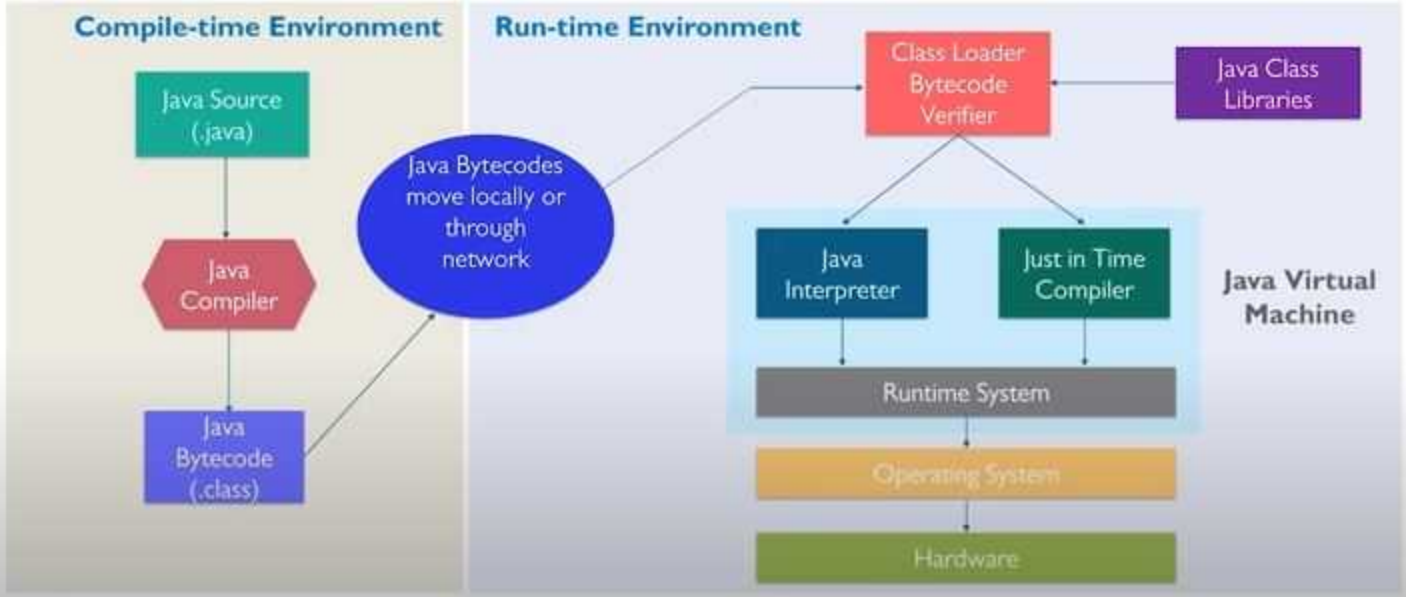
java MyFirstJavaProgram

Output: Hello World

Java Working

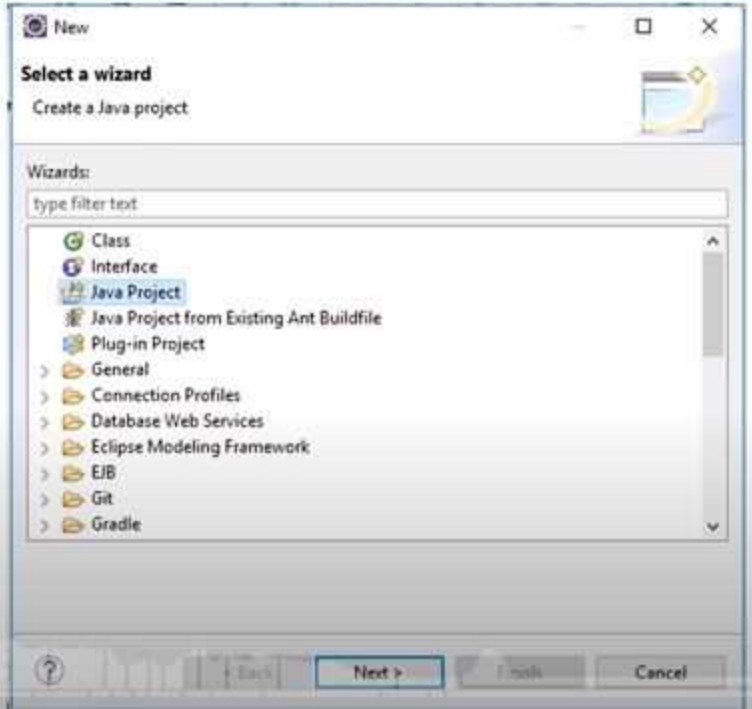


Java Working



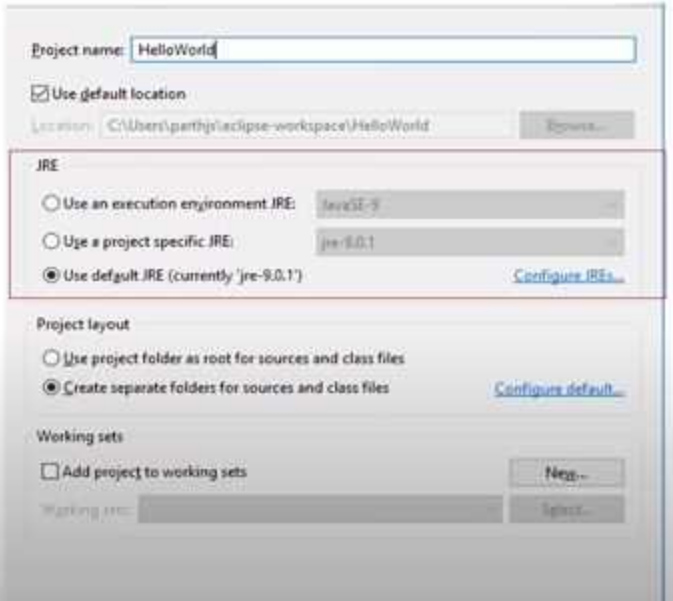
Create New Java Project

- Open Eclipse
- Click on File -> New -> Java Project

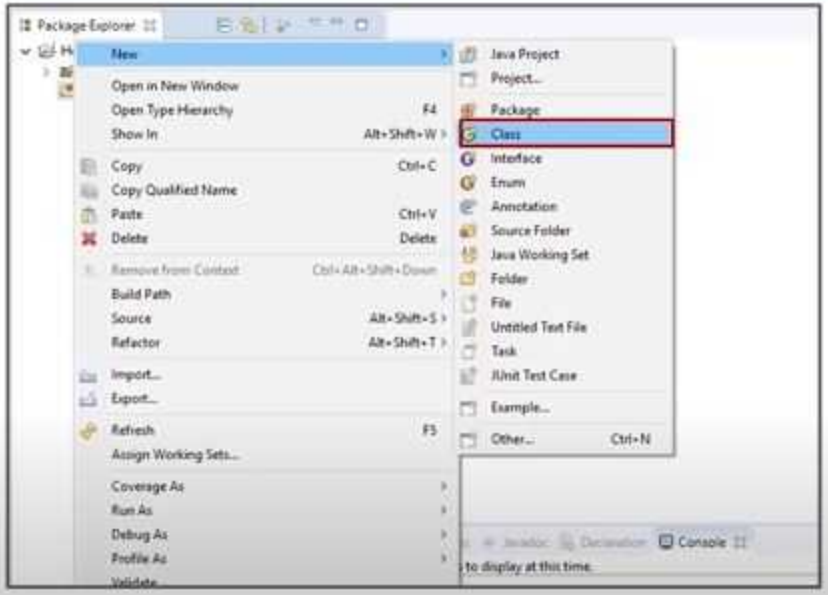


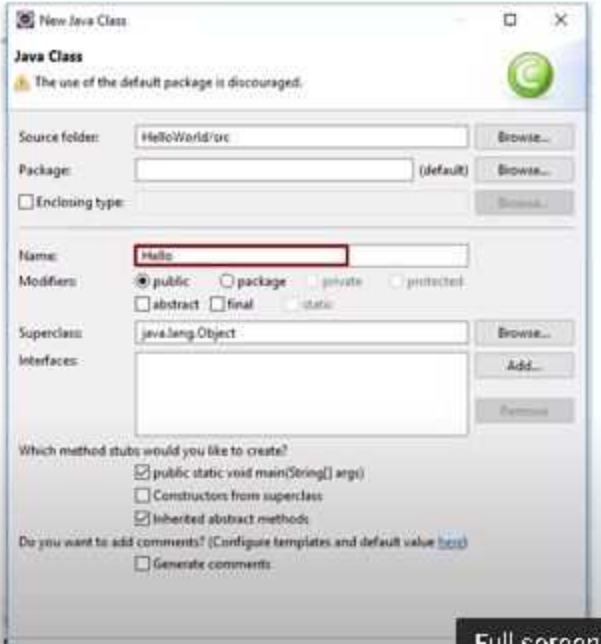
Check The JRE Version

Check the environment configurations to see whether the JRE provided is of the version java 9



Create A Class





Full screen (f)

First Java Program

public keyword is an access modifier which represents visibility, it means it is visible to all

static is a keyword, if we declare any method as static, it is known as static method. The core advantage of static method is that there is no need to create object to invoke the static method

class keyword is used to declare a class in java

Hello.java

```
1
2  public class Hello {
3
4      public static void main(String[] args) {
5          System.out.println("HelloWorld");
6      }
7
8  }
```

String[] args is used for command line argument. We will learn it later

void is the return type of the method, it means it doesn't return any value

main represents startup of the program

System.out.println() is used print statement. We will learn about the internal working of System.out.println statement later

Full screen (f

Fedora installation with virtual machine (VMware Workstation)

Copy the fedora Directory and VMware Workstation software to D: Drive

2 .Double Click on VMplayer 16.1.2.

3. Click on accept check box then click next.

4 .Click On next

5 .Uncheck the both check boxes

6 .Click on Install

7. Click on Next

8. click on finish

9.Open VMware Workstation from desktop and click on (open virtual machine) for path selection

10 .Choose fedora path from copied directory (D: Drive)

Fedora 20 is displaying below home button select and clicks on play

click on (take ownership) if ask

click on (I copied it) if ask

Click on Remind me later don't click(Download and install)

Click on (Not Listed?)

Type USER NAME : root

Type PASSWORD : fedora

Done installation VMware Workstation along with fedora 20 with Network Simulation

Introduction to NS-2

NS2 stands for Network Simulator Version 2.

Widely known as NS2, is simply an event driven simulation tool

It is an open-source event-driven simulator designed specifically for research in computer communication networks.

Useful in studying the dynamic nature of communication networks.

Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols andsimulating their corresponding behaviors.

Basic Architecture of NS2:

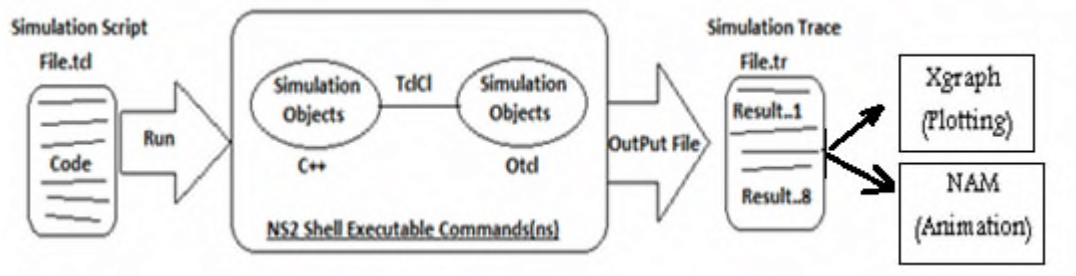


Figure 1: Basic Architecture of NS2

TCL – Tool Command Language:

Tcl is a very simple programming language and it is a general purpose scripting language. [Interpreter]. Tcl runs on most of the platforms such as Unix, Windows, and Mac. The strength of Tcl is its simplicity. It is not necessary to declare a data type for variable prior to the usage.

STRUCTURE OF TRACE FILES:

When tracing into an output ASCII file, the trace is organized in 12 fields as follows

in fig shown below, The meaning of the fields are:

Event	Time	From Node	To Node	PKT Type	PKT Size	Flags	Fid	Src Addr	Dest Addr	Seq Num	Pkt id

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 and dropped.

2. The second field gives the time at which the event occurs.

3. Gives the input node of the link at which the event occurs.

4. Gives the output node of the link at which the event occurs.

5. Gives the packet type (eg CBR or TCP)

6. Gives the packet size

7. Some flags

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 NAM display.

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 analysis purposes

12. The last field shows the unique id of the packet.

XGRAPH:

The xgraph program draws a graph on an x-display given data read from either data file or from standard input if no files are specified. It can display upto 64 independent data sets using different colors and line styles for each set. It annotates the graph with a title, axis labels, grid lines or tick marks, grid labels and a legend.

Options are listed here

/-bd <color> (Border)

This specifies the border color of the xgraph window.

/-bg <color> (Background)

This specifies the background color of the xgraph window.

/-fg<color> (Foreground)

This specifies the foreground color of the xgraph window.

/-lf <fontname> (LabelFont)

All axis labels and grid labels are drawn using this font.

/-t<string> (Title Text)

This string is centered at the top of the graph.

/-x <unit name> (XunitText)

This is the unit name for the x-axis. Its default is ―X‖.

/-y <unit name> (YunitText)

This is the unit name for the y-axis. Its default is ―Y‖.

Steps to draw the topology:

To start NSG2 open terminal :

java –jar NSG2.1.jar

It opens a graphical user interface on top of the window click on scenario,

select the type of connection (new wired scenario/wireless scenario).

3. To create *node*click on node tab and click on the screen, you can create any no of nodes as per the need.

Steps for execution:

1. Open gedit editor and type program. Program name should have the extension ― .tcl

[root@localhost ~]# gedit lab1.tcl

2. Open gedit editor and type awk program. Program name should have the extension ―.awk

[root@localhost ~]# gedit lab1.awk

3. Run the simulation program

[root@localhost~]# ns lab1.tcl

4. Here "ns" indicates network simulator. We get the topology shown in the snapshot. Now press the play button in the simulation window and the simulation will begins.

5. After simulation is completed run awk file to see the output

[root@localhost~]# awk –f lab1.awk lab1.tr

6. To see the trace file contents open the file as ,

[root@localhost~]# gedit out.tr

7. Change Bandwidth, Queue limit in the TCL file. Repeat the steps

NETWORK SIMULATION 2 SOFTWARE

Network Simulation

- Approaches
  - Experiment
    - Put all network Devices and measure the results/performance
    - Pros - realistic
    - Cons - Expensive/Sometime impossible
  - Mathematical model
    - Model devices using a graph model
    - Insight
    - Need to make assumptions
  - Simulation
    - Use programming to represent devices
    - Easy and can be easily verified
    - Not much insight, need to make assumptions

Network Simulation

- Network Simulation
  - Event Driven
    - Every event provide a reference to the next event (Example: using pointer)

- Simulation finishes
  - When there are no more events
  - At pre-specified time

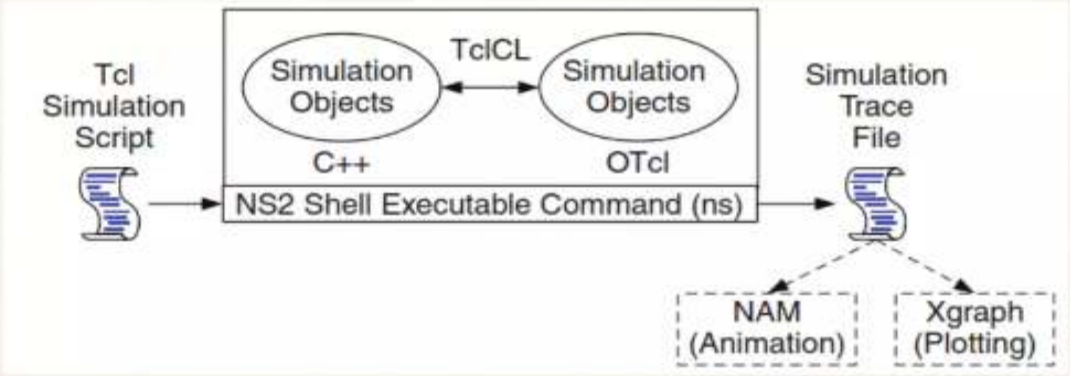
About NS2

- NS is a discrete event simulator
- It provides support for
  - Simulation of TCP
  - Routing
  - Huge number of protocols ranging from wired to wireless networks
  - Supports the addition of new entities like agent, packet, application, queue, protocol, routing, etc.

NS2 Architecture

- Network Simulator 2 is an event driven Simulator
- It consists of
  - C++ (Internally)
  - OTCL (User Interface)
  - TclCL (Interface between C++ and OTCL)

NS2 Architecture



PROGRAM-1

TITLE

Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.

AIM

Analyze the traffic between the nodes using different bandwidth, propagation delay and queue size of point to point duplex link and its effects on packet transmission.

DESCRIPTION

Ns2(network simulator2) is used for this experiment, in which point to point duplex link is created between the node with varying queuing capacity of node. Bandwidth, propagation delay of a point to point link and queuing capacity of a node is very important to minimize the affect on packet transmission. Ns2 simulated data traffic analyzed by setting the different bandwidth, propagation delay and queuing capacity of a link and correspondingly its effect on packet transmission is noted. Unix grep commands are used for analyzing the out.tr trace log file generated upon executing the ns2 simulation script for determining the packet drops.

INPUT

(a) Different Bandwidth, propagation delay & queuing capacity of node of duplex link
(b) Unix grep commands for analyzing the out.tr trace log file for determining the number of packet drop.

EXPECTED OUTPUT

(a) generate the out.tr trace log file and out.nam network animation file.
(b) number of packet drop

STEPS:

Steps to be followed:

Step 1: Click the Activities Panel on the top left side and search for **Terminal**

Step 2: Type the command in the **Terminal**

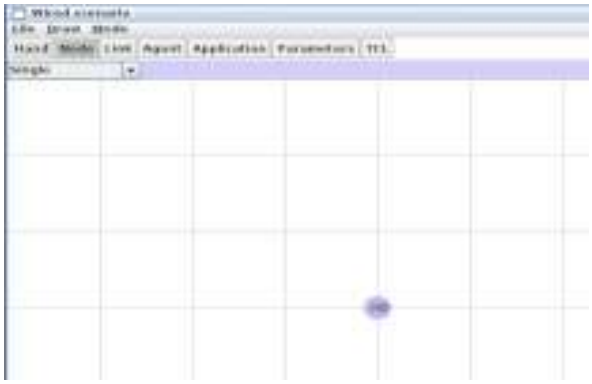
java -jar NSG2.1.jar

Step 3: Once Entered you will the redirected to a blue screen page there click on

Scenario -> New wired scenario

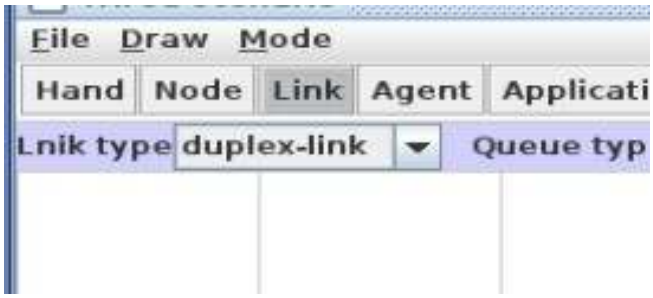
Step 4: Now you will have a blank screen here you must design your topology

Click on **Node** and click on the white screen to add the **Node**

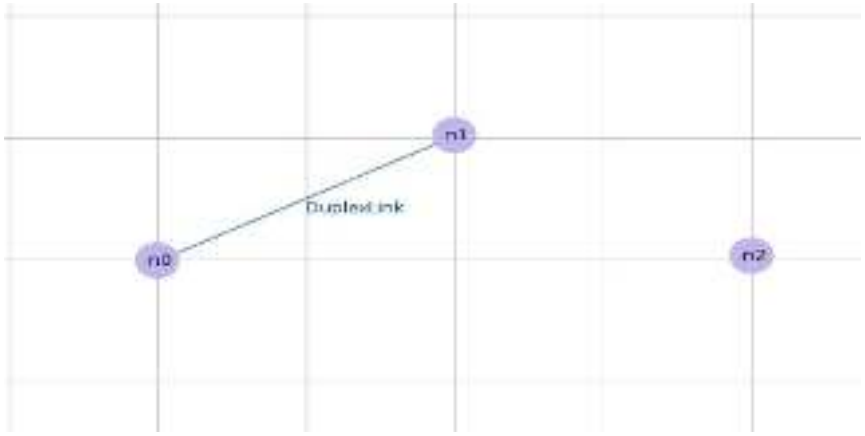


Similary add three **Nodes** shown below

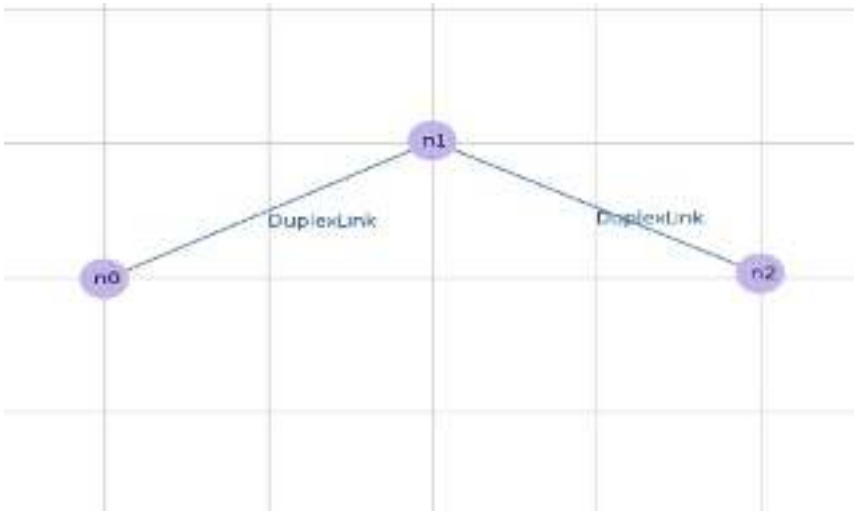
Step 5: Now to make the connection click on **Link -> duplex-link**



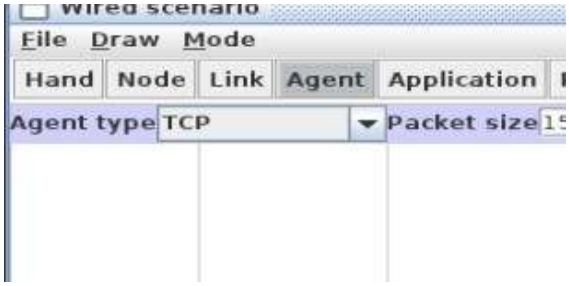
Step 6: Now Click on Node n0 and make the connection from n0 to n1



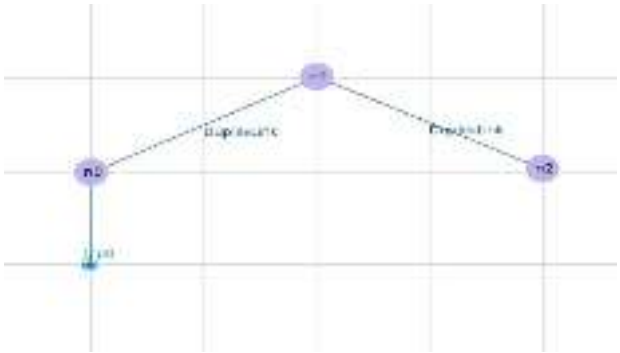
Similary make the connection from Node n1 to n2



Step 7: Go to **Agent** and select **TCP**



Click Node n0 and drag to bottom

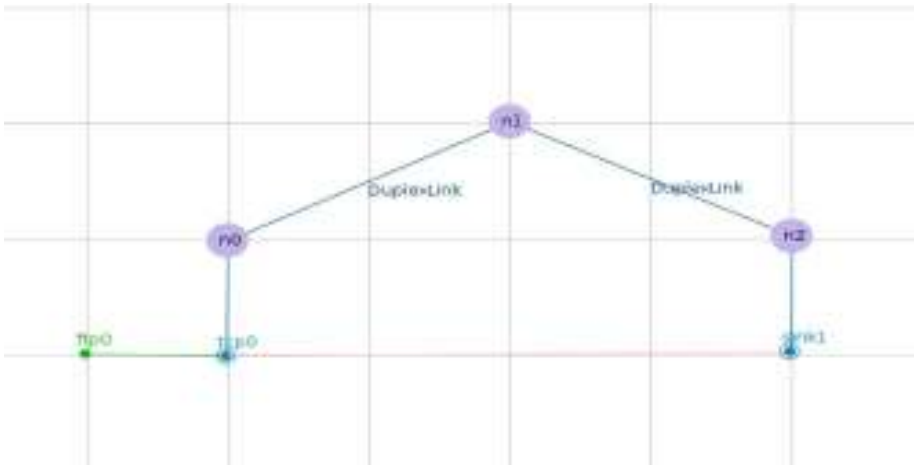


Similary choose **TCPSink** from the dropdown menu and select Node n2 and drag it

Now click on **tcp0** and next click on **sink1** to make a connection of orange line

Step 8: Choose **Application** and Application type as **FTP**

Click **tcp0** and drag to get the topology of



This is the final Topology

Step 9: Click **Parameters** and choose a trace file or leave it as default and click

1. **Save as Default->Done**

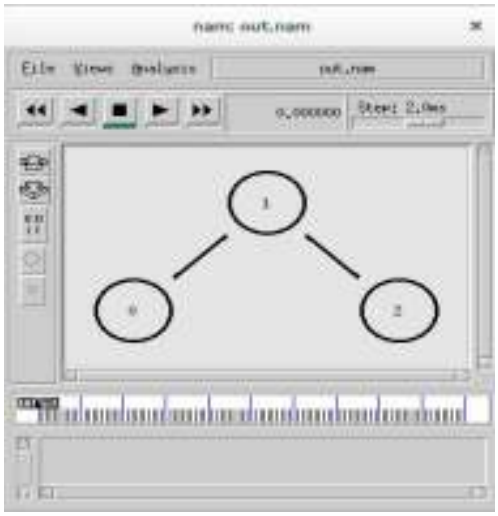
Step 10: Select **TCL** and name the file it as anything of our choice say for example exp1.tcl and exit

Step 11: Now in the terminal to run the program enter the following line of code ns progname.tcl

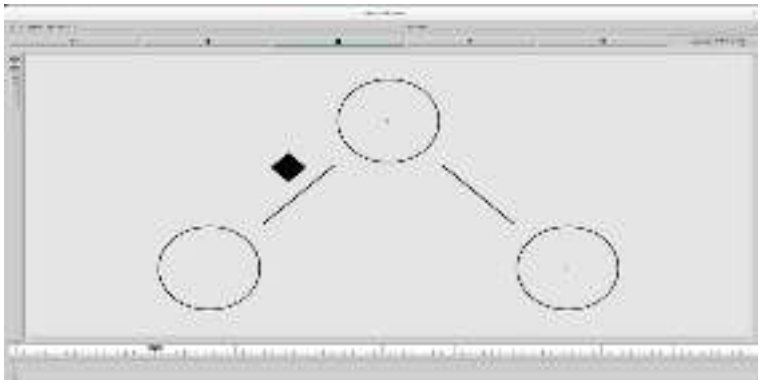
Ex: ns exp1.tcl

```
File  Edit  View  Search  Terminal  Help
[root@localhost ~]# java -jar NSG2.1.jar
java.awt.Color[r=238,g=238,b=238]
[root@localhost ~]# ns exp1.tcl
```

Click on Enter. You will get the output.nam



Now click on play forward and see the packets moving from sender Node n0 to Receiver Node n2



PROGRAM

Figure 2: NSG2 code generator screen

set ns [new Simulator]

```
set tracefile [open prog1.tr w]
$ns trace-all $tracefile

set namfile [open prog1.nam w]
$ns namtrace-all $namfile

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

$ns duplex-link $n0 $n1 100Mbps 10ms DropTail
$ns queue-limit $n0 $n1 5
$ns duplex-link $n1 $n2 100Mbps 10ms DropTail
$ns queue-limit $n1 $n2 3

$ns duplex-link-op $n0 $n1 orient right-down
$ns duplex-link-op $n1 $n2 orient left-down
set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns attach-agent $n0 $tcp
$ns attach-agent $n2 $sink
$ns connect $tcp $sink
$tcp set PacketSize_ 2500

set ftp [new Application/FTP]
$ftp attach-agent $tcp

$ns at 0.5 "$ftp start"
$ns at 2.0 "$ftp stop"
$ns at 2.5 "Finish"

proc Finish {} {
global ns tracefile namfile
$ns flush-trace
close $tracefile
close $namfile
exec nam prog1.nam &
exit 0
}
puts "simulation starts..."

$ns run
```

Step 12: Make a awk code by typing the command **gedit exp1.awk**

and the following line of code to it

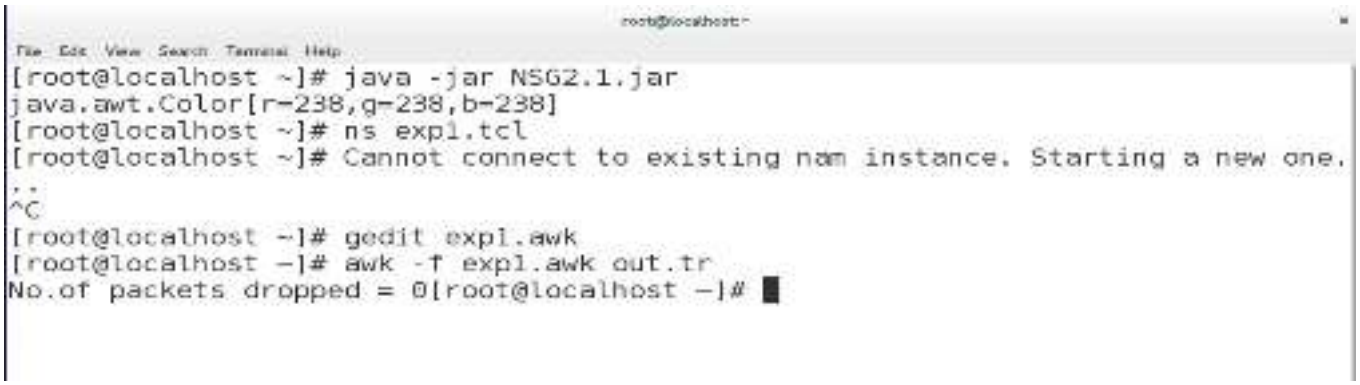
```
BEGIN{
        Count=0;
}
{
```

```
        if($4==”d”)
        {
                Count++;
        }
}
END{
printf(“No. of packets dropped = %d”,Count);
}
```

Step 13: To run the awk code type the following command in terminal

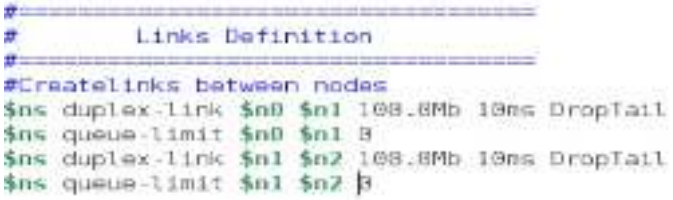
awk -f exp1.awk out.tr

Hit Enter ,You will get



Modifications:

Step 1: Open the tcl script and in the #Links Definition make the queue-limit as 0



Step 2:Now run the code

using **ns exp1.tcl**

and then use the command

awk -f exp1.awk out.tr

you will get a drop packet at Node n0

```
[root@localhost ~]# ns exp1.tcl
[root@localhost ~]# Missing required flag -x in: W -t 10.0

Missing required flag -y in: W -t 10.0

Parsing error in event.
^C
[root@localhost ~]# awk -f exp1.awk out.tr
No.of packets dropped = 1[root@localhost ~]#
```

VIVA QUESTIONS AND ANSWERS

1.Which are the different types of networks?
Personal Area **Network** (PAN)
Local Area **Network** (LAN)
Wireless Local Area **Network** (WLAN)
Campus Area **Network** (CAN)
Metropolitan Area **Network** (MAN)
Wide Area **Network** (WAN)
Storage-Area **Network** (SAN)

2.What is network topology?
Network topology is the arrangement of the elements (links, nodes, etc.) of a communication network

3.Which are the different types of network topologies?
Point-to-point topology is the simplest of all the network topologies. There is a direct link between two computers that want to communicate
Bus topology all the nodes are connected to one main cable which acts as a backbone
Star topology each computer is connected to a central hub using a point-to-point connection.
Ring topology the computers in the network are connected in a circular fashion, and the data travels in one direction.
Mesh topology every node has a direct point to point connection to every other node

4.Which are the different modes of data transmission?
Simplex: In this type of transmission mode, data can be sent only in one direction Half duplex: data can be transmitted in both directions, but not at the same time. Full duplex: data can be sent in both directions simultaneously.

5.What is a packet?

A packet is the unit of data that is routed between an origin and a destination on the Internet or any other packet-switched network.

6. Define jitter?
   Jitter is defined as a variation in the delay of received packets. At the sending side, packets are sent in a continuous stream with the packets spaced evenly apart. Due to network congestion, improper queuing, or configuration errors, this steady stream can become lumpy,or the delay between each packet can vary instead of remaining constant.

STUDENTS OBSERVATION:

PROGRAM –2

TITLE

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.

AIM

To understand the working principle of ICMP Ping message and deeper insights into the congestion scenario caused by successive ping message among nodes.

DESCRIPTION

Ping is the one of popular mechanism for internet control messaging protocol. Ping message is used for determining the reachability and aliveness of the remote/ destination machine in a network. In this experiment, network simulator2 is used for creating network topology consisting of 6 nodes interconnected by point to point duplex link. Nodes on the created topology issues ping command to the other nodes in the network and generate traffic. Node upon receiving the ping message will respond by sending a ping reply message to the requesting node and generate return traffic in the network. Successive ping message by different nodes generates huge traffic on the network and causes packet drop.

INPUT

(a) ping message from nodes.
(b) Unix ‘grep’ command for analyzing the out.tr trace log file for determining the number of packet drop.

EXPECTED OUTPUT

1. Ping response from corresponding nodes.
(b) Generate the out.tr trace log file and out.nam network animation file.
(c) number of packet drop

STEPS

Steps to be followed:

Step 1: Click the Activities Panel on the top left side and search for **Terminal**

Step 2: Type the command in the **Terminal**

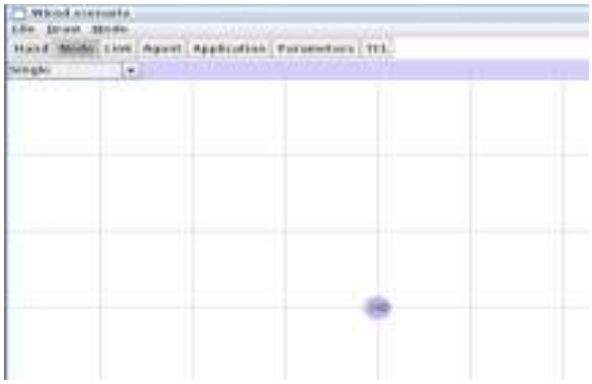
java -jar NSG2.1.jar

Step 3: Once Entered you will the redirected to a blue screen page there click on

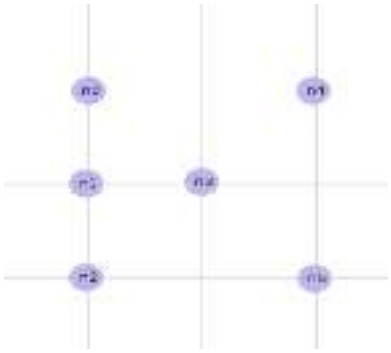
1. **Scenario -> New wired scenario**

Step 4: Now you will have a blank screen here you must design your topology

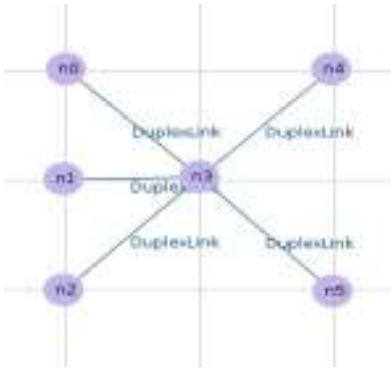
Click on **Node** and click on the white screen to add the **Node**



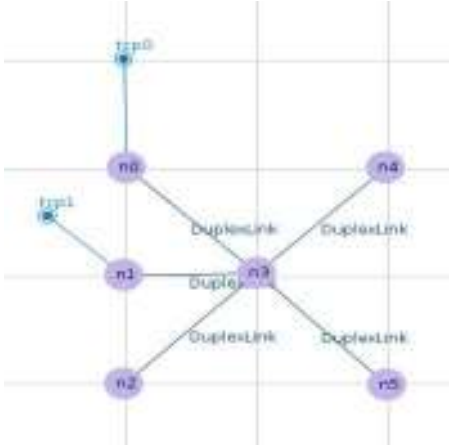
Similarly add 6 nodes as shown below



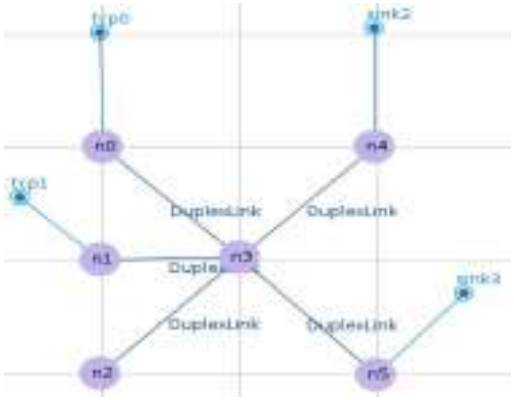
Step 5: Click **Link->duplex-link** and make the connection from Node n0 to n3, n1 to n3, n2 to n3, n3 to n4 and n3 to n5



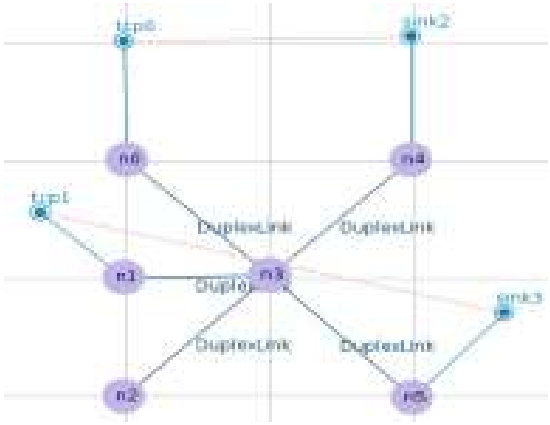
Step 6: Click **Agent->TCP** and click Node n0 and drag to some point make the same for n1



Similary Click **TCPSink** and click Node n4 and drag it and place it some place and do it for Node n5 also



Make the connection by clicking tcp0 and next click sink2 and similarly tcp1 and sink3



Step 7: Click Parameters and Click save as default and click on TCL and

save it.

PROGRAM

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>
#===================================
#     Simulation parameters setup
#===================================
set val(stop)   10.0                ;# time of simulation end


#===================================
#        Initialization
#===================================
#Create a ns simulator set ns [new Simulator]


#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile
```

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

#=====================================
#       Nodes Definition
#=====================================
#Create 6 nodes set n0 [$ns node] set n1 [$ns node] set n2 [$ns node] set n3
[$ns node] set n4 [$ns node] set n5 [$ns node]
#=====================================
#       Links Definition
#=====================================
#Createlinks between nodes
$ns duplex-link $n0 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n3 50

$ns duplex-link $n1 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n3 50
$ns duplex-link $n2 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
$ns duplex-link $n3 $n4 1.0Mb 10ms DropTail
$ns queue-limit $n3 $n4 2
$ns duplex-link $n3 $n5 1.0Mb 10ms DropTail
$ns queue-limit $n3 $n5 2

#Give node position (for NAM)
$ns duplex-link-op $n0 $n3 orient right-down
$ns duplex-link-op $n1 $n3 orient right
$ns duplex-link-op $n2 $n3 orient right-up
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $n5 orient right-down

#=====================================
#       Agents Definition
```

```
#================================== Agent/Ping instproc recv
{from rtt} {
$self instvar node_
puts "Node [$node_ id] receives response from $from with RTT = $rtt ms"
}
#Setup a TCP connection set p0 [new Agent/Ping]
$ns attach-agent $n0 $p0 set p3 [new Agent/Ping]
$ns attach-agent $n4 $p3
$ns connect $p0 $p3
$p0 set packetSize_ 1500

#Setup a TCP connection set p1 [new Agent/Ping]
$ns attach-agent $n1 $p1 set p4 [new Agent/Ping]
$ns attach-agent $n5 $p4
$ns connect $p1 $p4
$p1 set packetSize_ 1500
#=================================
#       Applications Definition
#=================================
$ns at 1.5 "$p0 send"
$ns at 1.5 "$p0 send"
$ns at 1.5 "$p0 send"
$ns at 1.5 "$p0 send"
$ns at 1.5 "$p1 send"

$ns at 1.5 "$p1 send"
$ns at 1.5 "$p1 send"
$ns at 1.5 "$p1 send"

#=================================
#       Termination
#=================================
#Define a 'finish' procedure proc finish {} {
global ns tracefile namfile
```

```
$ns flush-trace close $tracefile close $namfile

exec nam out.nam &

exit 0

}

$ns at $val(stop) "$ns nam-end-wireless $val(stop)"

$ns at $val(stop) "finish"

$ns at $val(stop) "puts \"done\" ; $ns halt"

$ns run
```

Step 8: Now open the tcl file by entering the following command **gedit exp2.tcl**

Step 9: Make the following changes

```
#=====================================
#       Agents Definition
#=====================================
Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts "Node [$node_ id] receives response from $from with RTT=$rtt
ms" }

#Setup a TCP
connection set p0 [new
Agent/Ping] $ns attach-
agent $n0 $p0 set p3
[new Agent/Ping] $ns
attach-agent $n4 $p3
$ns connect $p0 $p3
$p0 set packetSize_ 1500

#Setup a TCP
connection set p1 [new
Agent/Ping] $ns attach-
agent $n1 $p1 set p4
[new Agent/Ping] $ns

attach-agent $n5 $p4
$ns connect $p1 $p4
$p1 set packetSize_ 1500


#=====================================
#       Applications Definition
#=====================================
$ns at 1.0 "$p0 send"
$ns at 1.1 "$p0 send"
```

```
$ns at 1.2 "$p0 send"
$ns at 1.3 "$p0 send"
$ns at 1.0 "$p1 send"
$ns at 1.1 "$p1 send"
$ns at 1.2 "$p1 send"
$ns at 1.3 "$p1 send"
```

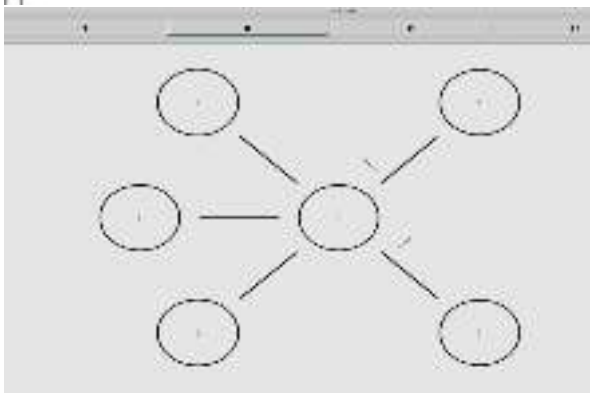
Step 10: Make a awk code by typing the command **gedit exp2.awk**

and the following line of code to it

```
BEGIN{
        Count=0;
}
{
        if($1==”d”)
        {
                Count++;
        }
}
END{
printf(“No. of packets dropped = %d”, Count);
}
```

Step 11: Now run the Command ns exp2.tcl in the terminal and press the play button

```
[root@localhost ~]# java -jar NSG2.1.jar
java.awt.Color[r=238,g=238,b=238]
[root@localhost ~]# gedit exp2.tcl
[root@localhost ~]# gedit test.awk
[root@localhost ~]# ns exp2.tcl
Node 0 receives response from 4 with RTT=40.3 ms
Node 1 receives response from 5 with RTT=40.3 ms
Node 0 receives response from 4 with RTT=40.3 ms
Node 1 receives response from 5 with RTT=40.3 ms
Node 0 receives response from 4 with RTT=40.3 ms
Node 1 receives response from 5 with RTT=40.3 ms
Node 0 receives response from 4 with RTT=40.3 ms
Node 1 receives response from 5 with RTT=40.3 ms
[root@localhost ~]# Cannot connect to existing nam instance. Starting a new one.
```



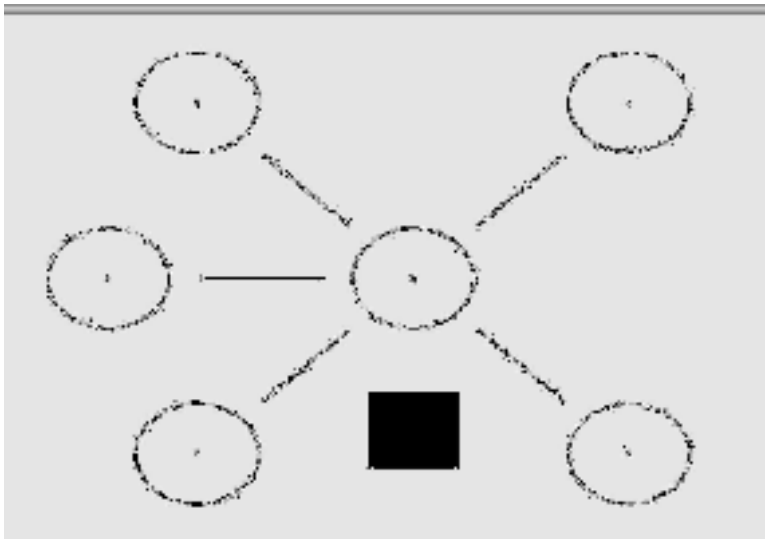
Step 12: Now run the Awk code by the command

awk -f exp2.awk exp2.tcl

```
[root@localhost ~]# awk -f test.awk out.tr
No. of packets dropped=0[root@localhost ~]#
```

Step 13: To get dropped packets make the Queue-limit

```
#=====================================
#       Links Definition
#=====================================
#Createlinks between nodes
$ns duplex-link $n0 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n3 50
$ns duplex-link $n1 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n3 50
$ns duplex-link $n2 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
$ns duplex-link $n3 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n3 $n4 0
$ns duplex-link $n3 $n5 100.0Mb 10ms DropTail
$ns queue-limit $n3 $n5 50
```



Now run the awk

```
File  Edit  View  Search  Terminal  Help
[root@localhost ~]# awk -f test.awk out.tr
No. of packets dropped=4[root@localhost ~]#
```

Viva Questions and Answers

1.Define Ping?

Ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network. It measures the round-trip time for messages sent from the originating host to a destination computer that are echoed back to the source.

2. What is network protocol?

Network protocols are formal standards and policies comprised of rules, procedures and formats that define communication between two or more devices over a network

3. Define Bandwidth?

Bandwidth is the amount of data that can be transmitted in a fixed amount of time

4. Define a router?

Routers perform the traffic directing functions on the Internet. Data sent through the internet, such as a web page or email, is in the form of data packets. A packet is typically forwarded from one router to another router through the networks until it reaches its destination node

5. Describe network congestion?

It is the reduced quality of service that occurs when a network node or link is carrying more data than it can handle. Typical effects include queuing delay, packet loss or the blocking of new connections

6. Which are the two congestion control mechanisms?

prevents the congestion from happening
removes congestion after it has taken place

STUDENTS OBSERBATION:

PROGRAM-3

TITLE

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

AIM

To understand working principle of Ethernet LAN and congestion scenario using multiple data traffic.

DESCRIPTION

Ethernet LAN denoted by IEEE 802.3 is one of the popular computer networking technology. In this experiment, ns2 simulator is used for creating Ethernet LAN and set the two different data traffic between pair of nodes using TCP as transport layer agent. Simulated data traffic between pair of nodes is analysed for determining the packet drop due to congestion in the network. Congestion window for each TCP traffic is plotted on graph using xgraph tool.

INPUT

Ethernet LAN Bandwidth, propagation delay, Queue Type and channel type
two TCP traffic between pair of nodes
initial Congestion window for both the traffic.
Unix grep command for analyzing the out.tr trace log file for determining the number of packet drop.

EXPECTED OUTPUT

generate the out.tr trace log file, winfile0, winfile1 and out.nam network animation file
Generate Windowsize_file0, WindowSize_file1 holding congestion window size of both the traffic at different instance of time.
number of packet drop.
Xgraph plotted graph of depicting the congestion window of both the traffic.

STEPS

Step 1: Click the Activities Panel on the top left side and search for **Terminal**

Step 2: Type the command in the **Terminal**

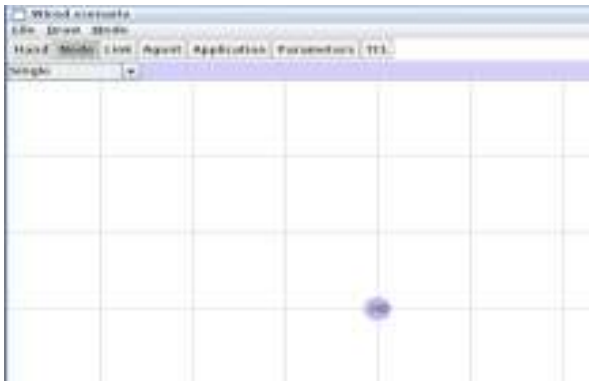
java -jar NSG2.1.jar

Step 3: Once Entered you will the redirected to a blue screen page there click on

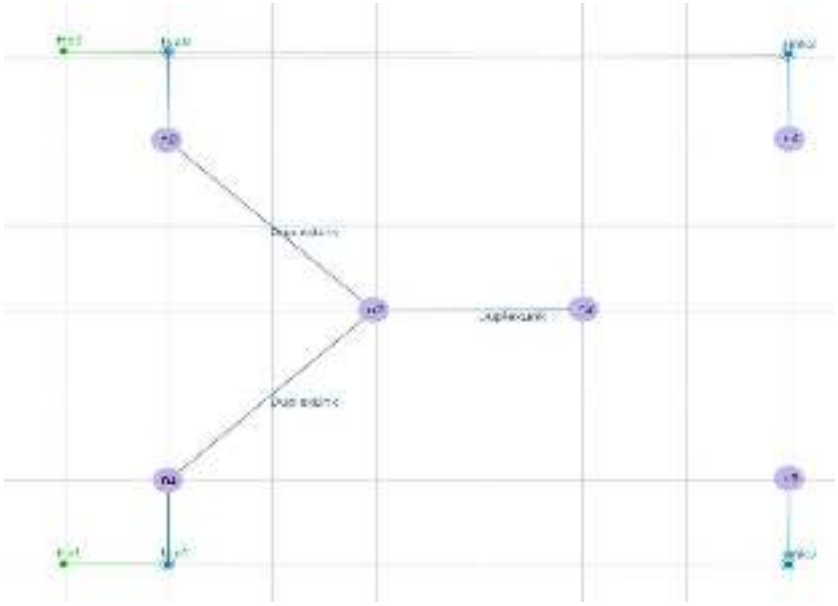
1. **Scenario -> New wired scenario**

Step 4: Now you will have a blank screen here you must design your topology

Click on **Node** and click on the white screen to add the **Node**



Similarly make the following topology



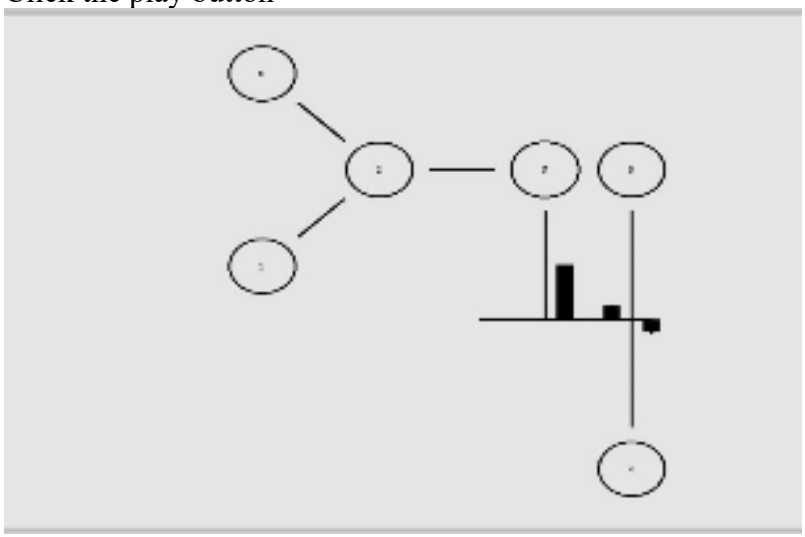
One thing to change is in Node n1 use TCPReno

Step 5: Go to Parameters->Save as Default -> Done
Step 6: Go to **TCL** and save it
Step 7: Go to Terminal and type **gedit ecp3.tcl**
And make the following changes

```
#Links Definition
set lan [$ns newLan "$n3 $n4 $n5" 1Mb 40ms LLQueue/DropTail Mac/802_3 channel]
#Give node position (for NAM)
```

```
#Applications definition
Set f1 [open f1.tr w]
$tcp0 attach $f1
Set f2 [open f2.tr w]
$tcp1 attach $f2

$tcp0 trace cwnd_
$tcp1 trace cwnd_
```

Step 8: In the terminal give the command **ns exp3.tcl**
Click the play button



Step 9: Update the AWK code

```
BEGIN{
```

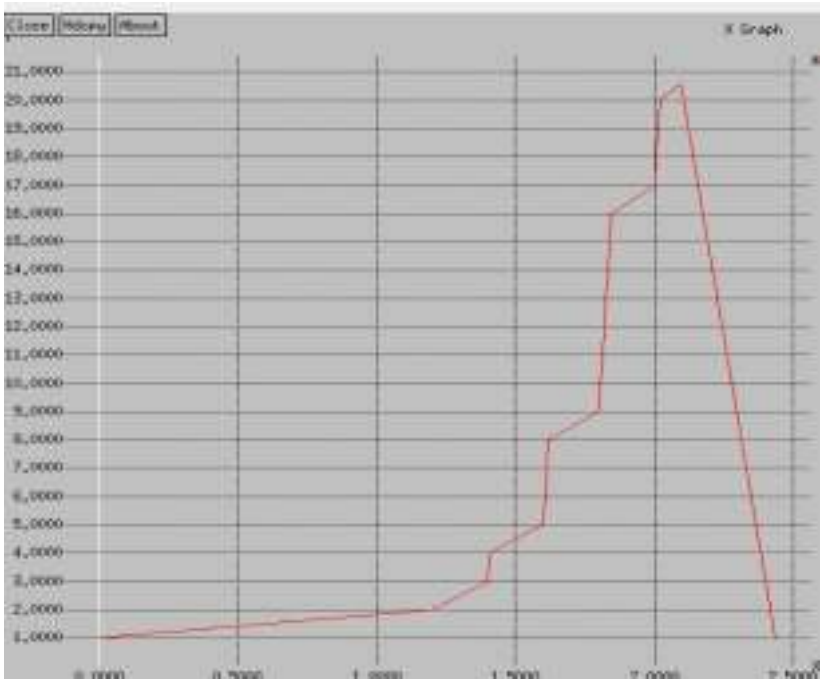
```
}
{
        if(“$6 == cwnd_”)
{
                printf(“%f\t%f\n”, $1, $7);
} }
END{
}
```

Step 10: Run the Awk code by the command

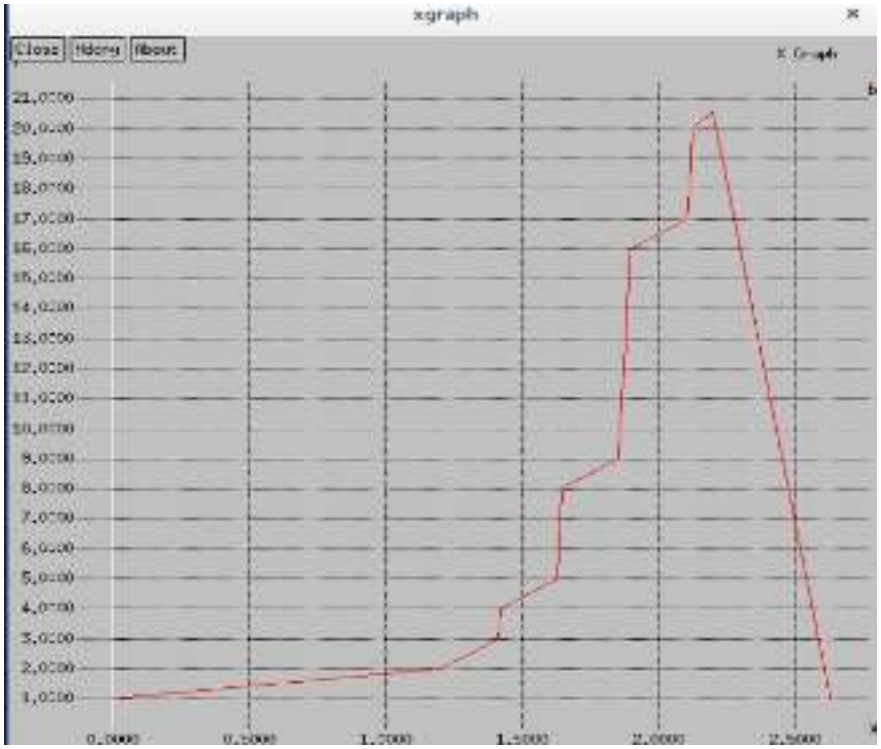
awk -f exp3.awk f1.tr
awk -f exp3.awk f2.tr

Step 11: We need the xgraph so assign the awk code as **awk -f exp3.awk f1.tr>a**
awk -f exp3.awk f2.tr>b

Step 12: Next to get the xgraph the command is
xgraph a

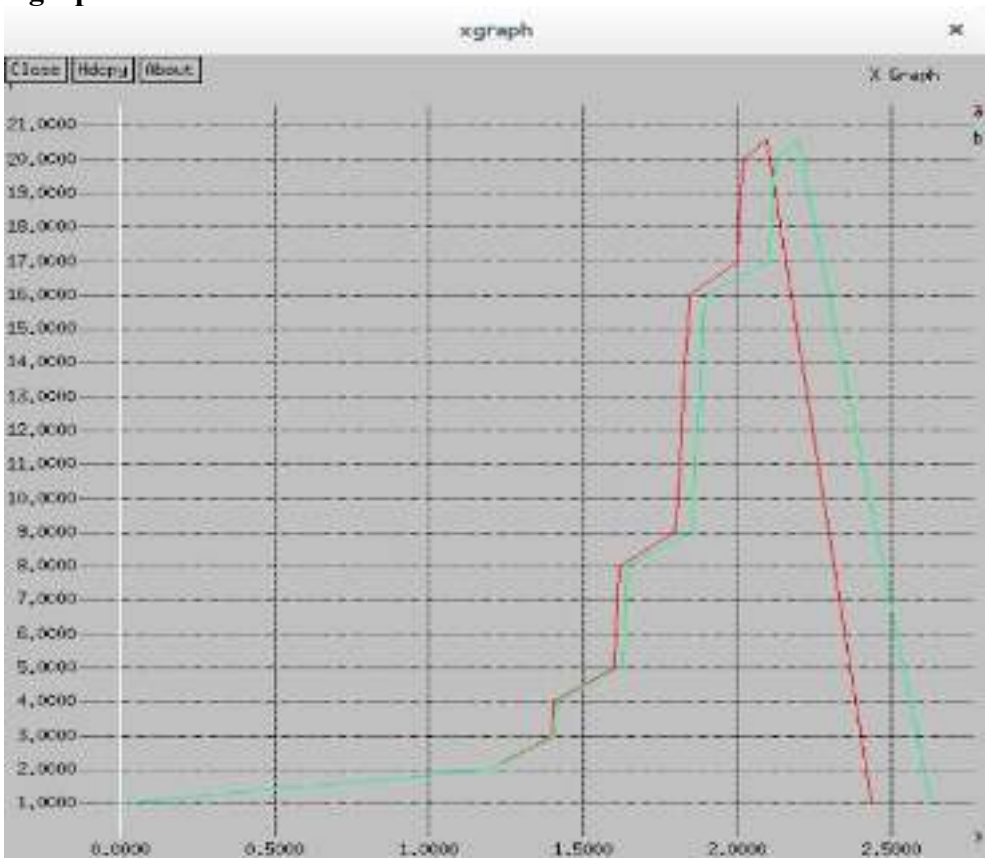


xgraph b



Finally comparing both

xgraph a b



VIVA QUESTIONS AND ANSWERS

What is Ethernet?
It is the standard way to connect computers on a network over a wired connection. It provides a simple interface for connecting multiple devices, such computers, routers, and switches. With a single router and a few Ethernet cables, you can create a LAN, which allows all connected devices to communicate with each other.

Which are the different layers of an OSI Model?
Layer 7: The application layer
Layer 6: The presentation layer
Layer 5: The session layer
Layer 4: The transport layer
Layer 3: The network layer
Layer 2: The data-link layer
Layer 1: The physical layer

Which are the two types of Internet Protocol (IP) traffic?
There are two types of Internet Protocol (IP) traffic. They are TCP or Transmission Control Protocol and UDP or User Datagram Protocol. TCP is connection oriented – once a connection is established, data can be sent bidirectional. UDP is a simpler, connectionless Internet protocol. Multiple messages are sent as packets in chunks using UDP.

Describe session?

A session is a semi-permanent interactive information interchange between two or more communicating devices or between a computer and user (see login session)

What is Demodulation?

It is the process of converting an analog signal to digital signal

PROGRAM 4

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

CRC generator using polynomials

If we consider the data unit 1001 and divisor or polynomial generator 1011their polynomial representation is:

• Now string of n 0s (one less than that of divisor) is appended to data. Now data is 1001000 and its corresponding polynomial representation is x6 + x3.

• The division of x6+x3 by x3+x+ 1 is shown in fig.

• The polynomial generator should have following properties:

1. It should have at least two terms.

2. The coefficient of the term x0 should be 1.

3. It should not be divisible by x.

4. It should be divisible by x+ 1.

Some Standard Generator Polynomials are shown below:

Name	Generator Polynomial
CRC – 8	x8 + x2 + x + 1
CRC – 10	x10 + x9 + x5 +x4 + x2 + 1
CRC - 16	x16 + x12 + x5 + 1

Program

```
import java.util.Scanner;
public class Crc {
static int data[],cs[];
static int g[]={1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1};
static int n, i, e,c,pos;
static int N=17;
static void xor() {
for(c=0;c<N;c++) cs[c]=((cs[c]==g[c])?0:1);
}
static void crc() {
for(i=0;i<N;i++) cs[i]=data[i];
do {
if(cs[0]==1) xor();
for(c=0;c<N-1;c++)
cs[c]=cs[c+1];
cs[c]=data[i++];
}while(i<=n+N-1);
}
public static void main(String[] args) {
                cs=new int[100];
                Scanner br=new Scanner(System.in);
                System.out.println("Enter no of Data bits");
                 n=br.nextInt();
                data=new int[100];
                System.out.println("\nEnter the data bits : ");
                for(int i=0;i<n;i++)
                data[i]=br.nextInt();
                System.out.println("\n\nCRC Divisor : ");
                for(int i=0;i<N;i++)
                                System.out.print(g[i]);
                for(i=n;i<n+N-1;i++)
                        data[i]=0;
                System.out.println("\n\nModified Data is :
                for(i=0;i<n+N-1;i++)
                        System.out.print(data[i]);
                crc();
                System.out.println("\n\nCRC Checksum is : ");
                for(int i=0;i<N-1;i++)
                        System.out.print(cs[i]);
                for(i=n;i<n+N-1;i++)
                data[i]=cs[i-n];
                System.out.println("\n\nFinal Codeword is :");
                for(i=0;i<n+N-1;i++)
                        System.out.print(data[i]);
                System.out.println("\n\nTest Error detection 0(yes) 1(no) ? : ");
                e=br.nextInt();
                if(e==0) {
                        System.out.println("Enter position where error is to inserted : ");
                        pos=br.nextInt();
                        data[pos]=(data[pos]==0)?1:0;
                        System.out.println("\nErroneous data
                        for(i=0;i<n+N-1;i++)
```

```
                        System.out.print(data[i]);
                        }
                crc();

                System.out.println("\n\nReceiver Checksum:");
                for(int i=0;i<N;i++)
                        System.out.print(cs[i]);
                for(i=0;i<N-1;i++)
                {
                        if(cs[i]!=0)
                                {
                                System.out.println("\n\nERROR in Received Codeword ");
                                System.exit(0);
                                }
                        }

                        System.out.println("\nNo Error in Received Codeword");
                        }
                }
```

/*Output of cyclic redundancy check program

OUTPUT:

```
$ gedit Crc.java
$ javac Crc.java
$ java Crc

Enter no of Data bits
4

Enter the data bits :
1 0 0 1

CRC Divisor :
10001000000100001


Modified Data is :
10010000000000000000


CRC Checksum is :
1001000100101001


Final Codeword is :
10011001000100101001


Test Error detection 0(yes) 1(no) ? :
1


Receiver Checksum:
00000000000000000
No Error in Received Codeword [root@localhost ~]# java Crc Enter no of Data bits
```

4

Enter the data bits :
1 0 0 1

CRC Divisor :
10001000000100001

Modified Data is :
10010000000000000000

CRC Checksum is :
1001000100101001

Final Codeword is :
10011001000100101001

Test Error detection 0(yes) 1(no) ? :
0
Enter position where error is to inserted :
2

Erroneous data
10111001000100101001

Receiver Checksum:
00100000010000100

ERROR in Received Codeword

Viva Question and Answers

1.What are the types of errors?

a. Single-Bit error: In a single-bit error, only one bit in the data unit has changed
b. Burst Error: A Burst error means that two or more bits in the data have changed.

2.What is Error Detection? What are its methods?

Data can be corrupted during transmission. For reliable communication, errors must be deducted and corrected. Error Detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination. The common Error Detection methods are

a. Vertical Redundancy Check (VRC)
b. Longitudinal Redundancy Check (VRC)
c. Cyclic Redundancy Check (VRC)
d. Checksum

3.What is CRC?
CRC, is the most powerful of the redundancy checking techniques, is based on binary division.

4. Compare Error Detection and Error Correction:
The correction of errors is more difficult than the detection. In error detection, checks only any error has occurred. In error correction, the exact number of bits that are corrupted and location in the message are known. The number of the errors and the size of the message are important factors.

5. What is Forward Error Correction?
Forward error correction is the process in which the receiver tries to guess the message by using redundant bits.

6. Define Retransmission?
Retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error-freed.

7. What are Cyclic Codes?
Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.

8. Define Encoder?
A device or program that uses predefined algorithms to encode, or compress audio or video data for storage or transmission use. A circuit that is used to convert between digital video and analog video.

9. Define Decoder?
A device or program that translates encoded data into its original format (e.g. it decodes the data). The term is often used in reference to MPEG-2 video and sound data, which must be decoded before it is output.

10. What is Bit Stuffing?
Bit stuffing is the process of adding one extra 0 whenever five consecutive Is follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

11. What is Error Control?
Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term error control refers primarily tomethods of error detection and retransmission.

STUDENTS OBSERVATIONS:

PROGRAM-5

TITLE

Develop a program to implement a sliding window protocol in the data link layer

DESCRIPTION :

The sliding window protocol is a crucial flow control mechanism used in the data link layer to ensure reliable and efficient data transmission between two communicating devices. It enables a sender to transmit multiple frames before needing an acknowledgment for the first frame, thereby utilizing network bandwidth more effectively. The "window" in this protocol represents the range of sequence numbers of frames that can be sent without awaiting an acknowledgment. As acknowledgments are received for transmitted frames, the window "slides" forward, allowing the sender to transmit new frames while keeping track of unacknowledged ones.

There are two main types of sliding window protocols: Go-Back-N and Selective Repeat. In Go-Back-N, if a frame is lost or corrupted, all subsequent frames are retransmitted, while in Selective Repeat, only the erroneous frames are retransmitted, improving efficiency. The sliding window protocol helps prevent data congestion by managing the flow of data based on network conditions and receiver capacity. It also ensures that frames are delivered in the correct sequence, providing both flow control and error control. This mechanism plays a vital role in maintaining the reliability and integrity of data transmission, particularly over unreliable or high-latency networks**.**

Program :

```
import java.util.Scanner;
import java.util.Random;

public class SlidingWindowProtocol {
   static int windowSize, totalFrames, sendBase, nextSeqNum;
   static boolean[] acknowledged;

   // Simulate sending a frame
   static void sendFrame(int frameNumber) {
System.out.println("Sending frame " + frameNumber);
   }

   // Simulate receiving acknowledgment with random loss
   static int receiveAck() {
      Random rand = new Random();
      // 90% chance of receiving ACK successfully
      if (rand.nextInt(100) < 90) {
         int ackFrame = sendBase + rand.nextInt(windowSize);
System.out.println("Acknowledgment received for frame " + ackFrame);
         return ackFrame;
      } else {
System.out.println("Acknowledgment lost!");
         return -1;  // No acknowledgment
      }
   }

   // Slide the window when an ACK is received
   static void slideWindow(int ackFrame) {
      while (sendBase<= ackFrame&&sendBase<totalFrames) {
         acknowledged[sendBase] = true;
System.out.println("Frame " + sendBase + " acknowledged.");
sendBase++;
      }
   }
```

```
    // Run the sliding window protocol
    static void runSlidingWindow() {
        while (sendBase<totalFrames) {
            // Send frames within the window
            while (nextSeqNum<sendBase + windowSize&&nextSeqNum<totalFrames) {
sendFrame(nextSeqNum);
nextSeqNum++;
            }

            // Simulate receiving an acknowledgment
            int ackFrame = receiveAck();
            if (ackFrame != -1) {
slideWindow(ackFrame);
            } else {
System.out.println("Timeout! Resending frames...");
            }

            // Simulate delay
            try {
Thread.sleep(1000);
            } catch (InterruptedException e) {
System.out.println(e);
            }
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input the window size and total number of frames
System.out.println("Enter the window size: ");
windowSize = sc.nextInt();

System.out.println("Enter the total number of frames to be sent: ");
totalFrames = sc.nextInt();

        // Initialize variables
        acknowledged = new boolean[totalFrames];
sendBase = 0;
nextSeqNum = 0;

        // Run the sliding window protocol
runSlidingWindow();

sc.close();
    }
}
```

Explanation:

Classes and Variables:

windowSize: Defines the size of the sliding window.

totalFrames: Total number of frames to be sent.

sendBase: The base of the current window (i.e., the first frame in the window).

nextSeqNum: The sequence number of the next frame to be sent.

acknowledged[]: An array to keep track of which frames have been acknowledged.

Methods:

sendFrame(): Simulates sending a frame by printing a message to the console.
receiveAck(): Simulates receiving an acknowledgment for a randomly selected frame. There's a 90% chance of receiving an acknowledgment and a 10% chance of loss.
slideWindow(): Moves the window forward once an acknowledgment is received.
runSlidingWindow(): The core logic that controls the flow of sending frames, receiving acknowledgments, and sliding the window.

Main Function:

The main() method prompts the user to enter the windowSize and totalFrames, initializes the protocol, and starts the sliding window process.

OUTPUT

```
Enter the window size:
4
Enter the total number of frames to be sent:
10
Sending frame 0
Sending frame 1
Sending frame 2
Sending frame 3
Acknowledgment received for frame 2
Frame 0 acknowledged.
Frame 1 acknowledged.
Frame 2 acknowledged.
Sending frame 4
Sending frame 5
Sending frame 6
Sending frame 7
Acknowledgment received for frame 5
Frame 3 acknowledged.
Frame 4 acknowledged.
Frame 5 acknowledged.
```

Key Features for Lab Demonstration:

Randomized Acknowledgment: Simulates realistic behavior where some frames may not be acknowledged due to network issues.
Window Sliding: Clearly demonstrates how the window slides forward after frames are acknowledged.
Delays and Timeouts: Timeouts are simulated using Thread.sleep() to mimic transmission delays.

Viva Question and Answers

1. *What is the Sliding Window Protocol*

Sliding Window Protocol is a flow control method used in data link layer to ensure that data is transmitted without overwhelming the receiver. It allows a sender to send multiple frames before needing an acknowledgment (ACK), using a "window" that defines the range of acceptable frames.

2. Why is flow control important in data communication?

Flow control is important because it ensures that the sender does not overwhelm the receiver with too much data at once. It prevents buffer overflow and ensures that data transmission occurs at an optimal rate.

3. What are the types of Sliding Window Protocols?

There are two main types of sliding window protocols:

1. Go-Back-N ARQ – The sender can send multiple frames, but if an error is detected, it resends all frames after the error.
2. Selective Repeat ARQ – The sender retransmits only the erroneous frame, not the entire set after an error.

4. How does Go-Back-N ARQ work?

In Go-Back-N, the sender can transmit several frames without waiting for an acknowledgment. However, if an error occurs or a frame is lost, the sender goes back to the sequence number of the unacknowledged frame and retransmits from that point onward.

5. What is Selective Repeat ARQ and how does it differ from Go-Back-N?

In Selective Repeat ARQ, only the erroneous frames are retransmitted rather than all frames after an error. This improves efficiency, as it minimizes the number of retransmissions. Unlike Go-Back-N, the receiver can accept and store out-of-order frames and only requests retransmission of missing frames.

6. What is the 'window size' in Sliding Window Protocol?

The window size refers to the number of frames the sender is allowed to send without receiving an acknowledgment. It dictates how much data can be in transit at one time. The size of the window is typically defined by the protocol and system configuration.

7. Explain the role of sequence numbers in Sliding Window Protocol?

Sequence numbers are used to uniquely identify frames. They help both the sender and receiver to keep track of which frames have been sent, received, and acknowledged. The sequence numbers wrap around once they reach a certain maximum, and this is known as the sequence number space.

8. How does acknowledgment work in Sliding Window Protocol?

Acknowledgments (ACKs) are sent by the receiver to indicate successful reception of frames. There are two types:

Cumulative Acknowledgment: ACK for a frame implicitly acknowledges all previous frames.

Selective Acknowledgment (used in Selective Repeat ARQ): ACK is sent only for specific frames that are received correctly.

9. What is the significance of the 'maximum window size'?

The maximum window size determines how many frames can be sent before waiting for an acknowledgment. It is limited by the number of bits available to represent the sequence number. For example, if the sequence number is represented by 3 bits, the window size cannot exceed $2^3 - 1 = 7$.

10. What are the advantages and disadvantages of Sliding Window Protocol?

Advantages:

- It allows efficient utilization of network resources by enabling multiple frames to be in transit.
- Minimizes the waiting time for acknowledgments, improving throughput.

Disadvantages:

- Go-Back-N can cause inefficiency when errors are frequent, as many frames might need retransmission.
- Requires both sender and receiver to maintain more state (such as sequence numbers and window size), which adds complexity.

These questions cover the basic to intermediate concepts of Sliding Window Protocols in the data link layer and are common topics

PROGRAM- 6

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

Bellman-Ford algorithm solves the single-source shortest-path problem in the general case in which edges of a given digraph can have negative weight as long as G contains no negative cycles.
This algorithm, like Dijkstra's algorithm uses the notion of edge relaxation but does not use with greedy method. Again, it uses d[u] as an upper bound on the distance d[u, v] from u to v. The algorithm progressively decreases an estimate d[v] on the weight of the shortest path from the source vertex s to each vertex v in V until it achieve the actual shortest-path. The algorithm returns Boolean TRUE if the given digraph contains no negative cycles that are reachable from source vertex s otherwise it returns Boolean FALSE.

BELLMAN-FORD (G, w, s)

1. INITIALIZE-SINGLE-SOURCE (G, s)
2. for each vertex i = 1 to V[G] - 1 do
3. for each edge (u, v) in E[G] do
4. RELAX (u, v, w)
5. For each edge (u, v) in E[G] do
6. if d[u] + w(u, v) < d[v] then
7. return FALSE
8. return TRUE
9. Asymptotic complexity:

• Average case (random data): O(|V ||E|)
• Worst case: O(|V ||E|)

Conclusion:
Thus, the Bellman-Ford algorithm runs in O(E) time.

Program

```
import java.util.*;
class DVT
	{
	public static void main(String args[])
		{
		int dist[][]=new int[20][20];
		int from[][]=new int[20][20];
		int costmat[][]=new int[10][10];
		int i,j,k,nodes;
		Scanner s=new Scanner(System.in);

		System.out.println("\nEnter the number of nodes :");
		nodes=s.nextInt();
		System.out.println("\nEnter the cost matrix :\n");
		for(i=1;i<=nodes;i++)
		{
		for( j=1;j<=nodes;j++)
		{
```

```
                costmat[i][j]=s.nextInt();
                costmat[i][i]=0;
                dist[i][j]=costmat[i][j];
                from[i][j]=j;

                }
        }
for( i=1;i<=nodes;i++)
        {
        for( j=1;j<=nodes;j++)
                {
                for( k=1;k<=nodes;k++)
                        {
                                if((dist[i][j])>dist[i][k]+dist[k][j])
                                        {
                                                dist[i][j]=dist[i][k]+dist[k][j];
                                                from[i][j]=k;
                                                }
                                        }
                                }
                        }
                for( i=1;i<=nodes;i++)
                        {
                        System.out.println("\n\nFromRouter Node :"+i);
                        System.out.println("\nDesti Node\tNext-Hop\tdistance\n");
                        for( j=1;j<=nodes;j++)
                                {
                                System.out.println(j +"\t \t "+from[i][j]+" \t\t"+dist[i][j]);
                                }
                        }
                        System.out.println("\n\n");
                }
        }
```

Output:

```
$javac  BellmanFord.java
$java BellmanFord
```

Enter the number of nodes :
3

Enter the cost matrix :

0 1 2
1 0 7
2 7 0

From Router Node :1

Desti Node	**Next-Hop**	**distance**
1	**1**	**0**
2	**2**	**1**
3	**3**	**2**

From Router Node :2

Desti Node	Next-Hop	distance
1	1	1
2	2	0
3	1	3

From Router Node :3

Desti Node	Next-Hop	distance
1	1	2
2	1	3
3	3	0

Enter the number of nodes :
4

Enter the cost matrix :

0 99 6 3
99 0 99 2
6 99 0 1
3 2 1 0

From Router Node: 1

Desti Node	Next-Hop	distance
1	1	0
2	4	5
3	4	4
4	4	3

From Router Node :2

Desti Node	Next-Hop	distance
1	4 5	
2 20		
3	4	3
4	4	2

From Router Node :3

Desti Node	Next-Hop	distance
1	4	4
2	4	3
3	3	0
4	4	1

From Router Node :4

Desti Node	Next-Hop	distance
1	3	
2	2	
3	3	1
4	4	0

Viva Questions and answers

1. **What is Bellman Ford Algorithm?**
   The Bellman-Ford algorithm computes single-source shortest paths in a weighted digraph (where some of the edge weights may be negative).

2. **Advantages of bellman-ford algorithm?**
   (a) Cost is minimized when building a network using BF algo.
   (b) Maximizes the performance of the system. Also finds min path weight.
   (c) It allows splitting of traffic between several paths. It thus increases system performance.

3. **What is the Difference between Routing Protocol and Routed Protocol?**
   Routing Protocol is responsible For Sending and Receiving a Route from One Router to another Router in the Network. When Ever We Will Enable a Routing protocol on the router, in That case Router Automatically creates A Route on the router. As for Example—RIP, IGRP, EIGRP, and OSPF
   Routed Protocol is responsible for provides the communication From Source device To Destination Device in the Network. As For Example—TCP/IP, IPX/SPX, apple talk

4. **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.

5. **What is strongly connected and weekly connected graph?**
   A digraph G for which each vertex u has a path to each other vertex v is said to be strongly connected.
   A digraph G whose underlying graph is connected but for which a pair of vertices u, v exists such that there is no path from u to v is said to be weakly connected.

6. **What is Complete Graph?**
   A complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge.

7. **List and define the different ways of representing a graph?**
   i. Adjacency Matrix
   ii. Adjacency List
   Adjacency Matrix is 2-Dimensional Array which has the size VxV, where V are the number of vertices in the graph.
   Adjacency List is the Array[] of Linked List, where array size is same as number of Vertices in the graph.

8. **What is minimum spanning tree?**

A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted (un)directed graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

9. What are the applications of MST?

Network design: telephone, electrical, hydraulic, TV cable, computer, road
Approximation algorithms for NP-hard problems traveling salesperson problem, Steiner tree.
Cluster analysis: k clustering problem can be viewed as finding an MST and deleting the k-1 most expensive edges.

PROGRAM 7

Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.

Socket is an interface which enables the client and the server to communicate and pass on information from one another. Sockets provide the communication mechanism between two computers using TCP. A client program creates a socket on its end of the communication and attempts to connect that socket to a server. When the connection is made, the server creates a socket object on its end of the communication. The client and the server can now communicate by writing to and reading from the socket.

Source Code:

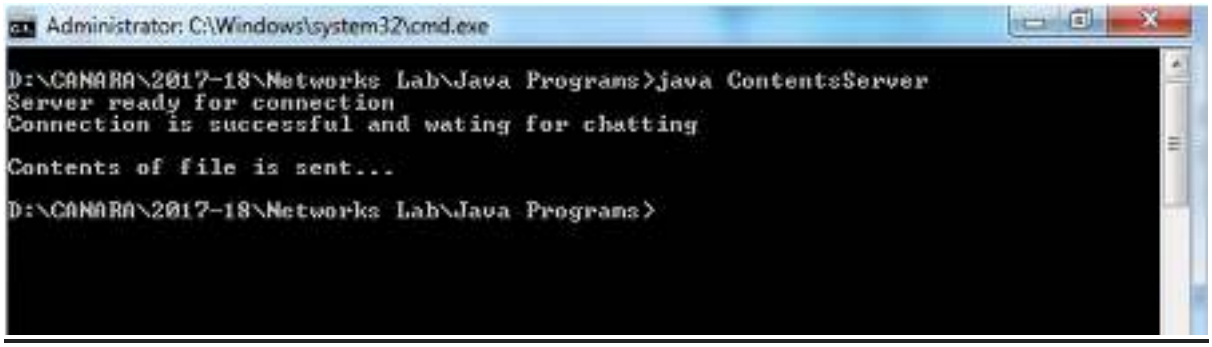
Server Program:

```
import java.net.*;
import java.io.*;
public class ContentsServer
{
 public static void main(String args[]) throws Exception
 {
// establishing the connection with the server
   ServerSocket sersock = new ServerSocket(4000);
   System.out.println("Server ready for connection");
   Socket sock = sersock.accept();  // binding with port: 4000
        System.out.println("Connection is successful and wating for chatting");

   // reading the file name from client
   InputStream istream = sock.getInputStream( );
         BufferedReader fileRead =new BufferedReader(new InputStreamReader(istream));
   String fname = fileRead.readLine( );
   // reading file contents
        BufferedReader contentRead = new BufferedReader(new FileReader(fname) );

 // keeping output stream ready to send the contents
   OutputStream ostream = sock.getOutputStream( );
   PrintWriter pwrite = new PrintWriter(ostream, true);

   String str;
        // reading line-by-line from file

   while((str = contentRead.readLine()) !=  null)
        {
                pwrite.println(str); // sending each line to client
    }
   System.out.println("Contents of the file is sent…");
   sock.close();  sersock.close();  // closing network sockets
   pwrite.close();  fileRead.close(); contentRead.close();
 }
```

Client Program:

```
import java.net.*;
import java.io.*;
public class ContentsClient
{
 public static void main( String args[ ] ) throws Exception
 {
   Socket sock = new Socket( "127.0.0.1", 4000);

    // reading the file name from keyboard. Uses input stream
   System.out.print("Enter the file name");
        BufferedReader keyRead = new BufferedReader(new InputStreamReader(System.in));
   String fname = keyRead.readLine();

         // sending the file name to server. Uses PrintWriter
   OutputStream  ostream = sock.getOutputStream( );
   PrintWriter pwrite = new PrintWriter(ostream, true);
   pwrite.println(fname);

         // receiving the contents from server.  Uses input stream

        System.out.println("Contents of the File:");
   InputStream istream = sock.getInputStream();
        BufferedReader socketRead = new BufferedReader(new InputStreamReader(istream));

   String str;
   while((str = socketRead.readLine())  !=  null)
        // reading line-by-line
   {
      System.out.println(str);
   }
   pwrite.close(); socketRead.close(); keyRead.close();
 }
}
```

Output:




Administrator: C:\Windows\system32\cmd.exe
D:\CANARA\2017-18\Networks Lab\Java Programs>javac ContentsClient.java
D:\CANARA\2017-18\Networks Lab\Java Programs>java ContentsClient
Enter the file name:
test.txt
Contents of the file:
CANARA ENGINEERING COLLEGE
MANGALURU-575219
D:\CANARA\2017-18\Networks Lab\Java Programs>

PROGRAM 8

Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.

A datagram socket is the one for sending or receiving point for a packet delivery service. Each packet sent or received on a datagram socket is individually addressed and routed. Multiple packets sent from one machine to another may be routed differently, and may arrive in any order.

Source Code:

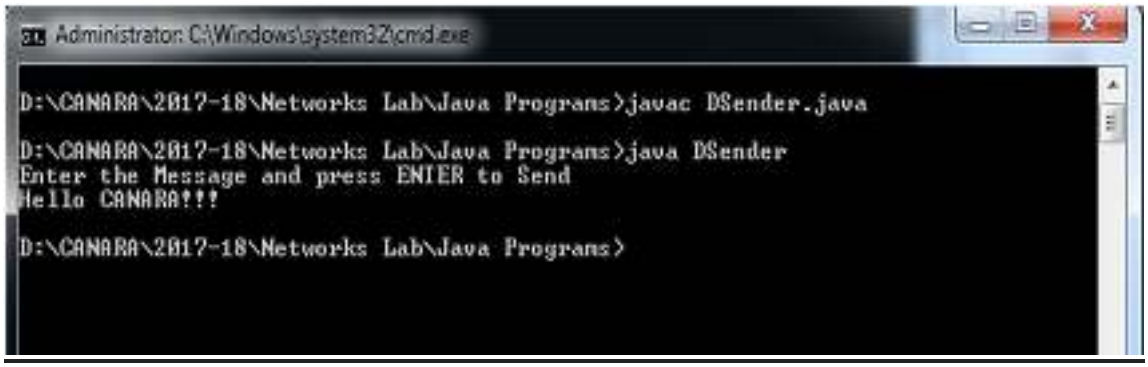
Server Program:

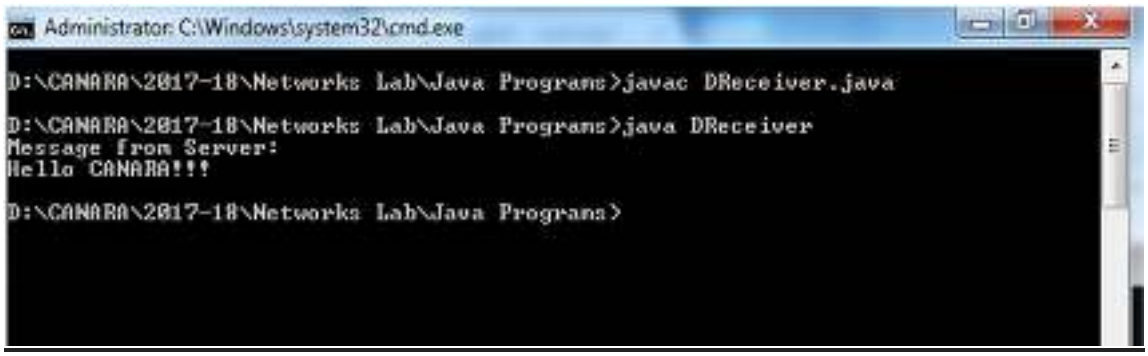
```
//DSender.java
import java.net.*;
import java.util.*;
public class DSender
{
        public static void main(String[] args) throws Exception
        {
                 DatagramSocket ds = new DatagramSocket();
                Scanner s=new Scanner(System.in);
                System.out.println("Enter the Message and press ENTER to Send");
                String str = s.nextLine();
                 InetAddress ip = InetAddress.getByName("127.0.0.1");

                DatagramPacket dp = new DatagramPacket(str.getBytes(), str.length(), ip, 21);
                ds.send(dp);
                ds.close();
        }
}
```

Client Program:

```
//DReceiver.java
import java.net.*;
public class DReceiver
{
         public static void main(String[] args) throws Exception
        {
                 DatagramSocket ds = new DatagramSocket(21);
                 byte[] buf = new byte[1024];
                          DatagramPacket dp = new DatagramPacket(buf, 1024);
                ds.receive(dp);
                        String str = new String(dp.getData(), 0, dp.getLength());
                System.out.println("Message from Server:");
                System.out.println(str);
                ds.close();
        }
        }
```

Output:


Administrator: C:\Windows\system32\cmd.exe
D:\CANARA\2017-18\Networks Lab\Java Programs>javac DSender.java
D:\CANARA\2017-18\Networks Lab\Java Programs>java DSender
Enter the Message and press ENTER to Send
Hello CANARA!!!
D:\CANARA\2017-18\Networks Lab\Java Programs>


Administrator: C:\Windows\system32\cmd.exe
D:\CANARA\2017-18\Networks Lab\Java Programs>javac DReceiver.java
D:\CANARA\2017-18\Networks Lab\Java Programs>java DReceiver
Message from Server:
Hello CANARA!!!
D:\CANARA\2017-18\Networks Lab\Java Programs>

PROGRAM 9

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

RSA is an example of public key cryptography. It was developed by Rivest, Shamir and Adelman. The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers.

The RSA algorithm's efficiency requires a fast method for performing the modular exponentiation operation. A less efficient, conventional method includes raising a number (the input) to a power (the secret or public key of the algorithm, denoted *e* and *d*, respectively) and taking the remainder of the division with *N*. A straight-forward implementation performs these two steps of the operation sequentially: first, raise it to the power and second, apply modulo. The RSA algorithm comprises of three steps, which are depicted below:

Key Generation Algorithm

1. Generate two large random primes, p and q, of approximately equal size such that their product n = p*q
2. Compute n = p*q and Euler's totient function (φ) phi(n) = (p-1)(q-1).
3. Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1.
4. Compute the secret exponent d, 1 < d < phi, such that e*d ≡ 1 (mod phi).
5. The public key is (e, n) and the private key is (d, n). The values of p, q, and phi should also be kept secret.

Encryption
Sender A does the following:-

1. Using the public key (e,n)
2. Represents the plaintext message as a positive integer M
3. Computes the cipher text C = M^e mod n.
4. Sends the cipher text C to B (Receiver).

Decryption
Recipient B does the following:-

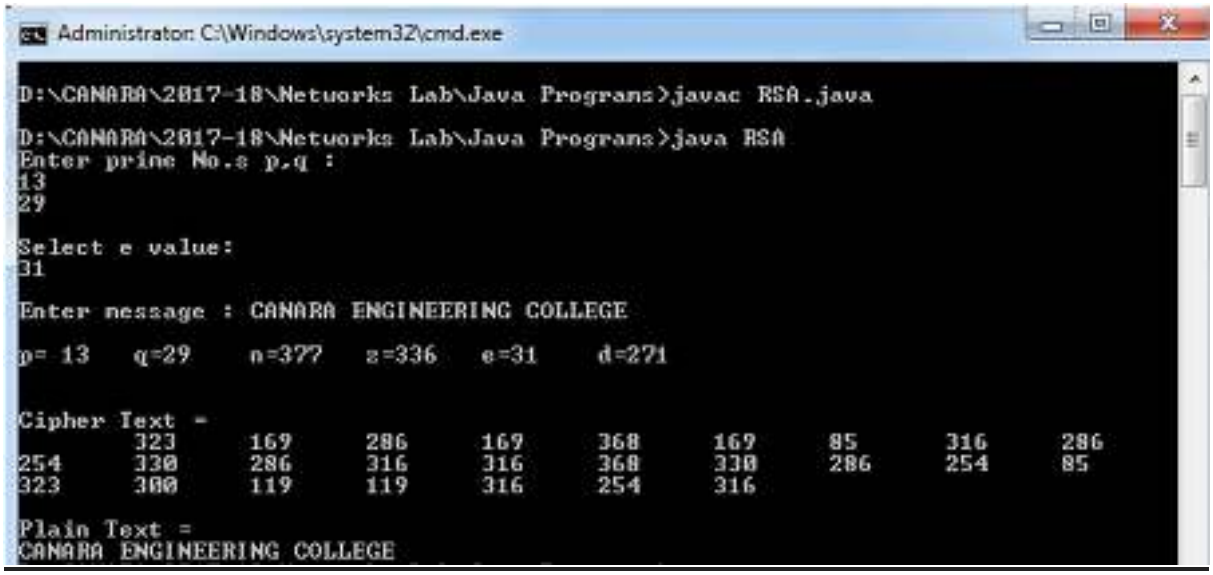
1. Uses his private key (d, n) to compute M = C^d mod n.
2. Extracts the plaintext from the integer representative m.

Source Code:

```
import java.util.*;
import java.io.*;
class RSA
{

        static int mult(int x,  int y,  int n)
        {
                int k=1;
                int j;
                 for (j=1; j<=y; j++) k = (k * x) % n;
                return ( int) k;
        }
```

```
        public static void main (String arg[])throws Exception
        {
                Scanner s=new Scanner(System.in);

                InputStreamReader r=new InputStreamReader(System.in);
                BufferedReader br=new BufferedReader(r);

                String msg1;
                int pt[]=new int[100];
                int ct[]=new int[100];
                int a,b, n, d, e,Z, p, q, i,temp,et;
                System.out.println("Enter prime No.s p,q :");
                p=s.nextInt();
                q=s.nextInt();
                n = p*q;
                Z=(p-1)*(q-1);

                System.out.println("\nSelect e value:");
                e=s.nextInt();

                System.out.printf("Enter message : ");
                msg1=br.readLine();
                char msg[]=msg1.toCharArray();

                for(i=0;i<msg.length;i++)
                  pt[i]=msg[i];
                for(d=1;d<Z;++d)
                        if(((e*d)%Z)==1) break;
                                System.out.println("p="+"
                                "+p+"\tq="+q+"\tn="+n+"\tz="+Z+"\te="+e+"\td="+d);

                System.out.println("\nCipher Text = ");
                for(i=0; i<msg.length; i++)
                         ct[i] = mult(pt[i], e,n);
                 for(i=0; i<msg.length; i++)
                         System.out.print("\t"+ct[i]);
                System.out.println("\nPlain Text = ");
                for(i=0; i<msg.length; i++)
                        pt[i] = mult(ct[i], d,n) ;
                for(i=0; i<msg.length; i++)
                System.out.print((char)pt[i]);
        }
}
```

Output:


Administrator: C:\Windows\system32\cmd.exe
D:\CANARA\2017-18\Networks Lab\Java Programs>javac RSA.java
D:\CANARA\2017-18\Networks Lab\Java Programs>java RSA
Enter prime No.s p,q :
13
29
Select e value:
31
Enter message : CANARA ENGINEERING COLLEGE
p= 13 q=29 n=377 z=336 e=31 d=271
Cipher Text =
323 169 286 169 368 169 85 316 286
254 330 286 316 316 368 330 286 254 85
323 300 119 119 316 254 316
Plain Text =
CANARA ENGINEERING COLLEGE

PROGRAM-10

Write a program for congestion control using leaky bucket algorithm.

Description

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

From network perspective, leaky bucket consists of a finite queue (bucket) where all the incoming packetsare stored in case there is space in the queue, otherwise the packets are discarded. In order to regulate theoutput flow, leaky bucket transmits one packet from the queue in a fixed time (e.g. at every clock tick). Inthe 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)).

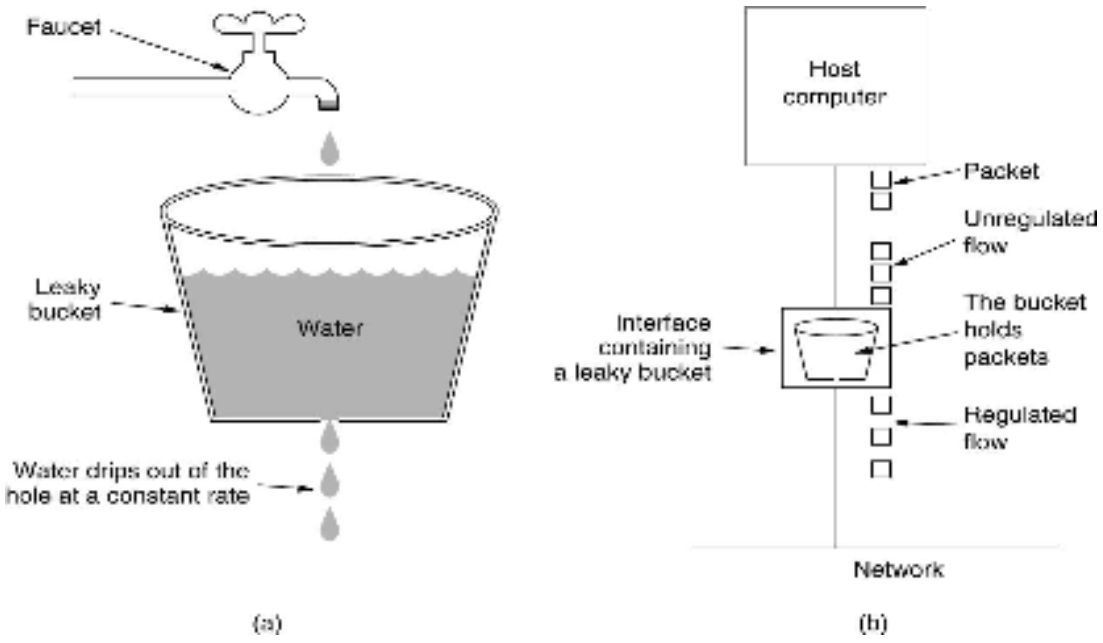


Figure :20 a) Leaky Bucket with water b) Leaky Bucket with packets

Program:

```
import java.util.*;
class pg12
{
        public static void main(String args[])
                {

                Scanner sc=new Scanner(System.in);
                int i,size,nop,opr,temp=0; int[]
        datarate=new int[100];
        System.out.println("enter the bucket size");
        size=sc.nextInt();
        System.out.println("enter the number of packets");
```

```
nop=sc.nextInt();
System.out.println("enter the dara rate");

for(i=0;i<nop;i++)
        datarate[i]=sc.nextInt();


System.out.println("enter the output rate");
opr=sc.nextInt();


for(i=0;i<nop;i++)
        {
                if(datarate[i]>size)
                        System.out.println("bucket overflow");

                else
                        {

                        temp=datarate[i];

                        while(temp>opr)
                        {
                                System.out.println("packet transmission"+opr);
                                temp=temp-opr;
                                }
                                System.out.println("packet transmission"+temp);
                                }
                        }


                }
        }
```

Output

```
$ gedit Leakybucket.java
$ javac Leakybucket.java
$ java Leakybucket


Enter the bucket size=
50
Enter the number of packets=
3
Enter the Data rate=
35 76 10
Enter output rate=
10
Packet transmitted 10
Packet transmitted 10
Packet transmitted 10
Packet transmitted 5
Bucket overflow
76
```

Packet transmitted 10

Viva Questions and answers

1. What is leaky bucket algorithm?
   In this algorithm, an interface is connected between a host and the network and the interface has a finite buffer space. If a packet arrives at the interface when the buffer is full then it is discarded by the interface. It is called leaky bucket algorithm because the outgoing rate of packet from the buffer is constant no matter how much incoming traffic is there at the interface.

2. Define a network congestion.
   When two or more nodes would simultaneously try to transmit packets to one node there is high probability that the number of packets would exceed the packet handling capacity of the network and lead to congestion.

3. List some ways to deal with congestion.
   Several ways to handle congestion
   Packet elimination Flow Control Buffer allocation Choke packets

4. What is meant by choke packets?
   A specialized packet that is used for flow control along a network. A router detects congestion by measuring the percentage of buffers in use, line utilization and average queue lengths. When it detects congestion, it sends chokepackets across the network to all the data sources associated with the congestion.

5. What is packet dropping?
   When a buffer becomes full a router can drop waiting packets- if not coupled with some other techniques, this can lead to greater congestion through retransmission.

6. What are the different techniques used to improve network congestion? Buffering
   Over-provisioning Traffic Shaping Packet scheduling

7. What is traffic shaping?
   Traffic shaping, also known as packet shaping, Quality of Service (QoS) or bandwidth management, is the manipulation and prioritization of network traffic to reduce the impact of heavy users or machines from effecting other users.

8. What is the advantage of leaky bucket algorithm?
   Leaky-bucket algorithm is needed to regulate the data flow. It is used to improve the lifetime of network and prevent from traffic flows increases performance of our network.

9. What are the different applications of leaky bucket algorithm?
   Leaky bucket algorithm is used to regulate the traffic. It can be used in telecom network either as traffic shaping or traffic policing.
   In Telecom networks, the control of traffic (i.e. sending more packets into the network) is done using this algorithm
   Network monitors traffic flows continuously to ensure they meet their trafficcontract. The process of monitoring and enforcing the traffic flow is called policing