Kammavari Sangham (R) 1952,
K.S.Group of Institutions

K S INSTITUTE OF TECHNOLOGY

#14,Raghuvanahalli, Kanakapura Main Road,Bengaluru-5600109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER NETWORK LABORATORY MANUAL
[15CSL57]

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2016 -2017)

SEMESTER – V

Prepared by
SANGEETHA.V M.Tech(Ph.D)
Associate Professor
Department of Computer Science & Engineering

DEPARTMENT VISION

To provide competent and responsible professionals in the field of Computer Science and Engineering with knowledge and skills required for country in its quest for development.

DEPARTMENT MISSION

- Inculcate strong theoretical and practical knowledge for continuous learning.
- Prepare students to find Computer Solutions for the society through research and entrepreneurship with professional ethics.
- Encourage team work in inter-disciplines and evolve as leaders with social concerns.

Lab Experiments

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.

PART A - Simulation Exercises

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.

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:

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

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

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.

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

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

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

Course outcomes: The students should be able to:

- Analyze and Compare various networking protocols.
- Demonstrate the working of different concepts of networking.
- Implement, analyze and evaluate networking protocols in NS2 / NS3

Conduction of Practical Examination:

1. All laboratory experiments are to be included for practical examination.
2. Students are allowed to pick one experiment from part A and part B with lot.
3. Strictly follow the instructions as printed on the cover page of answer script
4. Marks distribution: Procedure + Conduction + Viva: 80

   Part A: 10+25+5 =40

   Part B: 10+25+5 =40

5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

INTRODUCTION

Network simulators implemented in software are valuable tools for researchers to develop, test, and diagnose network protocols. Simulation is economical because it can carry out experiments without the actual hardware. It is flexible because it can, for example, simulate a link with any bandwidth and propagation delay. Simulation results are easier to analyze than experimental results because important information at critical points can be easily logged to help researchers diagnose network protocols.

Network simulators, however, have their limitations. A complete network simulator needs to simulate networking devices (e.g., hosts and routers) and application programs that generate network traffic. It also needs to provide network utility programs to configure, monitor, and gather statistics about a simulated network. Therefore, developing a complete network simulator is a large effort.

Different types of simulators

Some of the different types of simulators are as follows:

- NETSIM
- REAL
- NS
- OPNET
- NCTUns
- NIST
- CPSIM
- INSANE
- NEST

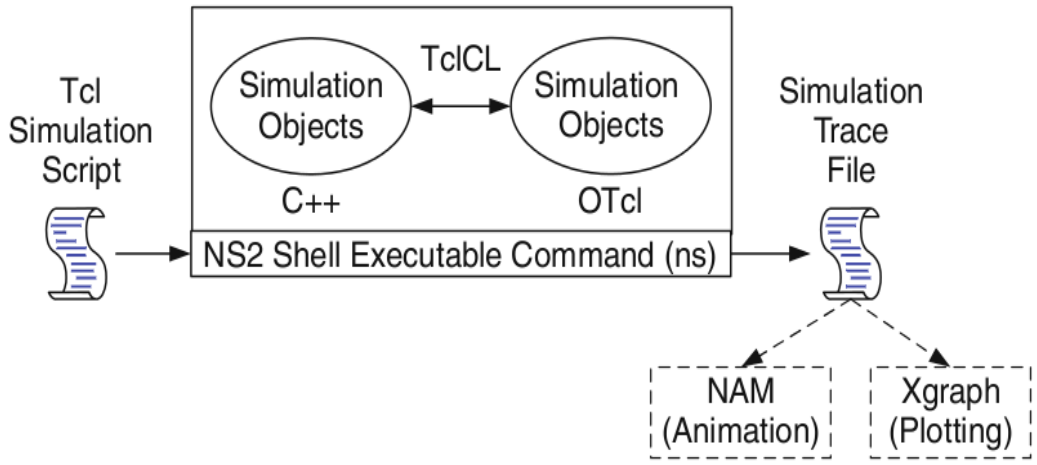
NS version 2.0(NS-2)

NS is a object-oriented discrete-event simulator for networking research based on REAL. Initially, NS version 1.0 was developed by the Network Research Group at the Lawrence Berkeley National Laboratory (LBNL). Its development is now part of the VINT project under which NS version 2.0 was released. At the time being, NS is well suited for packets switched networks and is used mostly for small-scale simulations of queuing algorithms, transport

protocol congestion control, and some multicast related work. It provides support for various implementations of TCP, routing, multicast protocols, link layer, MAC, ... It currently has memory limitations in the face of large simulations.

- NS2 is 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 behaviors.

Basic Architecture of NS2



Steps to Write TCL Script for Wired Network

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

Create the event scheduler

Initialization and Termination of TCL Script in NS-2

An ns simulation starts with the command

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

The first line in the tcl script, 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.

Open new files & turn on the tracing

To create 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 "tracefile" and "namfile" respectively.

Open the Trace file

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

Open the file "out.tr" to be used for writing, declared with the letter "w". Simulator method called trace-all that have as parameter the name of the file where the traces will go.

Open the NAM trace file

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

The simulator will 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.

Create the nodes

The way to define a node is

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

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

Set up the links

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-2, 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-2, 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:

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

Configure the traffic type (e.g., TCP, UDP, etc)

Traffic type -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.

Command to Create Source agent

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

The command to attach agent to source node of the tcp connection.

```
$ns attach-agent $n0 $tcp
```

Command to Create Destination agent and then connect source and destination agents

```
set sink [new Agent /TCPSink]
$ns attach-agent $n1 $sink
$ns connect $tcp $sink
```

Similarly Traffic type -UDP

```
set udp [new Agent/UDP]
$ns attach-agent $n2 $udp

set null [new Agent/Null]
$ns attach-agent $n4 $null

$ns connect $udp $null
```

Set the type of traffic generation (e.g., CBR, FTP)

Setup a FTP over TCP connection

The below shows the definition of a FTP application using a TCP agent

```
set tcp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set packetSize_ 100
$ftp set rate_ 0.01Mb
$ftp set random_ false
```

Setup a CBR over UDP connection

The below shows the definition of a CBR application using a UDP agent

```
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set packetSsize_ 100
$cbr set rate_ 0.01Mb
$cbr set random_ false
```

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 1.0  " $ftp start"
$ns at 2.0  "$ftp stop"
$ns at 3.0  "$cbr start"
$ns at 3.5  "$cbr stop"
```

Terminate the simulation

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

Define a 'finish' procedure

```
Proc finish { } {
global ns tracefile namfile
$ns flush-trace
Close $tracefile
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 5.0 "finish"
```

will be used to call "**finish**" at time 5sec. Indeed, the **at** method of the simulator allows us to schedule events explicitly.

The simulation can then begin using the command

```
$ns run
```

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.

Syntax:

```
awk  -f   filename.awk  filename.tr
```

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.

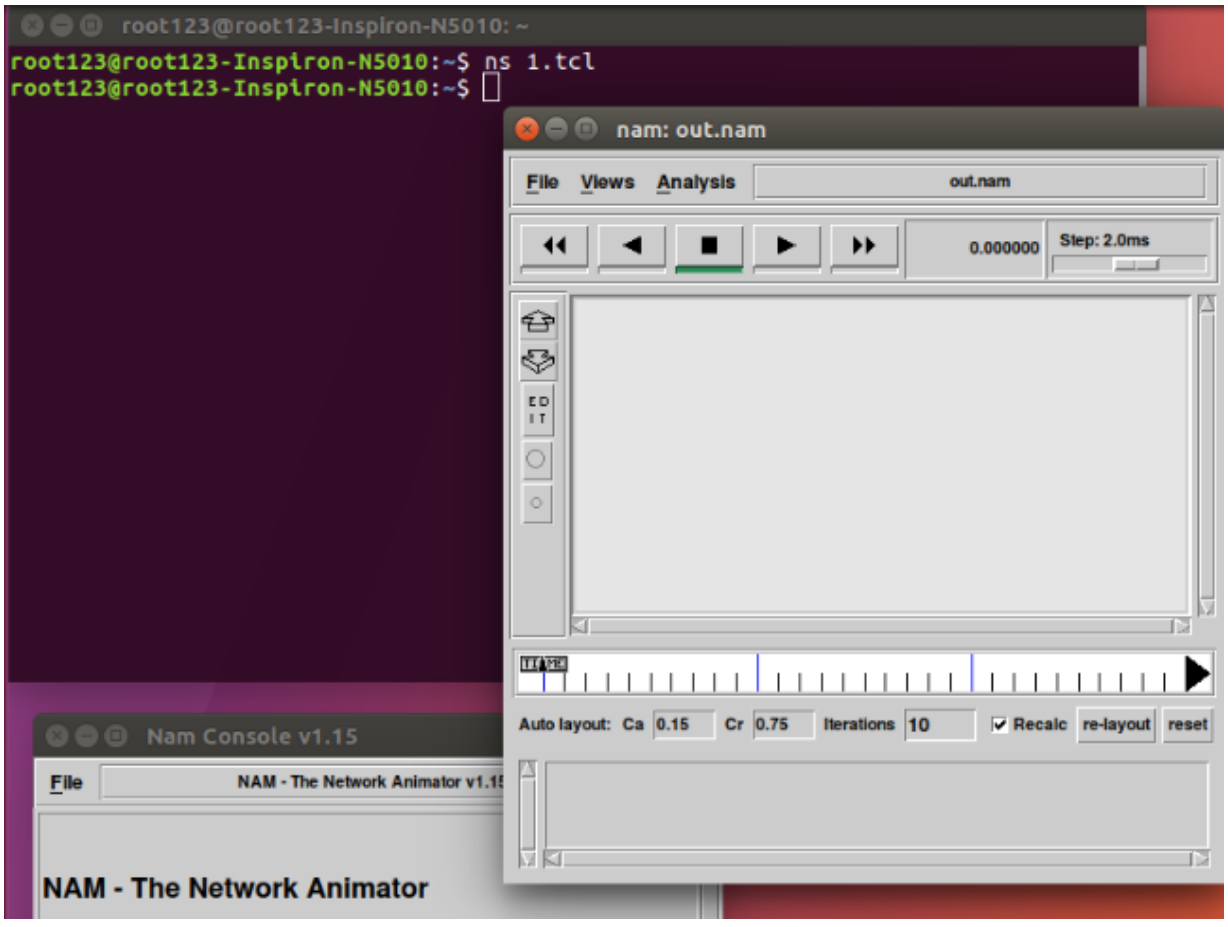
The BEGIN and END sections are optional and take the form

BEGIN {
Action
}
END {
Action
}

Sample TCL scripts

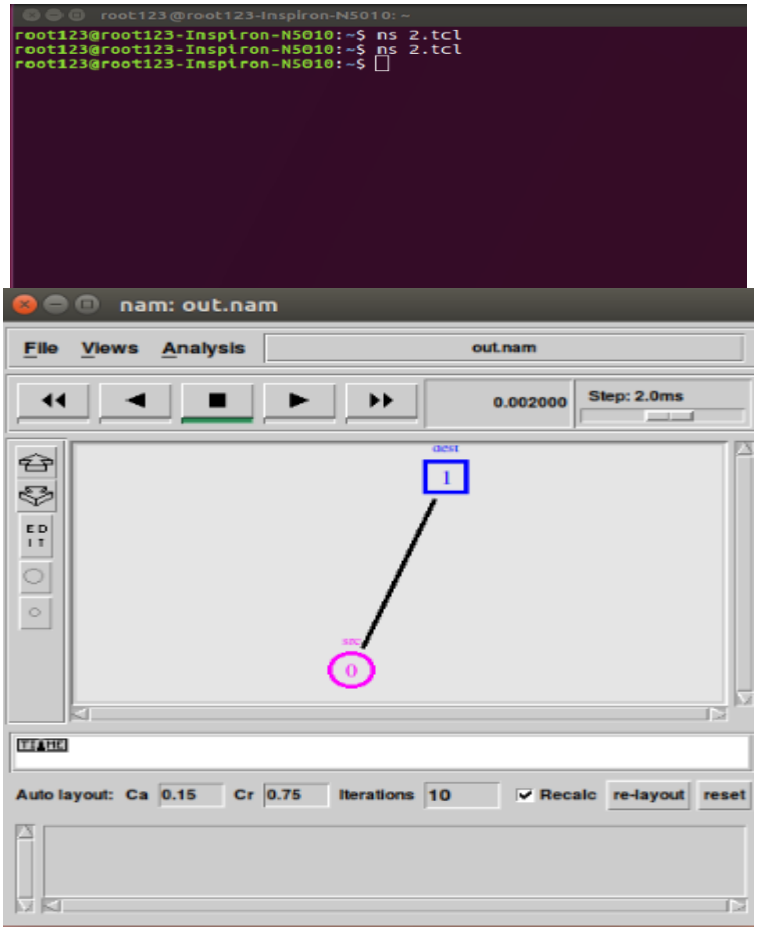
Example 1: 1.tcl

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
global ns nf
$ns flush-trace
close $nf
exec nam out.nam &
exit 0
}
$ns at 5.0 "finish"
$ns run
```



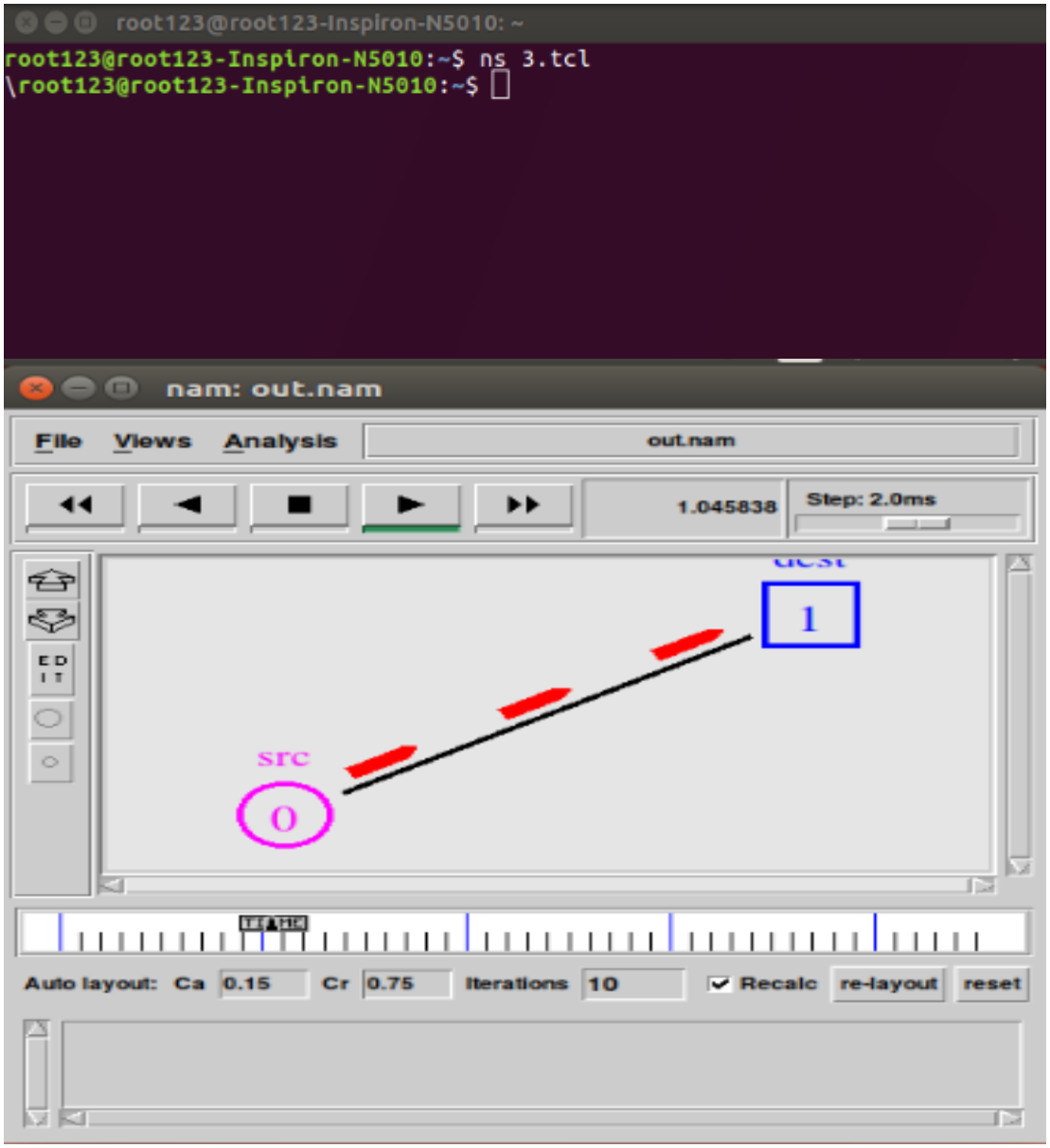
Example 2: 2.tcl

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set n0 [$ns node]
$n0 color "magenta"
$n0 label "src"
set n1 [$ns node]
$n1 color "blue"
$n1 label "dest"
$n1 shape square
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
proc finish {} {
global ns nf
$ns flush-trace
close $nf
exec nam out.nam &
exit 0
}
$ns at 5.0 "finish"
$ns run
```



Example 3: 3.tcl

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set n0 [$ns node]
$n0 color "magenta"
$n0 label "src"
set n1 [$ns node]
$n1 color "blue"
$n1 label "dest"
$n1 shape square
$ns color 1 "red"
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize 500
$cbr0 set interval  0.005
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$udp0 set class_ 1
$ns connect $udp0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
proc finish {} {
global ns nf
$ns flush-trace
close $nf
exec nam out.nam &
exit 0
}
$ns at 5.0 "finish"
$ns run
```


root123@root123-Inspiron-N5010: ~
root123@root123-Inspiron-N5010:~$ ns 3.tcl
\root123@root123-Inspiron-N5010:~$
nam: out.nam
File
Views
Analysis
out.nam
1.045838
Step: 2.0ms
1
src
0
Auto layout:
Ca
0.15
Cr
0.75
Iterations
10
Recalc
re-layout
reset

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.

Step1: Open text editor, type the below program and save with extention .tcl (**lab1.tcl**)

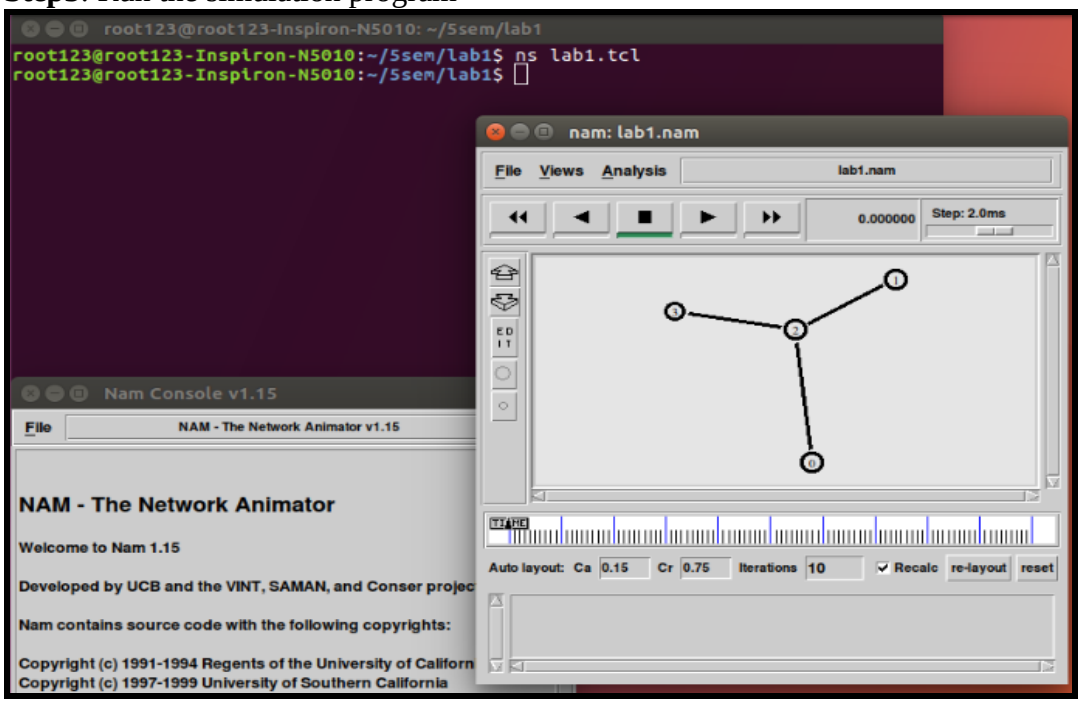
```
set ns [ new Simulator ]
set tf [ open lab1.tr w ]
$ns trace-all $tf
set nf [ open lab1.nam w ]
$ns namtrace-all $nf
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns color 1 "red"
$ns color 2 "blue"
$n0 label "Source/udp0"
$n1 label "Source/udp1"
$n2 label "Router"
$n3 label "Destination/Null"
$ns duplex-link $n0 $n2 100Mb 300ms DropTail
$ns duplex-link $n1 $n2 100Mb 300ms DropTail
$ns duplex-link $n2 $n3 1Mb 300ms DropTail
$ns set queue-limit $n0 $n2 50
$ns set queue-limit $n1 $n2 50
$ns set queue-limit $n2 $n3 5
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
set null3 [new Agent/Null]
$ns attach-agent $n3 $null3
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$udp0 set class_ 1
$udp1 set class_ 2
$ns connect $udp0 $null3
$ns connect $udp1 $null3
$cbr0 set packetSize_ 500Mb
$cbr0 set interval_ 0.005
$cbr1 set packetSize_ 500Mb
$cbr1 set interval_ 0.005
```

```
proc finish { } {
global ns nf tf
$ns flush-trace
exec nam lab1.nam &
close $tf
close $nf
exit 0
}
$ns at 0.1 "$cbr0 start"
$ns at 0.1 "$cbr1 start"
$ns at 10.0 "finish"
$ns run
```

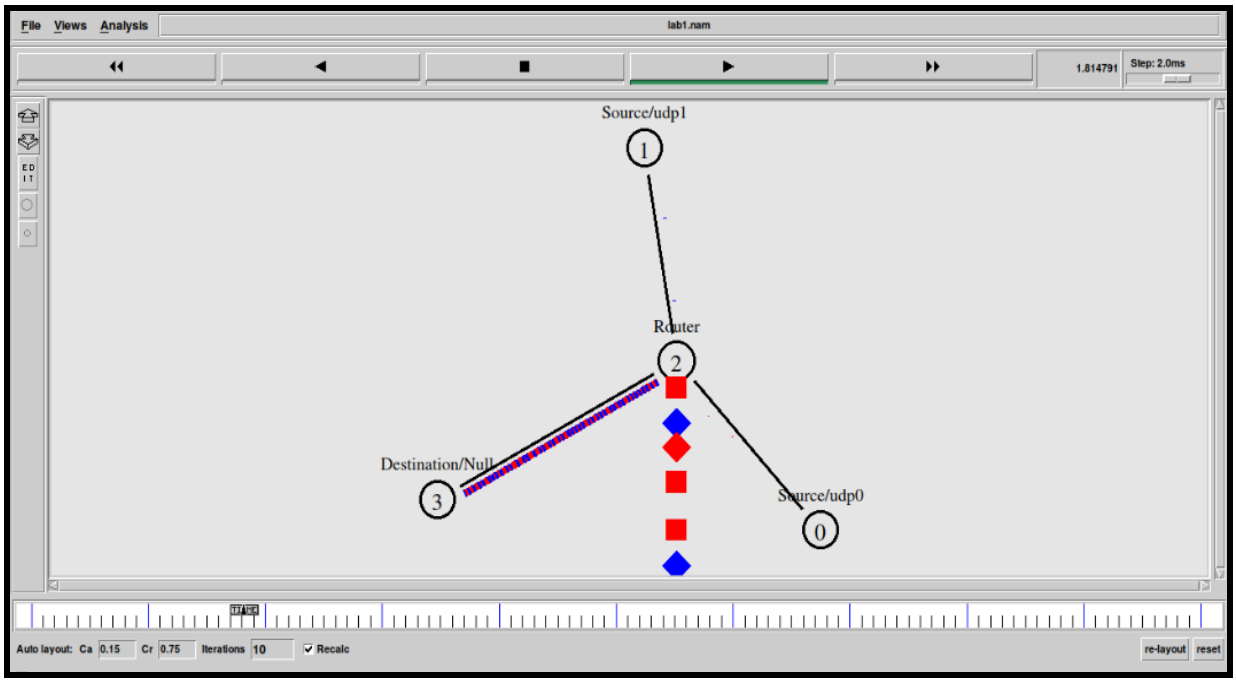
Step2: Open text editor, type the below program and save with extention .awk (**lab1.awk**)

```
BEGIN{
          count=0;
     }
{
     if($1=="d")
     count++
}
END{
     printf("The Total no of Packets Drop is :%d\n\n", count)
    }
```

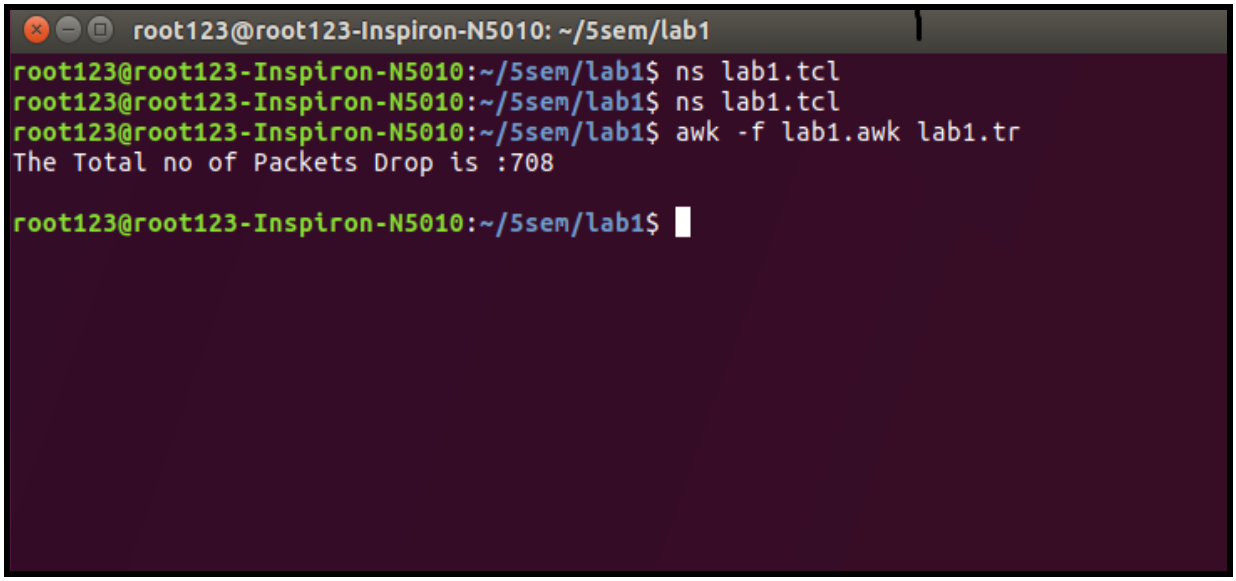
Step3: Run the simulation program



Step 4: Now press the play button in the simulation window and the simulation will begins.

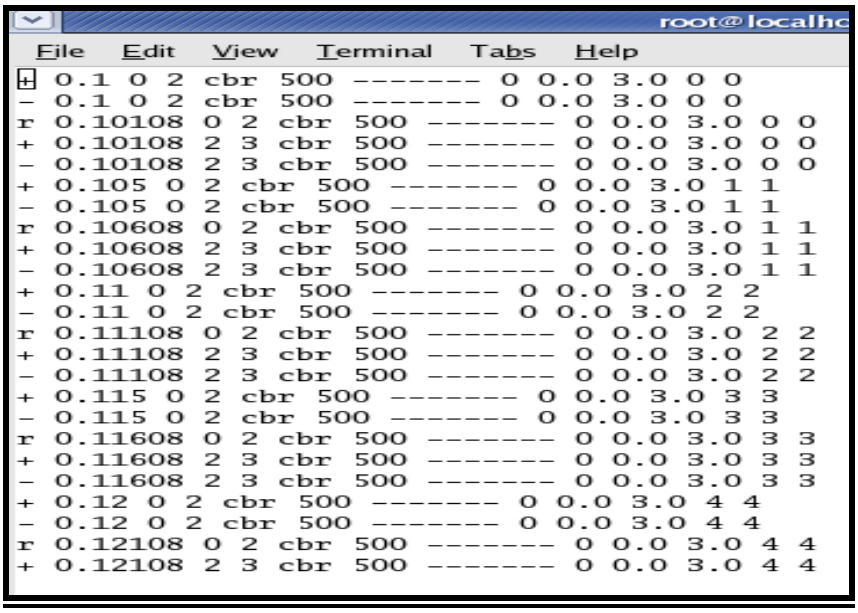


Step 5: After simulation is completed run **awk file** to see the output ,



Step 6: To see the trace file contents open the file as ,

[root@localhost~]# gedit lab1.tr



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.

Step1: Open text editor, type the below program and save with extention .tcl (**lab2.tcl**)

```
set ns [new Simulator]
set tf [open lab2.tr w]
$ns trace-all $tf
set nf [open lab2.nam w]
$ns namtrace-all $nf
set n0 [$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]
$n0 label "Ping0"
$n4 label "Ping4"
$n5 label "Ping5"
$n6 label "Ping6"
$n2 label "Router"
$ns color 1 "red"
$ns color 2 "green"
$ns duplex-link $n0 $n2 100Mb 300ms DropTail
$ns duplex-link $n1 $n2 1Mb 300ms DropTail
$ns duplex-link $n3 $n2 1Mb 300ms DropTail
$ns duplex-link $n5 $n2 100Mb 300ms DropTail
$ns duplex-link $n2 $n4 1Mb 300ms DropTail
$ns duplex-link $n2 $n6 1Mb 300ms DropTail
$ns queue-limit $n0 $n2 5
$ns queue-limit $n2 $n4 3
$ns queue-limit $n2 $n6 2
$ns queue-limit $n5 $n2 5
```

#The below code is used to connect between the ping agents to the node n0, n4 , n5 and n6.

```
set ping0 [new Agent/Ping]
$ns attach-agent $n0 $ping0
set ping4 [new Agent/Ping]
$ns attach-agent $n4 $ping4
set ping5 [new Agent/Ping]
$ns attach-agent $n5 $ping5
set ping6 [new Agent/Ping]
$ns attach-agent $n6 $ping6
$ping0 set packetSize_ 50000
$ping0 set interval_ 0.0001
```

```
$ping5 set packetSize_ 60000
$ping5 set interval_ 0.00001
$ping0 set class_ 1
$ping5 set class_ 2

$ns connect $ping0 $ping4
$ns connect $ping5 $ping6
```

#Define a 'recv' function for the class 'Agent/Ping' and the below function is executed when the ping agent receives a reply from the destination

```
Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts " The node [$node_ id] received an reply from $from with
round trip time of $rtt"
}

proc finish {} {
global ns nf tf
exec nam lab2.nam &
$ns flush-trace
close $tf
close $nf
exit 0
}
```

#Schedule events

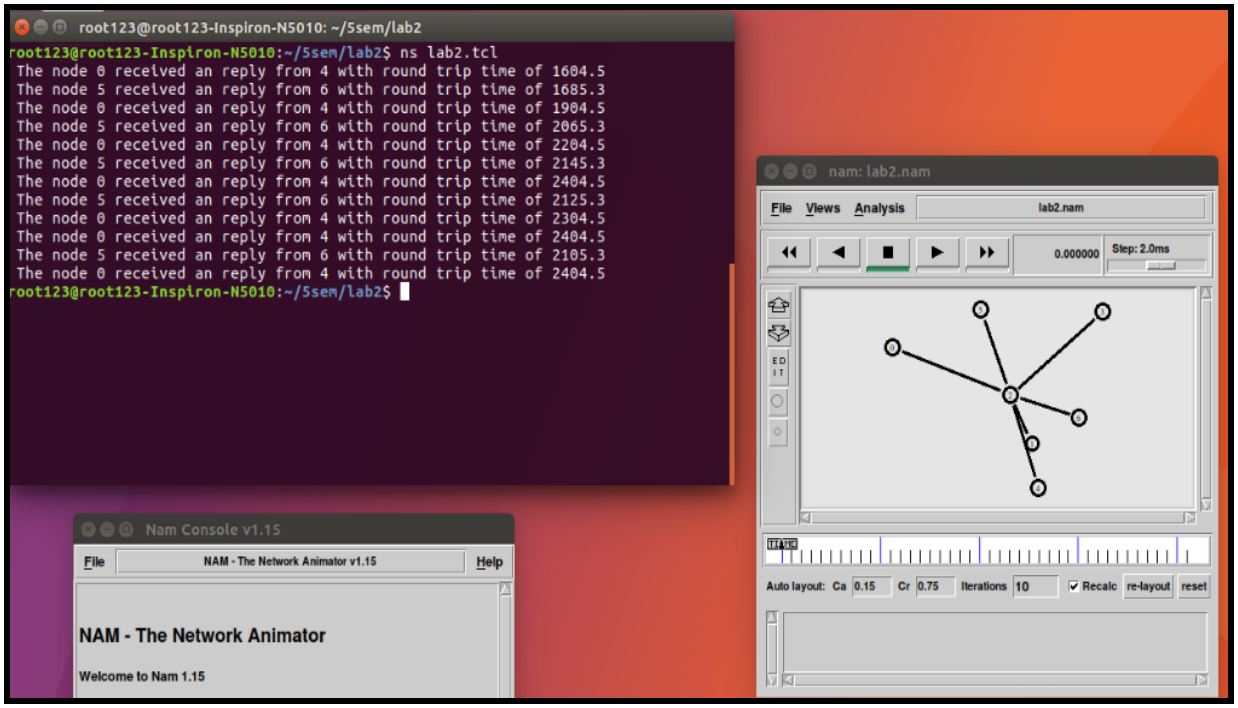
```
$ns at 0.1 "$ping0 send"
$ns at 0.2 "$ping0 send"
$ns at 0.3 "$ping0 send"
$ns at 0.4 "$ping0 send"
$ns at 0.5 "$ping0 send"
$ns at 0.6 "$ping0 send"
$ns at 0.7 "$ping0 send"
$ns at 0.8 "$ping0 send"
$ns at 0.9 "$ping0 send"
$ns at 1.0 "$ping0 send"
$ns at 1.1 "$ping0 send"
$ns at 1.2 "$ping0 send"
$ns at 1.3 "$ping0 send"
$ns at 1.4 "$ping0 send"
$ns at 1.5 "$ping0 send"
$ns at 1.6 "$ping0 send"
$ns at 1.7 "$ping0 send"
$ns at 1.8 "$ping0 send"
$ns at 0.1 "$ping5 send"
```

```
$ns at 0.2 "$ping5 send"
$ns at 0.3 "$ping5 send"
$ns at 0.4 "$ping5 send"
$ns at 0.5 "$ping5 send"
$ns at 0.6 "$ping5 send"
$ns at 0.7 "$ping5 send"
$ns at 0.8 "$ping5 send"
$ns at 0.9 "$ping5 send"
$ns at 1.0 "$ping5 send"
$ns at 1.1 "$ping5 send"
$ns at 1.2 "$ping5 send"
$ns at 1.3 "$ping5 send"
$ns at 1.4 "$ping5 send"
$ns at 1.5 "$ping5 send"
$ns at 1.6 "$ping5 send"
$ns at 1.7 "$ping5 send"
$ns at 1.8 "$ping5 send"
$ns at 5.0 "finish"
$ns run
```

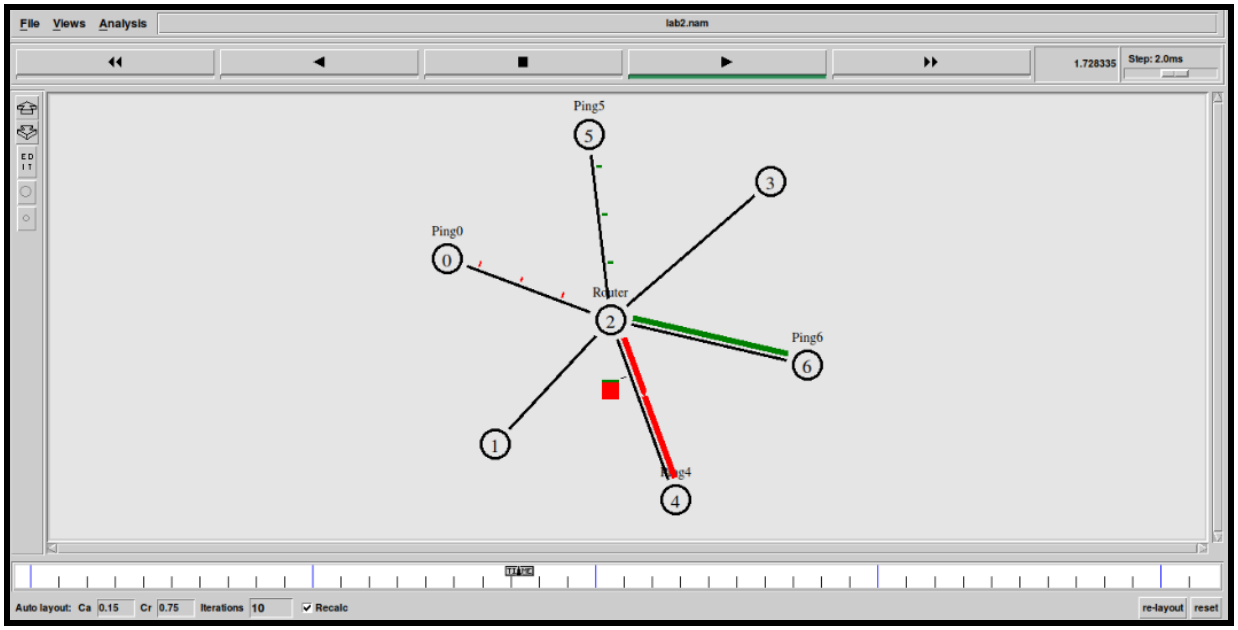
Step2: Open text editor, type the below program and save with extention .awk (**lab2.awk**)

```
BEGIN{
count=0;
}
{
     if($1=="d")
     count++
}
END{
     printf("The Total no of Packets Drop is :%d\n\n", count)
}
```

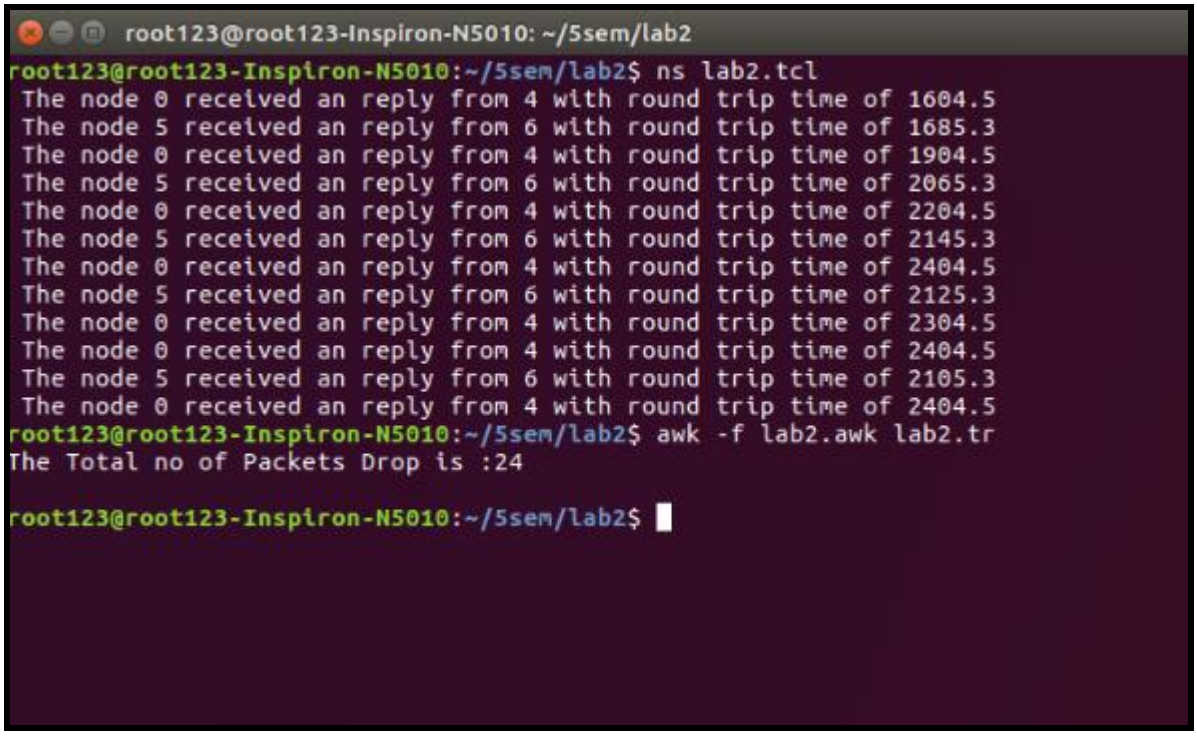
Step3: Run the simulation program



Step 4: Now press the play button in the simulation window and the simulation will begins.



Step 5: After simulation is completed run **awk file** to see the output ,



Step 6: To see the trace file contents open the file a ,

[root@localhost~]# gedit lab2.tr

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

Step1: Open text editor, type the below program and save with extention .tcl (**lab3.tcl**)

```
set ns [new Simulator]
set tf [open lab3.tr w]
$ns trace-all $tf
set nf [open lab3.nam w]
$ns namtrace-all $nf

# Create the nodes,color and label
set n0 [$ns node]
$n0 color "magenta"
$n0 label "src1"
set n1 [$ns node]
$n1 color "red"
set n2 [$ns node]
$n2 color "magenta"
$n2 label "src2"
set n3 [$ns node]
$n3 color "blue"
$n3 label "dest2"
set n4 [$ns node]
$n4 shape square
set n5 [$ns node]
$n5 color "blue"
$n5 label "dest1"

#Creates a lan from a set of nodes given by <nodelist>. Bandwidth, delay characteristics along with the link-layer, Interface queue, Mac layer and channel type for the lan also needs to be defined.

$ns make-lan "$n0 $n1 $n2 $n3 $n4" 50Mb 100ms LL Queue/DropTail
Mac/802_3

# Create the link
$ns duplex-link $n4 $n5 1Mb 1ms DropTail

# Create the node position
$ns duplex-link-op $n4 $n5 orient right

# Add a TCP sending module to node n0
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
```

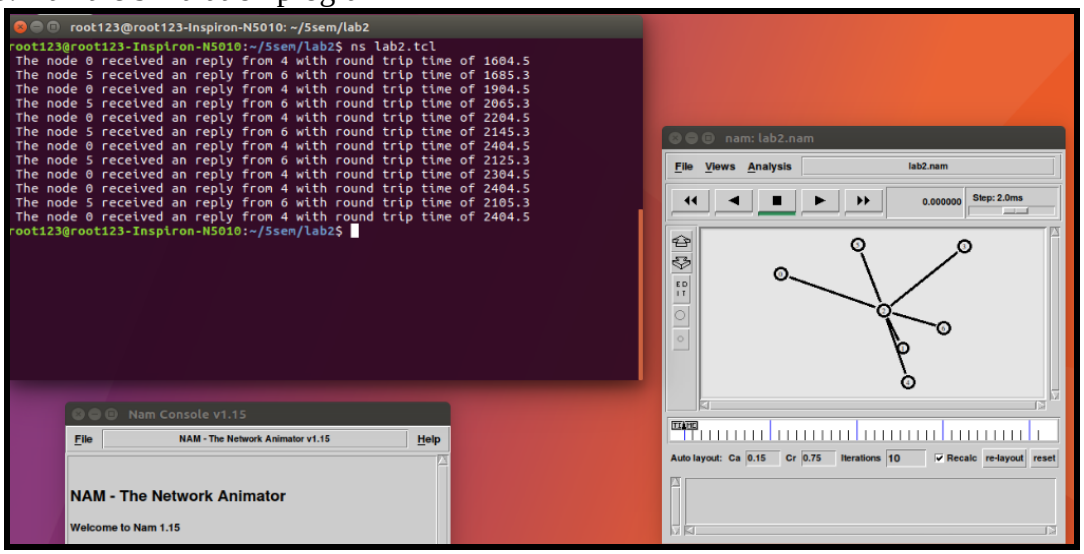
```
# Setup a FTP traffic generator on "tcp0"
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ftp0 set packetSize_ 500
$ftp0 set interval_ 0.0001


# Add a TCP receiving module to node n5
set sink0 [new Agent/TCPSink]
$ns attach-agent $n5 $sink0

# Direct traffic from "tcp0" to "sink1"
$ns connect $tcp0 $sink0

# Add a TCP sending module to node n2
set tcp1 [new Agent/TCP]
$ns attach-agent $n2 $tcp1

# Setup a FTP traffic generator on "tcp1"
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ftp1 set packetSize_ 600
$ftp1 set interval_ 0.001

# Add a TCP receiving module to node n3
set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink1

# Direct traffic from "tcp1" to "sink1"
$ns connect $tcp1 $sink1


set file1 [open file1.tr w]
$tcp0 attach $file1

set file2 [open file2.tr w]
$tcp1 attach $file2

$tcp0 trace cwnd_
$tcp1 trace cwnd_

# Define a 'finish' procedure
proc finish { } {
global ns nf tf
$ns flush-trace
close $tf
close $nf
```

```
exec nam lab3.nam &
exit 0
}

# Schedule start/stop times
$ns at 0.1 "$ftp0 start"
$ns at 5 "$ftp0 stop"
$ns at 7 "$ftp0 start"
$ns at 0.2 "$ftp1 start"
$ns at 8 "$ftp1 stop"
$ns at 14 "$ftp0 stop"
$ns at 10 "$ftp1 start"
$ns at 15 "$ftp1 stop"

# Set simulation end time
$ns at 16 "finish"

$ns run
```

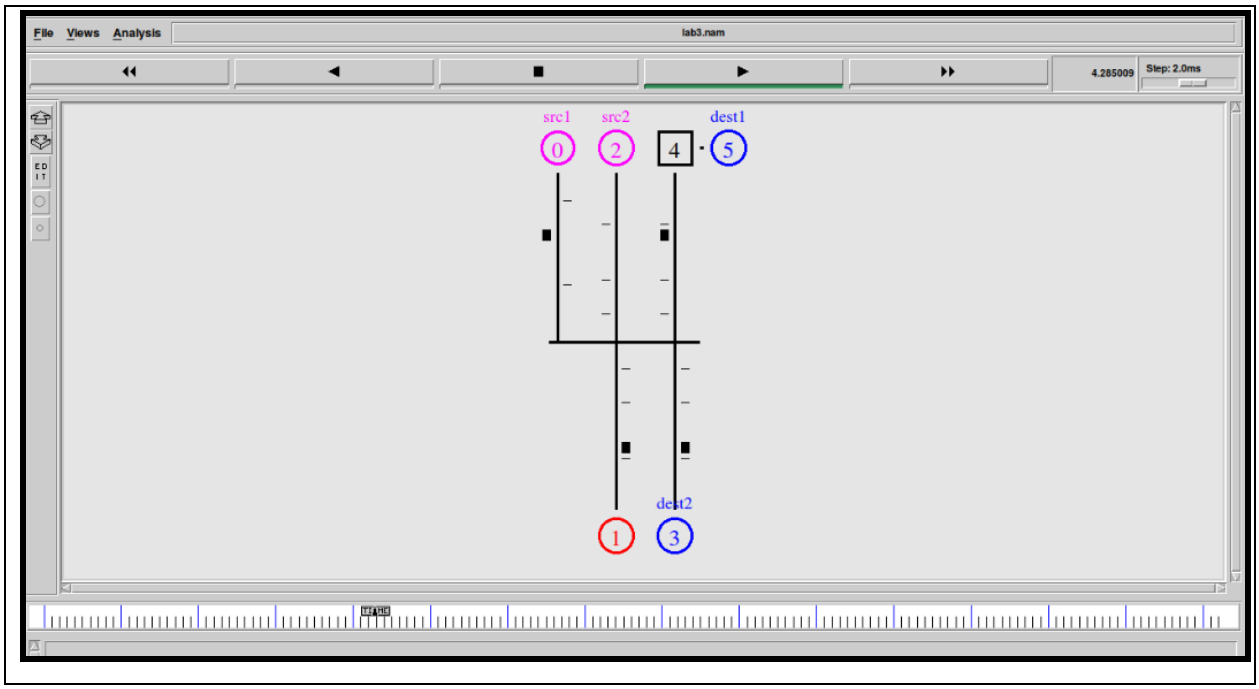
Step2: Open text editor, type the below program and save with extention .awk (**cwd.awk**)

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

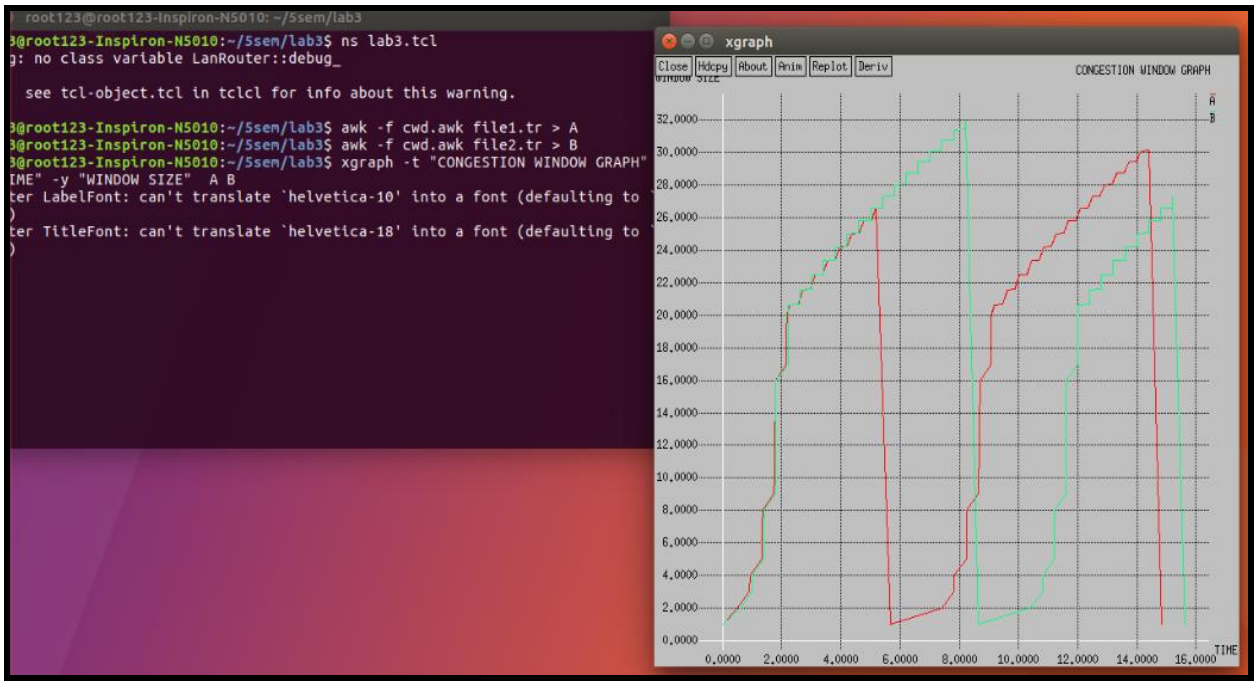
Step3: Run the simulation program



Step 4: Now press the play button in the simulation window and the simulation will begins.



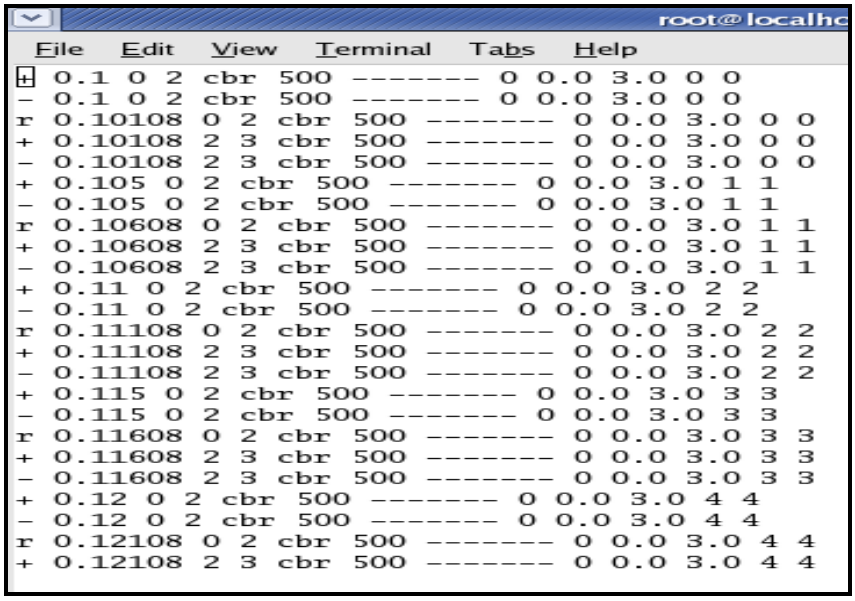
Step 5: After simulation is completed run **awk file** to see the output ,



Step 6: To see the trace file contents open the file as ,

[root@localhost~]# gedit file1.tr

[root@localhost~]# gedit file2.tr



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

Step1: Open text editor, type the below program and save with extention .tcl (**lab4.tcl**)

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

#setup trace support by opening file lab4.tr and call the
procedure trace-all
set tf [open lab4.tr w]
$ns trace-all $tf

#create a topology object that keeps track of movements of
mobilenodes within the topological boundary.
set topo [new Topography]
$topo load_flatgrid 1000 1000


set nf [open lab4.nam w]
$ns namtrace-all-wireless $nf 1000 1000

# creating a wireless node you MUST first select (configure)
the node configuration parameters to "become" a wireless node.

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


# Create god object
create-god 3

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

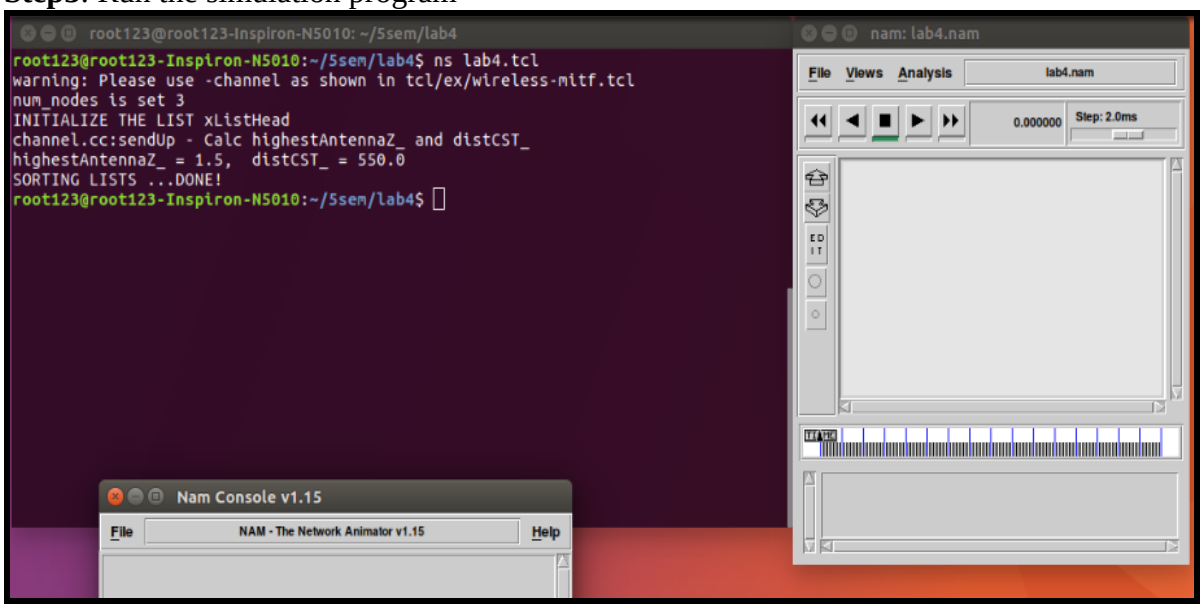
```
$n0 label "tcp0"
$n1 label "sink1/tcp1"
$n2 label "sink2"

$n0 set X_ 50
$n0 set Y_ 50
$n0 set Z_ 0

$n1 set X_ 100
$n1 set Y_ 100
$n1 set Z_ 0

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

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

set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0

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

set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1

set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2

$ns connect $tcp1 $sink2

$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"

$ns at 100 "$n1 setdest 550 550 15"
$ns at 190 "$n1 setdest 70 70 15"
```

```
proc finish { } {
global ns nf tf
$ns flush-trace
exec nam lab4.nam &
close $tf
exit 0
}
$ns at 250 "finish"
$ns run
```

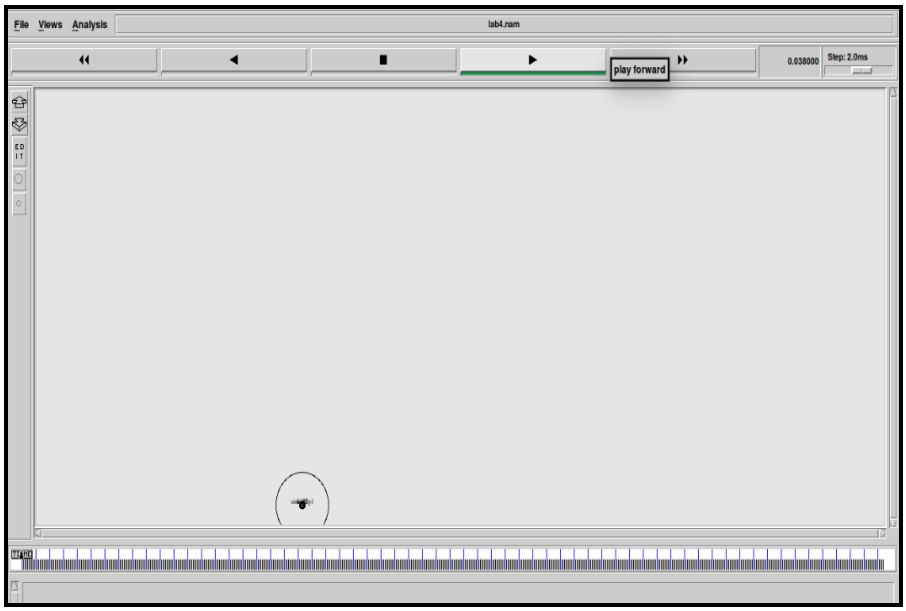
Step2: Open text editor, type the below program and save with extention .awk (**lab4.awk**)

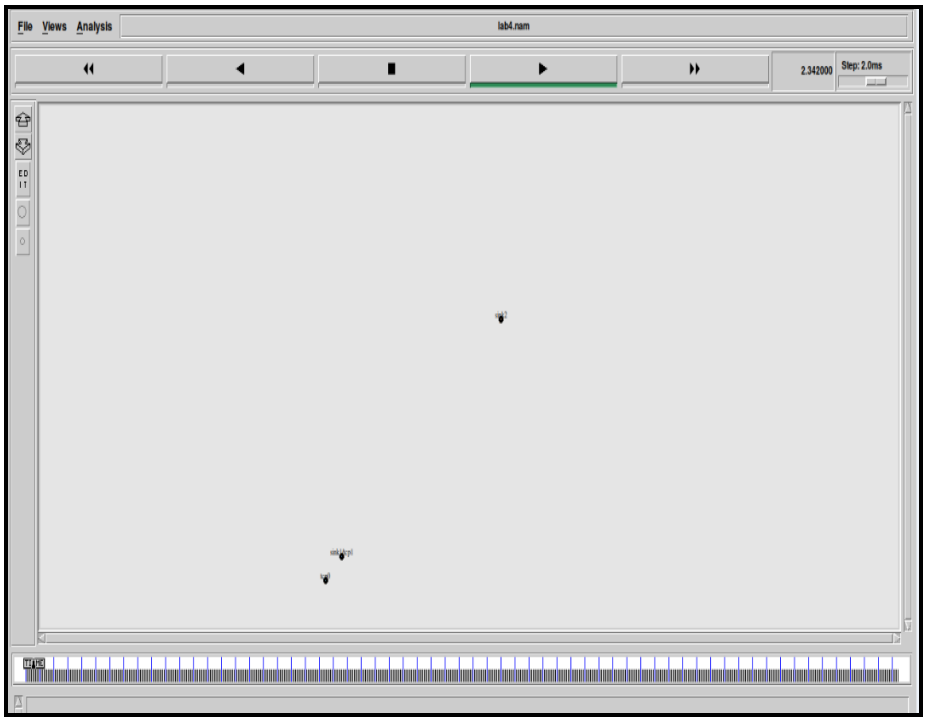
```
BEGIN{
     count1=0
     count2=0
     pack1=0
     pack2=0
     time1=0
     time2=0
}
{
     if($1 == "r" && $3 == "_1_" && $4 == "AGT")
     {
          count1++
          pack1=pack1+$8
          time1=$2
     }
     if($1 == "r" && $3 == "_2_" && $4 =="AGT")
     {
          count2++
     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

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

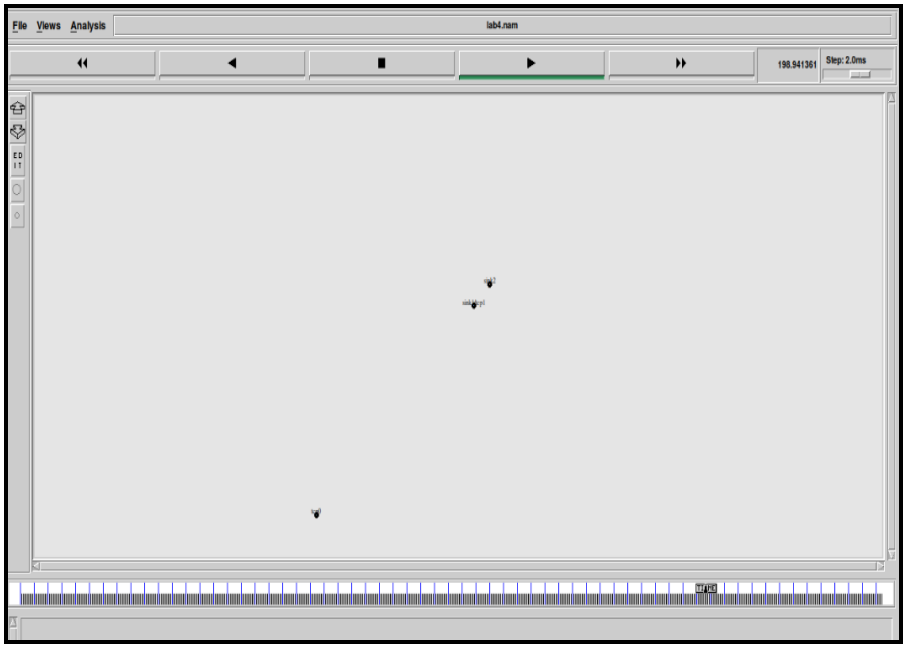
Step3: Run the simulation program



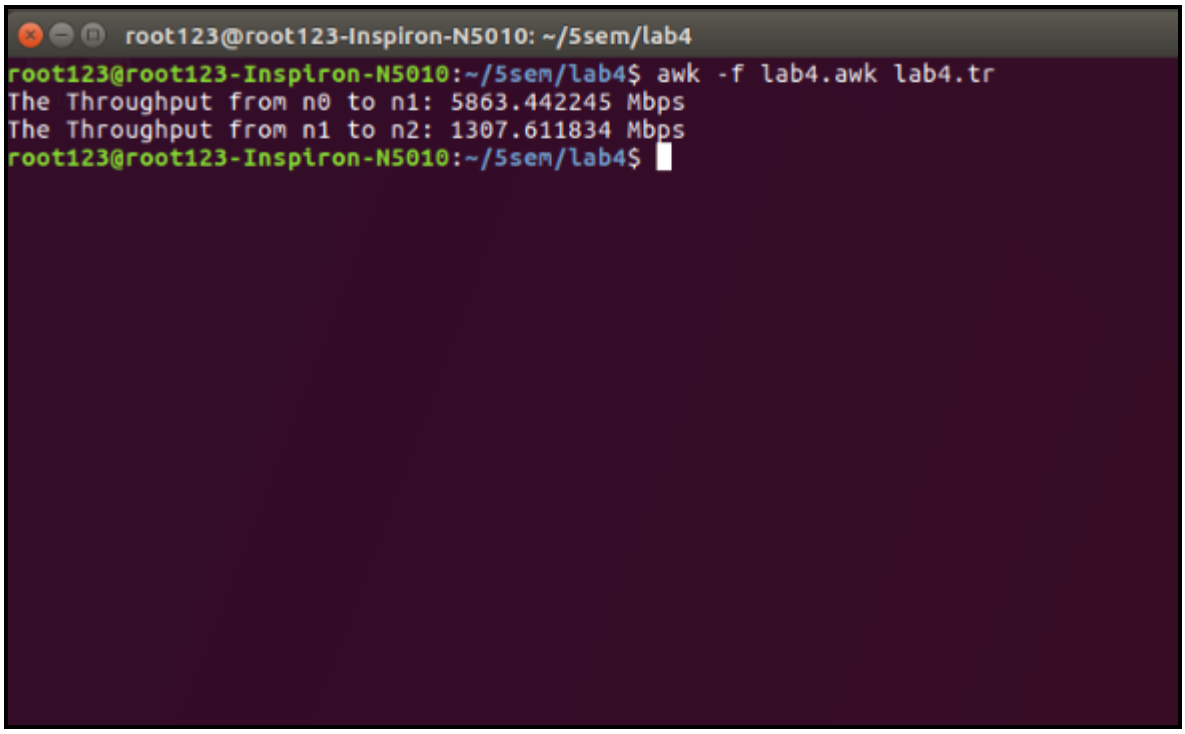
Step 4: Now press the play button in the simulation window and the simulation will begins.




File Views Analysis
lab4.nam
2.342000
Step: 2.0ms

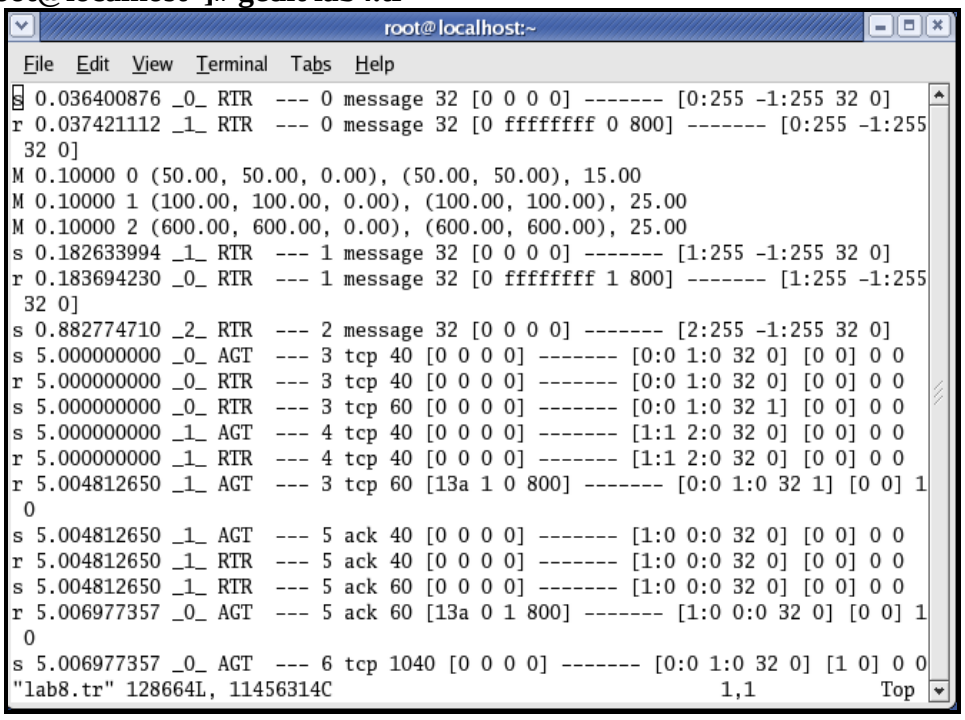

File Views Analysis
lab4.nam
198.941361
Step: 2.0ms

Step 5: After simulation is completed run **awk file** to see the output ,

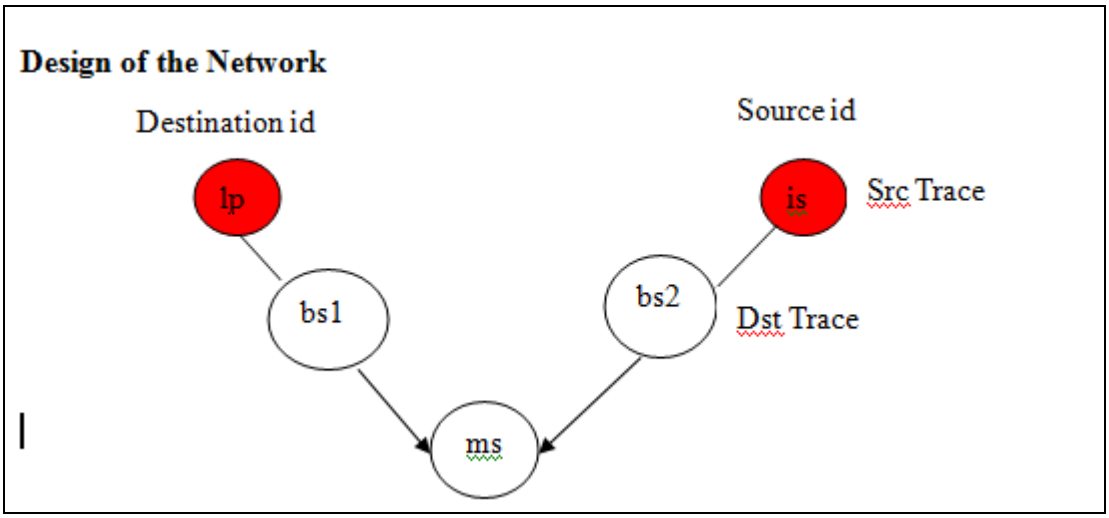


Step 6: To see the trace file contents open the file as ,

[root@localhost~]# gedit lab4.tr



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



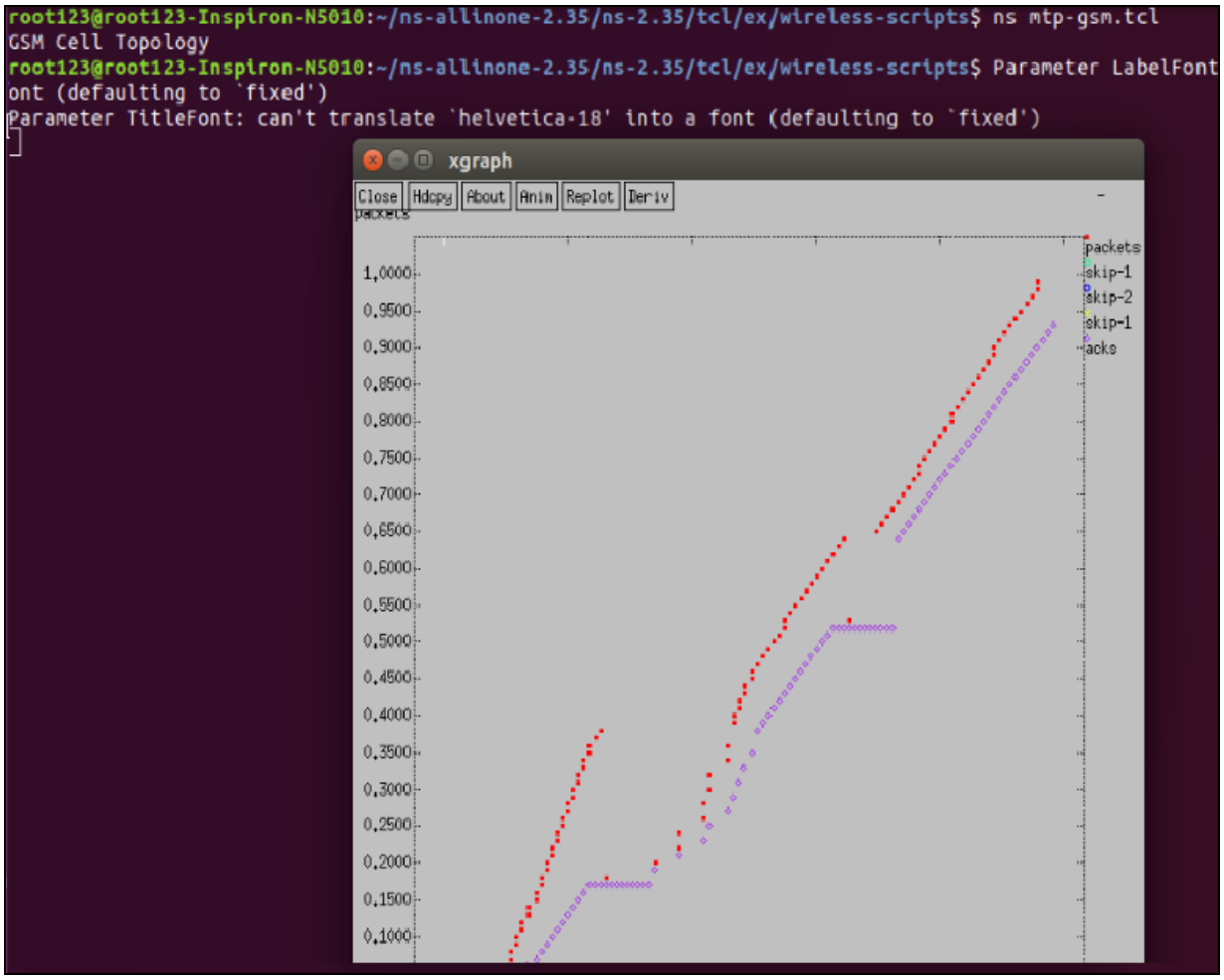
Step1: Open text editor, type the below program and save with extention .tcl (**gsm.tcl**)

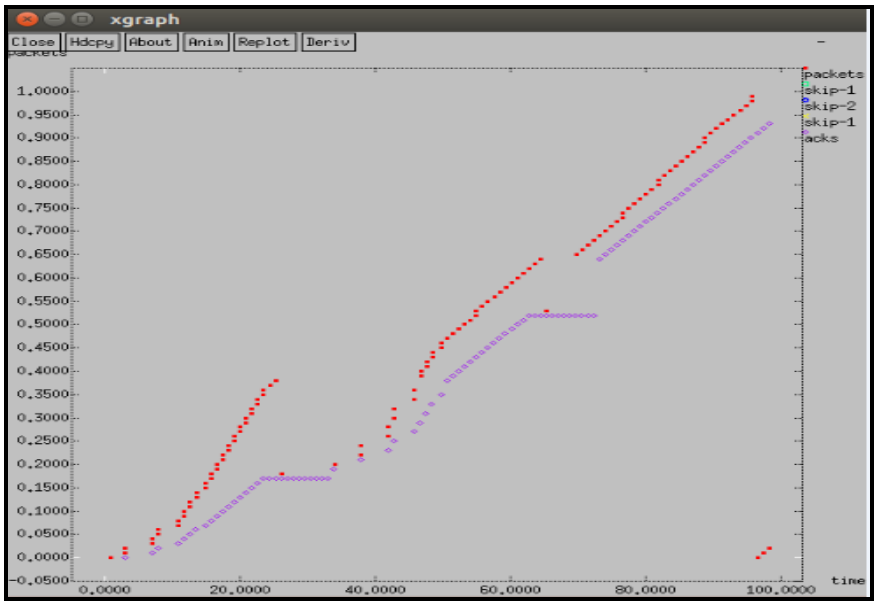
```
# Stop time
set stop 100

#type of link
set type gsm

# AQM parameters
set minth 30
set maxth 0

# 1 for Adaptive RED, 0 for plain RED
set adaptive 1

# number of long-lived TCP flows
set flows 0

# window for long-lived traffic
set window 30

# Plotting statistics
set opt(wrap) 100
set opt(srcTrace) is
set opt(dstTrace) bs2


#default downlink bandwidth in bps
```

```
set bwDL(gsm)  9600

#default downlink propagation delay in seconds
set propDL(gsm)  .500

set ns [new Simulator]

set tf [open out.tr w]
$ns trace-all $tf

set nodes(is) [$ns node]
set nodes(ms) [$ns node]
set nodes(bs1) [$ns node]
set nodes(bs2) [$ns node]
set nodes(lp) [$ns node]


proc cell_topo {} {
  global ns nodes
  $ns duplex-link $nodes(lp) $nodes(bs1) 3Mbps 10ms DropTail
  $ns duplex-link $nodes(bs1) $nodes(ms) 1 1 RED
  $ns duplex-link $nodes(ms) $nodes(bs2) 1 1 RED
  $ns duplex-link $nodes(bs2) $nodes(is) 3Mbps 50ms DropTail
  puts "GSM Cell Topology"
}

proc set_link_params {t} {
  global ns nodes bwDL propDL
  $ns bandwidth $nodes(bs1) $nodes(ms) $bwDL($t) duplex
  $ns bandwidth $nodes(bs2) $nodes(ms) $bwDL($t) duplex
  $ns delay $nodes(bs1) $nodes(ms) $propDL($t) duplex
  $ns delay $nodes(bs2) $nodes(ms) $propDL($t) duplex
  $ns queue-limit $nodes(bs1) $nodes(ms) 10
  $ns queue-limit $nodes(bs2) $nodes(ms) 10
}

# RED and TCP parameters
Queue/RED set adaptive_ $adaptive
Queue/RED set thresh_ $minth
Queue/RED set maxthresh_ $maxth
Agent/TCP set window_ $window

source web.tcl

#Create topology
switch $type {
 gsm -
```

```
 cdma {cell_topo}
 }
  set_link_params $type
  $ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]
  $ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]

# Set up forward TCP connection
if {$flows == 0} {
     set tcp1 [$ns create-connection TCP/Sack1 $nodes(is)
                                TCPSink/Sack1 $nodes(lp) 0]
     set ftp1 [[set tcp1] attach-app FTP]
     $ns at 0.8 "[set ftp1] start"
}



proc stop {} {
     global nodes opt tf
     set wrap $opt(wrap)
     set sid [$nodes($opt(srcTrace)) id]
     set did [$nodes($opt(dstTrace)) id]
     set a "out.tr"
     set GETRC "../../../bin/getrc"
     set RAW2XG "../../../bin/raw2xg"
     exec $GETRC -s $sid -d $did -f 0 out.tr | \
          $RAW2XG -s 0.01  -m $wrap -r > plot.xgr
     exec $GETRC -s $did -d $sid -f 0 out.tr | \
          $RAW2XG -a -s 0.01 -m $wrap  >> plot.xgr
     exec xgraph -x time -y packets plot.xgr &
     exit 0
}
$ns at $stop "stop"
$ns run
```

Step 2: Run the simulation program and see the output that display graph

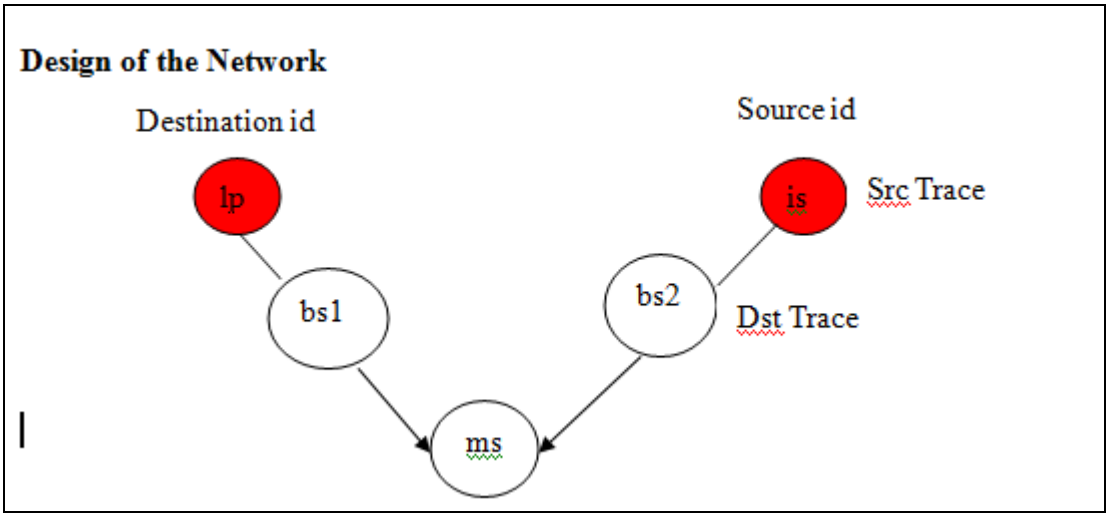




Step 3: To see the trace file contents open the file as ,

[root@localhost~]# gedit out.tr

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



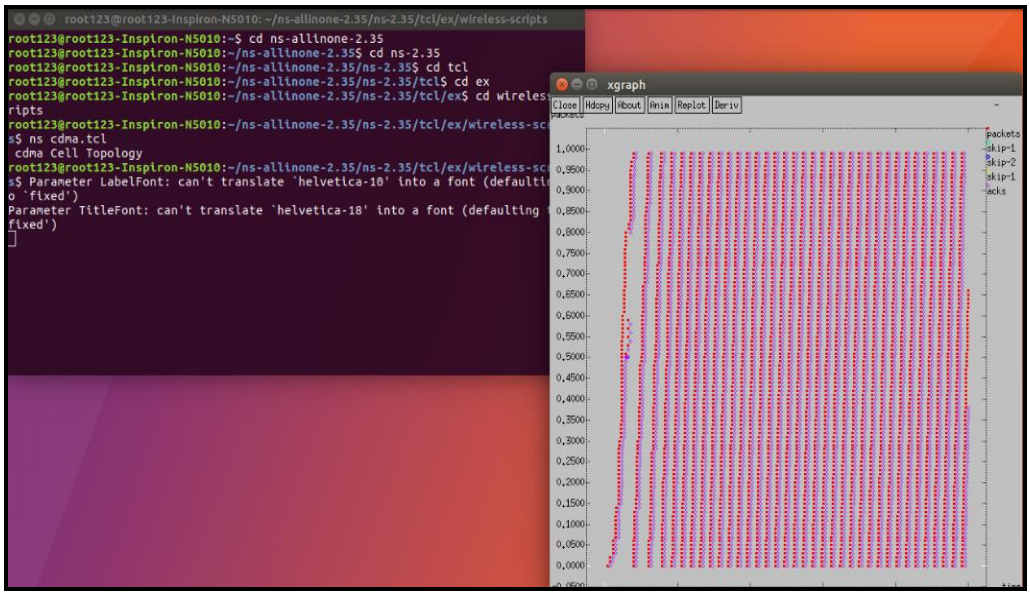
Step1: Open text editor, type the below program and save with extention .tcl (**gsm.tcl**)

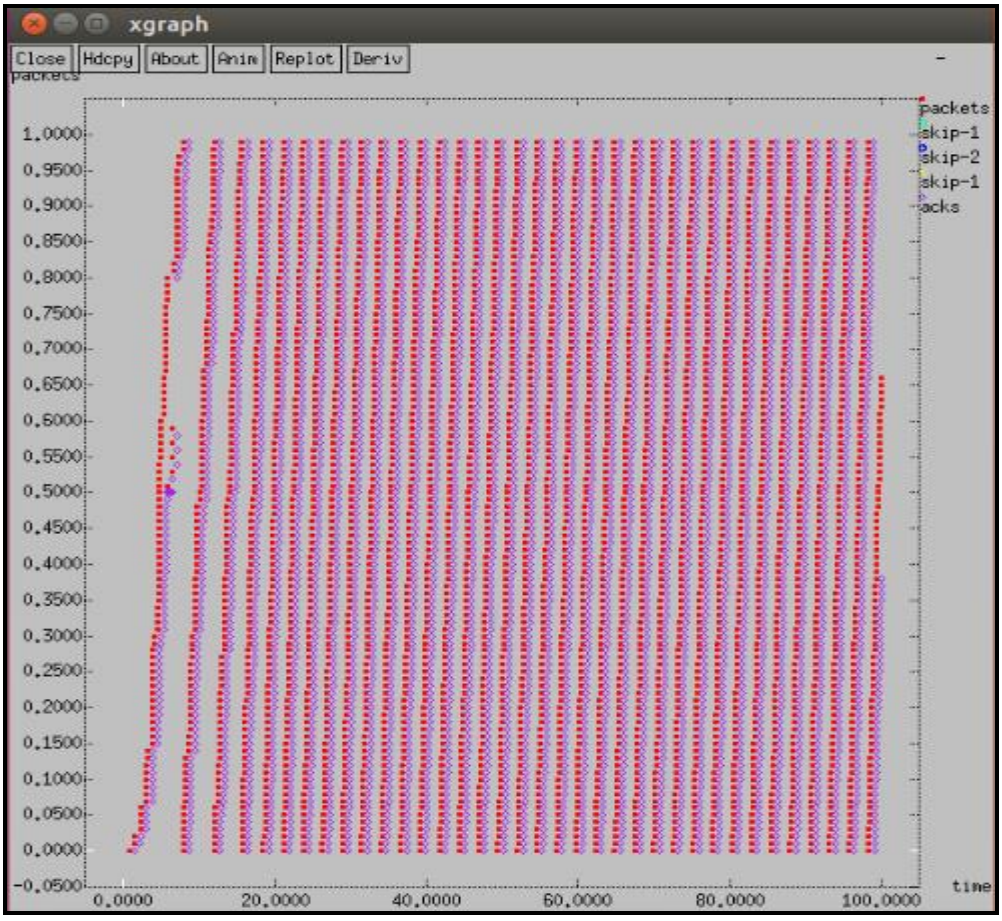
```
# Stop time
set stop 100

#type of link
set type cdma

# AQM parameters
set minth 30
set maxth 0

# 1 for Adaptive RED, 0 for plain RED
set adaptive 1

# number of long-lived TCP flows
set flows 0

# window for long-lived traffic
set window 30

# Plotting statistics
set opt(wrap) 100
set opt(srcTrace) is
set opt(dstTrace) bs2


#default downlink bandwidth in bps
```

```
set bwDL(cdma)  384000

#default downlink propagation delay in seconds
set propDL(cdma)  .150

set ns [new Simulator]

set tf [open out.tr w]
$ns trace-all $tf

set nodes(is) [$ns node]
set nodes(ms) [$ns node]
set nodes(bs1) [$ns node]
set nodes(bs2) [$ns node]
set nodes(lp) [$ns node]
proc cell_topo {} {
global ns nodes
$ns duplex-link $nodes(lp) $nodes(bs1) 3Mbps 10ms DropTail
$ns duplex-link $nodes(bs1) $nodes(ms) 1 1 RED
$ns duplex-link $nodes(ms) $nodes(bs2) 1 1 RED
$ns duplex-link $nodes(bs2) $nodes(is) 3Mbps 50ms DropTail
puts " cdma Cell Topology"
}

proc set_link_para {t} {
global ns nodes bwDL propDL
$ns bandwidth $nodes(bs1) $nodes(ms) $bwDL($t) duplex
$ns bandwidth $nodes(bs2) $nodes(ms) $bwDL($t) duplex
$ns delay $nodes(bs1) $nodes(ms) $propDL($t) duplex
$ns delay $nodes(bs2) $nodes(ms) $propDL($t) duplex
$ns queue-limit $nodes(bs1) $nodes(ms) 20
$ns queue-limit $nodes(bs2) $nodes(ms) 20
}

# RED and TCP parameters
Queue/RED set adaptive_ $adaptive
Queue/RED set thresh_ $minth
Queue/RED set maxthresh_ $maxth
Agent/TCP set window_ $window

source web.tcl

#Create topology
switch $type {
cdma {cell_topo}
}
```

```
set_link_para $type
$ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]
$ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]

# Set up forward TCP connection
if {$flows == 0} {
set tcp1 [$ns create-connection TCP/Sack1 $nodes(is)
TCPSink/Sack1 $nodes(lp) 0]
set ftp1 [[set tcp1] attach-app FTP]
$ns at 0.8 "[set ftp1] start"
}

proc stop {} {
global nodes opt tf
set wrap $opt(wrap)
set sid [$nodes($opt(srcTrace)) id]
set did [$nodes($opt(dstTrace)) id]
set a "out.tr"
set GETRC "../../../bin/getrc"
set RAW2XG "../../../bin/raw2xg"
exec $GETRC -s $sid -d $did -f 0 out.tr | \
     $RAW2XG -s 0.01 -m $wrap -r > plot.xgr
exec $GETRC -s $did -d $sid -f 0 out.tr | \
     $RAW2XG -a -s 0.01 -m $wrap >> plot.xgr
exec xgraph -x time -y packets plot.xgr &
exit 0
}
$ns at $stop "stop"
$ns run
```

Step 2: Run the simulation program and see the output that display graph





Step 3: To see the trace file contents open the file as ,

[root@localhost~]# gedit out.tr

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

Theory

CRC(Cyclic Redundancy Check) is an error detecting technique used in digital networks and storage devices to detect the accidental changes to raw data. It cannot be used for correcting errors. The sender and the receiver agree upon a fixed polynomial called generator polynomial. The CRC does error checking via polynomial division.

CRC-CCITT (Cyclic Redundancy Check – Committee Consultative International Telephone and Telegraph). The standard agreed generator polynomial is $x^{16}+x^{12}+x^5+x^0$

The generated polynomial $g(x) = x^{16}+x^{12}+x^5+x^0$

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1

→ 17 bits.

So the g(x) value is 10001000000100001

Algorithm:

1. Given a bit string (message to be sent), append 16 0^S to the end of it (the number of 0^s is the same as the degree of the generator polynomial) let this string + 0^S be called as modified string B
2. Divide B by agreed on polynomial g(x) and determine the remainder R(x). The 16-bit remainder received is called as checksum.
3. The message string is appended with checksum and sent to the receiver.
4. At the receiver side, the received message is divided by generator polynomial g(x).
5. If the remainder is 0, the receiver concludes that there is no error occurred otherwise, the receiver concludes an error occurred and requires a retransmission.

PROGRAM

```
import java.io.*;
class Crc1
{
public static void main(String args[]) throws IOException
{
BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
int[] data;
int[] div;
int[] divisor;
int[] rem;
int[] crc;
int data_bits, divisor_bits, tot_length;
System.out.println("Enter number of Message data bits:");
data_bits=Integer.parseInt(br.readLine());
data=new int[data_bits];
System.out.println("Enter Message data bits:");
for(int i=0;i<data_bits;i++)
data[i]=Integer.parseInt(br.readLine());
System.out.println("Enter number of bits in Generator
polynomial:");
divisor_bits=Integer.parseInt(br.readLine());
divisor=new int[divisor_bits];
System.out.println("Enter Generator polynomial bits:");
for(int i=0;i<divisor_bits;i++)
divisor[i]=Integer.parseInt(br.readLine());
tot_length=data_bits+divisor_bits-1;
div=new int[tot_length];
rem=new int[tot_length];
crc=new int[tot_length];
for(int i=0;i<data.length;i++)
div[i]=data[i];
System.out.print("Message(after appending 0's)are:");
for(int i=0;i<div.length;i++)
System.out.print(div[i]);
System.out.println();
for(int j=0;j<div.length;j++)
{
rem[j]=div[j];
}
rem=divide(div,divisor,rem);
for(int i=0;i<div.length;i++)
{
crc[i]=(div[i]^rem[i]);
}
```

```
System.out.println();
System.out.print("CRC code to be sent to receiver:");
for(int i=0;i<crc.length;i++)
System.out.print(crc[i]);
System.out.println();
System.out.println("Enter the received CRC code of
"+tot_length+" bits:");
for(int i=0;i<crc.length;i++)
crc[i]=Integer.parseInt(br.readLine());
for(int j=0;j<crc.length;j++)
{
rem[j]=crc[j];
}
rem=divide(crc,divisor,rem);
for(int i=0;i<rem.length;i++)
{
if(rem[i]!=0)
{
System.out.println("Error in received data");
break;
}
if(i==rem.length-1)
System.out.println("No error in received data");
}
}
static int[] divide(int div[],int divisor[],int rem[])
{
int cur=0;
while(true)
{
for(int i=0;i<divisor.length;i++)
rem[cur+i]=(rem[cur+i]^divisor[i]);
while(rem[cur]==0&&cur!=rem.length-1)
cur++;
if((rem.length-cur)<divisor.length)
break;
}
return rem;
}
}
```

Output

```
root123@root123-Inspiron-N5010:~/5sem$ javac Crc1.java
root123@root123-Inspiron-N5010:~/5sem$ java Crc1
Enter number of Message data bits:
5
Enter Message data bits:
1
0
1
1
0
Enter number of bits in Generator polynomial:
4
Enter Generator polynomial bits:
1
0
0
1
Message(after appending 0's)are:10110000

CRC code to be sent to receiver:10110100
Enter the received CRC code of 8 bits:
1
0
1
1
0
1
0
0
No error in received data
root123@root123-Inspiron-N5010:~/5sem$
```

```
root123@root123-Inspiron-N5010:~/5sem$ javac Crc1.java
root123@root123-Inspiron-N5010:~/5sem$ java Crc1
Enter number of Message data bits:
5
Enter Message data bits:
1
0
1
1
0
Enter number of bits in Generator polynomial:
4
Enter Generator polynomial bits:
1
0
0
1
Message(after appending 0's)are:10110000

CRC code to be sent to receiver:10110100
Enter the received CRC code of 8 bits:
1
1
1
1
0
1
0
0
Error in received data
root123@root123-Inspiron-N5010:~/5sem$
```

For the same Message try again the output with generator polynomial = 17 bits [10001000000100001]

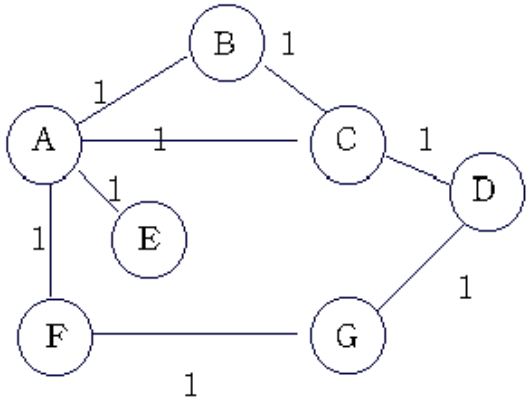
2.Write a program to find the shortest path between vertices using bellman-ford algorithm.

Theory

Routing algorithm is a part of network layer software which is responsible for deciding which output line an incoming packet should be transmitted on.

The distance vector routing algorithm is sometimes called by other names, including the distributed Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962). Two algorithms in particular, distance vector routing and link state routing are the most popular. In distance vector routing, each router maintains a routing table that contains two parts: the preferred out going line to use for that destination, and an estimate of the time or distance to that destination. The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar. The Routing tables are shared among the neighbors, and the tables at the router are updated, such that the router will know the shortest path to the destination.

Consider the graph shown below with the Cost Matrix



	A	B	C	D	E	F	G
A	**0**	**1**	**1**	∞	**1**	**1**	∞
B	**1**	**0**	**1**	∞	∞	∞	∞
C	**1**	**1**	**0**	**1**	∞	∞	∞
D	∞	∞	**1**	**0**	∞	∞	**1**
E	**1**	∞	∞	∞	**0**	∞	∞
F	**1**	∞	∞	∞	∞	**0**	**1**
G	∞	∞	∞	**1**	∞	**1**	**0**

In practice, each node's forwarding table consists of a set of triples of the form:(Destination, Cost, HopCount). For example, the below table shows the routing table maintained at node B for the graph with minimum one hop.

Destination	**Cost**	Hop Count
A	1	1
C	1	1
D	∞	0
E	∞	0
F	∞	0
G	∞	0

The Count to Infinity Problem.

Distance vector routing algorithm reacts rapidly to good news, but leisurely to bad news. Consider a router whose best route to destination *X* is large. If on the next exchange neighbor *A* suddenly reports a short delay to *X*, the router just switches over to using the line to *A* to send traffic to *X*. In one vector exchange, the good news is processed.

To see how fast good news propagates, consider the five node (linear) subnet of following figure, where the delay metric is the number of hops. Suppose A is down initially and all the other routers know this. In other words, they have all recorded the delay to A as infinity.

A	B	C	D	E		A	B	C	D	E	
∞	∞	∞	∞	∞	Initially	1	2	3	4	Initially
1	∞	∞	∞	∞	After 1 exchange	3	2	3	4	After 1 exchange
1	2	∞	∞	After 2 exchange	3	3	3	4	After 2 exchange
1	2	3	∞	After 3 exchange	5	3	5	4	After 3 exchange
1	2	3	4	After 4 exchange	5	6	5	6	After 4 exchange
					7	6	7	6	After 5 exchange
					7	8	7	8	After 6 exchange
						:			
					∞	∞	∞	∞	

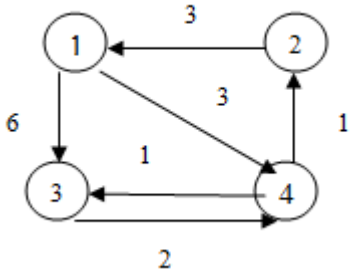
Many ad hoc solutions to the count to infinity problem have been proposed in the literature, each one more complicated and less useful than the one before it. The **split horizon** algorithm works the same way as distance vector routing, except that the distance to *X* is not reported on line that packets for *X* are sent on (actually, it is reported as infinity). In the initial state of right figure, for example, *C* tells *D* the truth about distance to *A* but *C* tells *B* that its distance to *A* is infinite. Similarly, *D* tells the truth to *E* but lies to *C.*

Program

```
import java.util.Scanner;
public class BellmanFords
{
private int distances[];
private int numberofvertices;
public static final int MAX_VALUE=999;

public BellmanFords(int  numberofvertices)
{
this.numberofvertices= numberofvertices;
distances=new int[ numberofvertices+1];
}
public void BellmanFordsEvaluation(int source,int
destination,int adjacencymatrix[][])
{
for(int node=1;node<=numberofvertices;node++)
{
distances[node]=MAX_VALUE;
}
distances[source]=0;
for(int node=1;node<=numberofvertices-1;node++)
{
for(int sourcenode=1;sourcenode<=numberofvertices;sourcenode++)
{
for(int
destinationnode=1;destinationnode<=numberofvertices;destination
node++)
{
if(adjacencymatrix[sourcenode][destinationnode]!=MAX_VALUE)
{
if(distances[destinationnode]>distances[sourcenode]+adjacencyma
trix[sourcenode][destinationnode])
distances[destinationnode]=distances[sourcenode]+adjacencymatri
x[sourcenode][destinationnode];
}
}
}
}
for(int vertex=1;vertex<=numberofvertices;vertex++)
{
if(vertex==destination)
System.out.println("The shortest distance from a source node "
+source+ "to destination node " +vertex+ "is : "
+distances[vertex]);
}
```

```
}
public static void main(String[]args)
{
int numberofvertices=0;
int source,destination;
Scanner scanner=new Scanner(System.in);
System.out.println("Enter the number of vertices in graph: ");
numberofvertices=scanner.nextInt();
int adjacencymatrix[][]=new
int[numberofvertices+1][numberofvertices+1];
System.out.println("Enter the adjacency matrix:");
for(int sourcenode=1;sourcenode<=numberofvertices;sourcenode++)
{
for(int
destinationnode=1;destinationnode<=numberofvertices;destination
node++)
{
adjacencymatrix[sourcenode][destinationnode]=scanner.nextInt();
if(sourcenode==destinationnode)
{
adjacencymatrix[sourcenode][destinationnode]=0;
continue;
}
if(adjacencymatrix[sourcenode][destinationnode]==0)
{
adjacencymatrix[sourcenode][destinationnode]=MAX_VALUE;
}
}
}
System.out.println("Enter the source vertex");
source=scanner.nextInt();
System.out.println("Enter the destination vertex");
destination=scanner.nextInt();
BellmanFords bellmanfords=new BellmanFords(numberofvertices);
bellmanfords.BellmanFordsEvaluation(source,destination,adjacenc
ymatrix);
scanner.close();
}
}
```

OUTPUT




```
root123@root123-Inspiron-N5010:~/5sem$ javac BellmanFords.java
root123@root123-Inspiron-N5010:~/5sem$ java BellmanFords
Enter the number of vertices in graph:
4
Enter the adjacency matrix:
0 999 6 3
3 0 999 999
999 999 0 2
999 1 1 0
Enter the source vertex
2
Enter the destination vertex
4
The shortest distance from a source node 2to destination node 4is : 6
root123@root123-Inspiron-N5010:~/5sem$
```

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.

Theory

A socket represents a single connection between exactly two pieces of software (so-called point-to-point connection). More than two pieces of software can communicate with client/server or distributed systems by using multiple sockets.

Socket-based software usually runs on two separate computers on the network, but sockets can also be used to communicate locally (interprocess) on a single computer. Sockets are bidirectional, meaning that either side of the connection is capable of both sending and receiving data. Sometimes the one application that initiates communication is termed the "client" and the other application the "server,"

Clients and servers that communicate via a reliable channel, such as a TCP socket, have a dedicated point-to-point channel between themselves, or at least the illusion of one. To communicate, they establish a connection, transmit the data, and then close the connection. All data sent over the channel is received in the same order in which it was sent. This is guaranteed by the channel.

Algorithm (Server Side)

1. Start.
2. Create a socket
3. Bind the socket to an address
4. Listen to the connection
5. accept connection
6. Receive filename and transfer contents of file with client.
7. Stop.

Algorithm (Client Side)

1. Start.
2. Create a socket
3. Connect the socket to the address of the server
4. Send the filename of required
5. Read the contents of the file sent by server
6. Stop.

Server Program

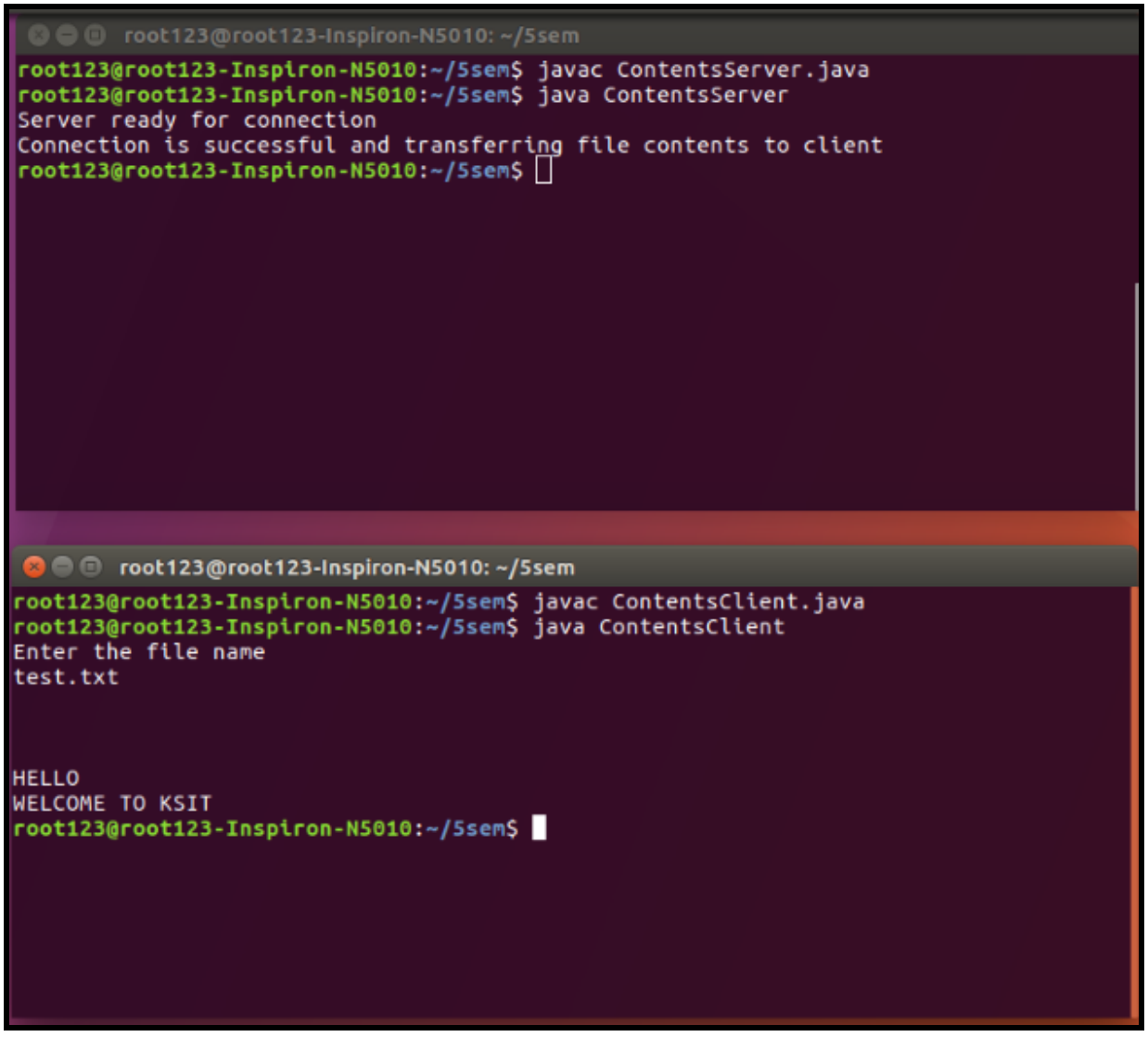
```
import java.io.*;
import java.net.*;
public class ContentsServer
{
public static void main(String[] args) throws IOException
{
ServerSocket sersock=new ServerSocket(4000);
System.out.println("Server ready for connection");
Socket sock=sersock.accept();
System.out.println("Connection is successful and transferring
file contents to client");
InputStream istream=sock.getInputStream();
BufferedReader fileRead=new BufferedReader(new
InputStreamReader(istream));
String fname=fileRead.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);
}
sock.close();
pwrite.close();
}
}
```

Client Program

```
import java.io.*;
import java.net.*;
public class ContentsClient
{
public static void main(String[] args) throws IOException
{
Socket sock=new Socket( "127.0.0.1",4000);
System.out.println("Enter the file name");
BufferedReader keyRead=new BufferedReader(new
InputStreamReader(System.in));
String fname=keyRead.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)
{
System.out.println(str);
}
pwrite.close();
}
}
```

OUTPUT



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

Theory

A datagram is an independent, self-contained message sent over the network whose arrival, arrival time, and content are not guaranteed. Applications that communicate via datagrams send and receive completely independent packets of information.

Clients and Servers do not have and do not need a dedicated point-to-point channel. The delivery of datagrams to their destinations is not guaranteed. Nor is the order of their arrival. The server continuously receives datagram packets over a datagram socket. Each datagram packet received by the server indicates a client request for a “MESSAGE”. When the server receives a datagram, it replies by sending a datagram packet "MESSAGE" back to the client. The client sends a single datagram packet to the server indicating that the client would like to receive a message of the moment. The client then waits for the server to send a datagram packet in response.

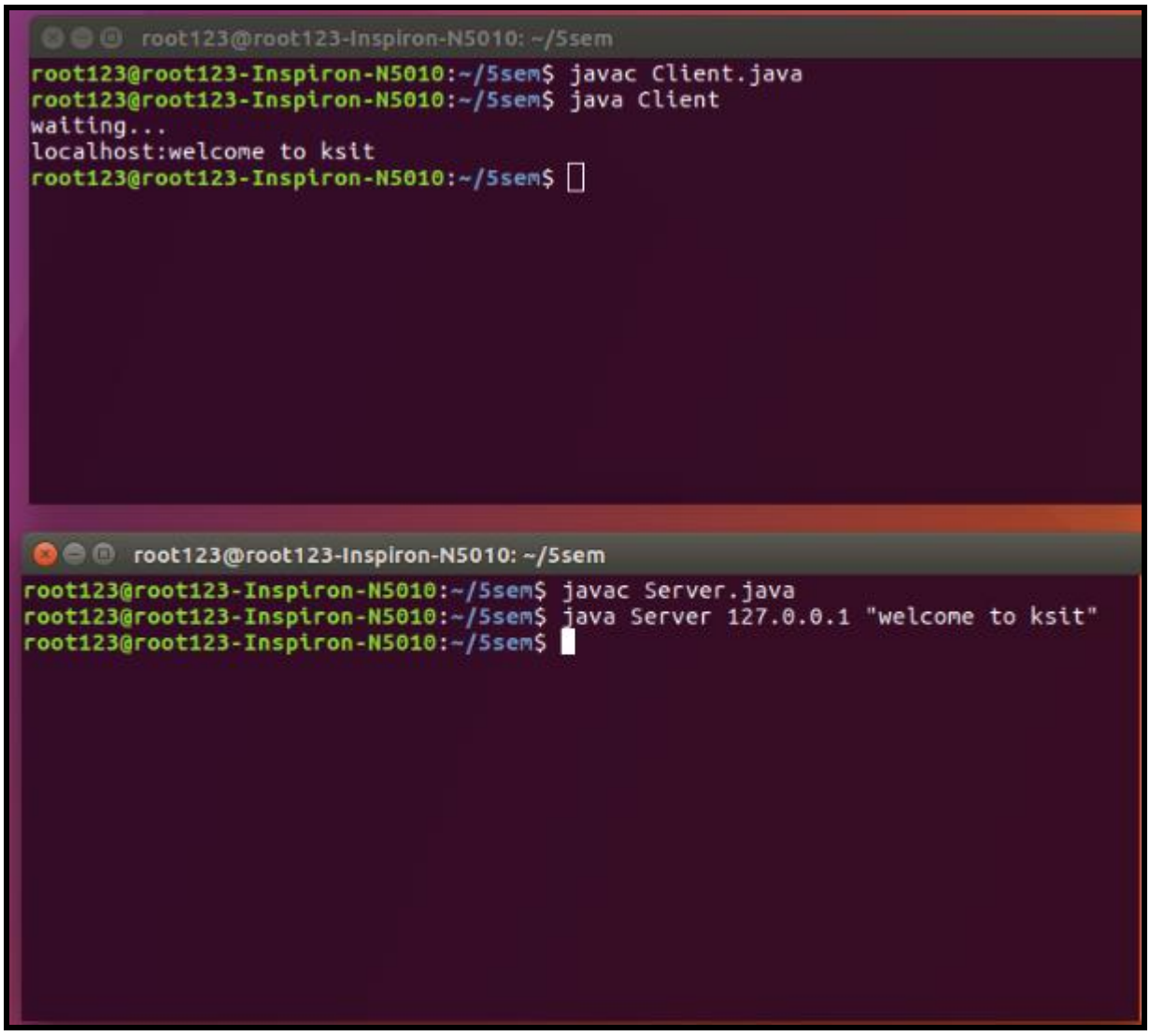
Server Program

```
import java.io.*;
import java.net.*;
public class Server
 {
public static void main(String[] args) throws IOException {
InetAddress addr=InetAddress.getByName(args[0]);
byte[] buf=args[1].getBytes();
DatagramPacket packet=new
DatagramPacket(buf,buf.length,addr,4444);
DatagramSocket socket=new DatagramSocket();
socket.send(packet);
}
}
```

Client Program

```
import java.io.*;
import java.net.*;
public class Client
 {
public static void main(String[] args) throws IOException {
DatagramSocket socket=new DatagramSocket(4444);
byte[] buf=new byte[256];
DatagramPacket packet=new DatagramPacket(buf,buf.length);
System.out.println("waiting...");
socket.receive(packet);
String s=new String(packet.getData(),0,packet.getLength());
System.out.println(packet.getAddress().getHostName() + ":" +s);
}
}
```

OUTPUT


root123@root123-Inspiron-N5010: ~/5sem
root123@root123-Inspiron-N5010:~/5sem$ javac Client.java
root123@root123-Inspiron-N5010:~/5sem$ java Client
waiting...
localhost:welcome to ksit
root123@root123-Inspiron-N5010:~/5sem$
root123@root123-Inspiron-N5010: ~/5sem
root123@root123-Inspiron-N5010:~/5sem$ javac Server.java
root123@root123-Inspiron-N5010:~/5sem$ java Server 127.0.0.1 "welcome to ksit"
root123@root123-Inspiron-N5010:~/5sem$

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

Theory

RSA(Rivest–Shamir–Adleman) algorithm is asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes that the Public Key is given to everyone and Private key is kept private. The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024 bit keys could be broken in the near future.

Algorithm

1. Generate two large random primes, P and Q, of approximately equal size.
2. Compute $N = P \times Q$
3. Compute $Z = (P-1) \times (Q-1)$.
4. Choose an integer E, $1 < E < Z$, such that GCD $(E, Z) = 1$
5. Compute the secret exponent D, $1 < D < Z$, such that $E \times D \equiv 1 \pmod{Z}$
6. The public key is (N, E) and the private key is (N, D).

An example of RSA encryption

1. Select primes $P=11$, $Q=3$.
2. $N = P \times Q = 11 \times 3 = 33$
   $Z = (P-1) \times (Q-1) = 10 \times 2 = 20$
3. Lets choose $E=3$
   Check GCD$(E, P-1)$ = GCD(3, 10) = 1 (i.e. 3 and 10 have no common factors except 1),
   and check GCD$(E, Q-1)$ = GCD(3, 2) = 1
   therefore GCD(E, Z) = GCD(3, 20) = 1
4. Compute D such that $E \times D \equiv 1 \pmod{Z}$
   compute $D = E^{-1} \bmod Z = 3^{-1} \bmod 20$
   find a value for D such that Z divides $((E \times D)-1)$
   find D such that 20 divides 3D-1.
   Simple testing (D = 1, 2, ...) gives D = 7
   Check: $(E \times D)-1 = 3.7 - 1 = 20$, which is divisible by Z.
5. Public key = (N, E) = (33, 3)
   Private key = (N, D) = (33, 7).

Now say we want to encrypt the message m = 7,

Cipher code = $M^E \bmod N$

$= 7^3 \bmod 33$

$= 343 \bmod 33$

$= 13$.

Hence the ciphertext c = 13.

To check decryption we compute Message' = $C^D \bmod N$

$= 13^7 \bmod 33$

$= 7$.

PROGRAM

```
import java.math.BigInteger;
import java.util.*;
class rsa
{
public static void main(String args[])
{
Scanner ip=new Scanner(System.in);
int p,q,n,e=1,j;
int d=1,i1;
int t2,t1;
int pt[]=new int[10];
int ct[]=new int[10];
int rt[]=new int[10];
int temp[]=new int[10];
String i=new String();
System.out.println("Enter the two prime numbers:");
p=ip.nextInt();
q=ip.nextInt();
System.out.println("Enter the message to be sent");
i=ip.next();
i1=i.length();
n=p*q;
t1=p-1;
t2=q-1;
System.out.println("\n--------------------");
System.out.println("Sender side");
while((t1*t2)%e==0)
{
e++;
}
System.out.println("Public key(e)="+e);
System.out.println("----------------------");
for(j=0;j<i1;j++)
{
pt[j]=(i.charAt(j))-96;
//System.out.println("Plain text="+pt[j]);
ct[j]=((int)Math.pow(pt[j],e))%n;
System.out.println("Cipher Text="+ct[j]);
}
System.out.println("Transmitted Message:");

for(j=0;j<i1;j++)
{
temp[j]=ct[j]+96;
System.out.print((char)temp[j]);
```

```
}
System.out.println("\n\n---------------------");
System.out.println("Receiver side:");
while((d*e)%(t1*t2)!=1)
{
d++;
}
System.out.println("Private key(d)="+d);
System.out.println("------------------");
for(j=0;j<i1;j++)
{
BigInteger very_big_no=BigInteger.valueOf(ct[j]);
very_big_no=very_big_no.pow(d);
very_big_no=very_big_no.mod(BigInteger.valueOf(n));
rt[j]=very_big_no.intValue();
System.out.println("Plain Text="+rt[j]);
}
System.out.println("\n------------------------");
System.out.println("Decrypted Message:");
for(j=0;j<i1;j++)
{
rt[j]=rt[j]+96;
System.out.println((char)rt[j]);
}
System.out.println("\n-------------------------");
ip.close();
}
}
```

OUTPUT

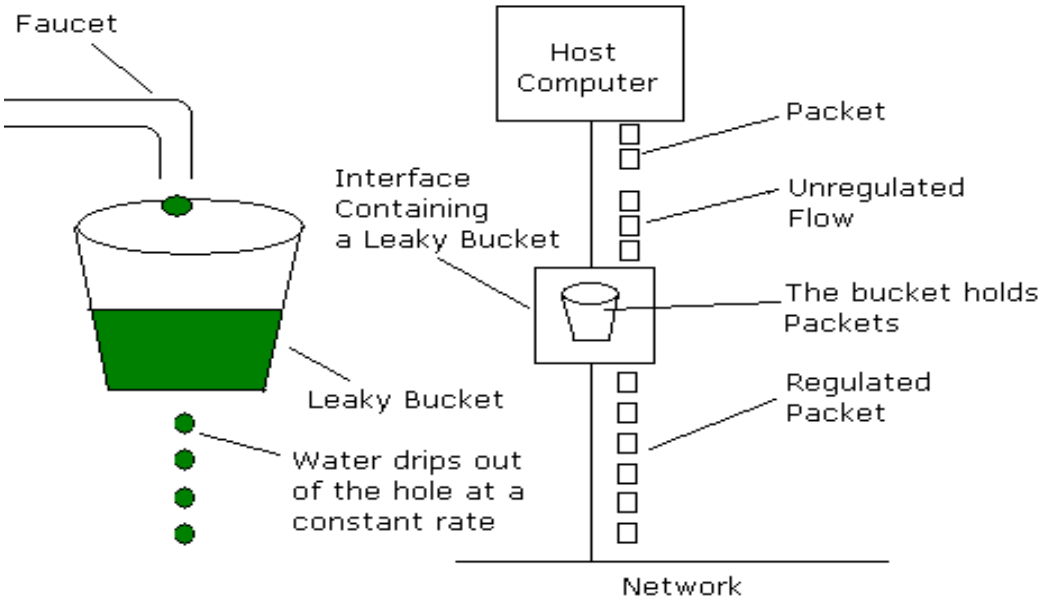
```
root123@root123-Inspiron-N5010:~/5sem$ javac rsa.java
root123@root123-Inspiron-N5010:~/5sem$ java rsa
Enter the two prime numbers:
3
11
Enter the message to be sent
welcome

--------------------
Sender side
Public key(e)=3
----------------------
Cipher Text=23
Cipher Text=26
Cipher Text=12
Cipher Text=27
Cipher Text=9
Cipher Text=19
Cipher Text=26
Transmitted Message:
wzl{isz

--------------------
Receiver side:
Private key(d)=7
-----------------
Plain Text=23
Plain Text=5
Plain Text=12
Plain Text=3
Plain Text=15
Plain Text=13
Plain Text=5

-----------------------
Decrypted Message:
w
e
l
c
o
m
e
```

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

Theory

In Leaky bucket algorithm, Each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. In other words, if one or more process are already queued, the new packet is unceremoniously discarded. This arrangement can be built into the hardware interface or simulated by the host operating system. In fact it is nothing other than a single server queuing system with constant service time.

The host is allowed to put one packet per clock tick onto the network. This mechanism turns an uneven flow of packet from the user process inside the host into an even flow of packet onto the network, smoothing out bursts and greatly reducing the chances of congestion.



PROGRAM

```
import java.lang.*;
import java.util.Random;
import java.io.*;
import java.util.Scanner;
class leaky_bucket
{
public static void main(String args[])
{
int drop=0,mini,nsec,p_remain=0;
int o_rate,b_size,i,packet[];
packet = new int[100];
Scanner in = new Scanner(System.in);
System.out.println("Enter Bucket size:");
b_size = in.nextInt();
System.out.println("Enter the Output rate:");
o_rate = in.nextInt();
System.out.println("Enter the number of seconds you want to
simulate:");
nsec = in.nextInt();
Random rand = new Random();
for(i=0;i<nsec;i++)
packet[i]=((rand.nextInt(9)+1)*10);
System.out.println("Seconds|packets  Received|packets
Sent|packets Left|packets Dropped");
System.out.println("-----------------------------------------
--------------------------------");
for(i=0;i<nsec;i++)
     {
          p_remain+=packet[i];
          if(p_remain>b_size)
          {
               drop=p_remain-b_size;
               p_remain=b_size;
               System.out.print(i+1+"        ");
               System.out.print(packet[i]+"       ");
               mini=Math.min(p_remain,o_rate);
               System.out.print(mini+"       ");
               p_remain=p_remain-mini;
               System.out.print(p_remain+"        ");
               System.out.print(drop+"       ");
               System.out.println();
               drop=0;
          }
     }
```

```
while(p_remain!=0)
     {
          if(p_remain>b_size)
          {
               drop=p_remain-b_size;
               p_remain=b_size;
          }
          mini=Math.min(p_remain,o_rate);
          System.out.print("       "+p_remain+"        "+mini);
          p_remain=p_remain-mini;
          System.out.println(p_remain+"           "+drop);
          drop=0;
     }
}
}
```

OUTPUT

```
root123@root123-Inspiron-N5010:~$ javac leaky_bucket.java
root123@root123-Inspiron-N5010:~$ java leaky_bucket
Enter Bucket size:
20
Enter the Output rate:
4
Enter the number of seconds you want to simulate:
5
Seconds|packets  Received|packets  Sent|packets Left|packets Dropped
--------------------------------------------------------------------------
1               40              4               16              20
2               30              4               16              26
3               50              4               16              46
4               30              4               16              26
5               10              4               16              6
                16              412             0
                12              48              0
                8               44              0
                4               40              0
root123@root123-Inspiron-N5010:~$
```

Viva Questions

1. Explain the functions of OSI layers ?
2. Differentiate between TCP/IP Layers and OSI Layers
3. Why header is required?
4. What is the use of adding header and trailer to frames?
5. What is encapsulation?
6. Why fragmentation requires?
7. What is MTU?
8. Which layer imposes MTU?
9. Differentiate between flow control and congestion control.
10. Differentiate between Point-to-Point Connection and End-to-End connections.
11. What are protocols running in different layers?
12. What is Protocol Stack?
13. Differentiate between TCP and UDP.
14. Differentiate between Connectionless and connection oriented connection.
15. Why frame sorting is required?
16. What is meant by subnet?
17. What is meant by Gateway?
18. What is an IP address?
19. What is MAC address?
20. Why IP address is required when we have MAC address?
21. What is meant by port?
22. What are ephemerical port number and well known port numbers?
23. What is a socket?
24. What are the parameters of socket()?
25. Describe bind(), listen(), accept(),connect(), send() and recv().
26. What are system calls? Mention few of them.
27. What is IPC? Name three techniques.
28. Explain mkfifo(), open(), close() with parameters.
29. What is meant by file descriptor?

30. What is meant by traffic shaping?
31. How do you classify congestion control algorithms?
32. Differentiate between Leaky bucket and Token bucket.
33. How do you implement Leaky bucket?
34. How do you generate bursty traffic?
35. What is the polynomial used in CRC-CCITT?
36. What are the other error detection algorithms?
37. What is difference between CRC and Hamming code?
38. Why Hamming code is called 7,4 code?
39. What is odd parity and even parity?
40. What is meant by syndrome?
41. What is generator matrix?
42. What are Routing algorithms?
43. How do you classify routing algorithms? Give examples for each.
44. What are drawbacks in distance vector algorithm?
45. How routers update distances to each of its neighbor?
46. How do you overcome count to infinity problem?
47. What is cryptography?
48. How do you classify cryptographic algorithms?
49. What is public key?
50. What is private key?
51. What are key, ciphertext and plaintext?
52. What is simulation?
53. What are advantages of simulation?
54. Differentiate between Simulation and Emulation.
55. What is meant by router?
56. What is meant by bridge?
57. What is meant by switch?
58. What is meant by hub?
59. Differentiate between route, bridge, switch and hub.
60. What is ping and telnet?

61. What is FTP?
62. What is BER?
63. What is meant by congestion window?
64. What is BSS?
65. What is incoming throughput and outgoing throughput?
66. What is collision?
67. How do you generate multiple traffics across different sender-receiver pairs?
68. How do you setup Ethernet LAN?
69. What is meant by mobile host?
70. Name few other Network simulators
71. Differentiate between logical and physical address.
72. Which address gets affected if a system moves from one place to another place?
73. What is ICMP? What are uses of ICMP? Name few.
74. Which layer implements security for data?
75. What is flow control and error control?
76. What are the protocols used for error control in data link layer.
77. What is the difference between?
    a. Transferring frames from source to destination.
    b. Transferring packets from source to destination.
    c. Transferring segments from source to destination.
78. Why the message is divided into segments?
79. How do we define End to End communication messages using 5 tuples?
80. What is Routing? What are the different routing protocols used in N/W layer?
81. What is meant by gateway?
82. What are the different N/W topologies?
83. What is the topology used in your lab?
84. What are the three address used to identify a system? Name them? What is their importance?
85. What is the need of MAC protocols? Give examples.
86. Difference between CSMA/CA and CSMA/CD.
87. Why CSMA/CD is not used in WLANS?

88. How CSMA/CA is adopted in WLANS?

Viva Questions with answers

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

2) What are the layers of the OSI reference model?
There are 7 OSI layers: Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer and Application Layer.

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

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

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

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

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

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

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

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

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

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

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

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

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

15) What is the job of the Network Layer under the OSI reference model?
The Network layer is responsible for data routing, packet switching and control of network congestion. Routers operate under this layer.

16) How does a network topology affect your decision in setting up a network?
Network topology dictates what media you must use to interconnect devices. It also serves as basis on what materials, connector and terminations that is applicable for the setup.

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

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

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

20) What is WAN?

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

21) What is the importance of the OSI Physical Layer?
The physical layer does the conversion from data bits to electrical signal, and vice versa. This is where network devices and cable types are considered and setup.

22) How many layers are there under TCP/IP?
There are four layers: the Network Layer, Internet Layer, Transport Layer and Application Layer.

23) What are proxy servers and how do they protect computer networks?
Proxy servers primarily prevent external users who identifying the IP addresses of an internal network. Without knowledge of the correct IP address, even the physical location of the network cannot be identified. Proxy servers can make a network virtually invisible to external users.

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

25) What is the importance of implementing a Fault Tolerance System? Are there limitations?
A fault tolerance system ensures continuous data availability. This is done by eliminating a single point of failure. However, this type of system would not be able to protect data in some cases, such as in accidental deletions.

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

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

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 fault-tolerant network topology.

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.