Sri Shridevi Charitable Trust (R.)
SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY
Sira Road, Tumkur - 572 106, Karnataka, India.
Phone: 0816 - 2212629 | Principal: 0816 - 2212627, 9686114899 | Telefax: 0816 - 2212628
Email: info@shridevienginеering.org, principal@shridevienginеering.org | Website: www.shridevienginеering.org
(Approved by AICTE, New Delhi, Recognised by Govt. of Karnataka and Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

COMPUTER NETWORKS

(BCS502)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

PROGRAM OUTCOMES (POS)

- Engineering knowledge: Apply knowledge of mathematics, science, and engineering to solve engineering problems.
- Problem analysis: Identify, formulate, and analyse complex engineering problems to reach substantiated conclusions.
- Design/development of solutions: Design solutions for engineering problems and systems to meet desired specifications.
- Conduct investigations of complex problems: Identify, formulate, analyse, and synthesize information to solve complex engineering problems.
- Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
- The engineer and society: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal, and cultural issues.
- Environment and sustainability: Understand the impact of engineering solutions on society and the environment, and demonstrate knowledge of sustainable development.

- Ethics: Apply ethical principles and commit to professional ethics and responsibilities.
- Individual and team work: Function effectively as an individual and as a member or leader in diverse teams.
- Communication: Communicate effectively on complex engineering activities with the engineering community and society.
- Project management and finance: Use modern engineering tools, techniques, skills, and management principles to work in teams and manage projects in multidisciplinary environments.
- Life-long learning: Recognize the need for and ability to engage in lifelong learning to address contemporary issues.

Vision

Envisioning SHRIDEVI INSTITUE OF ENGINEERING & TECHNOLOGY as a world class organization meeting global standards where students acquire the knowledge and skills as per the NEP 2020 to be successful as they continue their education at the Institute and enter the competitive workforce.

Mission

- To pursue excellence in a academic knowledge, skills, and behavior for each student resulting in measured improvement against local, national and world-class standards.
- To promote the development of the institute's teaching and learning activities through the stimulation of excellence in teaching and the enhancement of student's independent learning skills.
- To create a conductive environment for optimal Student Recruitment, Higher student Retention, Maximum campus growth and excellent Human capital development.

Quality Policy

Shridevi Institute of Engineering and Technology strives to excel in imparting quality education included with ethical, moral and spiritual values to engineering and management students to make them globally competitive.

Quality Objectives

- To strive hard for academic excellence.
- To achieve maximum placements and to promote entrepreneurship development through proper training and awareness programs.
- To enhance the industry institute interaction.
- To promote and to encourage R and D activities.
- To Synergise Spiritual and Moral values

CONTENT LIST

Introduction

Introduction to NS-2

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

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

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

Features of NS-2:

- It is a discrete event simulator for networking research.
- It provides substantial support to simulate protocols like TCP, FTP, UDP & DSR.
- It simulates wired and wireless network.
- It is primarily UNIX based.
- Uses TCL as its scripting language.
- Otcl: Object oriented support Tcl
- TclCL: Tcl with Classes and OTcl linkage
- Discrete event scheduler

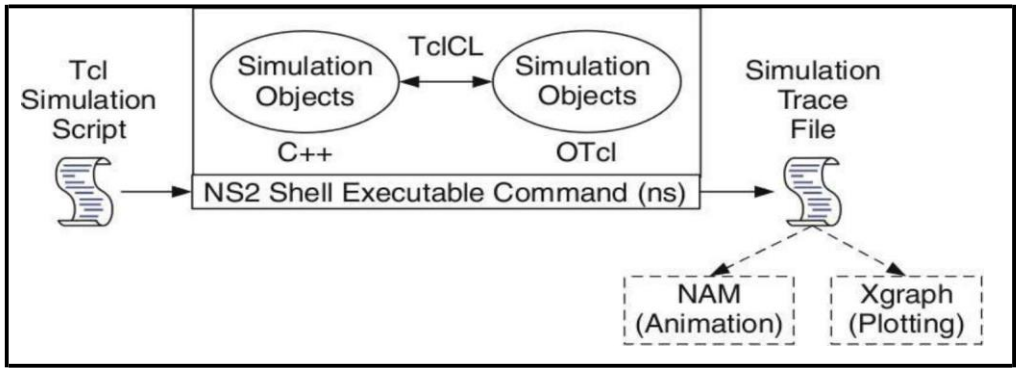


Fig. 1 Basic Architecture of Network Simulator

Why two languages? (TCL and C++)

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

Tcl scripting

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

Structure of NS-2 Program:

- Creating a Simulator Object
- Setting up files for trace & NAM
- Tracing files using their commands
- Closing trace file and starting NAM
- Creating LINK & NODE topology &Orientation of links

Working of NS-2

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

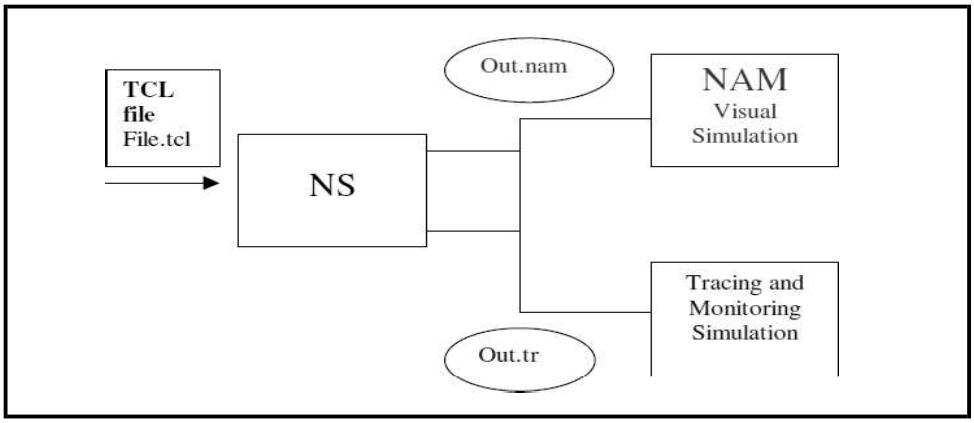


Fig 2. Working of Network Simulator 2

Trace file and NamTrace file:

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

Network Animator Window

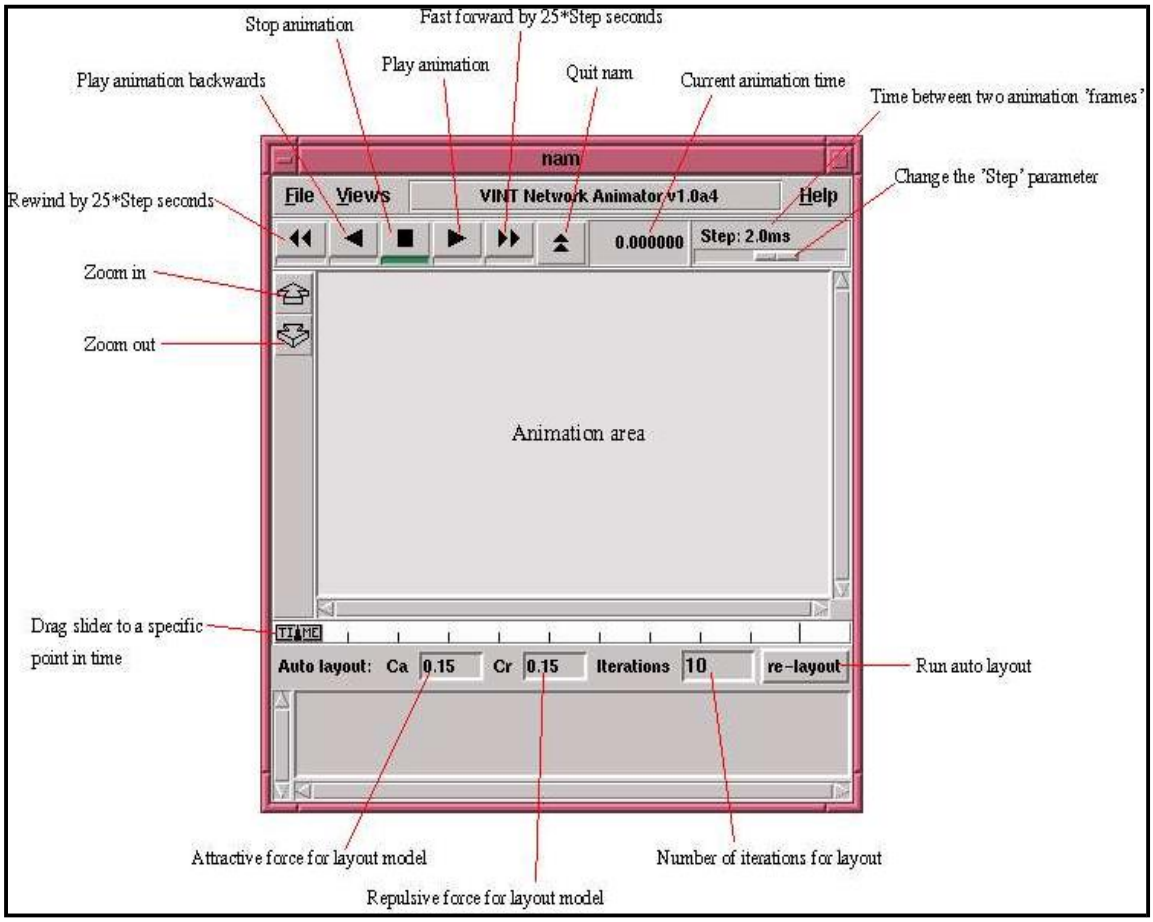


Fig. 3 Details of NAM Window

Advantages and Disadvantages of NS2:

Advantages

1. Open Source
2. Complex scenarios can be easily tested.
3. Results can be quickly obtained – more ideas can be tested in a smaller time frame.
4. Supported protocols
5. Supported platforms
6. Modularity

Disadvantages

1. Limitation in designing large scale systems
2. May be slow compared to real time network and computationally expensive
3. Does not reflect reality in large and complex networks.
4. Statistical uncertainty in results

Steps to create a scenario file:

Step1: Declare Simulator and setting output file

Step2: Setting Node and Link

Step3: Setting Agent

Step4: Setting Application

Step5: Setting Simulation time and schedules

Step6: Declare finish.

Step 1: Declare Simulator and setting

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

Step 2: Setting Node and Link

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

Step 3: Setting Agent

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

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

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

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

Step 4: Setting Application

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

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

Step 5: Setting time schedule for simulation

Time schedule of a simulation is set as below:

$ns at 1.0 "$cbr start" $ns at 3.5 "$cbr stop"	cbr transmits data from 1.0[sec] to 3.5[sec]
$ns at 1.5 "$ftp start" $ns at 3.0 "$ftp stop"	ftp transmits data from 1.5[sec] to 3.0[sec].

Step 6: Declare finish

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

$ns at 4.0 "finish"	The finish function is used to output data file at the end of simulation.

proc finish {} {
global ns file namfile tcpfile
$ns flush-trace
close $file
close $namfile
close $tcpfile
exit 0
}	

Execute Simulation and start Nam

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

ns sample.tcl
nam out.nam

View trace file (out.tr)

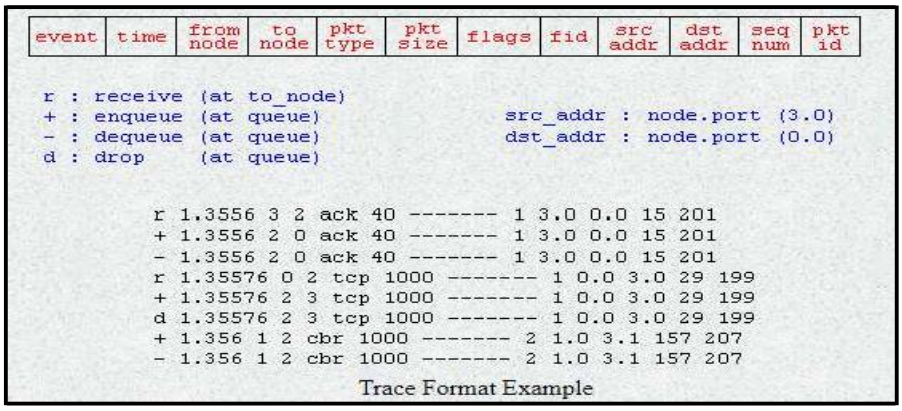


Figure 4 . Details of Trace Window

1. The first field is the event type. It is given by one of four possible symbols r, +, -, d which correspond respectively to receive (at the output of the link), enqueued, dequeued 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.

AWK file:

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

- Programs in awk are different from programs in most other languages, because awk programs are **data-driven;** that is, we describe the data to work with, and then what to do when we find it. Most other languages are **procedural.** When working with procedural languages, it is usually much harder to clearly describe the data of our program will process.

- For this reason, awk programs are often refreshingly easy to both write and read. When we run awk, we can specify an awk **program** that tells awk what to do. The program consists of a series of **rules**. (It may also contain **function definitions**, an advanced feature which we will ignore for now.

- Each rule specifies one pattern to search for, and one action to perform when that pattern is found). Syntactically, a rule consists of a pattern followed by an action. The action is enclosed in curly braces to separate it from the pattern. Rules are usually separated by newlines. Therefore, an awk program looks like this:

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

- Since we are dealing with column oriented data, AWK is probably the easiest tool we can use to format our data. AWK is a simple scripting language that scans through a file line by line. It allows to access any column in the current line by using special variables $1,$2, $3, etc. for the first, second and third columns. The definition of each column of the trace file is shown above, so we can use the AWK script to check the value of each column and collect the data we need.

- The BEGIN and END sections are only executed once (before and after the file has been processed). The middle section is executed for each line of the file. The AWK script keeps three variables to store the throughput in Mb/s for flow 1, flow 2, and the total. For each line of the trace file, it checks to see if a TCP packet ($5 == "tcp") is received ($1 == "r") at node 3 ($4 == "3"), which is our destination node. If so, it increments the count for the appropriate flow, using the size of the particular packet (in bytes) from column 6. After each second, it prints the total while converting bytes to Mb.

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

<u>To run awk script</u>

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

<u>XGRAPH:</u>

- Plotting purposes.
- Comes together with NS2 installation package.
- Running Xgraph

Xgraph <inputfile1>...<inputfile$_n$> -bg <color> -t <graph_title> -x <xtitle> -y <ytitle>

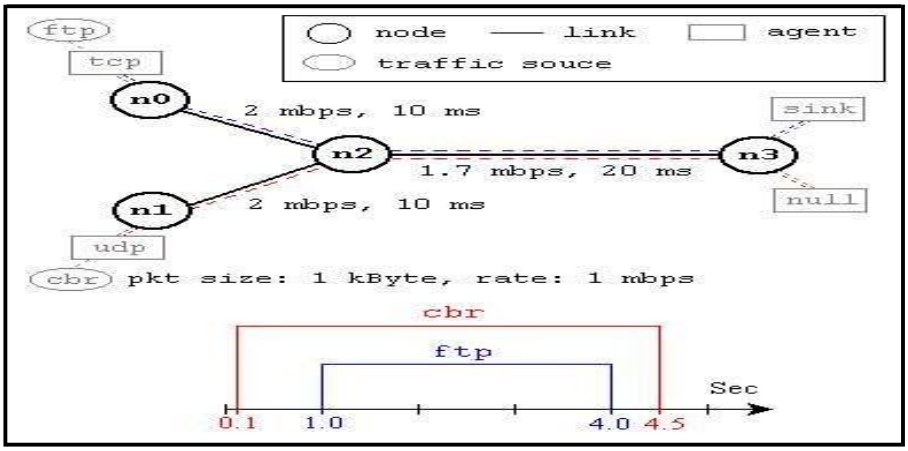


Fig 5. A Simple Network Topology and Simulation Scenario

This network consists of 4 nodes (n0, n1, n2, n3) as shown in Fig. 5. The duplex links between n0 and n2, and n1 and n2 have 2 Mbps of bandwidth and 10 ms of delay. The duplex link between n2 and n3 has 1.7 Mbps of bandwidth and 20 ms of delay. Each node uses a DropTail queue, of which the maximum size is 10. A "tcp" agent is attached to n0, and a connection is established to a tcp "sink" agent attached to n3. As default, the maximum size of a packet that a "tcp" agent can generate is 1KByte.

A tcp "sink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets. A "udp" agent that is attached to n1 is connected to a "null" agent attached to n3. A "null" agent just frees the packets received. A "ftp" and a "cbr" traffic generator are attached to "tcp" and "udp" agents respectively, and the "cbr" is configured to generate 1KByte packets at the rate of 1 Mbps.

The "cbr" is set to start at 0.1 sec and stop at 4.5 sec, and "ftp" is set to start at 1.0 sec and stop at 4.0 sec.

An Example Simulation Script

```
#Create a simulator object
set ns [new Simulator]
#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf
#Define a 'finish' procedure
proc finish {} {
 global ns nf
```

```
$ns flush-trace
#Close the NAM trace file
close $nf
#Execute NAM on the trace file
exec nam out.nam & exit 0
}
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail
#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10
#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5
#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1
$udp set null [new
Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2
#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false
#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
```

```
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"
$ns at 5.0 "finish"
#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"
#Run the simulation
$ns run
```

Program No. 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.

Theory:

- Create a simulator object.
- We open a file for writing that is going to be used for the trace data.
- We now attach the agent to the nodes.
- Now we attach the application to run on top of these nodes
- We now connect the agent and the application for its working
- Set the simulation time
- The next step is to add a 'finish' procedure that closes the trace file and starts nam.

```
set ns [ new Simulator ]
set tf [ open lab1.tr w ]
$ns trace-all $tf
set nf [ open lab1.nam w ]
$ns namtrace-all $nf
# The below code is used to create the nodes.
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
#This is used to give color to the packets.
$ns color 1 "red"
$ns color 2 "blue"
$n0 label "Source/udp0"
$n1 label "Source/udp1"
$n2 label "Router"
$n3 label "Destination/Null"
#Vary the below Bandwidth and see the number of packets dropped.
$ns duplex-link $n0 $n2 10Mb 300ms DropTail
$ns duplex-link $n1 $n2 10Mb 300ms DropTail
$ns duplex-link $n2 $n3 1Mb 300ms DropTail
#The below code is used to set the queue size b/w the nodes
$ns set queue-limit $n0 $n2 10
$ns set queue-limit $n1 $n2 10
$ns set queue-limit $n2 $n3 5
#The below code is used to attach an UDP agent to n0, UDP agent to n1 and null agent to n3.
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0 set null [new Agent/Null]
$ns attach-agent $n3 $null set udp1 [new Agent/UDP]
```

```
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
#The below code sets the udp0 packets to red and udp1 packets to blue color
$udp0 set class_ 1
$udp1 set class_ 2
#The below code is used to connect the agents.
$ns connect $udp0 $null
$ns connect $udp1 $null
#The below code is used to set the packet size to 500
$cbr1 set packetSize_ 500Mb
#The below code is used to set the interval of the packets, i.e., Data rate of the packets.
#if the data rate is high then packets drops are high.
$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
```

AWK file:

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

```
BEGIN{

#include<stdio.h>

count=0;

}

{

if($1=="d")

#d stands for the packets drops.

count++

} END{

printf("The Total no of Packets Dropped due to Congestion :%d\n\n", count)

}
```

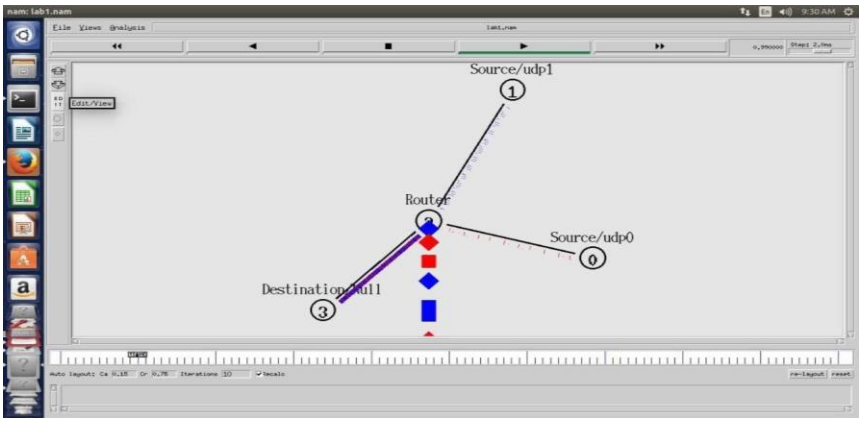
Steps for execution

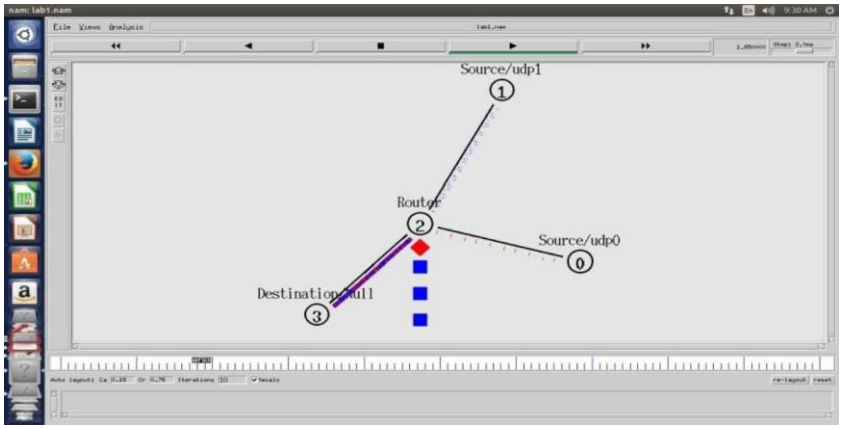
1. Open gedit and type program. Program name should have the extension " .tcl
   [root@localhost ~] gedit lab1.tcl
2. Save the program.
3. Open gedit and type awk program. Program name should have the extension ".awk "
   [root@localhost ~] gedit lab1.awk
4. Save the program.
5. Run the simulation program [root@localhost~] **ns lab1.tcl**
6. Here "ns" indicates network simulator. We get the topology shown in the snapshot.
7. Now press the play button in the simulation window and the simulation will begin.
8. After simulation is completed run awk file to see the output ,
   [root@localhost~] awk –f lab1.awk lab1.tr
9. To see the trace file contents open the file as ,
   [root@localhost~] gedit lab1.tr

Output:

The Total no of packets Dropped due to congestion: 456

Topology





Program No. 2: Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion

```
set ns [new Simulator]
set nf [open lab2.nam w]
$ns namtrace-all $nf
set nd [open lab2.tr w]
$ns trace-all $nd
proc finish {} {
global ns nf nd
$ns flush-trace
close  $nf
close $nd
exec nam lab2.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
$ns duplex-link $n1 $n0 1Mb 10ms DropTail
$ns duplex-link $n2 $n0 1Mb 10ms DropTail
$ns duplex-link $n3 $n0 1Mb 10ms DropTail
$ns duplex-link $n4 $n0 1Mb 10ms DropTail
$ns duplex-link $n5 $n0 1Mb 10ms DropTail
$ns duplex-link $n6 $n0 1Mb 10ms DropTail
Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts "node [$node_ id] recieved ping answer from \
$from with round-trip-time $rtt ms."
}
set p1 [new Agent/Ping]
set p2 [new Agent/Ping]
set p3 [new Agent/Ping]
set p4 [new Agent/Ping]
set p5 [new Agent/Ping]
set p6 [new Agent/Ping]
$ns attach-agent $n1 $p1
$ns attach-agent $n2 $p2
$ns attach-agent $n3 $p3
$ns attach-agent $n4 $p4
$ns attach-agent $n5 $p5
$ns attach-agent $n6 $p6
```

```
$ns queue-limit $n0 $n4 3
$ns queue-limit $n0 $n5 2
$ns queue-limit $n0 $n6 2
$ns connect $p1 $p4
$ns connect $p2 $p5
$ns connect $p3 $p6
$ns at 0.2 "$p1 send"
$ns at 0.4 "$p2 send"
$ns at 0.6 "$p3 send"
$ns at 1.0 "$p4 send"
$ns at 1.2 "$p5 send"
$ns at 1.4 "$p6 send"
$ns at 2.0 "finish"
$ns run
```

AWK file:

```
BEGIN {
count=0;
}
{
event=$1;
if(event=="d")
{
count++;
}
}
END {
printf("No of packets dropped : %d\n",count);
}
```

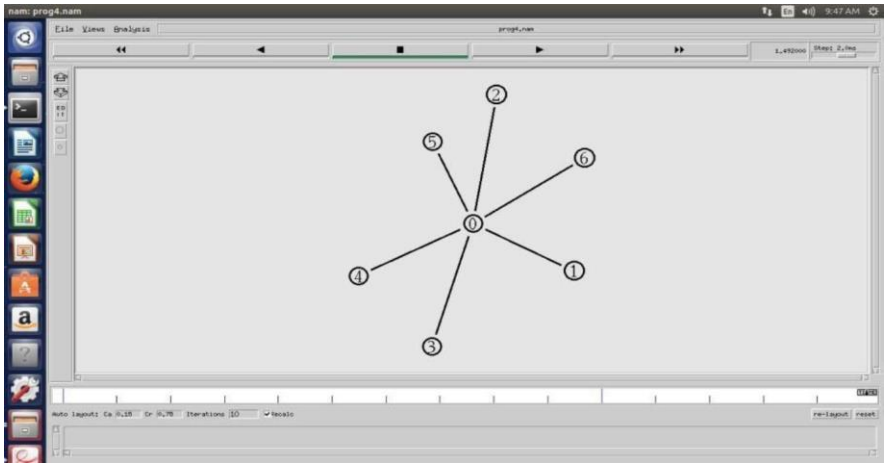
Output Commands:

[root@localhost ~]# ns lab2.tcl
[root@localhost ~]# awk -f lab2.awk lab2.tr

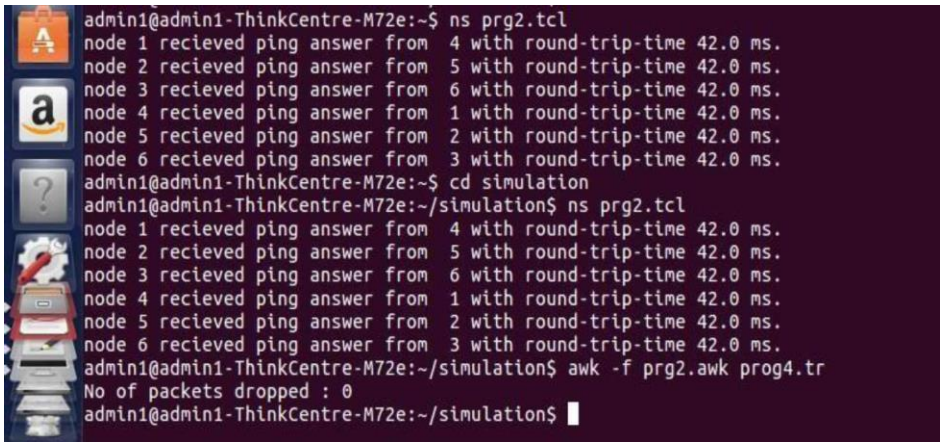
Output:

The Total no of packets dropped due to congestion: 6

Snapshot 1:



Snapshot_2:


admin1@admin1-ThinkCentre-M72e:~$ ns prg2.tcl
node 1 recieved ping answer from 4 with round-trip-time 42.0 ms.
node 2 recieved ping answer from 5 with round-trip-time 42.0 ms.
node 3 recieved ping answer from 6 with round-trip-time 42.0 ms.
node 4 recieved ping answer from 1 with round-trip-time 42.0 ms.
node 5 recieved ping answer from 2 with round-trip-time 42.0 ms.
node 6 recieved ping answer from 3 with round-trip-time 42.0 ms.
admin1@admin1-ThinkCentre-M72e:~$ cd simulation
admin1@admin1-ThinkCentre-M72e:~/simulation$ ns prg2.tcl
node 1 recieved ping answer from 4 with round-trip-time 42.0 ms.
node 2 recieved ping answer from 5 with round-trip-time 42.0 ms.
node 3 recieved ping answer from 6 with round-trip-time 42.0 ms.
node 4 recieved ping answer from 1 with round-trip-time 42.0 ms.
node 5 recieved ping answer from 2 with round-trip-time 42.0 ms.
node 6 recieved ping answer from 3 with round-trip-time 42.0 ms.
admin1@admin1-ThinkCentre-M72e:~/simulation$ awk -f prg2.awk prog4.tr
No of packets dropped : 0
admin1@admin1-ThinkCentre-M72e:~/simulation$

Program No. 3: Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.

```
set ns [new Simulator]
set nf [open lab3.nam w]
$ns namtrace-all $nf
set nd [open lab3.tr w]
$ns trace-all $nd
$ns color 1 Blue
$ns color 2 Red
proc finish { } {
global ns nf nd
$ns flush-trace
close $nf
close $nd
exec nam lab3.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
$n7 shape box
$n7 color Blue
$n8 shape hexagon
$n8 color Red
$ns duplex-link $n1 $n0 2Mb 10ms DropTail
$ns duplex-link $n2 $n0 2Mb 10ms DropTail
$ns duplex-link $n0 $n3 1Mb 20ms DropTail
$ns make-lan "$n3 $n4 $n5 $n6 $n7 $n8" 512Kb 40ms LL Queue/DropTail Mac/802_3
$ns duplex-link-op $n1 $n0 orient right-down
$ns duplex-link-op $n2 $n0 orient right-up
$ns duplex-link-op $n0 $n3 orient right
$ns queue-limit $n0 $n3 20
set tcp1 [new Agent/TCP/Vegas]
$ns attach-agent $n1 $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n7 $sink1
$ns connect $tcp1 $sink1
$tcp1 set class_ 1
$tcp1 set packetSize_ 55
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
```

```
set tfile [open cwnd.tr w]
$tcp1 attach $tfile
$tcp1 trace cwnd_
$ns at 0.5 "$ftp1 start"
$ns at 1.0 "$ftp2 start"
$ns at 5.0 "$ftp2 stop"
$ns at 5.0 "$ftp1 stop"
$ns at 5.5 "finish"
$ns run
```

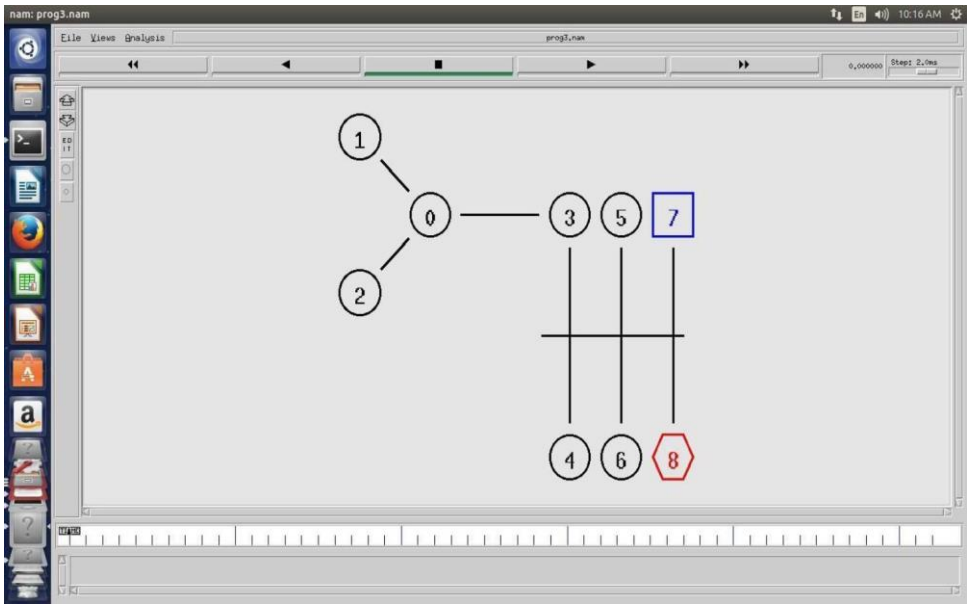
AWK File:

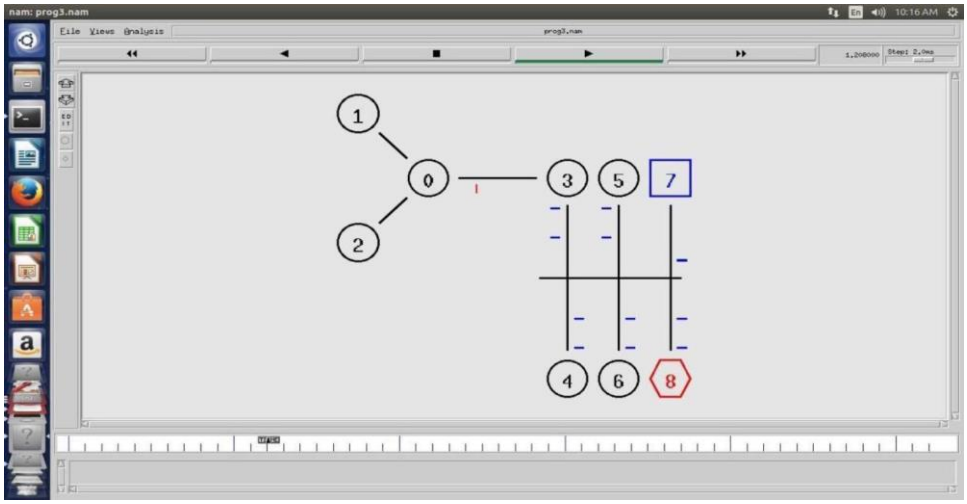
```
BEGIN {

}

{

if($6=="cwnd_")

 {
printf("%f\t%f\n",$1,$7);

}

}

END {

}
```

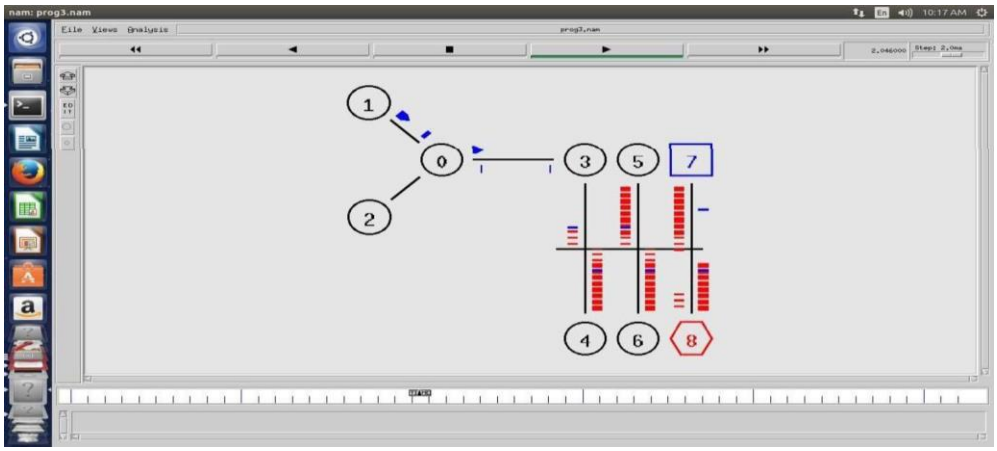
Output

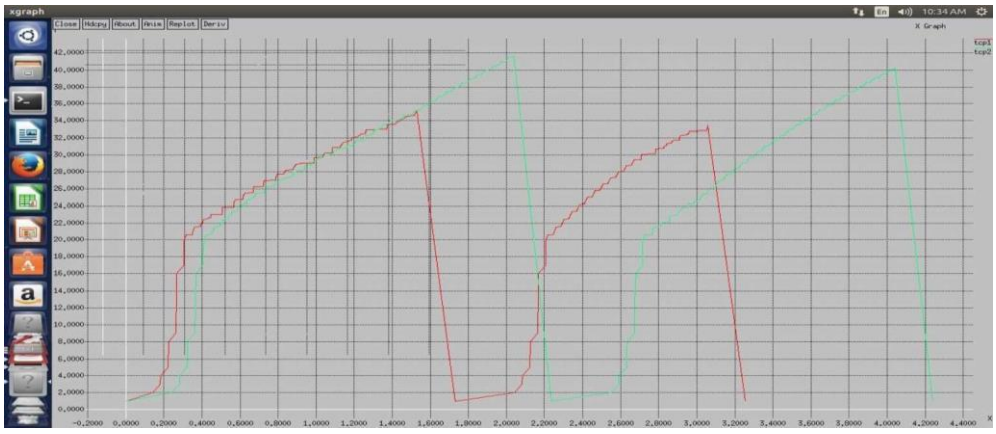
```
[root@localhost ~]# ns lab3.tcl

[root@localhost ~]# awk -f lab3.awk file1.tr>tcp1

[root@localhost ~]# awk -f lab3.awk file2.tr>tcp2

[root@localhost ~]# xgraph -x “time” -y “convalue” tcp1 tcp2
```

Topology


nam: prog3.nam
10:16 AM
File Views Analysis
1
0
2
3
5
7
4
6
8


nam: prog3.nam
10:16 AM
File Views Analysis
1
0
2
3
5
7
4
6
8


nam: prog3.nam
File Views Analysis
1
0
2
3
5
7
4
6
8


xgraph
Close Hdcpy About Mesh Replot Deriv
X Graph

Java Programs

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

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

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

1. Compile Java Program from Command Prompt
   [root@host ~]# javac Filename.java

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

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

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

Program 4. Write a program for error detecting code using CRC-CCITT (16- bits).

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

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

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

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

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

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

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

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

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

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

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

	CRC-CCITT	**CRC-16**	**CRC-32**
Checksum Width	16 bits	16 bits	32 bits
Generator Polynomial	1000100000010000 1	11000000000000101	10000010011000001000111011011011 1

International Standard CRC Polynomials

Algorithm:-

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

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

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

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

/* CRC */

```
import java.util.*;
public class Crc
{
   void div(int a[],int k)
{
int gp[]={1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,1};
int count=0;
for(int i=0;i<k;i++)
{
if(a[i]==gp[0])
{
for(int j=i;j<17+i;j++)
{
a[j]=a[j]^gp[count++];
}
count=0;
}
}
}
public static void main(String args[])
{
int a[]=new int[100];
int b[]=new int[100];
int len,k;
Crc ob=new Crc();
System.out.println("Enter the length of Data Frame:");
Scanner sc=new Scanner(System.in);
len=sc.nextInt();
int flag=0;
System.out.println("Enter the Message:");
for(int i=0;i<len;i++)
{
a[i]=sc.nextInt();
}
for(int i=0;i<16;i++)
{
a[len++]=0;
}
k=len-16;
for(int i=0;i<len;i++)
{
b[i]=a[i];
}
ob.div(a,k);
for(int i=0;i<len;i++)
a[i]=a[i]^b[i];
System.out.println("Data to be transmitted: ");
for(int i=0;i<len;i++)
{
   System.out.print(a[i]+" ");
}
System.out.println();
```

```
System.out.println("Enter the Reveived Data: ");
for(int i=0;i<len;i++)
{
a[i]=sc.nextInt();
}
ob.div(a, k);
for(int i=0;i<len;i++)
{
if(a[i]!=0)
{
flag=1; break;
}
}
if(flag==1)
System.out.println("error in data");
else
System.out.println("no error");
}
}
```

Output1

```
Enter the length of Data Frame:
4
Enter the Message:
1 0 1 1
Data to be transmitted:
1 0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1 1
Enter the Reveived Data:
1 0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1 1
no error
```

Output2

```
Enter the length of Data Frame:
4
Enter the Message:
1 0 1 1
Data to be transmitted:
1 0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1 1
Enter the Reveived Data:
1 0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 0 0 1
error in data
```

Program 5. Develop a program to implement a sliding window protocol in the data link layer.

Sliding Window Protocol

The Sliding Window Protocol is a key mechanism used in the Data Link Layer of the OSI model to manage the reliable transmission of data across a network. It ensures that data frames are sent and received in order without loss or duplication, which is crucial for maintaining the integrity of the data being transmitted.

Key Concepts of the Sliding Window Protocol

1. **Window Size**:
   - The protocol uses a "window" to limit the number of frames that can be sent before receiving an acknowledgment (ACK) from the receiver. The window size is the maximum number of frames that can be "in transit" at any given time.
2. **Sliding Window**:
   - As frames are sent and acknowledged, the window "slides" forward to allow new frames to be sent. If an acknowledgment for a frame is received, the window moves to the next set of frames. If an acknowledgment is not received, the sender may need to retransmit the frame(s), depending on the specific variant of the protocol.
3. **Acknowledgment (ACK) and Negative Acknowledgment (NAK)**:
   - The receiver sends back an acknowledgment (ACK) for each successfully received frame. If a frame is missing or corrupted, a negative acknowledgment (NAK) might be sent, or the receiver might just ignore the frame, prompting the sender to retransmit after a timeout.
4. **Sequence Numbers**:
   - Each frame is given a unique sequence number to keep track of the order of frames. This allows both the sender and receiver to know which frames have been sent, received, or missed.

Variants of Sliding Window Protocol

1. **Go-Back-N ARQ**:
   - In this variant, the sender can send several frames specified by the window size without waiting for an acknowledgment. However, if an acknowledgment is not received for a frame, the sender goes back and retransmits that frame and all subsequent frames, even if some of them were received correctly.
2. **Selective Repeat ARQ**:
   - Unlike Go-Back-N, in Selective Repeat, the sender only retransmits the frames that were not acknowledged or were acknowledged with errors. This is more efficient but requires more complex logic at both the sender and receiver.

/* Sliding Window*/

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

class SlidingWindowProtocol {
    private int windowSize;
    private int[] frames;
    private int sentIndex = 0;
    private int ackIndex = 0;
    private Random random;
```

```
public SlidingWindowProtocol(int totalFrames, int windowSize) {
    this.windowSize = windowSize;
    frames = new int[totalFrames];
    random = new Random();
    for (int i = 0; i < totalFrames; i++) {
        frames[i] = i + 1;
    }
}

public void sendFrames() {
    while (ackIndex < frames.length) {
        // Send frames within the window
        for (int i = 0; i < windowSize && sentIndex < frames.length; i++) {
            System.out.println("Sending Frame: " + frames[sentIndex]);
            sentIndex++;
        }

        // Simulate the acknowledgment
        receiveAcks();
    }
    System.out.println("All frames sent and acknowledged successfully.");
}

private void receiveAcks() {
    for (int i = ackIndex; i < sentIndex; i++) {
        if (random.nextBoolean()) {  // Simulate random ACK loss or success
            System.out.println("ACK received for Frame: " + frames[i]);
            ackIndex++;
        } else {
            System.out.println("ACK lost for Frame: " + frames[i]);
            System.out.println("Resending from Frame: " + frames[i]);
            sentIndex = ackIndex; // Reset the window to the last acknowledged frame
            break;
        }
    }
}

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

    System.out.print("Enter the total number of frames to send: ");
    int totalFrames = scanner.nextInt();

    System.out.print("Enter the window size: ");
    int windowSize = scanner.nextInt();
```

```
        SlidingWindowProtocol protocol = new SlidingWindowProtocol(totalFrames, windowSize);
        protocol.sendFrames();
    }
}
```

Output

```
Enter the total number of frames
to send: 8
Enter the window size: 4

Sending Frame: 1
Sending Frame: 2
Sending Frame: 3
Sending Frame: 4

ACK received for Frame: 1
ACK received for Frame: 2
ACK received for Frame: 3
ACK lost for Frame: 4
Resending from Frame: 4

Sending Frame: 4
Sending Frame: 5
Sending Frame: 6
Sending Frame: 7

ACK received for Frame: 4
ACK received for Frame: 5
ACK received for Frame: 6
ACK received for Frame: 7

Sending Frame: 8

ACK received for Frame: 8


All frames sent and acknowledged successfully.
```

Program 6: Write a program to find the shortest path between vertices using bellman-ford algorithm.

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

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

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

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

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

Implementation Algorithm:

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

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

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

}
```

Output1

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

Output2:

```
enter the no.of values
4
enter the weighted matrix:
0  4 999 5
999 0 999 999
999 -10 0 999
```

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

Output3
enter the no.of values
4
enter the weighted matrix:
0 4 5 999
999 0 999 7
999 7 0 999
999 999 -15 0
enter the source vertex:
1
the graph contains -ve edge cycle

Program 7:Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present. Implement the above program using as message queues or FIFOs as IPC channels.

The term network programming refers to writing programs that execute across multiple devices (computers), in which the devices are all connected to each other using a network.

The java.net package of the J2SE APIs contains a collection of classes and interfaces that provide the low-level communication details, allowing you to write programs that focus on solving the problem at hand.

The java.net package provides support for the two common network protocols

- **TCP –** TCP stands for Transmission Control Protocol, which allows for reliable communication between two applications. TCP is typically used over the Internet Protocol, which is referred to as TCP/IP.
- **UDP –** UDP stands for User Datagram Protocol, a connection-less protocol that allows for packets of data to be transmitted between applications.

Sockets are a protocol independent method of creating a connection between processes. Sockets can be either

- Connection based or connectionless: Is a connection established before communication or does each packet describe the destination?
- Packet based or streams based: Are there message boundaries or is it one stream?
- Reliable or unreliable: Can messages be lost, duplicated, reordered, or corrupted?

Socket characteristics

Sockets are characterized by their domain, type and transport protocol. Common domains are:

- AF_UNIX: address format is UNIX pathname
- AF_INET: address format is host and port number

Common types are:

- virtual circuit: received in order transmitted and reliably
- datagram: arbitrary order, unreliable

Each socket type has one or more protocols. Ex:

- TCP/IP (virtual circuits)
- UDP (datagram)

Use of sockets:

- Connection–based sockets communicate client-server: the server waits for a connection from the client
- Connectionless sockets are peer-to-peer: each process is symmetric.

Socket is an interface which enables the client and the server to communicate and pass on information from one another. Sockets provide the communication mechanism between two computers using TCP.

A client program creates a socket on its end of the communication and attempts to connect that socket to a server. When the connection is made, the server creates a socket object on its end of the communication.

The client and the server can now communicate by writing to and reading from the socket. The java.net.Socket class represents a socket, and the java.net.ServerSocket class provides a mechanism for the server program to listen for clients and establish connections with them.

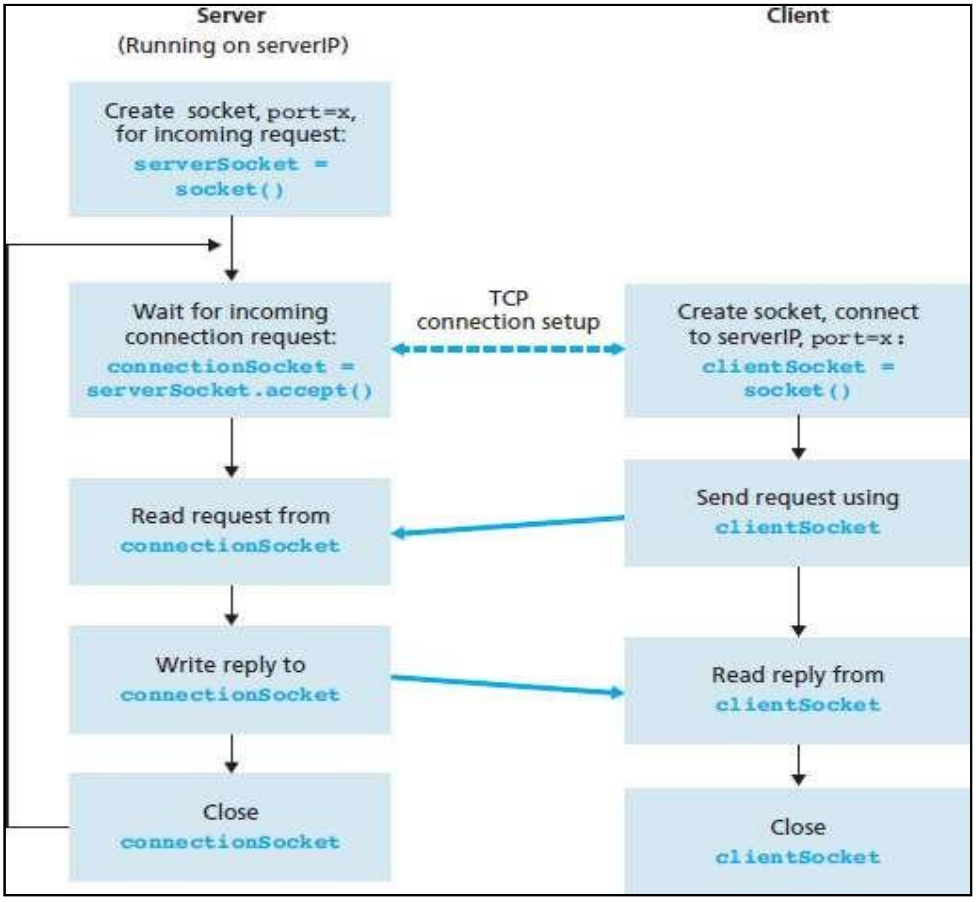


Fig. 6 The client-server application using TCP

As shown in the Fig. 1, the following steps occur when establishing a TCP connection between two computers using sockets –

- The server instantiates a ServerSocket object, denoting which port number communication is to occur on.
- The server invokes the accept() method of the ServerSocket class. This method waits until a client connects to the server on the given port.
- After the server is waiting, a client instantiates a Socket object, specifying the server name and the port number to connect to.
- The constructor of the Socket class attempts to connect the client to the specified server and the port number. If communication is established, the client now has a Socket object capable of communicating with the server.
- On the server side, the accept() method returns a reference to a new socket on the server that is connected to the client's socket.

Algorithm (Client Side)

1. Start.
2. Create a socket using socket() system call.
3. Connect the socket to the address of the server using connect() system call.
4. Send the filename of required file using send() system call.
5. Read the contents of the file sent by server by recv() system call.
6. Stop.

Algorithm (Server Side)

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

TCP Client

At client side:

```
/*TCPClient*/
import java.net.*;
import java.io.*;
public class TCPClient
{
	public static void main(String args[]) throws Exception {
		Socket sock=new Socket("127.0.0.1",4000);
		System.out.println("Enter the filename");
		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();
            socketRead.close();
            keyRead.close();
        }
}
```

TCP Server

At server side:

```
/* TCPServer */
import java.net.*;
import java.io.*;
public class TCPServer
{
    public static void main(String args[]) throws Exception {
        ServerSocket sersock=new ServerSocket(4000);
        System.out.println("Server ready for Connection");
        Socket sock=sersock.accept();
        System.out.println("Connection is Successful and waiting for chatting");
        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();
        sersock.close();
        pwrite.close();
        fileRead.close();
        contentRead.close();

    }
}
```

Note: Create two different files TcpClient.java and TcpServer.java. Follow the steps given:

1. Open a terminal run the server program and provide the filename to send
2. Open one more terminal run the client program and provide the IP address of the server. We can give localhost address "127.0.0.1" as it is running on same machine or give the IP address of the machine.
3. Send any start bit to start sending file.

First Method of Executing TCP/IP sockets, write a client – server program

Output1

```
student@student:~/naveen$ javac TCPServer.java
student@student:~/naveen$ java TCPServer
Server ready for Connection
Connection is Successful and waiting for chatting

student@student:~/naveen$
student@student:~/naveen$ javac TCPClient.java
student@student:~/naveen$  java  TCPClient
Enter the filename
abc.txt
atme college of engineering,mysuru
student@student:~/naveen$
```

Second Method of Executing TCP/IP sockets, write a client – server program

Output2

TCPServer

first run TCPServer program .you will get below message that server is started and ready to connect with client

Server ready for Connection

TCPClient

Next run TCPClient program

Enter the filename
/home/student/naveen/abc.txt
hello atme college of engineering

Program 8: Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.

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

Datagram packets are used to implement a connectionless packet delivery service supported by the UDP protocol. Each message is transferred from source machine to destination based on information contained within that packet. That means, each packet needs to have destination address and each packet might be routed differently, and might arrive in any order. Packet delivery is not guaranteed.

Java supports datagram communication through the following classes:

- DatagramPacket
- DatagramSocket

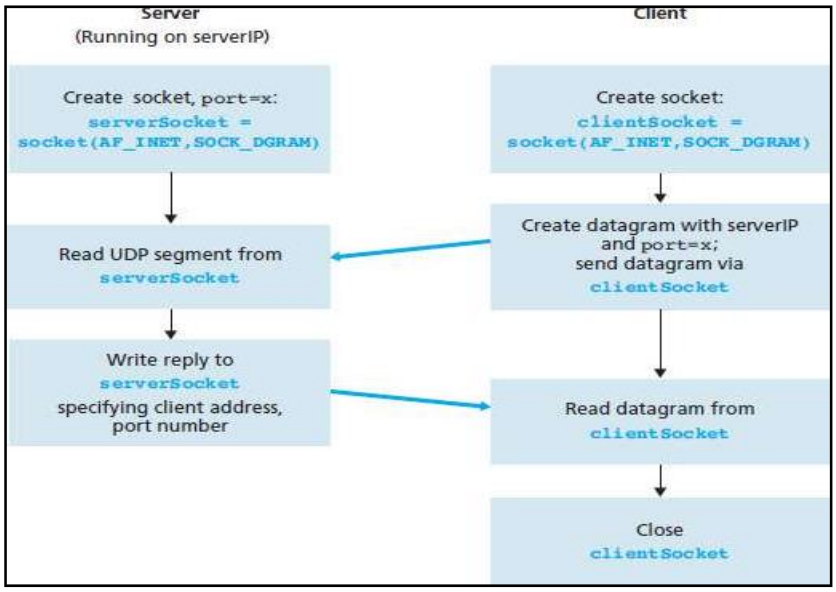


Fig -7-UDP client/server communication flow:

/* Datagram Socket Program */

UDP Server

```
import java.io.*;
import java.net.*;
public class UDPServer
{
  public static void main(String[] args)
 {
DatagramSocket skt=null;
try
{
System.out.println("server is started");
skt=new DatagramSocket(6788);
byte[] buffer = new byte[1000];
while(true)
{
DatagramPacket request = new DatagramPacket(buffer,buffer.length);
skt.receive(request);
String[] message = (new String(request.getData())).split(" ");
byte[] sendMsg= (message[1].toUpperCase()+ " from server to client").getBytes();
DatagramPacket reply = new
DatagramPacket(sendMsg,sendMsg.length,request.getAddress(),request.getPort());
skt.send(reply);
}
}
catch(Exception ex)
{
System.out.println(ex.getMessage());
}
}
}
```

UDP Client

```
import java.io.*;
import java.net.*;
public class UDPClient
{
 public static void main(String[] args)
{
DatagramSocket skt;
try
{
skt=new DatagramSocket();
String msg= "atme college ";
byte[] b = msg.getBytes();
InetAddress host=InetAddress.getByName("127.0.0.1");
int serverSocket=6788;
DatagramPacket request =new DatagramPacket (b,b.length,host,serverSocket);
skt.send(request);
byte[] buffer =new byte[1000];
DatagramPacket reply= new DatagramPacket(buffer,buffer.length);
skt.receive(reply);
System.out.println("client received:" +new String(reply.getData()));
```

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

Output1:

client received : COLLEGE from server to client

Program 9: Write a program for simple RSA algorithm to encrypt and decrypt the data.

RSA is algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. Asymmetric means that there are two different keys. This is also called *public key cryptography*, because one of them can be given to everyone. The other key must be kept private.

- It is based on the fact that finding the factors of an integer is hard (the factoring problem). RSA stands for **Ron Rivest, Adi Shamir** and **Leonard Adleman**, who first publicly described it in 1978. A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key.
- The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message.
- RSA involves a public key and private key. The public key can be known to everyone; it is used to encrypt messages. Messages encrypted using the public key can only be decrypted with the private key. The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers.
- The RSA algorithm's efficiency requires a fast method for performing the modular exponentiation operation. A less efficient, conventional method includes raising a number (the input) to a power (the secret or public key of the algorithm, denoted *e* and *d*, respectively) and taking the remainder of the division with *N*. A straight-forward implementation performs these two steps of the operation sequentially: first, raise it to the power and second, apply modulo.

Basically RSA is cryptographic algorithm which is meant to encrypt the data, generally used in network security applications while we are sending the data from one source to destination. The concept of RSA algorithm starts with a two keyconcepts, it uses two keys (asymmetric keys) one

- is considered as the public key and another is a private key.

It was developed because using the symmetric encryption algorithm is easy but the key distribution is difficult, so the concept of two key concept appears to be more efficient. The whole algorithm depends on the fact that "It is not possible to judge another key when attacker

- gets one key" Here in two keys, one key is taken as the public key another as a private key.

- Public key is available for everyone to access, so whenever sender want to send the data to receiver, he uses the public key of receiver (as it is available for use to all) and encrypts the data using the key, this encrypted data is called cipher text, when receiver receives the cipher text, he can decrypt the data using his private key. Here even if theattacker knows the encryption algorithm, he can't do anything until the keys are available.

A very simple example of RSA encryption

1. Select primes p = 11, q = 3.

2. n = p*q = 11*3 = 33

   phi = (p-1)(q-1) = 10*2 = 20

3. 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, phi) = gcd(e, (p-1)(q-1)) = gcd(3, 20) = 1

4. Compute d such that ed ≡ 1 (mod phi)

i.e. compute d = e^-1 mod phi = 3^-1 mod 20

i.e. find a value for d such that phi divides (ed-1)

i.e. find d such that 20 divides 3d-1. Simple testing (d = 1, 2, ...) gives d = 7

Check: ed-1 = 3*7 - 1 = 20, which is divisible by phi.

5. Public key = (n, e) = (33, 3) Private key = (n, d) = (33, 7).

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

$c = m^{\wedge e} \bmod n = 7^{\wedge 3} \bmod 33 = 343 \bmod 33 = 13.$

Hence the ciphertext c = 13.

To check decryption we compute

$m' = c^{\wedge d} \bmod n = 13^{\wedge 7} \bmod 33 = 7.$

Note that we don't have to calculate the full value of 13 to the power 7 here. We can make use of the fact that a = bc mod n = (b mod n).(c mod n) mod n so we can break down a potentially large number into its components and combine the results of easier, smaller calculations to calculate the final value.

One way of calculating m' is as follows:-

$m' = 13^{\wedge 7} \bmod 33 = 13^{\wedge (3+3+1)} \bmod 33 = 13^{\wedge 3}.13^{\wedge 3}.13 \bmod 33$

= $(13^{\wedge 3} \bmod 33).(13^{\wedge 3} \bmod 33).(13 \bmod 33) \bmod 33$

= (2197 mod 33).(2197 mod 33).(13 mod 33) mod 33

= 19.19.13 mod 33 = 4693 mod 33

= 7.

Key Generation Algorithm

1. Generate two large random primes, p and q, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.

2. Compute n = pq and (φ) phi = (p-1)(q-1).

3. Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1.

4. Compute the secret exponent d, 1 < d < phi, such that $ed \equiv 1 \pmod{phi}$.

5. The public key is (n, e) and the private key is (n, d). The values of p, q, and phi should also be kept secret.
    - n is known as the modulus.
    - e is known as the public exponent or encryption exponent.
    - d is known as the secret exponent or decryption exponent.

Note: It is possible to find a smaller d by using lcm(p-1,q-1) instead of phi, lcm(p-1,q-1) = phi /gcd(p-1,q-1)).

Encryption

Sender A does the following:-

1. Obtains the recipient B's public key (n, e).
2. Represents the plaintext message as a positive integer m.
3. Computes the ciphertext $c = m^{\wedge e} \bmod n$.
4. Sends the ciphertext c to B.

Decryption

Recipient B does the following:-

1. Uses his private key (n, d) to compute $m = c^{\wedge d} \bmod n$.
2. Extracts the plaintext from the integer representative m.

/* RSA Key Generation */

Source Code:

```
import java.util.*;
import java.io.*;
public class rsa
{
  static int gcd(int m,int n)
{
while(n!=0)
{
  int r=m%n;
m=n;
n=r;
}
return m;
}
public static void main(String args[])
{
int p=0,q=0,n=0,e=0,d=0,phi=0;
int nummes[]=new int[100];
int encrypted[]=new int[100];
int decrypted[]=new int[100];
int i=0,j=0,nofelem=0;
Scanner sc=new Scanner(System.in);
String message ;
System.out.println("Enter the Message tobe encrypted:");
message= sc.nextLine();
System.out.println("Enter value of p and q\n");
 p=sc.nextInt();
q=sc.nextInt();
n=p*q;
phi=(p-1)*(q-1);
for(i=2;i<phi;i++)
if(gcd(i,phi)==1)
break;
e=i;
for(i=2;i<phi;i++)
if((e*i-1)%phi==0)
break;
d=i;
for(i=0;i<message.length();i++)
{
char c = message.charAt(i);
int a =(int)c;
nummes[i]=c-96;
}
nofelem=message.length();
for(i=0;i<nofelem;i++)
{
encrypted[i]=1;
for(j=0;j<e;j++)
encrypted[i] =(encrypted[i]*nummes[i])%n;
}
```

```
System.out.println("\n Encrypted message\n");
for(i=0;i<nofelem;i++)
{
System.out.print(encrypted[i]);
System.out.print((char)(encrypted[i]+96));
}
for(i=0;i<nofelem;i++)
{
decrypted[i]=1;  for(j=0;j<d;j++)

decrypted[i]=(decrypted[i]*encrypted[i])%n;
}

System.out.println("\n Decrypted message\n ");
for(i=0;i<nofelem;i++)
System.out.print((char)(decrypted[i]+96)); return;
}
}
```

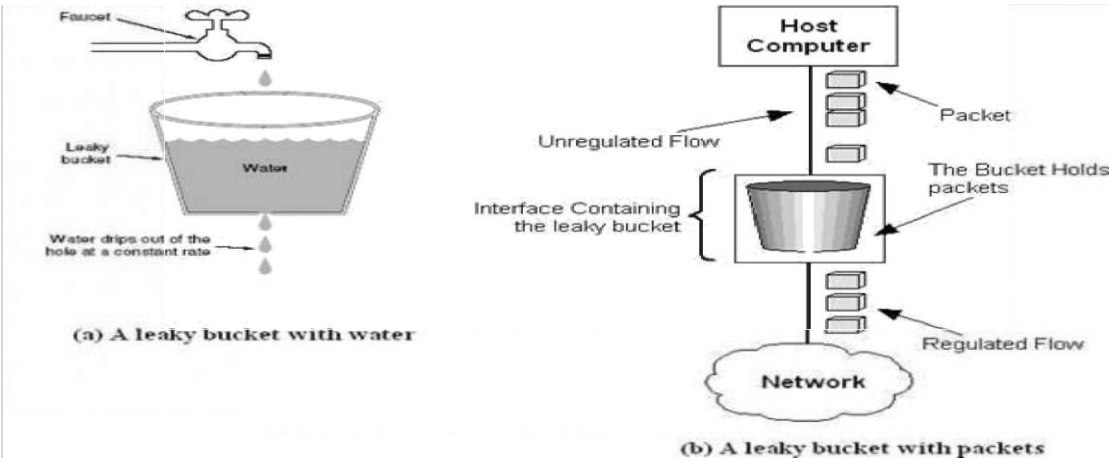
Output

```
Enter the text:
hello
Enter the value of P and Q :
5
7
Encrypted Text is: 8 h 10 j 17 q 17 q 15 o
Decrypted Text is: hello
```

Program 10: Write a program for congestion control using leaky bucket algorithm.

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

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



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

The leaky-bucket algorithm:

The algorithm can be conceptually understood as follows:

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

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

/* Leaky Bucket */

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

```
drop=0;
System.out.println();
}
for(;count!=0;i++)
{
if(count>cap)
{
drop=count-cap;
count=cap;
}
System.out.print(i+1);
System.out.print("\t\t0");
mini=min(count,process);
System.out.print("\t\t"+mini);
count=count-mini;
System.out.print("\t\t"+count);
System.out.print("\t\t"+drop);
System.out.println();
}
}
}
```

Output1

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

Output2

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

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

VIVA QUESTIONS AND ANSWERS

1) What is a Link?

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

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

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

3) What is backbone network?

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

4) What is a LAN?

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

5) What is a node?

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

6) What are routers?

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

7) What is point to point link?

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

8) What is anonymous FTP?

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

9) What is subnet mask?

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

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

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

11) What is data encapsulation?

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

12) Describe Network Topology

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

13) What is VPN?

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

14) Briefly describe NAT.

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

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

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

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

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

17) What is RIP?

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

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

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

19) What is NIC?

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

20) What is WAN?

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

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

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

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

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

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

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

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

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

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

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

26) What does 10Base-T mean?

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

27) What is a private IP address?

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

29) What is DoS?

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

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

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

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

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

32) What is the advantage of address sharing?

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

33) What are MAC addresses?

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

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

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

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

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

36) What is the main purpose of OSPF?

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

37) What are firewalls?

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

38) Describe star topology

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

39) What are gateways?

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

40) What is the disadvantage of a star topology?

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

41) What is SLIP?

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

42) Give some examples of private network addresses.

10.0.0.0 with a subnet mask of 255.0.0.0

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

43) What is tracert?

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

44) What are the functions of a network administrator?

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

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

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

46) What is Hybrid Network?

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

47) What is DHCP?

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

48) What is the main job of the ARP?

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

49) What is TCP/IP?

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

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

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

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

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

52) What is the use of a default gateway?

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

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

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

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

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

55) What is netstat?

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

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

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

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

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

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

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

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

59) What is ICMP?

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

60) What is Ping?

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

61) What is peer to peer?

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

62) What is DNS?

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

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

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

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

A hub acts as a multiport repeater. However, as more and more devices connect to it, it would not be able to efficiently manage the volume of traffic that passes through it. A switch provides a better alternative that can improve the performance especially when high 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-relatedproblems can occur?

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

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

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

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

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

91) Explain profile in terms of networking concept?

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

92) What is sneakernet?

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

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

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

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

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

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

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

96) What is one basic requirement for establishing VLANs?

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

97) What is IPv6?

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

98) What is RSA algorithm?

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

99) What is mesh topology?

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

100) What is ns2?

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

101) What is simulation?

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

102) Explain basic architecture of NS-2.

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

103) What is trace file?

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

104) What is nam file?

A visual aid showing how packets flow along the network.

105) Why awk file is used?

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

106) What is xgraph?

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

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

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