BMS Institute of Technology and Management

Yelahanka Bengaluru -560064

ISE Dept.
Transform Here

Department of Information Science and Engineering

COMPUTER NETWORK LABORATORY Manual

(Subject Code: 18CSL57)

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

BMS INSTITUTE OF TECHNOLOGYAND MANAGEMENT

VISION

To emerge as one of the finest technical institutions of higher learning, to develop engineering professionals who are technically competent, ethical and environment friendly for betterment of the society.

MISSION

Accomplish a stimulating learning environment through high quality academic instruction, innovation and industry-institute interface.

ABOUT INSTITUTION

In view of the growing demand for technical education and with the goal of establishing a premier technical education on par with international standards, a new technical institution by name 'BMS Institute of Technology and Management' was established in 2002. Currently, BMSIT & M offers eight UG, three PG programs and Ph.D. /M.Sc. (Engg.) in seven disciplines. BMSIT & M considers research to be of equal importance as academics for the betterment of an institution. Research culture has been embraced well by the faculty members and research scholars at BMSIT and M. In this report, we present an overview of the research activities of Information Science and Engineering, BMSIT & M.

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

VISION

Emerge as Centre of Learning in the field of information Science & engineering with technical competency to serve the society.

MISSION

To provide excellent learning environment Through Balanced Curriculum, Best Teaching Methods, Innovation, Mentoring and Industry Institute Interaction.

ABOUT DEPARTMENT

The Department of Information Science and Engineering started in the Year 2010 with an approved intake of 60 and enhanced to 180 from the academic year 2019-20. The Department has qualified and professionally dedicated faculty member practice OBE in the academic deliverables. The faculties have published research articles in various National, International, IEEE Conferences and Journals.

The department has modern laboratories to serve the teaching and research needs of the students as well as faculty members. The Department has been organizing conferences, workshops, expert lectures and student centric activities to encourage students and faculty to install lifelong learning. Few of our students are working for consultancy projects along with few faculty members. The staffs are encouraged to attend the 10 days internship to bridge the gap between the academics and industry. The department has admirable research ambiance.

Laboratory in-charges (2020-21)

1. Mr. Gireesh Babu C N, Assistant Professor, ISE
2. Mrs. Amudha, Programmer, ISE

Laboratory Instructors

1. Arpitha H. M, Assistant Instructor

COURSE OUTCOMES

Students will be able to:

CO1: Apply the Knowledge of Data Communications

CO2: Analyze various networking protocols.

CO3: Implement and evaluate networking protocols in NS2 / NS3 and JAVA programming language

CO4: Communicate effectively with the engineering community & society with respect to computer networks.

PROGRAMME SPECIFIC OUTCOMES

Graduates will be able to:-

PSO-1: Apply the knowledge of information technology to develop software solutions.

PSO-2: Design and Develop hardware systems, manage and monitor resources in the Product life cycle.

PROGRAMME EDUCATION OBJECTIVES

Graduates will be able to:-

PEO-1: Successful professional career in Information Technology Industry.

PEO-2: Pursue higher studies & research for advancement of knowledge in IT industry.

PEO-3: Exhibit professionalism and team work with social concern.

PROGRAMME OUTCOMES

Program outcomes are narrower statements that describe what students are expected to know and be able to do by the time of graduation. These relate to the skills, knowledge and behavior. Bachelor of Engineering Graduation students of Information science and Engineering program at B M S Institute of Technology will attain the following program outcomes.

Program Outcomes:

After the successful completion of the course, the graduate will be able to

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: 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.

PO4: 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.

PO5: 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.

PO6: 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.

PO7: 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.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: 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.

PO11: 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.

PO12: 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.

CONTENTS:

COMPUTER NETWORK LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)
SEMESTER – V

Laboratory Code	15CSL57	IA Marks	20
Number of Lecture hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives:	This course will enable students to

- Demonstrate operation of network and its management commands
- Simulate and demonstrate the performance of GSM and CDMA
- Implement data link layer and transport layer protocols.

Description (If any):

For the experiments below modify the topology and parameters set for the experiment and take multiple rounds of reading and analyze the results available in log files. Plot necessary graphs and conclude. Use NS2/NS3.

Lab Experiments:

PART A

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. 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.

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

4. Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.

5. Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.

6. Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or equivalent environment.

PART B: Implement the following in Java:

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

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

9. 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.

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

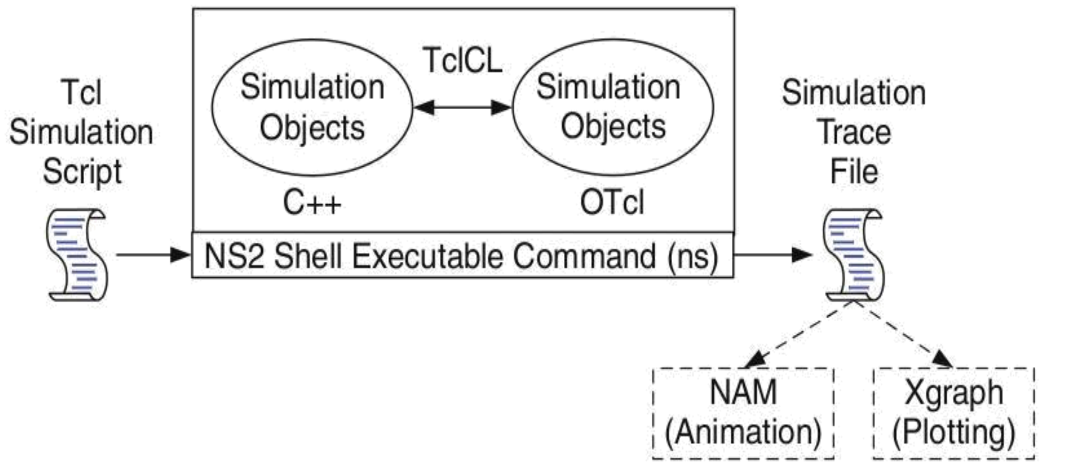
11. Write a program for simple RSA algorithm to encrypt and decrypt the data.

12. Write a program for congestion control using leaky bucket algorithm.

Introduction to NS-2

- Widely known as NS2, is simply an event driven simulation tool.
- 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 and simulating their corresponding behaviours.

Basic Architecture of NS2



Tcl scripting

- Tcl 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.

Basics of TCL

Syntax: command arg1 arg2 arg3

- **Hello World!**
```
puts stdout{Hello, World!}
Hello, World!
```
- **Variables** Command Substitution
```
 set a 5      set len [string length foobar]
set b $a      set len [expr [string length foobar] + 9]
```

- **Simple Arithmetic** expr
```
7.2 / 4
```
- **Procedures**
```
proc Diag {a b} {
set c [expr sqrt($a * $a + $b * $b)]
return $c }
        puts "Diagonal of a 3, 4 right triangle is [Diag 3 4]"
    Output: Diagonal of a 3, 4 right triangle is 5.0
```

- **Loops**

```
while{$i < $n} {                for {set i 0} {$i < $n} {incr i} {
. . .                           . . .
          }                     }
```

Wired TCL Script Components

- Create the event scheduler
- Open new files & turn on the tracing
- Create the nodes
- Setup the links
- Configure the traffic type (e.g., TCP, UDP, etc)
- Set the time of traffic generation (e.g., CBR, FTP)
- Terminate the simulation

NS Simulator Preliminaries.

1. Initialization and termination aspects of the ns simulator.
2. Definition of network nodes, links, queues and topology.
3. Definition of agents and of applications.
4. The nam visualization tool.
5. Tracing and random variables.

Initialization and Termination of TCL Script in NS-2 An ns simulation starts with the command

```
set ns [new Simulator]
```

Which is thus the first line in the tcl script? This line declares a new variable as using the set command, you can call this variable as you wish, In general people declares it as ns because it is an instance of the Simulator class, so an object the code[new Simulator] is indeed the installation of the class Simulator using the reserved word new.

In order to have output files with data on the simulation (trace files) or files used for visualization (nam files), we need to create the files using “open” command:

```
#Open the Trace file            set tracefile1 [open out.tr w]
                                $ns trace-all $tracefile1

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

The above creates a data trace file called “out.tr” and a nam visualization trace file called “out.nam”. Within the tcl script, these files are not called explicitly by their names, but instead by pointers that are declared above and called “tracefile1” and “namfile” respectively. Remark that they begins with a # symbol. The second line open the file “out.tr” to be used for writing, declared with the letter “w”. The third line uses a simulator method called trace-all that have as parameter the name of the file where the traces will go.

The last line tells the simulator to record all simulation traces in NAM input format. It also gives the file name that the trace will be written to later by the command $ns flush-trace. In our case, this will be the file pointed at by the pointer “$namfile”,i.e the file “out.tr”.

The termination of the program is done using a “finish” procedure.

#Define a ‘finish’ procedure

```
Proc finish { } {
global ns tracefile1 namfile
$ns flush-trace
Close $tracefile1
Close $namfile
Exec nam out.nam &
Exit 0
```

The word proc declares a procedure in this case called **finish** and without arguments. The word **global** is used to tell that we are using variables declared outside the procedure. The simulator method “**flush-trace”** will dump the traces on the respective files. The tcl command “**close”** closes the trace files defined before and **exec** executes the nam program for visualization. The command **exit** will ends the application and return the number 0 as status to the system. Zero is the default for a clean exit. Other values can be used to say that is a exit because something fails.

At the end of ns program we should call the procedure “finish” and specify at what time the termination should occur. For example,

```
$ns at 125.0 “finish”
```

will be used to call “**finish**” at time 125sec.Indeed, the **at** method of the simulator allows us to schedule events explicitly.

The simulation can then begin using the command

```
$ns run
```

Definition of a network of links and nodes

The way to define a node is

```
set n0 [$ns node]
```

We created a node that is printed by the variable n0. When we shall refer to that node in the script we shall thus write $n0.

Once we define several nodes, we can define the links that connect them. An example of a definition of a link is:

```
$ns duplex-link $n0 $n2 10Mb 10ms DropTail
```

Which means that $n0 and $n2 are connected using a bi-directional link that has 10ms of propagation delay and a capacity of 10Mb per sec for each direction.

To define a directional link instead of a bi-directional one, we should replace “duplex-link” by “simplex-link”.

In NS, an output queue of a node is implemented as a part of each link whose input is that node. The definition of the link then includes the way to handle overflow at that queue.

In our case, if the buffer capacity of the output queue is exceeded then the last packet to arrive is dropped. Many alternative options exist, such as the RED (Random Early Discard) mechanism, the FQ (Fair Queuing), the DRR (Deficit Round Robin), the stochastic Fair Queuing (SFQ) and the CBQ (which including a priority and a round-robin scheduler).

In ns, an output queue of a node is implemented as a part of each link whose input is that node. We should also define the buffer capacity of the queue related to each link. An example would be:

```
#set Queue Size of link (n0-n2) to 20

$ns queue-limit $n0 $n2 20
```

Agents and Applications

We need to define routing (sources, destinations) the agents (protocols) the application that use them.

FTP over TCP

TCP is a dynamic reliable congestion control protocol. It uses Acknowledgements created by the destination to know whether packets are well received.

There are number variants of the TCP protocol, such as Tahoe, Reno, NewReno, Vegas.
The type of agent appears in the first line:

```
set tcp [new Agent/TCP]
```

The command **$ns attach-agent $n0 $tcp** defines the source node of the tcp connection.

The command

```
set sink [new Agent/TCPSink]
```

Defines the behaviour of the destination node of TCP and assigns to it a pointer called sink. **#Setup a UDP connection**

```
set udp [new Agent/UDP]

$ns attach-agent $n1 $udp

set null [new Agent/Null]

$ns attach-agent $n5 $null

$ns connect $udp $null

$udp set fid_2
```

#setup a CBR over UDP connection

```
set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set packetsize_ 100

$cbr set rate_ 0.01Mb

$cbr set random_ false
```

Above shows the definition of a CBR application using a UDP agent

The command **$ns attach-agent $n4 $sink** defines the destination node. The command **$ns connect $tcp $sink** finally makes the TCP connection between the source and destinationnodes.TCP has many parameters with initial fixed defaults values that can be changed if mentioned explicitly.

For example, the default TCP packet size has a size of 1000bytes.This can be changed to another value, say 552bytes, using the command **$tcp set packetSize_ 552**. When we have several flows, we may wish to distinguish them so that we can identify them with different colors in the visualization part. This is done by the command **$tcp set fid_1** that assigns to the TCP connection a flow identification of "1".We shall later give the flowidentification of "2" to the UDP connection.

CBR over UDP

A UDP source and destination is defined in a similar way as in the case of TCP.

Instead of defining the rate in the command $cbr set rate_ 0.01Mb, one can define the time interval between transmission of packets using the command.

```
$cbr set interval_ 0.005
```

The packet size can be set to some value using

```
$cbr set packetSize_ <packet size>
```

Scheduling Events

NS is a discrete event based simulation. The tcp script defines when event should occur. The initializing command set ns [new Simulator] creates an event scheduler, and events are then scheduled using the format:

```
$ns at <time><event>
```

The scheduler is started when running ns that is through the command $ns run. The beginning and end of the FTP and CBR application can be done through the following command

```
$ns at 0.1 "$cbr start"

$ns at 1.0 " $ftp start"

$ns at 124.0 "$ftp stop"
```

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.

Syntax:

```
Xgraph [options] file-name
```

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".

Awk- An Advanced

awk is a programmable, pattern-matching, and processing tool available in UNIX. It works equally well with text and numbers.

awk is not just a command, but a programming language too. In other words, awk utility is a pattern scanning and processing language. It searches one or more files to see if they contain lines that match specified patterns and then perform associated actions, such as writing the line to the standard output or incrementing a counter each time it finds a match.

Syntax: **awk option ‘selection_criteria {action}’ file(s)**

Here, selection_criteria filters input and select lines for the action component to act upon. The selection_criteria is enclosed within single quotes and the action within the curly braces. Both the selection_criteria and action forms an awk program.

Example: $ awk ‘/manager/ {print}’ emp.lst

Variables

Awk allows the user to use variables of there choice. You can now print a serial number, using the variable kount, and apply it those directors drawing a salary exceeding 6700:

$ awk –F”|” ‘$3 == “director” && $6 > 6700 { count
=count+1
printf “ %3f %20s %-12s %d\n”, count,$2,$3,$6 }’ empn.lst THE –f
OPTION: STORING awk PROGRAMS INA FILE

You should holds large awk programs in separate file and provide them with the awk extension for easier identification. Let’s first store the previous program in the file empawk.awk:

$ cat empawk.awk

Observe that this time we haven’t used quotes to enclose the awk program. You can now use awk with the –f *filename* option to obtain the same output:

```
Awk –F”|” –f empawk.awk empn.lst
```

THE BEGIN AND END SECTIONS

Awk statements are usually applied to all lines selected by the address, and if there are no addresses, then they are applied to every line of input. But, if you have to print something before processing the first line, for example, a heading, then the BEGIN section can be used gainfully. Similarly, the end section useful in printing some totals after processing is over. The BEGIN and END sections are optional and take the form

BEGIN {action}
END {action}

These two sections, when present, are delimited by the body of the awk program. You can use them to print a suitable heading at the beginning and the average salary at the end.

BUILT-IN VARIABLES

Awk has several built-in variables. They are all assigned automatically, though it is also possible for a user to reassign some of them. You have already used NR, which signifies the record number of the current line. We’ll now have a brief look at some of the other variable. ***The FS Variable:***as stated elsewhere, awk uses a contiguous string of spaces as the defaultfield delimiter.

FS redefines this field separator, which in the sample database happens to be the |. When used at all, it must occur in the BEGIN section so that the body of the program knows its value before it starts processing:

BEGIN {FS="|"}
This is an alternative to the –F option which does the same thing.
The OFS Variable:when you used the print statement with comma-separated arguments, eachargument was separated from the other by a space. This is awk's default output field separator, and can reassigned using the variable OFS in the BEGIN section:

BEGIN { OFS="~" }
When you reassign this variable with a ~ (tilde), awk will use this character for delimiting the print arguments. This is a useful variable for creating lines with delimited fields.
The NF variable:NF comes in quite handy for cleaning up a database of lines that don'tcontain the right number of fields. By using it on a file, say emp.lst, you can locate those lines not having 6 fields, and which have crept in due to faulty data entry:
$awk 'BEGIN {FS = "|"}
NF! =6 {
Print "Record No ", NR, "has", "fields"}' empx.lst
The FILENAME Variable:FILENAME stores the name of the current file being processed.Like grep and sed, awk can also handle multiple filenames in the command line. By default, awk doesn't print the filename, but you can instruct it to do so:
'$6<4000 {print FILENAME, $0 }'
With FILENAME, you can device logic that does different things depending on the file that is processed.

NS2 Installation

- NS2 is a free simulation tool.
- It runs on various platforms including UNIX (or Linux), Windows, and Mac systems.
- NS2 source codes are distributed in two forms: the all-in-one suite and the component-wise.
- 'all-in-one' package provides an "install" script which configures the NS2 environment and creates NS2 executable file using the "make" utility.

<u>NS-2 installation steps in Linux</u>

- Go to **Computer File System** now paste the zip file **"ns-allinone-2.34.tar.gz"** into opt folder.
- Now **unzip** the file by typing the following **command** [root@localhost opt] # **tar -xzvf ns-allinone-2.34.tar.gz**
- After the files get extracted, we get ns-allinone-2.34 folder as well as zip file ns-allinone-2.34.tar.gz
  [root@localhost opt] # **ns-allinone-2.34 ns-allinone-2.34.tar.gz**
- Now go to ns-allinone-2.33 folder and install it
  [root@localhost opt] # **cd ns-allinone-2.34**
  [root@localhost ns-allinone-2.33] # **./install**
- Once the installation is completed successfully we get certain pathnames in that terminal which must be pasted in **".bash_profile"** file.

- First **minimize the terminal** where installation is done and **open a new terminal** and open the file **".bash_profile"**
  [root@localhost ~] # **vi .bash_profile**
- When we open this file, we get a line in that file which is shown below
  PATH=$PATH:$HOME/bin

To this line we must paste the path which is present in the previous terminal where **ns wasinstalled**. First put **":"** then paste the path in-front of bin. That path is shown below.
":/opt/ns-allinone-2.33/bin:/opt/ns-allinone-2.33/tcl8.4.18/unix:/opt/ns-allinone-2.33/tk8.4.18/unix".

- In the next line type **"LD_LIBRARY_PATH=$LD_LIBRARY_PATH:"** and paste the **two paths** separated by **":"** which are present in the previous terminal i.e **Important notices section (1)**

"/opt/ns-allinone-2.33/otcl-1.13:/opt/ns-allinone-2.33/lib"

- In the next line type **"TCL_LIBRARY=$TCL_LIBRARY:"** and paste the path which is present in previous terminal i.e **Important Notices section (2)**

"/opt/ns-allinone-2.33/tcl8.4.18/library"

- In the next line type **"export LD_LIBRARY_PATH"**
- In the next line type **"export TCL_LIBRARY"**
- The next two lines are already present the file **"export PATH"** and **"unset USERNAME"**
- **Save the program (ESC + shift : wq and press enter)**
- Now in the terminal where we have opened **.bash_profile** file, type the following command to **check if path is updated correctly or not**

```
[root@localhost ~] # vi .bash_profile [root@localhost
~] # source .bash_profile
```

- If **path is updated properly**, then we will **get the prompt** as shown below

```
[root@localhost ~] #
```

- Now open the previous terminal where you have installed **ns**

```
[root@localhost  ns-allinone-2.33] #
```

- Here we need to configure three packages **"ns-2.33", "nam-1.13" and "xgraph-12.1"**
- **First**, configure **"ns-2.33"** package as shown below

```
[root@localhost ns-allinone-2.33] # cd ns-2.33
[root@localhost ns-2.33] # ./configure [root@localhost
ns-2.33] # make clean [root@localhost ns-2.33] # make
[root@localhost ns-2.33] # make install [root@localhost
ns-2.33] # ns
                         %
```

- If we get **"%"** symbol it indicates that **ns-2.33 configuration** was **successful**.
- **Second,** configure **"nam-1.13"** package as shown below

```
[root@localhost ns-2.33] # cd . . [root@localhost ns-
allinone-2.33] # cd nam-1.13 [root@localhost nam-1.13] #
./configure [root@localhost nam-1.13] # make clean
[root@localhost nam-1.13] # make [root@localhost nam-
1.13] # make install [root@localhost nam-1.13] # ns
```

PART-A

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.

```
#===================================
#     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 p1.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open p1.nam w]
$ns namtrace-all $namfile

#===================================
#        Nodes Definition
#===================================
#Create 3 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

#===================================
#        Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50
$ns duplex-link $n1 $n2 0.1Mb 10ms DropTail
$ns queue-limit $n1 $n2 50

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

#===================================
#        Agents Definition
#===================================
#Setup a UDP connection
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set null1 [new Agent/Null]
$ns attach-agent $n2 $null1
$ns connect $udp0 $null1
$udp0 set packetSize_ 1500
```

```
#===================================
#       Applications Definition
#===================================
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 1000
$cbr0 set rate_ 1.0Mb
$cbr0 set random_ null
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr0 stop"

#===================================
#       Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
global ns tracefile namfile
$ns flush-trace
close $tracefile
close $namfile
exec nam p1.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
```

AWK file: (Open a new editor using "vi command" and write awk file and save with ".awk" extension)

#immediately after BEGIN should open braces „{„

```
BEGIN{ c=0;}
{
   if($1= ="d")
  {        c++;
           printf("%s\t%s\n",$5,$11);
   }
 }
END{ printf("The number of packets dropped is %d\n",c); }
```

Steps for execution

- Open gedit editor and type program. Program name should have the extension " .tcl "

  `[root@localhost ~]# gedit lab1.tcl`
- Save the program and close the file.
- Open gedit editor and type awk program. Program name should have the extension ".awk "

  `[root@localhost ~]# gedit lab1.awk`
- Save the program and close the file.
- Run the simulation program

  `[root@localhost~]# ns lab1.tcl`
- 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.
- After simulation is completed run awk file to see the output ,

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

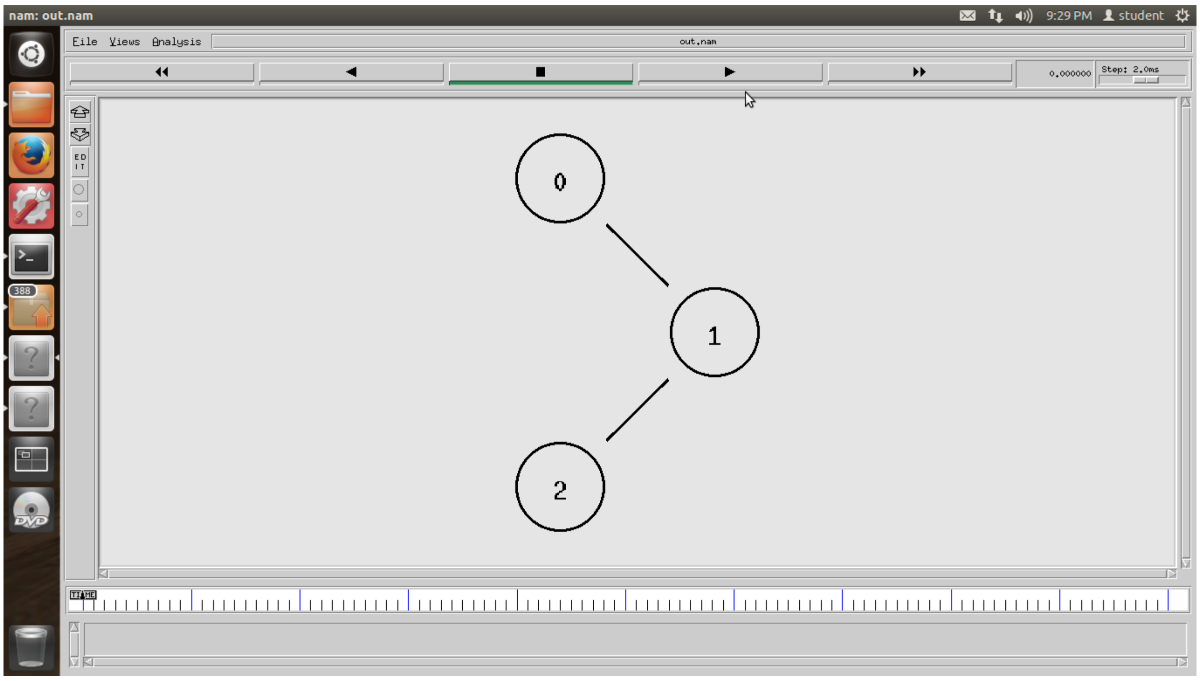
- To see the trace file contents open the file as ,

```
[root@localhost~]# gedit lab1.tr
```

Trace file contains 12 columns:

Event type, Event time, From Node, To Node, Packet Type, Packet Size, Flags (indicated by --------), Flow ID, Source address, Destination address, Sequence ID, Packet ID

<u>Output</u>



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.**

```
set ns [ new Simulator ]
set nf [ open lab2.nam w ]
$ns namtrace-all $nf
set tf [ open lab2.tr w ]
$ns trace-all $tf
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n4 shape box
$ns duplex-link $n0 $n4 1005Mb 1ms DropTail
$ns duplex-link $n1 $n4 50Mb 1ms DropTail
$ns duplex-link $n2 $n4 2000Mb 1ms DropTail
$ns duplex-link $n3 $n4 200Mb 1ms DropTail
```

```
 $ns duplex-link $n4 $n5 1Mb 1ms DropTail
Set p1 [new Agent/Ping]
$ns attach-agent $n0 $p1
$p1 set packetSize_ 50000
$p1 set interval_ 0.0001

set p2 [new Agent/Ping]
$ns attach-agent $n1 $p2
set p3 [new Agent/Ping]
$ns attach-agent $n2 $p3
$p3 set packetSize_ 30000
$p3 set interval_ 0.00001
set p4 [new Agent/Ping]
$ns attach-agent $n3 $p4
set p5 [new Agent/Ping]
$ns attach-agent $n5 $p5
$ns queue-limit $n0 $n4 5
$ns queue-limit $n2 $n4 3
$ns queue-limit $n4 $n5 2
Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts "node [$node_ id] received answer from $from with round trip time $rtt msec" }
# please provide space between $node_ and id. No space between $ and from. No #space
between and $ and rtt */
$ns connect $p1 $p5 $ns
connect $p3 $p4 proc finish {
}
{ global ns nf tf
$ns flush-trace close $nf
close $tf
exec nam lab2.nam & exit 0
}
$ns at 0.1 "$p1 send"
$ns at 0.2 "$p1 send"
$ns at 0.3 "$p1 send"
$ns at 0.4 "$p1 send"
$ns at 0.5 "$p1 send"
$ns at 0.6 "$p1 send"
$ns at 0.7 "$p1 send"
$ns at 0.8 "$p1 send"
$ns at 0.9 "$p1 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"
```

```
        $ns at 1.4 "$p1 send"
        $ns at 1.5 "$p1 send"
        $ns at 1.6 "$p1 send"
        $ns at 1.7 "$p1 send"
        $ns at 1.8 "$p1 send"
        $ns at 1.9 "$p1 send"
        $ns at 2.0 "$p1 send"
        $ns at 2.1 "$p1 send"
        $ns at 2.2 "$p1 send"
        $ns at 2.3 "$p1 send"
        $ns at 2.4 "$p1 send"
        $ns at 2.5 "$p1 send"
        $ns at 2.6 "$p1 send"
        $ns at 2.7 "$p1 send"
        $ns at 2.8 "$p1 send"
        $ns at 2.9 "$p1 send"
        $ns at 0.1 "$p3 send"
        $ns at 0.2 "$p3 send"
        $ns at 0.3 "$p3 send"
        $ns at 0.4 "$p3 send"
        $ns at 0.5 "$p3 send"
        $ns at 0.6 "$p3 send"
        $ns at 0.7 "$p3 send"
        $ns at 0.8 "$p3 send"
        $ns at 0.9 "$p3 send"
        $ns at 1.0 "$p3 send"
        $ns at 1.1 "$p3 send"
        $ns at 1.2 "$p3 send"
        $ns at 1.3 "$p3 send"
        $ns at 1.4 "$p3 send"
        $ns at 1.5 "$p3 send"
        $ns at 1.6 "$p3 send"
        $ns at 1.7 "$p3 send"
        $ns at 1.8 "$p3 send"
        $ns at 1.9 "$p3 send"
        $ns at 2.0 "$p3 send"
        $ns at 2.1 "$p3 send"
        $ns at 2.2 "$p3 send"
        $ns at 2.3 "$p3 send"
        $ns at 2.4 "$p3 send"
        $ns at 2.5 "$p3 send"
        $ns at 2.6 "$p3 send"
$ ns at 2.7 "$p3 send"
$ns at 2.8 "$p3 send"
```

```
$ns at 2.9 "$p3 send"
$ns at 3.0 "finish"
$ns run
```

AWK file: (Open a new editor using “gedit command” and write awk file and save with “.awk” extension)

```
BEGIN{
drop=0;
}
{
if($1= ="d" )
{
drop++;
}
} END{
printf("Total number of %s packets dropped due to congestion =%d\n",$5,drop);
}
```

Steps for execution

Open gedit editor and type program. Program name should have the extension “ .tcl ”

```
[root@localhost ~]# gedit lab2.tcl
```

Save the program and close the file.

Open gedit editor and type awk program. Program name should have the extension “.awk ”

```
[root@localhost ~]# gedit lab2.awk
```

Save the program and close the file.

Run the simulation program

```
[root@localhost~]# ns lab2.tcl
```

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.

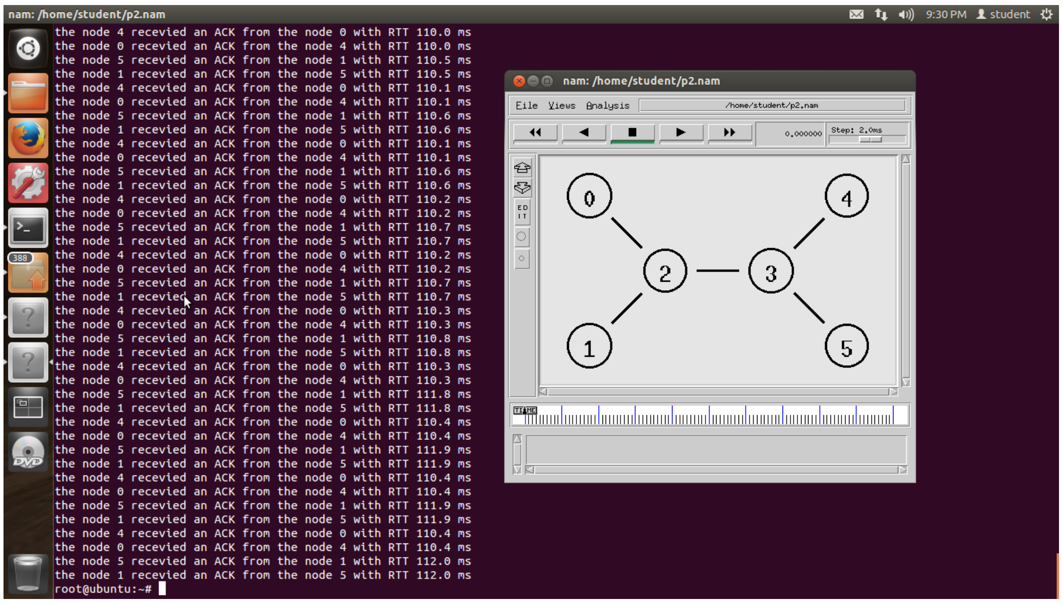
After simulation is completed run awk file to see the output ,

```
[root@localhost~]# awk –f lab2.awk lab2.tr
```

To see the trace file contents open the file as ,

```
[root@localhost~]# gedit lab2.tr
```

<u>Output</u>



```
the node 4 recevied an ACK from the node 0 with RTT 110.0 ms
the node 0 recevied an ACK from the node 4 with RTT 110.0 ms
the node 5 recevied an ACK from the node 1 with RTT 110.5 ms
the node 1 recevied an ACK from the node 5 with RTT 110.5 ms
the node 4 recevied an ACK from the node 0 with RTT 110.1 ms
the node 0 recevied an ACK from the node 4 with RTT 110.1 ms
the node 5 recevied an ACK from the node 1 with RTT 110.6 ms
the node 1 recevied an ACK from the node 5 with RTT 110.6 ms
the node 4 recevied an ACK from the node 0 with RTT 110.1 ms
the node 0 recevied an ACK from the node 4 with RTT 110.1 ms
the node 5 recevied an ACK from the node 1 with RTT 110.6 ms
the node 1 recevied an ACK from the node 5 with RTT 110.6 ms
the node 4 recevied an ACK from the node 0 with RTT 110.2 ms
the node 0 recevied an ACK from the node 4 with RTT 110.2 ms
the node 5 recevied an ACK from the node 1 with RTT 110.7 ms
the node 1 recevied an ACK from the node 5 with RTT 110.7 ms
the node 4 recevied an ACK from the node 0 with RTT 110.2 ms
the node 0 recevied an ACK from the node 4 with RTT 110.2 ms
the node 5 recevied an ACK from the node 1 with RTT 110.7 ms
the node 1 recevied an ACK from the node 5 with RTT 110.7 ms
the node 4 recevied an ACK from the node 0 with RTT 110.3 ms
the node 0 recevied an ACK from the node 4 with RTT 110.3 ms
the node 5 recevied an ACK from the node 1 with RTT 110.8 ms
the node 1 recevied an ACK from the node 5 with RTT 110.8 ms
the node 4 recevied an ACK from the node 0 with RTT 110.3 ms
the node 0 recevied an ACK from the node 4 with RTT 110.3 ms
the node 5 recevied an ACK from the node 1 with RTT 111.8 ms
the node 1 recevied an ACK from the node 5 with RTT 111.8 ms
the node 4 recevied an ACK from the node 0 with RTT 110.4 ms
the node 0 recevied an ACK from the node 4 with RTT 110.4 ms
the node 5 recevied an ACK from the node 1 with RTT 111.9 ms
the node 1 recevied an ACK from the node 5 with RTT 111.9 ms
the node 4 recevied an ACK from the node 0 with RTT 110.4 ms
the node 0 recevied an ACK from the node 4 with RTT 110.4 ms
the node 5 recevied an ACK from the node 1 with RTT 111.9 ms
the node 1 recevied an ACK from the node 5 with RTT 111.9 ms
the node 4 recevied an ACK from the node 0 with RTT 110.4 ms
the node 0 recevied an ACK from the node 4 with RTT 110.4 ms
the node 5 recevied an ACK from the node 1 with RTT 112.0 ms
the node 1 recevied an ACK from the node 5 with RTT 112.0 ms
root@ubuntu:~#
```

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

```
set ns [new Simulator]
set f [open p3.tr w]
set nf [open p3.nam w]
$ns trace-all $f
$ns namtrace-all $nf

proc finish {} {
global ns f nf
$ns flush-trace
close $f
close $nf

exec nam p3.nam &

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 [$ns node]
set n7 [$ns node]

$n0 set lable "TCP1"
$n3 set lable "SINK1"
$n4 set lable "TCP2"
$n7 set lable "sink2"

$ns color 1 red

$ns color 2 blue


$ns make-lan "$n0 $n1 $n2 $n3
$n4 $n5 $n6 $n7 " 10mb 30ms
LL Queue/DropTail Mac/802_3

set tcp1 [new Agent/TCP]
$ns attach-agent $n0 $tcp1

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

set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink1
```

```
$ns connect $tcp1 $sink1

$ftp1 set Type_ FTP
$tcp1 set class_ 1


set tcp2 [new Agent/TCP]
$ns attach-agent $n4 $tcp2

set telnet [new
Application/Telnet]
$telnet attach-agent $tcp2

set sink2 [new Agent/TCPSink]
$ns attach-agent $n7 $sink2

$ns connect $tcp2 $sink2

$telnet set Type_ Telnet
$tcp2 set class_ 2

set outFile1 [open
congestion1.xg w]
set outFile2 [open
congestion2.xg w]

puts $outFile1 "TitleText:
congestionWidthPlot-tcp"
puts $outFile1 "XUnitText:
SimulationTime(secs)"
puts $outFile1 "YUnitText:
CongestionWindowSize"

puts $outFile2 "TitleText:
congestionWidthPlot-tcp2"
puts $outFile2 "XUnitText:
SimulationTime(secs)"
puts $outFile2 "YUnitText:
CongestionWindowSize"


$ns at 0.1 "$ftp1 start"

$ns at 1.0 "$telnet start"
$ns at 49.0 "$telnet stop"
$ns at 49.0 "$ftp1 stop"

proc findWindowSize
{tcpSource outFile} {
global ns f nf
set now [$ns now]
set wSize [$tcpSource set cwnd_
]
puts $outFile "$now $wSize"
$ns at [expr $now + 0.1]
"findWindowSize $tcpSource
$outFile" }

$ns at 0.0 "findWindowSize
$tcp1 $outFile1"
$ns at 1.0 "findWindowSize
$tcp2 $outFile2"
$ns at 49.8 "plotWindow"


proc plotWindow {} {
exec xgraph congestion1.xg    -
bg white -fg black -lw 2
400*400 &
exec xgraph congestion2.xg    -
bg white -fg black -lw 2
400*400 &
exit 0

}

#$ns at 50.0 "finish"

$ns run

}

#$ns at 50.0 "finish"

$ns run
```

AWK file: (Open a new editor using "gedit command" and write awk file and save with ".awk" extension)

cwnd:- means congestion window

```
BEGIN {
        }
        {
        if($6= ="cwnd_")                      # don"t leave space after writing
        cwnd_ printf("%f\t%f\t\n",$1,$7);     # you must put \n in printf
        }
        E
        N
        D
        {
        }
```

Steps for execution

- Open gedit editor and type program. Program name should have the extension " .tcl "

  [root@localhost ~]# gedit lab3.tcl
- Save the program and close the file.
- Open gedit editor and type awk program. Program name should have the extension ".awk "

  [root@localhost ~]# gedit lab3.awk
- Save the program and close the file.
- Run the simulation program

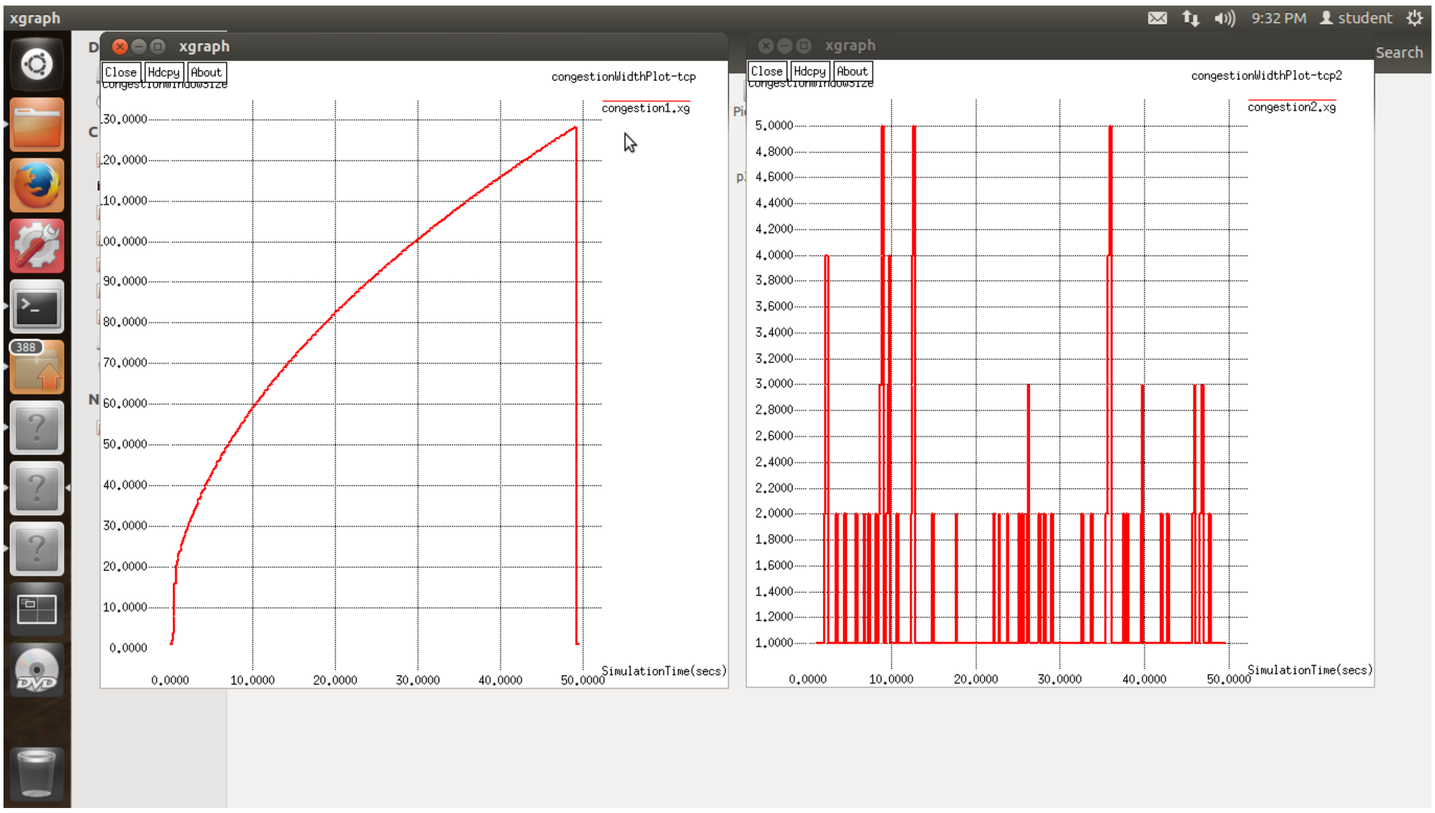
  [root@localhost~]# ns lab3.tcl
- Here "ns" indicates network simulator. We get the topology shown in the snapshot.
- Now press the play button in the simulation window and the simulation will begins.
- After simulation is completed run awk file to see the output ,

  [root@localhost~]# awk –f lab3.awk file1.tr > a1

  [root@localhost~]# awk –f lab3.awk file2.tr > a2

  [root@localhost~]# xgraph a1 a2\
- Here we are using the congestion window trace files i.e. file1.tr and file2.tr and we are redirecting the contents of those files to new files say a1 and a2 using output redirection operator (>).
- To see the trace file contents open the file as ,

<u>Output</u>



4. Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.

```
set ns [ new Simulator ]

set f [open p4.tr w]
set nf [open p4.nam w]
$ns trace-all $f

$ns namtrace-all $nf

proc finish {} {
global ns f nf
$ns flush-trace
close $f
close $nf
exec nam p4.nam &

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 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
set n9 [$ns node]

$n0 label "TCP Source"
$n2 label "UDP Source"
$n3 label "TCP Sink"

$n5 label "Null Agent"

$ns color 1 red

$ns color 2 blue

$ns make-lan "$n0 $n1 $n2 $n3 $n4" 1Mb 10ms LL Queue/DropTail Mac/802_3

$ns make-lan "$n5 $n6 $n7 $n8 $n9" 1Mb 10ms LL Queue/DropTail Mac/802_3

$ns duplex-link $n2 $n6 2.5Mb 10ms DropTail

set tcp0 [new Agent/TCP]
$ns attach-agent $n3 $tcp0

set telnet [new Application/Telnet]
$telnet attach-agent $tcp0
```

```
$telnet set packetSize_ 500

set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0

set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0

set null0 [new Agent/Null]
$ns attach-agent $n7 $null0

$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.001


$ns connect $tcp0 $sink0
$ns connect $udp0 $null0

$tcp0 set class_ 1

$udp0 set class_ 2

set err [new ErrorModel]
$ns lossmodel $err $n2 $n6
$err set rate_ 0.01
$ns set dataRate_ 10Mb

$ns at 0.1 "$cbr0 start"

$ns at 0.5 "$telnet start"
$ns at 9.0 "$telnet stop"
$ns at 9.0 "$cbr0 stop"
$ns at 10.0 "finish"

$ns run
```

#To run the awk file use command : awk -f 4.awk p4.tr

```
BEGIN {
cbrPktReceived=0;
totalPktReceived=0;
ftpPktReceived=0;

throughput=0;
}
{

if(($1=="+") && ($3=="2") && ($4=="6") && ($5=="cbr"))
cbrPktReceived++;

if(($1=="+") && ($3=="2") && ($4=="6") && ($5=="tcp"))
ftpPktReceived++;
totalPktReceived=cbrPktReceived+ftpPktReceived;
}

END {
throughput=((totalPktReceived*500*8)/(1000000))
printf "the throughput is:%dMbps\n",throughput
}
```

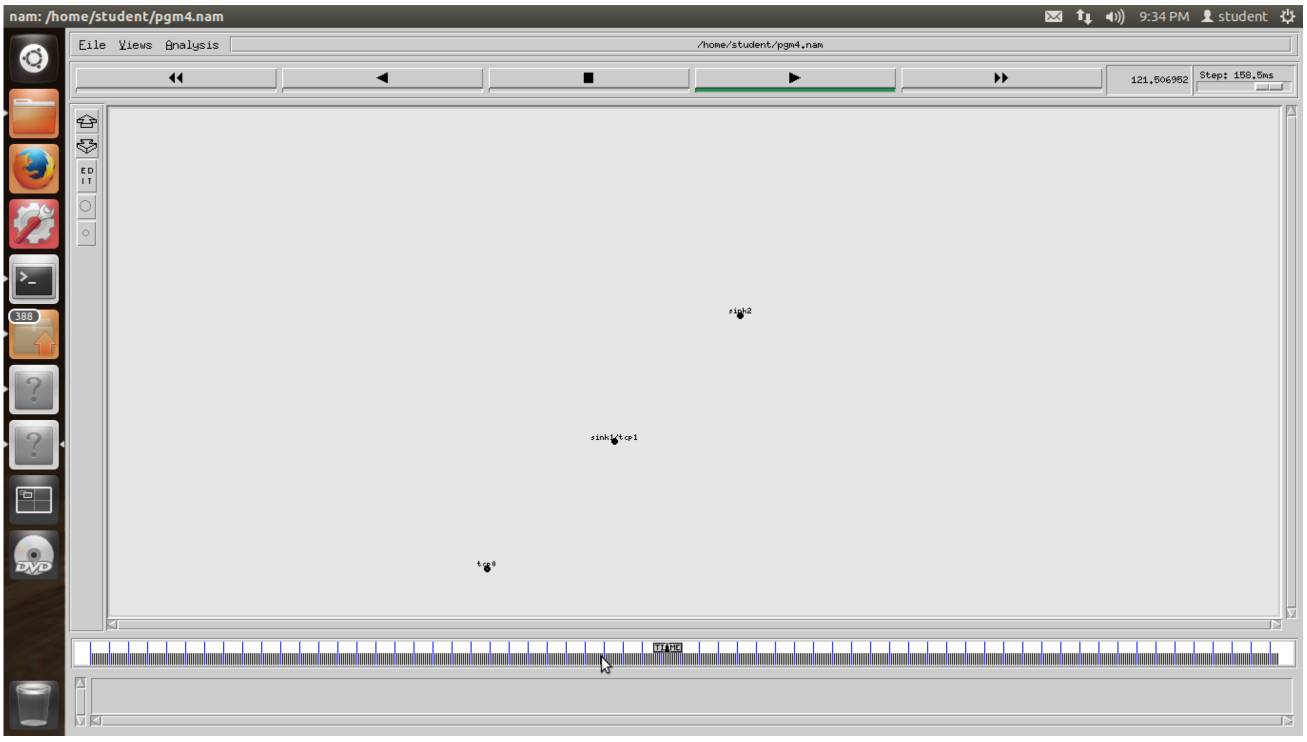
AWK file: (Open a new editor using “gedit command” and write awk file and save with “.awk” extension)

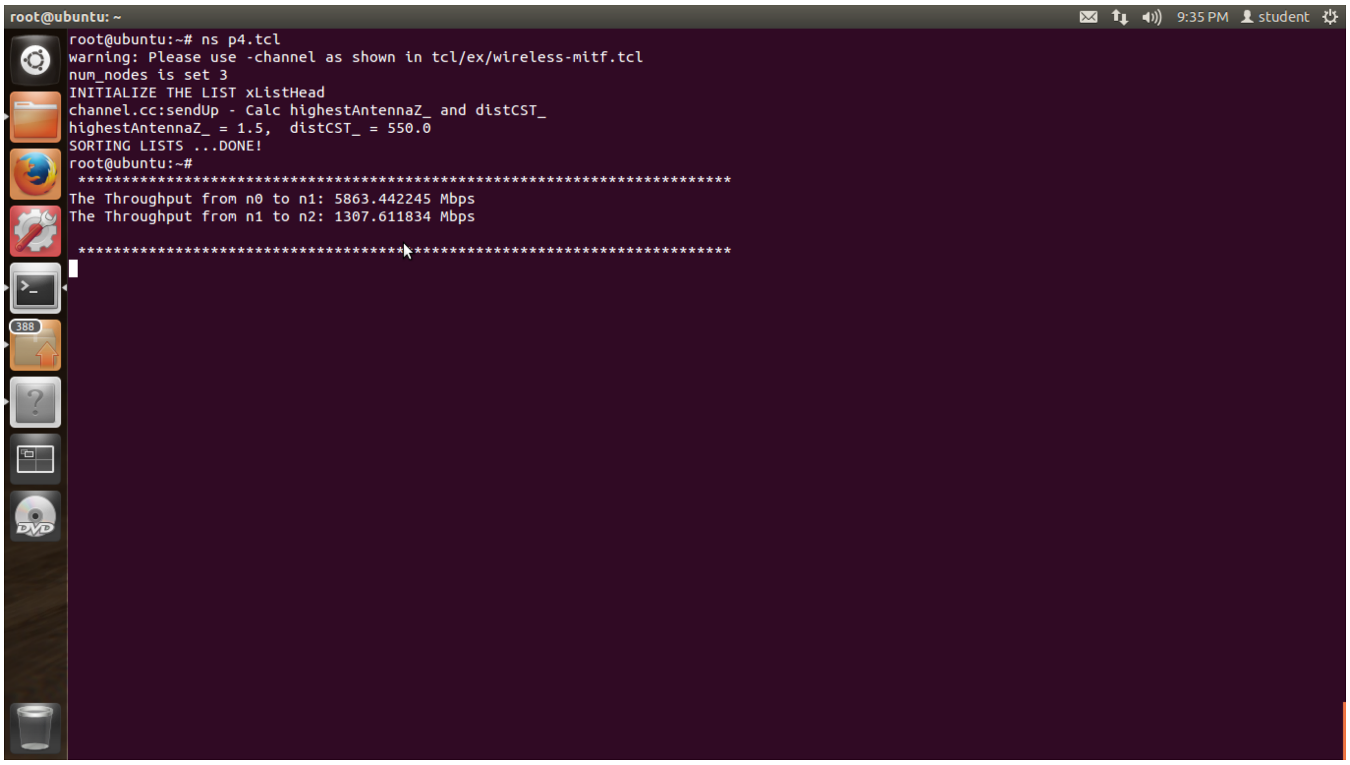
```
BEGIN{
count1=0 count2=0 pack1=0 pack2=0 time1=0 time2=0
}
{
if($1= ="r"&& $3= ="_1_" && $4= ="AGT")
{
count1++ pack1=pack1+$8 time1=$2
}
if($1= ="r" && $3= ="_2_" && $4= ="AGT")
{
count2++ pack2=pack2+$8 time2=$2
}
}

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)));
}
```

Steps for execution

- Open gedit editor and type program. Program name should have the extension “ .tcl ”
  `[root@localhost ~]# gedit lab4.tcl`
- Save the program and close the file.
- Open gedit editor and type awk program. Program name should have the extension “.awk ”
  `[root@localhost ~]# gedit lab4.awk`
- Save the program and close the file.
- Run the simulation program
  `[root@localhost~]# ns lab4.tcl`
- 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.
- After simulation is completed run awk file to see the output ,
  `[root@localhost~]# awk –f lab4.awk lab4.tr`
- To see the trace file contents open the file as ,
  `[root@localhost~]# gedit lab4.tr`

Output





```
root@ubuntu:~# ns p4.tcl
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
num_nodes is set 3
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
root@ubuntu:~#
 **************************************************************************
The Throughput from n0 to n1: 5863.442245 Mbps
The Throughput from n1 to n2: 1307.611834 Mbps

 **************************************************************************
```

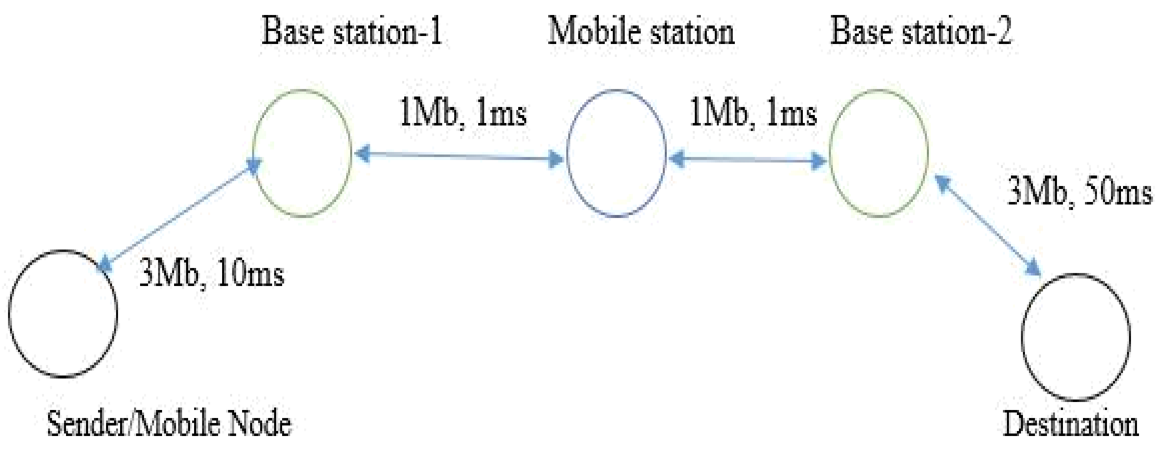
5. **Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.**

Second Generation (2G) technology is based on the technology known as global system for mobile communication (GSM). This technology enabled various networks to provide services like text messages, picture messages and MMS. The technologies used in 2G are either TDMA (Time Division Multiple Access) which divides signal into different time slots or CDMA (Code Division Multiple Access) which allocates a special code to each user so as to communicate over a multiplex physical channel.

GSM uses a variation of time division multiple access (TDMA). 2G networks developed as a replacement for first generation (1G) analog cellular networks, and the GSM standard originally described as a digital, circuit-switched network optimized for fullduplex voice telephony. This expanded over time to include data communications, first by circuit-switched transport, then by packet data transport via GPRS (General Packet Radio Services).

GSM can be implemented on all the versions of NS2 (Since year 2004: ns-2.27, and later versions of NS2)

Design:

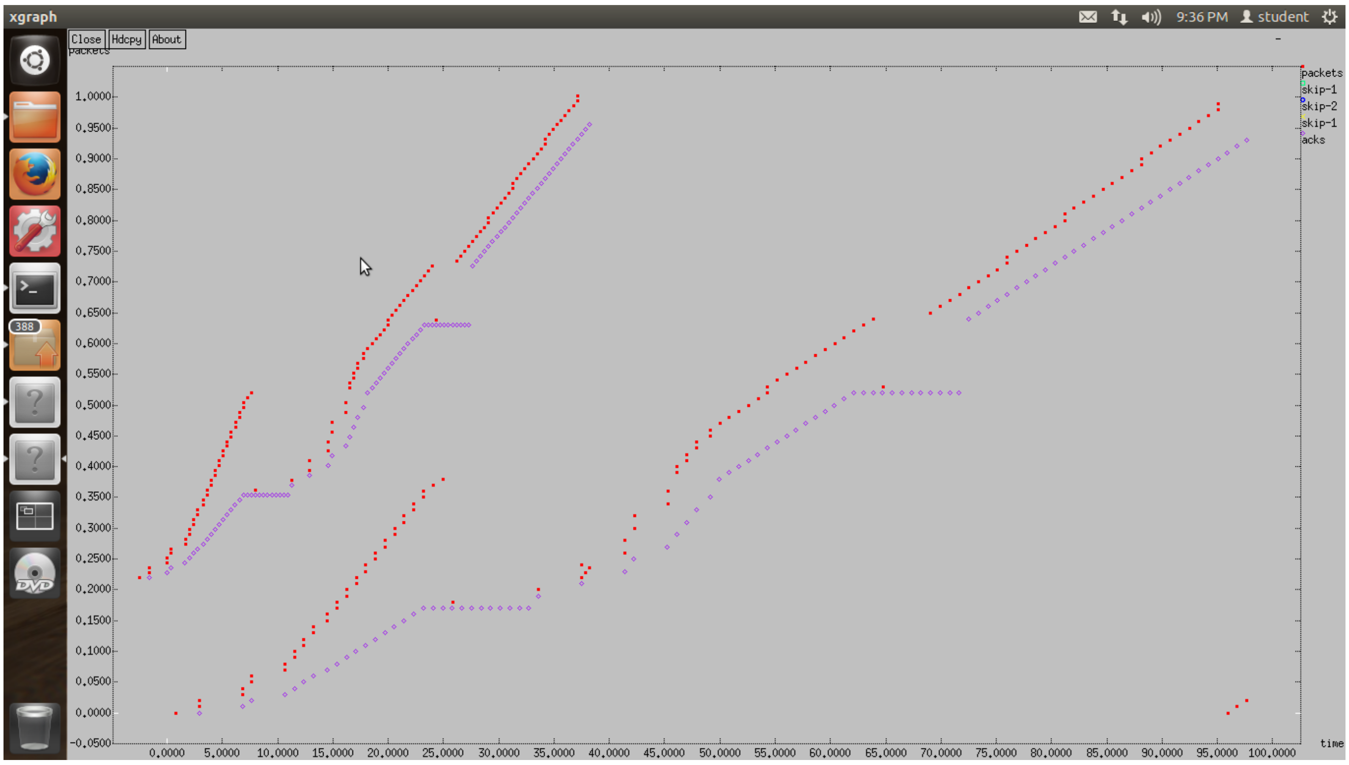


Program

```
set val(chan)              Channel/WirelessChannel
set val(type)              GSM
set val(prop)              Propagation/TwoRayGround
set val(netif)             Phy/WirelessPhy
set val(mac)               Mac/802_11
set val(ifq)        Queue/DropTail/PriQueue
set val(ll)         LL
set val(ant)        Antenna/OmniAntenna
set val(x)            1500
set val(y)            1500
set val(ifqlen)         1000
set val(adhocRouting)   AODV
set val(nn)            10
set val(stop)              5.0

set f0 [open out02.tr w]
set f1 [open lost02.tr w]
set f2 [open delay02.tr w]

set ns_                                   [new Simulator]
set topo                        [new Topography]

set tracefd                     [open out.tr w]
set namtrace        [open out.nam w]

$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

$topo load_flatgrid $val(x) $val(y)

set god_ [create-god $val(nn)]
$ns_ color 0 red
$ns_ node-config -adhocRouting AODV \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
                                -energyModel EnergyModel \
                                -initialEnergy 100 \
                                 -rxPower 0.3 \
                                 -txPower 0.6 \
                                 -topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF




for {set i 0} {$i < $val(nn) } {incr i} {
                    set node_($i) [$ns_ node]

}


set X1(0) 1035.201
set Y1(0) 444.699
set X1(1) 244.365
set Y1(1) 521.418
set X1(2) -18.1268
set Y1(2) 300.612
set X1(3) 723.89
set Y1(3) 343.533
set X1(4) 122.34
set Y1(4) 311.755
set X1(5) 373.498
set Y1(5) 472.206
set X1(6) 548.549
set Y1(6) 361.062
set X1(7) 389.995
set Y1(7) 381.178
set X1(8) 494.798
set Y1(8) 477.771
set X1(9) 275.01
set Y1(9) 381.99


for {set i 0} {$i < $val(nn) } {incr i} {
                    $node_($i) set X_ $X1($i)
$node_($i) set Y_ $Y1($i)
$node_($i) set Z_ 0.0



}

puts "---------------------------------------"
set m 0
puts "----------------------------------------"
puts "|    Node       |  One hop neighbour    |"
puts "----------------------------------------"
for {set i 0} {$i < $val(nn) } {incr i} {

set k 0
for {set j 0} {$j < $val(nn) } {incr j} {



set a [ expr $X1($j)-$X1($i)]
set b [ expr $a*$a]
set c [ expr $Y1($j)-$Y1($i)]
set d [ expr $c*$c]
set e [ expr $b+$d]
set f 0.5
set g [expr pow($e,$f)]
#puts "Distance from node($i) --to--node($j)----------->$g"
if {$g <= 200 && $i != $j} {

puts "|    node($i)     |     node($j)       |"
set nei($m) $j

set k [expr $k+1]
set m [ expr $m+1]
}

}
puts "----------------------------------------"
}


puts "Loading connection pattern..."


puts "Loading scenario file..."
for {set i 0} {$i < $val(nn) } {incr i} {

$ns_ initial_node_pos $node_($i) 45
```

```
}

for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ at $val(stop).0 "$node_($i) reset";
}

set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(2) $udp_(0)
set sink [new Agent/LossMonitor]
$ns_ attach-agent $node_(3) $sink
set cbr1_(0) [new Application/Traffic/CBR]
$cbr1_(0) set packetSize_ 1000
$cbr1_(0) set interval_ 0.1
$cbr1_(0) set maxpkts_ 1000
$cbr1_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $sink
$ns_ at 1.00 "$cbr1_(0) start"

set holdtime 0
set holdseq 0

set holdrate1 0

proc record {} {
global sink  f0 f1 f2 holdtime holdseq holdrate1

set ns [Simulator instance]
set time 0.9 ;#Set Sampling Time to 0.9 Sec

set bw0 [$sink set bytes_]
set bw1 [$sink set nlost_]

set bw2 [$sink set lastPktTime_]
set bw3 [$sink set npkts_]

set now [$ns now]

# Record Bit Rate in Trace Files
puts $f0 "$now [expr (($bw0+$holdrate1)*8)/(2*$time*1000000)]"


# Record Packet Loss Rate in File
puts $f1 "$now [expr $bw1/$time]"

if { $bw3 > $holdseq } {
puts $f2 "$now [expr ($bw2 - $holdtime)/($bw3 - $holdseq)]"
} else {
puts $f2 "$now [expr ($bw3 - $holdseq)]"
}

$sink set bytes_ 0
$sink set nlost_ 0

set holdtime $bw2
set holdseq $bw3
set  holdrate1 $bw0
$ns at [expr $now+$time] "record"   ;# Schedule Record after $time interval sec
}


# Start Recording at Time 0
$ns_ at 0.0 "record"

source link.tcl

proc stop {} {
global ns_ tracefd f0 f1 f2

# Close Trace Files
close $f0
close $f1
close $f2
exec nam out.nam
# Plot Recorded Statistics

exec xgraph out02.tr -geometry -x TIME -y thr -t Throughput 800x400 &
exec xgraph lost02.tr  -geometry -x TIME -y loss -t Packet_loss 800x400 &
exec xgraph delay02.tr  -geometry -x TIME -y delay -t End-to-End-Delay 800x400 &

$ns_ flush-trace

}

$ns_ at $val(stop) "stop"
$ns_ at  $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp "
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

puts "Starting Simulation..."
$ns_ run

#link code

$ns_ at 0.1 "$node_(0) setdest 786 813 20"
$ns_ at 0.1 "$node_(1) setdest 895 890 20"
$ns_ at 0.1 "$node_(2) setdest 633 669 20"
$ns_ at 0.1 "$node_(3) setdest 1375 712 20"
$ns_ at 0.1 "$node_(4) setdest 773 680 20"
$ns_ at 0.1 "$node_(5) setdest 1024 841 20"
$ns_ at 0.1 "$node_(6) setdest 1199 730 20"
$ns_ at 0.1 "$node_(7) setdest 1041 750 20"
$ns_ at 0.1 "$node_(8) setdest 1146 846 20"
$ns_ at 0.1 "$node_(9) setdest 926 751 20"



$ns_ at 0.5 "$node_(2) add-mark m blue square"
$ns_ at 0.5 "$node_(3) add-mark m blue square"

$ns_ at 0.5 "$node_(2) label source"
$ns_ at 0.5 "$node_(3) label Destination"
```

Output :

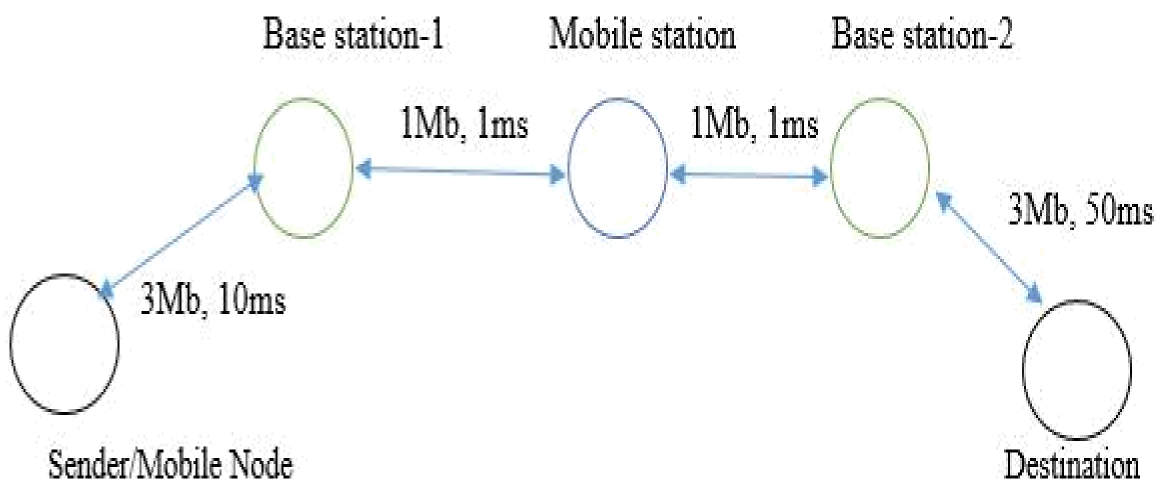


6. **Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or equivalent environment.**

3G networks developed as a replacement for second generation (2G) GSM standard network with full duplex voice telephony. CDMA is used as the access method in many mobile phone standards. IS-95, also called cdma One, and its 3G evolution CDMA2000, are often simply referred to as CDMA, but UMTS(The Universal Mobile Telecommunications System is a third generation mobile cellular system for networks based on the GSM standard.), the 3G standard used by GSM carriers, also uses wideband CDMA. Long-Term Evolution (LTE) is a standard for high-speed wireless communication which uses CDMA network technology.

3G technology generally refers to the standard of accessibility and speed of mobile devices. The standards of the technology were set by the International Telecommunication Union (ITU). This technology enables use of various services like GPS (Global Positioning System), mobile television and video conferencing. It not only enables them to be used worldwide, but also provides with better bandwidth and increased speed. The main aim of this technology is to allow much better coverage and growth with minimum investment.

CDMA can be implemented on all the versions of NS2 (Since year 2004: ns-2.27, and later versions of NS2)



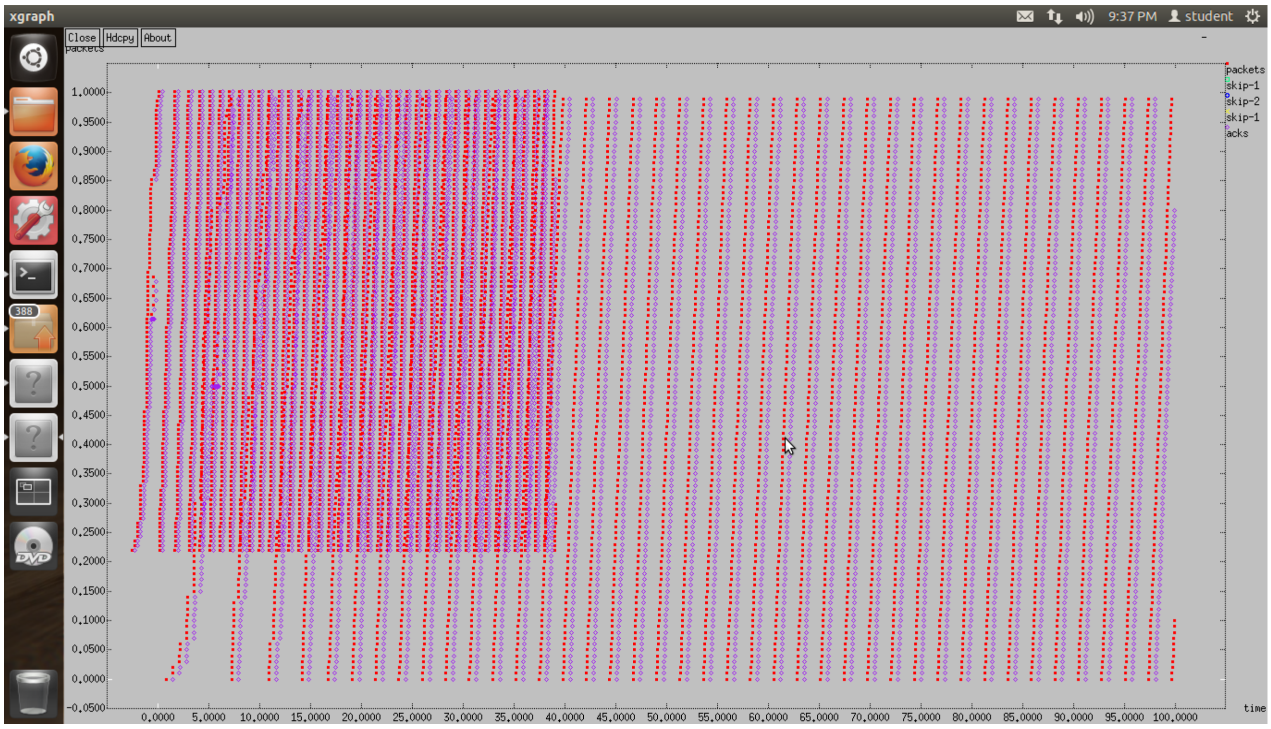
Program:

```
puts "Enter number of nodes"
set tnn [gets stdin]
set val(chan)             Channel/WirelessChannel
set val(prop)             Propagation/TwoRayGround
set val(netif)            Phy/WirelessPhy
set val(mac)              Mac/802_11
set val(ifq)       Queue/DropTail/PriQueue
set val(ll)        LL
set val(ant)       Antenna/OmniAntenna
set val(x)           1500
set val(y)           1500
set val(ifqlen)        1000
set val(adhocRouting)   AODV
set val(nn)           $tnn
set val(stop)             10.0

Mac/802_11 set cdma_code_bw_start_                  0      ;# cdma code for bw request (start)
Mac/802_11 set cdma_code_bw_stop_           63     ;# cdma code for bw request (stop)
Mac/802_11 set cdma_code_init_start_                64      ;# cdma code for initial request (start)
Mac/802_11 set cdma_code_init_stop_                 127      ;# cdma code for initial request (stop)
Mac/802_11 set cdma_code_cqich_start_       128     ;# cdma code for cqich request (start)
Mac/802_11 set cdma_code_cqich_stop_                195      ;# cdma code for cqich request (stop)
Mac/802_11 set cdma_code_handover_start_    196     ;# cdma code for handover request (start)
Mac/802_11 set cdma_code_handover_stop_     255      ;# cdma code for handover request (stop)

set f0 [open out02.tr w]
set f1 [open lost02.tr w]
set f2 [open delay02.tr w]

set ns_                   [new Simulator]
set topo           [new Topography]

set tracefd               [open out.tr w]
set namtrace       [open out.nam w]

$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
```

```
$topo load_flatgrid $val(x) $val(y)

set god_ [create-god $val(nn)]
$ns_ color 0 red
$ns_ node-config -adhocRouting AODV \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
                -energyModel EnergyModel \
                -initialEnergy 100 \
                 -rxPower 0.3 \
                 -txPower 0.6 \
                 -topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF

for {set i 0} {$i < $val(nn) } {incr i} {
        set node_($i) [$ns_ node]
$node_($i) set X_  [expr rand() * 500]
$node_($i) set Y_ [expr rand() * 500]
$node_($i) set Z_ 0.000000000000;
}

for {set i 0} {$i < $val(nn) } {incr i} {
set xx  [expr rand() * 1500]
set yy [expr rand() * 1000]
$ns_ at 0.1 "$node_($i) setdest $xx 4yy 5"
}

puts "Loading connection pattern..."

puts "Loading scenario file..."
for {set i 0} {$i < $val(nn) } {incr i} {

$ns_ initial_node_pos $node_($i) 55
}

for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ at $val(stop).0 "$node_($i) reset";
}

puts "Enter source node"
set source [gets stdin]
puts "Enter destination node"
set dest [gets stdin]

set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_($source) $udp_(0)
set sink [new Agent/LossMonitor]
$ns_ attach-agent $node_($dest) $sink
```

```
set cbr1_(0) [new Application/Traffic/CBR]
$cbr1_(0) set packetSize_ 1000
$cbr1_(0) set interval_ 0.1
$cbr1_(0) set maxpkts_ 10000
$cbr1_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $sink
$ns_ at 1.00 "$cbr1_(0) start"

set holdtime 0
set holdseq 0

set holdrate1 0

proc record {} {
global sink  f0 f1 f2 holdtime holdseq holdrate1

set ns [Simulator instance]
set time 0.9 ;#Set Sampling Time to 0.9 Sec

set bw0 [$sink set bytes_]
set bw1 [$sink set nlost_]

set bw2 [$sink set lastPktTime_]
set bw3 [$sink set npkts_]

set now [$ns now]

# Record Bit Rate in Trace Files
puts $f0 "$now [expr (($bw0+$holdrate1)*8)/(2*$time*1000000)]"

# Record Packet Loss Rate in File
puts $f1 "$now [expr $bw1/$time]"

if { $bw3 > $holdseq } {
puts $f2 "$now [expr ($bw2 - $holdtime)/($bw3 - $holdseq)]"
} else {
puts $f2 "$now [expr ($bw3 - $holdseq)]"
}

$sink set bytes_ 0
$sink set nlost_ 0

set holdtime $bw2
set holdseq $bw3

set  holdrate1 $bw0
$ns at [expr $now+$time] "record"   ;# Schedule Record after $time interval sec
}

# Start Recording at Time 0
$ns_ at 0.0 "record"

source link.tcl

proc stop {} {
global ns_ tracefd f0 f1 f2
# Close Trace Files
close $f0
close $f1
close $f2
exec nam out.nam
exec xgraph out02.tr -geometry -x TIME -y thr -t Throughput 800x400 &
exec xgraph lost02.tr  -geometry -x TIME -y loss -t Packet_loss 800x400 &
exec xgraph delay02.tr  -geometry -x TIME -y delay -t End-to-End-Delay 800x400 &
$ns_ flush-trace

}

$ns_ at $val(stop) "stop"
$ns_ at  $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp "
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

puts "Starting Simulation..."
$ns_ run

#link   code

$ns_ at 0.5 "$node_($source) add-mark m blue square"
$ns_ at 0.5 "$node_($dest) add-mark m magenta square"

$ns_ at 0.5 "$node_($source) label SENDER"
$ns_ at 0.5 "$node_($dest) label RECEIVER"

$ns_ at 0.01 "$ns_ trace-annotate \"Network Deployment\""
```

Output :



PART-B

Java is a general-purpose computer programming language that is simple, concurrent, class object language. The compiled Java code can run on all platforms that support Java without the need for recompilation hence Java is called as "write once, 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.

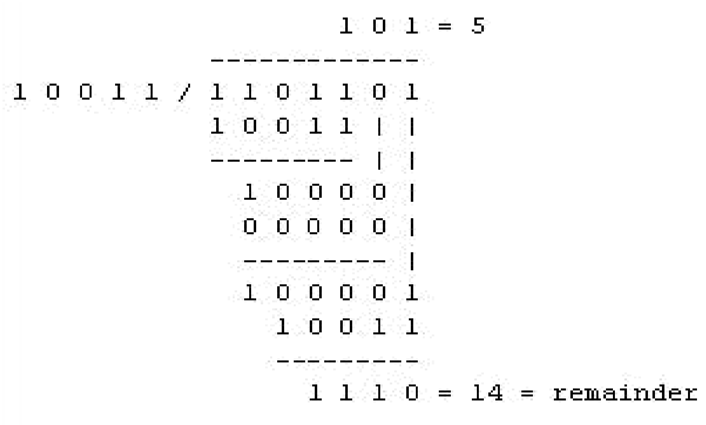
2. Run Java program from Command Prompt
   [root@host ~]# java Filename

   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).

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

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.

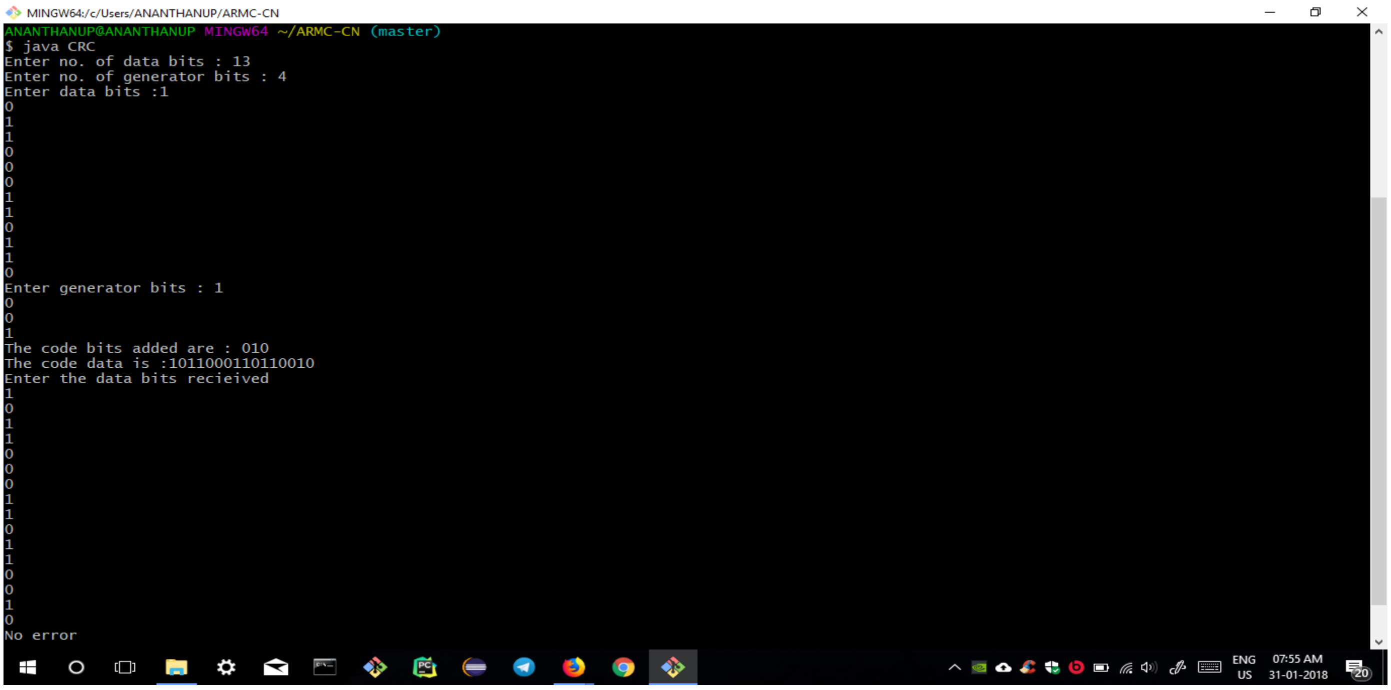
	CRC-CCITT	**CRC-16**	**CRC-32**
Checksum Width	16 bits	16 bits	32 bits
Generator Polynomial	10001000000100001	11000000000000101	100000100110000010001110110110111

International Standard CRC Polynomials

Source Code:

```
import java.util.*;
public class CRC {
        public static void main(String args[])
        {
                Scanner sc=new Scanner(System.in);
                int m,g[],n,d[],z[],r[],msb,i,j,k;
                System.out.println("Enter no. of bits in Dataword(d) : ");
                n=sc.nextInt();
                System.out.println("Enter no. of generator bits : ");
                 m=sc.nextInt();
                d=new int[n+m];
                g=new int[m];
                System.out.println("Enter Dataword bits(one bit per line):");
                for(i=0;i<n;i++)
                        d[i]=sc.nextInt();
                System.out.println("Enter generator bits(one bit per line):");
                for(j=0;j<m;j++)
                        g[j]=sc.nextInt();
                for(i=0;i<m-1;i++)
                        d[n+i]=0;
                r=new int[m+n];
                for(i=0;i<m;i++)
                        r[i]=d[i];
                z=new int[m];
                for(i=0;i<m;i++)
                        z[i]=0;
                for(i=0;i<n;i++)
                {
                        k=0;
                        msb=r[i];
                        for(j=i;j<m+i;j++)
                        {
                                if(msb==0)
                                        r[j]=xor(r[j],z[k]);
                                else
                                        r[j]=xor(r[j],g[k]);
                                k++;
                        }
                        r[m+i]=d[m+i];
                }
                System.out.println("The redundant(r) bits added are : ");
                for(i=n;i<n+m-1;i++)
                {
                        d[i]=r[i];
                        System.out.println(d[i]);
                }System.out.println();
                System.out.println("The codeword(c=d+r) is :");
                for(i=0;i<n+m-1;i++)
                {
                        System.out.println(d[i]);
                }
                int c[]=new int[n+m];
                System.out.println();
                System.out.println("Enter the data bits recieived(one bit per line)");
                for(i=0;i<n+(m-1);i++)
                        c[i]=sc.nextInt();
                int count=0;
                for(i=0;i<n+m-1;i++)
                {       if(c[i]==d[i])
                        count++;
                else
                        break;
                }

                if(count==n+m-1)
                        System.out.println("No error");
                else{
                        System.out.println("Error present");
                        System.out.println("Error occurs at "+(count+1));
                }       }
        public static int xor(int x,int y)
        {
                if(x==y)
                        return(0);
                else
                        return(1);
        }
}
```

Output:



```
ANANTHANUP@ANANTHANUP MINGW64 ~/ARMC-CN (master)
$ java CRC
Enter no. of data bits : 13
Enter no. of generator bits : 4
Enter data bits :1
0
1
1
0
0
0
1
1
0
1
1
0
Enter generator bits : 1
0
0
1
The code bits added are : 010
The code data is :1011000110110010
Enter the data bits recieived
1
0
1
1
0
0
0
1
1
0
1
1
0
0
1
0
No error
```

2. **Write a program to find the shortest path between vertices using 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.

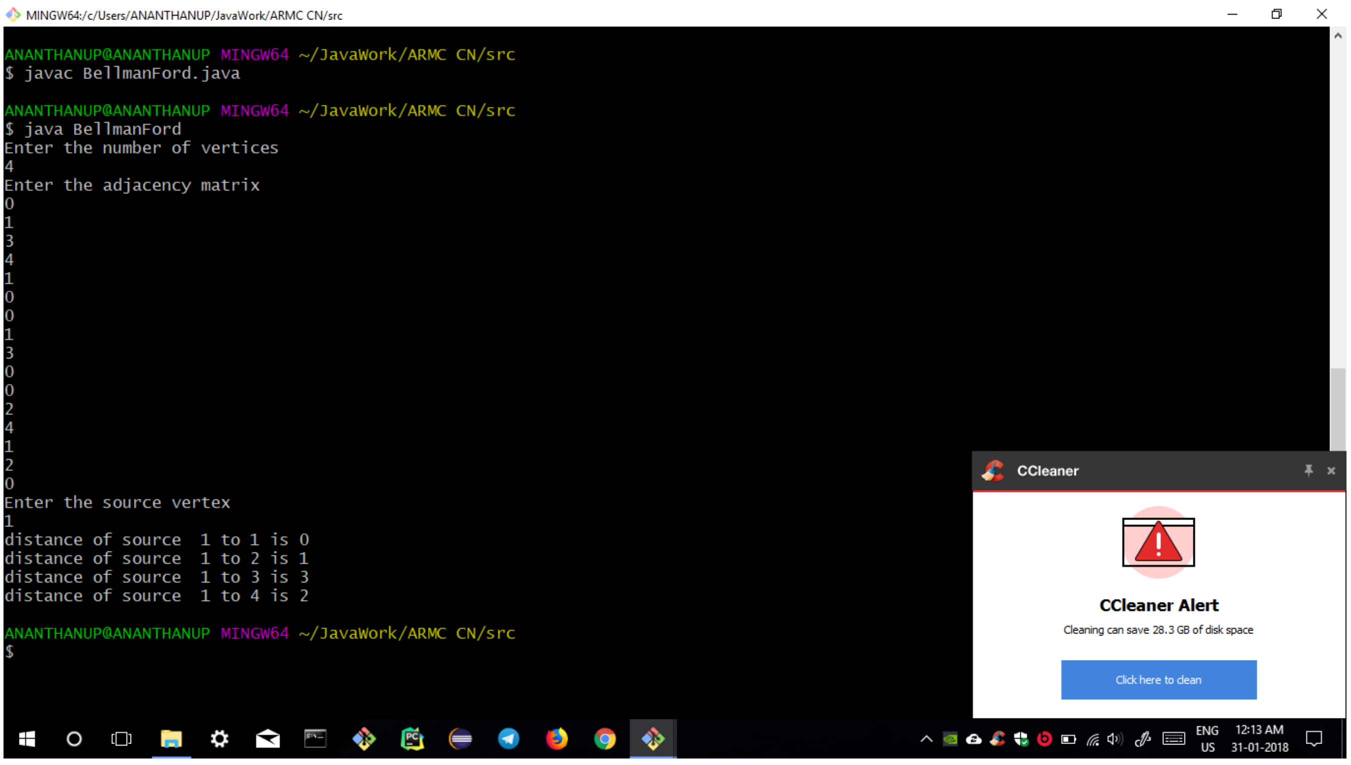
The Bellman–Ford algorithm is an algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph. It is slower than Dijkstra'salgorithm for the same problem, but more versatile, as it is capable of handling graphs in which some of the edge weights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

Source code:

```
import java.util.Scanner;

public class BellmanFord
{
private int d[];
private int n;
public static final int MAX_VALUE = 999;

public BellmanFord(int n)
{
this.n = n;
d = new int[n + 1];
}

public void BellmanFordEvaluation(int source, int a[][])
{
for (int node = 1; node <= n; node++)
{
d[node] = MAX_VALUE;
}

d[source] = 0;
for (int node = 1; node <= n - 1; node++)
{
```

```
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 negative egde cycle");
}
}
}

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

public static void main(String[] arg)
{
int n = 0;
int source;
Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of vertices");
n = scanner.nextInt();

int a[][] = new int[n + 1][n + 1];
System.out.println("Enter the adjacency matrix");
for (int i = 1; i <= n; i++)
{
for (int j = 1; j <= n; j++)
{
a[i][j] = scanner.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");
source = scanner.nextInt();

BellmanFord bellmanford = new BellmanFord(n);
bellmanford.BellmanFordEvaluation(source, a);
scanner.close();
}
}
```

Output:



```
MINGW64:/c/Users/ANANTHANUP/JavaWork/ARMC CN/src

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ javac BellmanFord.java

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ java BellmanFord
Enter the number of vertices
4
Enter the adjacency matrix
0
1
3
4
1
0
0
1
3
0
0
2
4
1
2
0
Enter the source vertex
1
distance of source  1 to 1 is 0
distance of source  1 to 2 is 1
distance of source  1 to 3 is 3
distance of source  1 to 4 is 2

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$
```

3. **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. Implement the above program using as message queues or FIFOs as IPC channels.**

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:

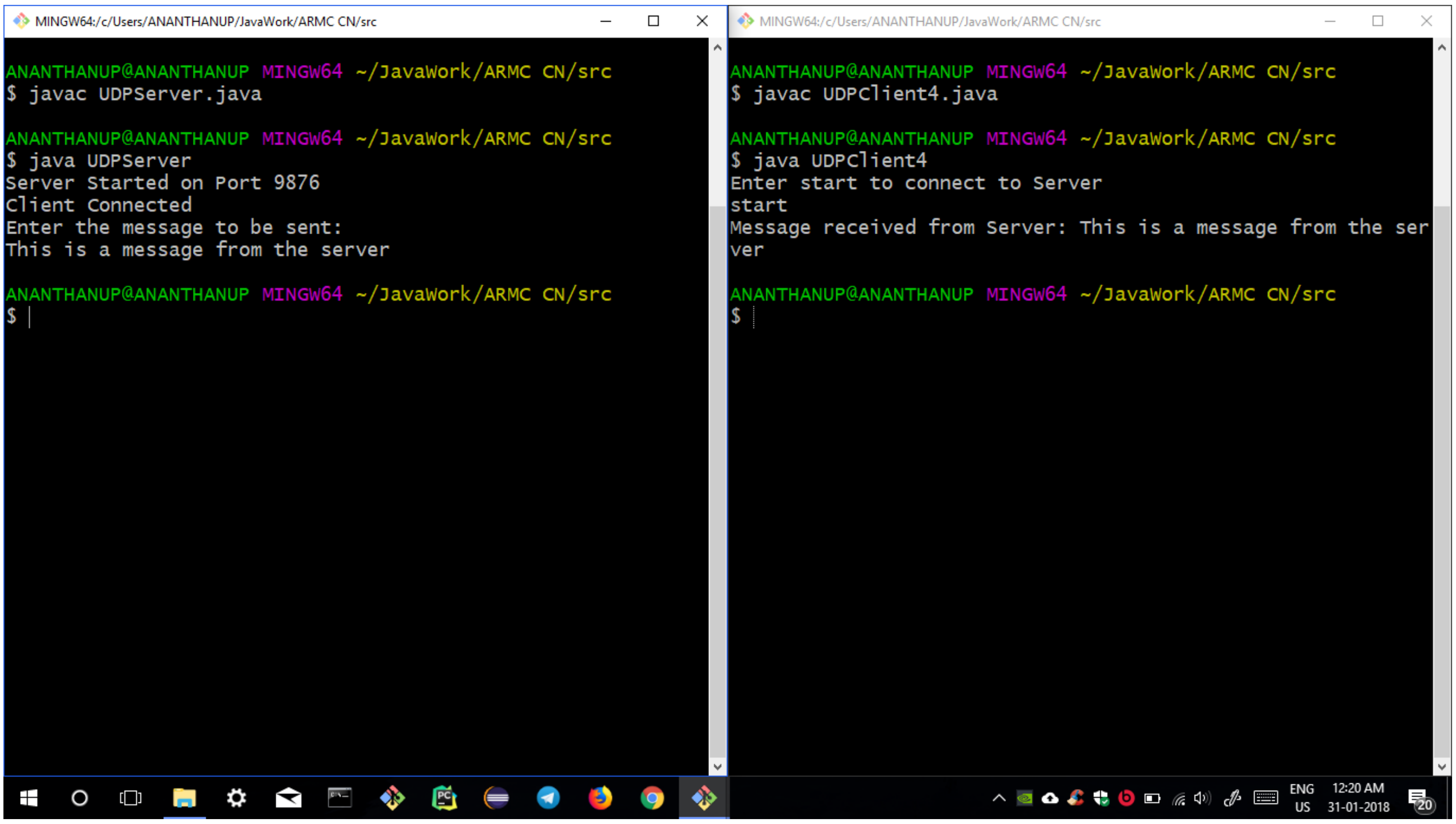
TCP Client

```
import java.net.*;
import java.io.*;
public class TCPClient
{
public static void main( String args[ ] ) throws Exception
{
Socket sock = new Socket( "127.0.0.1", 4000);
System.out.print("Enter the file name\n");
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String fname = br.readLine();
OutputStream  ostream = sock.getOutputStream( );
PrintWriter pwrite = new PrintWriter(ostream, true);
pwrite.println(fname);

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(); br.close(); sock.close();
}
}
```

TCP Server

```
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
public class TCPServer
{
public static void main(String args[]) throws Exception
{
ServerSocket sersock = new ServerSocket(4000);
System.out.println("Server ready for connection");
Socket sock = sersock.accept();
System.out.println("Connection successful | wating for file name");
InputStream istream = sock.getInputStream( );
BufferedReader br =new BufferedReader(new InputStreamReader(istream));
String fname = br.readLine( );
BufferedReader contentRead = new BufferedReader(new FileReader(fname) );
OutputStream ostream = sock.getOutputStream( );
PrintWriter pwrite = new PrintWriter(ostream, true);
String str;
while((str = contentRead.readLine()) !=  null)
{
    pwrite.println(str);
}
System.out.println("File Contents sent successfully");

sock.close();  sersock.close();
pwrite.close();  br.close(); contentRead.close();
}
}
```

Output:



```
ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ javac UDPServer.java

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ java UDPServer
Server Started on Port 9876
Client Connected
Enter the message to be sent:
This is a message from the server

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$
```

```
ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ javac UDPClient4.java

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ java UDPClient4
Enter start to connect to Server
start
Message received from Server: This is a message from the ser
ver

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$
```

4. **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:

UDP Client

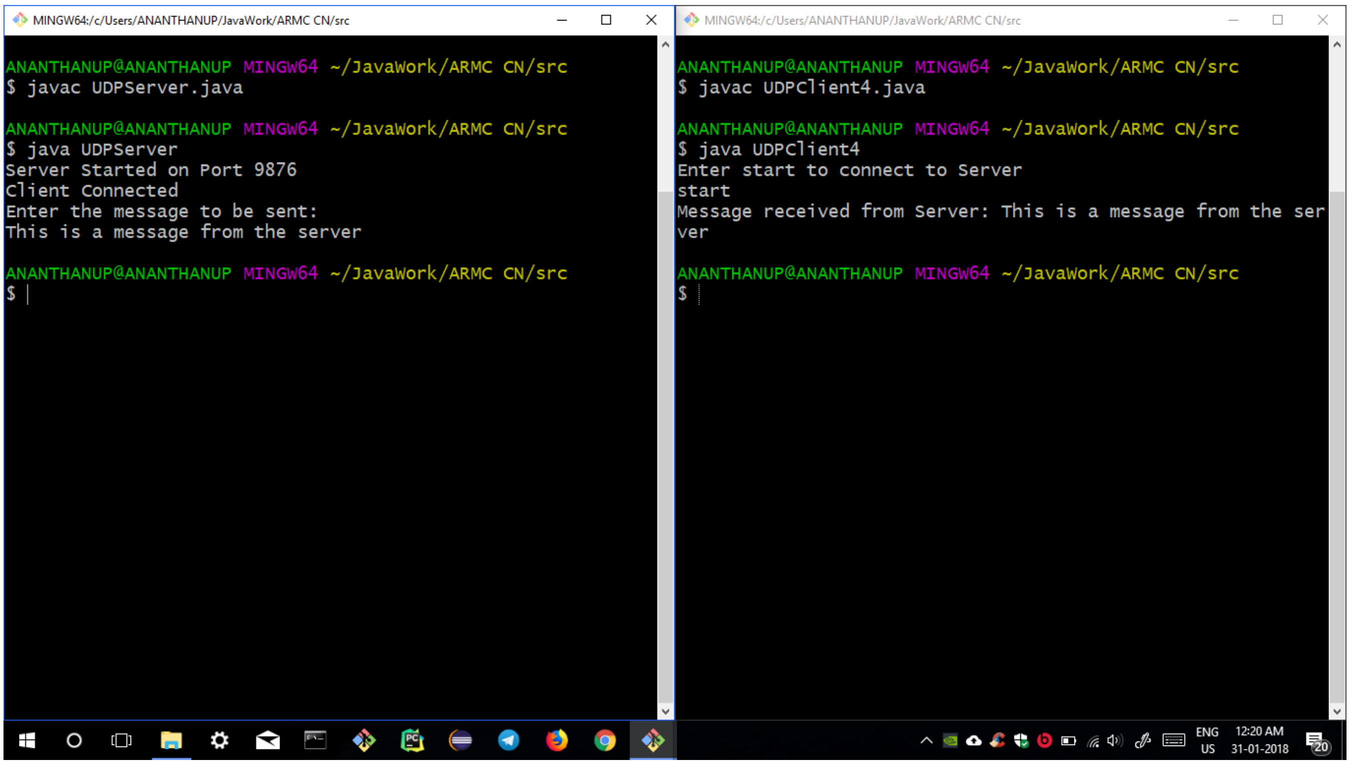
```
import java.io.*;
import java.net.*;

class UDPClient
{
public static void main(String args[]) throws Exception
{
BufferedReader inFromUser =
new BufferedReader(new InputStreamReader(System.in));
DatagramSocket clientSocket = new DatagramSocket();
InetAddress IPAddress = InetAddress.getByName("localhost");
byte[] sendData = new byte[1024];
byte[] receiveData = new byte[1024];
System.out.println("Enter 'START' to connect to Server");
String sentence = inFromUser.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());
System.out.println("Message received from Server: " + modifiedSentence);
clientSocket.close();
}
}
```

UDP Server

```
import java.io.*;
import java.net.*;
import java.util.Scanner;

class UDPServer
{
public static void main(String args[]) throws Exception
{
DatagramSocket serverSocket = new DatagramSocket(9876);
System.out.println("Server Started on Port 9876");
byte[] receiveData = new byte[1024];
byte[] sendData = new byte[1024];
while(true)
{
```

```
DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
serverSocket.receive(receivePacket);
receivePacket.getData();
InetAddress IPAddress = receivePacket.getAddress();
int port = receivePacket.getPort();
System.out.println("Client Connected");
Scanner input = new Scanner(System.in);
System.out.println("Enter the message to be sent: ");
String message = input.nextLine();
sendData = message.getBytes();
DatagramPacket sendPacket =
new DatagramPacket(sendData, sendData.length, IPAddress, port);
serverSocket.send(sendPacket);
System.exit(0);
}

}
}
```

Output:



```
ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ javac UDPServer.java

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ java UDPServer
Server Started on Port 9876
Client Connected
Enter the message to be sent:
This is a message from the server

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$
```

```
ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ javac UDPClient4.java

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$ java UDPClient4
Enter start to connect to Server
start
Message received from Server: This is a message from the ser
ver

ANANTHANUP@ANANTHANUP MINGW64 ~/JavaWork/ARMC CN/src
$
```

MINGW64:/c/Users/ANANTHANUP/JavaWork/ARMC CN/src
MINGW64:/c/Users/ANANTHANUP/JavaWork/ARMC CN/src
ENG US
12:20 AM
31-01-2018

5. 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:-

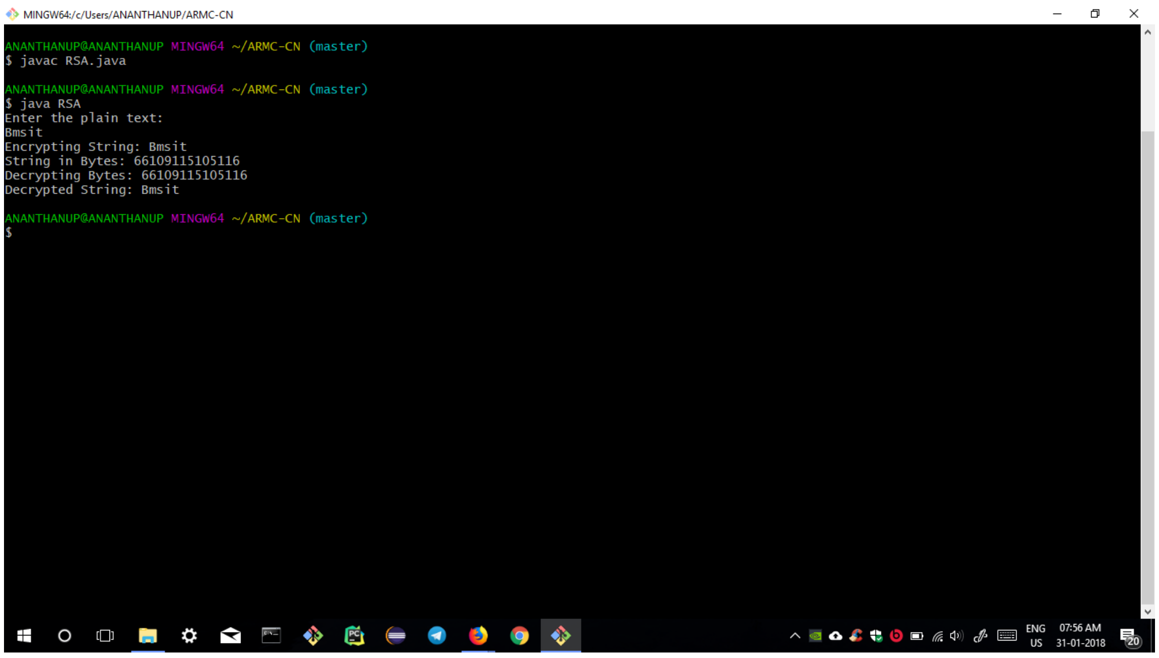
1. Uses his private key (d, n) to compute M = C^d mod n.
2. Extracts the plaintext from the integer representative m.

Source Code:

```
import java.math.BigInteger;
import java.io.*;

public class RSA
{
        // BigInteger is used to store and manipulate very large numbers
        BigInteger p, q, d, e, n, z;
        // p and q -> Large Prime Numbers
        // e -> Public Key
        // d -> Private Key
        BufferedReader keyin = new BufferedReader(new InputStreamReader(System.in));
        String msg, rmsg, code;
        // msg -> String to Encrypt
        //code -> Encrypted String
        // rmsg -> Decrypted String
        int size;
        //Size of the Message
        BigInteger m, c;
        // m and c -> used to store the ASCII values of individual characters in BigInteger format to
encrypt or decrypt

        void read()throws IOException
        {
                System.out.println("Enter the large prime numbers(p and q: Such that p*q > 127):");
                p = new BigInteger(keyin.readLine()); // Read p
                q = new BigInteger(keyin.readLine()); // Read q
                n = p.multiply(q); // n = p*q
                z = (p.subtract(BigInteger.ONE)).multiply(q.subtract(BigInteger.ONE)); // z = (p-1)(q-1)
                System.out.println("Enter the public exponent (e):");
                e = new BigInteger(keyin.readLine()); // Read Public Key (e)
                d = new BigInteger("0"); // Set Private Key (d) to 0
                BigInteger temp;
                // do-while loop to generate private key (d) such that (e*d) mod z = 1
                do
                {
                        d = d.add(BigInteger.ONE); // d++
                        temp = (d.multiply(e)).mod(z); // computing (e*d) mod z
                }while(!temp.equals(BigInteger.ONE)); // the value of d is accepted if temp==1
                System.out.println("Enter Message to Encrypt:");
                msg = keyin.readLine();
                size = msg.length();
                code = "";
                rmsg = "";
        }
        void encrypt()
        {
                for(int i = 0; i < size; i++)
                {
                        m = BigInteger.valueOf((int)msg.charAt(i)); //Copy each character of msg into m
                        c = m.modPow(e, n); // Calculate c = (m^e) mod n
                        code += (char)c.intValue(); // Add encrypted character into code
                }
        }
        void decrypt()
        {
                for(int i = 0; i < size; i++)
                {
                        c = BigInteger.valueOf((int)code.charAt(i)); //Copy each character of code into c
                        m = c.modPow(d, n); // Calculate m = (c^d) mod n
```

```
                    rmsg += (char) m.intValue(); // Add decrypted character into message
                }
        }
        void show()
        {
                System.out.println("\nThe Message Entered at Sender's end is \"" + msg + "\"");
                System.out.println("The Encrypted Message sent to the Receiver is \"" + code + "\"");
                System.out.println("The Decrypted Message at Receiver's end is \"" + rmsg + "\"");
        }
        public static void main(String args[])throws IOException
        {
                RSA obj = new RSA();
                obj.read();
                obj.encrypt();
                obj.decrypt();
                obj.show();
        }
}
```

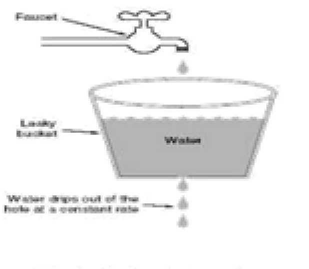
Output:



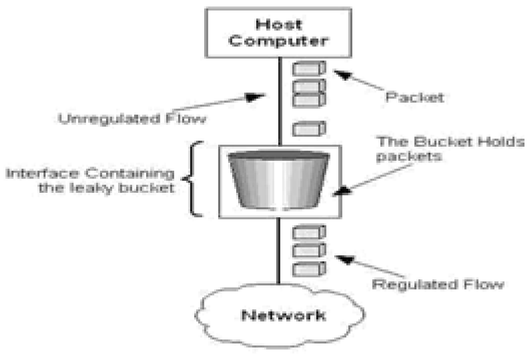
```
ANANTHANUP@ANANTHANUP MINGW64 ~/ARMC-CN (master)
$ javac RSA.java

ANANTHANUP@ANANTHANUP MINGW64 ~/ARMC-CN (master)
$ java RSA
Enter the plain text:
Bmsit
Encrypting String: Bmsit
String in Bytes: 66109115105116
Decrypting Bytes: 66109115105116
Decrypted String: Bmsit

ANANTHANUP@ANANTHANUP MINGW64 ~/ARMC-CN (master)
$
```

6. **Write a program for 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)).

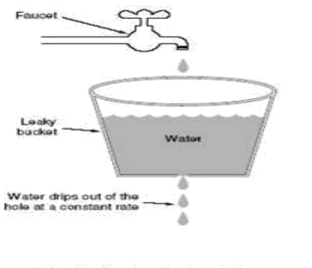


(a) A leaky bucket with water

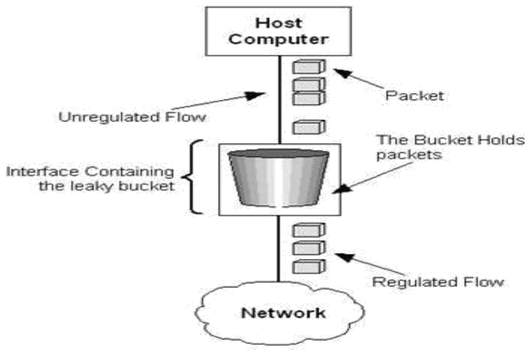


(b) A leaky bucket with packets

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 for some time the bucket becomes empty without permitting the transmission of any packet.



(a) A leaky bucket with water



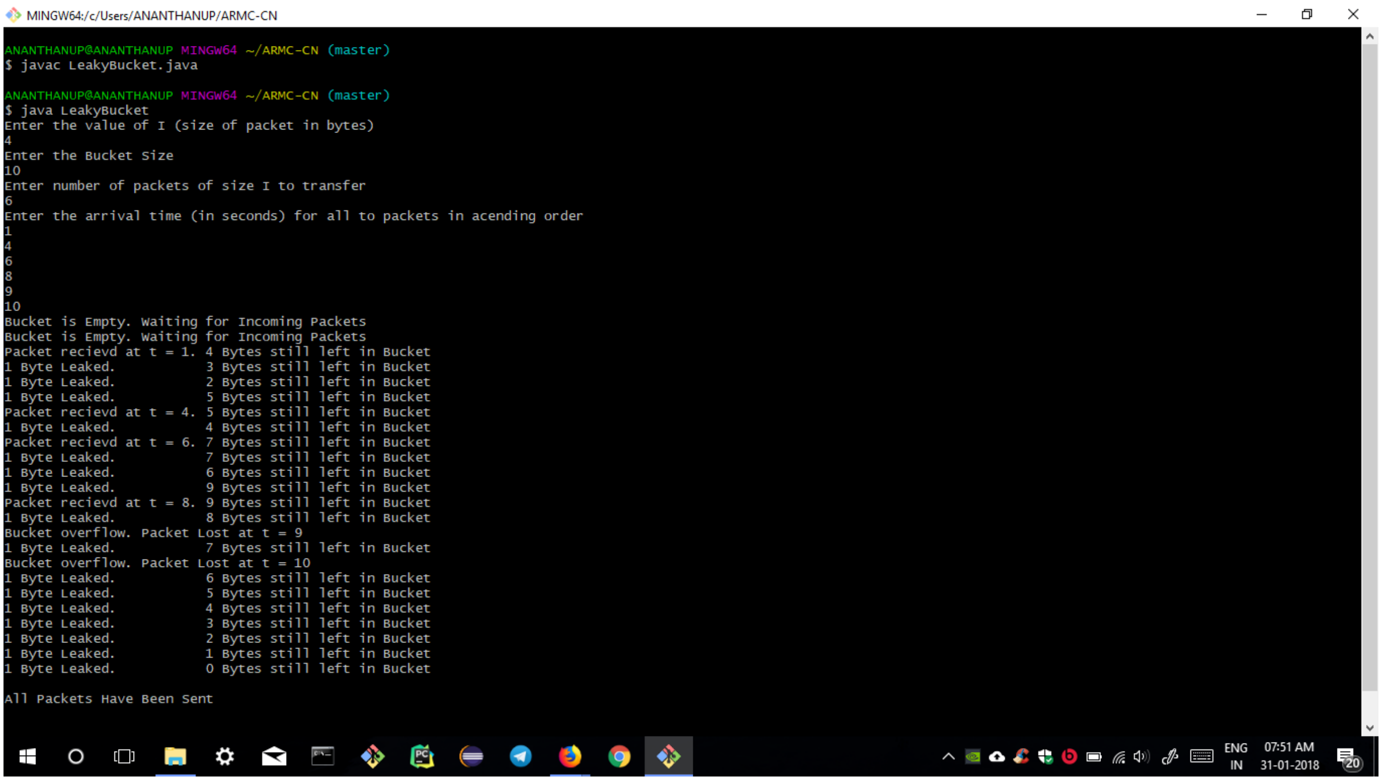
(b) A leaky bucket with packets

Source Code:

```
import java.util.*;

public class LeakyBucket
{
	int I, B, N;
	int dt[];
	int t, p, bs;
	boolean inCtrl = true;
	boolean outCtrl = true;
	void read()
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the value of I (size of packet in bytes)");
		I = sc.nextInt();
		System.out.println("Enter the Bucket Size");
		B = sc.nextInt();
		System.out.println("Enter number of packets of size I to transfer");
		N = sc.nextInt();
		dt = new int[N];
		System.out.println("Enter the arrival time (in seconds) for all to packets in acending order");
		for(int i = 0; i < N; i++)
			dt[i] = sc.nextInt();
		p = bs = 0;
		t = 0;
		sc.close();
	}
	void insert()
	{
		if(p < N)
		{
			if(dt[p] == t)
			{
				if(bs+I <= B)
				{
					bs+=I;
					System.out.println("Packet recievd at t = " + t + ". " + bs + " Bytes still
left in Bucket");
				}
				else
					System.out.println("Bucket overflow. Packet Lost at t = " + t);
				p++;
			}
			t++;
		}
		else
			inCtrl = false;
	}
	void delete()
	{
		if(bs > 0)
		{
			bs--;
			System.out.println("1 Byte Leaked.\t\t " + bs + " Bytes still left in Bucket");
		}
		else
		{
			System.out.println("Bucket is Empty. Waiting for Incoming Packets");
		}
		if(p == N && bs == 0)
			outCtrl = false;
	}
	public static void main(String[] args)
	{
		LeakyBucket bkt = new LeakyBucket();
		bkt.read();
		InsertThread insThread = new InsertThread(bkt); //Call insert thread
		DeleteThread delThread = new DeleteThread(bkt); //Call delete thread
	}
}

class InsertThread implements Runnable
{
	LeakyBucket bkt1;
	public InsertThread(LeakyBucket bkt)
	{
		bkt1 = bkt;
		Thread insThread = new Thread(this);
		insThread.start();
	}
	public void run()
	{
		try
		{
			while(bkt1.inCtrl)
			{
				bkt1.insert();
				Thread.sleep(1000);
			}
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
	}
}

class DeleteThread implements Runnable
{
	LeakyBucket bkt1; //bkt1 is an instance of Bucket class

	public DeleteThread(LeakyBucket bkt)
	{
		this.bkt1 = bkt;  //create a reference
		 Thread delThread = new Thread(this);
		delThread.start();
	}
	public void run()
	{
		try
		{
```

```
                        while(bkt1.outCtrl)
                        {
                                bkt1.delete();
                                Thread.sleep(1000);
                        }
                        System.out.println("\nAll Packets Have Been Sent");
                }
                catch(Exception e)
                {
                        e.printStackTrace();
                }
        }
}
```

Output:



MINGW64:/c/Users/ANANTHANUP/ARMC-CN

```
ANANTHANUP@ANANTHANUP MINGW64 ~/ARMC-CN (master)
$ javac LeakyBucket.java

ANANTHANUP@ANANTHANUP MINGW64 ~/ARMC-CN (master)
$ java LeakyBucket
Enter the value of I (size of packet in bytes)
4
Enter the Bucket Size
10
Enter number of packets of size I to transfer
6
Enter the arrival time (in seconds) for all to packets in acending order
1
4
6
8
9
10
Bucket is Empty. Waiting for Incoming Packets
Bucket is Empty. Waiting for Incoming Packets
Packet recievd at t = 1. 4 Bytes still left in Bucket
1 Byte Leaked.           3 Bytes still left in Bucket
1 Byte Leaked.           2 Bytes still left in Bucket
1 Byte Leaked.           5 Bytes still left in Bucket
Packet recievd at t = 4. 5 Bytes still left in Bucket
1 Byte Leaked.           4 Bytes still left in Bucket
Packet recievd at t = 6. 7 Bytes still left in Bucket
1 Byte Leaked.           7 Bytes still left in Bucket
1 Byte Leaked.           6 Bytes still left in Bucket
1 Byte Leaked.           9 Bytes still left in Bucket
Packet recievd at t = 8. 9 Bytes still left in Bucket
1 Byte Leaked.           8 Bytes still left in Bucket
Bucket overflow. Packet Lost at t = 9
1 Byte Leaked.           7 Bytes still left in Bucket
Bucket overflow. Packet Lost at t = 10
1 Byte Leaked.           6 Bytes still left in Bucket
1 Byte Leaked.           5 Bytes still left in Bucket
1 Byte Leaked.           4 Bytes still left in Bucket
1 Byte Leaked.           3 Bytes still left in Bucket
1 Byte Leaked.           2 Bytes still left in Bucket
1 Byte Leaked.           1 Bytes still left in Bucket
1 Byte Leaked.           0 Bytes still left in Bucket

All Packets Have Been Sent
```

VIVA QUESTION AND ANSWER

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.

28) What is NOS?

NOS, or Network Operating System, is specialized software whose main task is to provide network connectivity to a computer in order for it to be able to communicate with other computers and connected devices.

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 the external 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 traffic volume 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.

88) When troubleshooting computer network problems, what common hardware-related problems 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.

OPEN ENDED EXPERIMENT

1. Write a program to find the shortest path between vertices using dijkstra's algorithm.
2. Write a program for Diffie-Hellman Key Exchange algorithm to encrypt and decrypt the data.

3. Simulate a four node point-to-point network with the links connected as follows: a) n0 – n2, n1 – n2 and n2 – n3. Apply TCP agent between n0-n3 and UDP between n1-n3. Apply relevant applications over TCP and UDP agents, changing the parameter and determine the number of packets sent by TCP / UDP.

4. Simulate an Ethernet LAN using n nodes (6-10), change error rate and data rate and compare throughput.

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA – BANGALORE - 64
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Rubrics used for Continuous Evaluation in every lab session (Record: 10M)

Parameter	Allocated Marks	LOW	MEDIUM	HIGH
Execution	05	The given program was not code/debug/execute in the lab session	The given program was coded & debugged but not executed in the lab session	The given program was coded, debugged and executed in the lab session
		0 marks	**1** -3 **mark**	**4-5marks**
Viva-voce	02	The student did not answered any viva questions asked	The student answered few viva questions asked	The student answered all viva questions asked
		0 marks	**1 marks**	**2 marks**
Record writing	03	The record was not submitted in the lab session	The record was submitted in the lab session but was incomplete (no Algorithm / wrong Algorithm & no Flowchart / wrong Flowchart)	Completed record was submitted in the lab session
		0 marks	**1-2marks**	**3 marks**

Every program is evaluated for 10 Marks and average of all programs for 10 marks will be calculated. The final average is converted to 5 Marks.

Level 2: Lab Internal test (10 marks)

Two Lab Internals will be conducted:

1. **1stcycle test:** Mid semester after completion of 50% of university specified experiment.
2. **2ndcycle test:** End of semester after completion of all the university specified experiment

   In each cycle test the student pick a program from the pool and execute that program. The student should answer the Viva-voce asked. The marks are awarded for each lab internals based on rubrics defined in table 2.11 average marks of both the Internal test is considered for awarding final internal assessment marks.

Rubrics used for Lab Internal Test

Parameter	Allocated Marks	LOW	MEDIUM	HIGH
Program Write-up	03	The Student was not able to write the program & Algorithm or flowchart	The Student was able to write the program & Algorithm or flowchart with mistakes	The Student was able to write the program& Algorithm or flowchart correctly
		0 marks	**1-2 marks**	**3 marks**
Execution	05	The Student was not able to code/ debug/ execute the program	The Student was partially able to code/debug/execute the program	The Student was able to code/debug/execute the program
		0 marks	**1-2 marks**	**3-5 marks**
Viva-voce	02	The student did not answered any viva questions asked	The student answered few viva questions asked	The student answered all viva questions asked
		0mark	**1 mark**	**2 marks**

Level 3: Project Based Learning (10 marks)

Team of 4 students are assigned a mini project which is not there in syllabus (content beyond syllabus) and applied to real time scenario. Students should complete the project and submit the report to the lab course coordinator. On Open day Exhibition student prepare posters and show demo of their project which will be evaluated by External Examiner. This we called as Project based learning and convert the final marks to 10 Marks which is included in internal marks.

Parameter	Allocated Marks	LOW	MEDIUM	HIGH
Relevance to Society/ Industry	02	Not relevant to industry or society Needs	Little relevance to Industry/Society	High relevant to Industry/social need
		0 marks	**1 mark**	**2 marks**
Tools and Technology	02	Student not aware of latest tools and technology used	Student aware of latest tools and technology but not used	Student aware of latest tools and technology and used
		0 marks	**1 mark**	**2 marks**
Implementation	05	Project is not completed	Project is partially completed	Project is fully completed
		0 marks	**1-2 marks**	**3-5 marks**
Viva-voce	03	The student did not answered any viva questions	The student answered few viva questions asked	The student answered all viva questions asked

		asked		
		0marks	**1 mark**	**2 marks**
Project and Finance Management	03	The Student not specified the work schedule and budget of project	The Student specifies work schedule but not budget or vice versa	The student specifies clearly about work schedule and budget
		0marks	**1 mark**	**2 -3 marks**
Report with Unplagarised content (Ethics)	03	Report was submitted and Plagiarism is more than 35%	The report was well organized and submitted. plagiarism content is between 25% to 35%	The report was well organized and submitted. plagiarism content is less than 25%
		0marks	**1 mark**	**2-3 marks**
Team work	02	No coordination between team members and not able to work in team	-	The student is able to work in teams
		0marks		**1-2 marks**

Level 4: Assignment (10 marks)

Assignments based on Lab programs (open ended Experiments) or topics will be given. Students have to complete the assignment and will be evaluated based on the following rubrics**.**

Parameter	**Allocated Marks**	**LOW**	**MEDIUM**	**HIGH**
Demo	**05**	The Student was not able to demonstrated the program	The Student was partially able to demonstrate program	The Student was able to demonstrate the program and complete successfully
		0 Marks	**1-2 Marks**	**3-5 Marks**
Presentation	**02**	Not able to present well	Able to present somewhat	Able to present well
		0 Marks	**1 Mark**	**2 Marks**
Report	03	Report Not submitted for the given assignment	Report submitted and is not organized and not fully completed	Report Completed fully and well organized and submitted
		0 Marks	**1-5 marks**	**6-10 marks**