RN SHETTY TRUST®
RNS INSTITUTE OF TECHNOLOGY
Autonomous Institution Affiliated to VTU, Recognized by GOK, Approved by AICTE
(NAAC 'A+ Grade' Accredited, NBA Accredited (UG - CSE, ECE, ISE, EIE and EEE)
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098
Ph:(080)28611880,28611881 URL: www.rnsit.ac.in

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ESTD : 2001
An Institute with a Difference

COMPUTER NETWORKS LABORATORY

For Fifth Semester B.E-2022 Batch [VTU/CBCS, 2022 syllabus]

Subject Code – BCS502

NAME : _____

SECTION : _____

USN : _____

Vision

Building RNSIT into a World Class Institution

Mission

To impart high quality education in Engineering, Technology and Management with a Difference, Enabling Students to Excel in their Career by

1. Attracting quality Students and preparing them with a strong foundation in fundamentals so as to achieve distinctions in various walks of life leading to outstanding contributions
2. Imparting value based, need based, choice based and skill based professional education to the aspiring youth and carving them into disciplined, World class Professionals with social responsibility
3. Promoting excellence in Teaching, Research and Consultancy that galvanizes academic consciousness among Faculty and Students
4. Exposing Students to emerging frontiers of knowledge in various domains and make them suitable for Industry, Entrepreneurship, Higher studies, and Research & Development
5. Providing freedom of action and choice for all the Stake holders with better visibility

COMPUTER NETWORKS LABORATORY-BCS502
INTERNAL EVALUATION SHEET

EVALUATION (MAX MARKS 50)			
TEST **A**	**REGULAR EVALUATION** **B**	**RECORD** **C**	**TOTAL MARKS** **A+B+C**
5	**5**	**10**	**20**

R1: REGULAR LAB EVALUATION WRITE UP RUBRIC (MAX MARKS 10)				
Sl. No.	**Parameters**	**Good**	**Average**	**Needs improvement**
a.	**Understanding of problem (3 marks)**	Clear understanding of problem statement while designing and implementing the program (3)	Problem statement is understood clearly but few mistakes while designing and implementing program (2)	Problem statement is not clearly understood while designing the program (1)
b.	**Writing program (4 marks)**	Program handles all possible conditions (4)	Average condition is defined and verified. (3)	Program does not handle possible conditions (1)
c.	**Result and documentation (3 marks)**	Meticulous documentation and all conditions are taken care (3)	Acceptable documentation shown (2)	Documentation does not take care all conditions (1)

R2: REGULAR LAB EVALUATION VIVA RUBRIC (MAX MARKS 10)					
Sl. No.	**Parameter**	**Excellent**	**Good**	**Average**	**Needs Improvement**
a.	**Conceptual understanding (10 marks)**	Answers 80% of the viva questions asked (10)	Answers 60% of the viva questions asked (7)	Answers 30% of the viva questions asked (4)	Unable to relate the concepts (1)

R3: REGULAR LAB PROGRAM EXECUTION RUBRIC (MAX MARKS 10)				
Sl. No.	**Parameters**	**Excellent**	**Good**	**Needs Improvement**
a.	**Design, implementation and demonstration (5 marks)**	Program follows syntax and semantics of C programming language. Demonstrates the complete knowledge of the program written (5)	Program has few logical errors, moderately demonstrates all possible concepts implemented in programs (3)	Syntax and semantics of C programming is not clear (1)
b.	**Result and documentation (5 marks)**	All test cases are successful, all errors are debugged with own practical knowledge and clear documentation according to the guidelines (5)	Moderately debugs the programs , few test case are unsuccessful and Partial documentation (3)	Test cases are not taken care , unable to debug the errors and no proper documentation (1)

R4: RECORD EVALUATION RUBRIC (MAX MARKS 10)					
Sl. No.	**Parameter**	**Excellent**	**Good**	**Average**	**Needs Improvement**
a.	**Documentation (10 marks)**	Meticulous record writing including program, comments and test cases as per the guidelines mentioned (20)	Write up contains program and test cases, but comments are not included (18)	Write up contains only program (15)	Program written with few mistakes (10)

TEST /LAB INTERNALS MARKS (MAX MARKS 5)

TEST #	Write up 5	Execution 10	Viva 5	Sign	Total 20	Avg. 40	Final 5
TEST-1							
TEST-2							

REGULAR LAB EVALUATION (MAX MARKS 5)

Lab program	Date of Execution	Additional programs	Write up (5)	Execution (5)	Viva (5)	Total 15	Teacher Signature
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
Total Marks		**105**				**5**	

Final Marks obtained from test (05) + regular evaluation (05)	10	Lab in charge : **HOD:**
Record(10)	10	
Total Marks Obtained		

PREFACE

We have developed this comprehensive laboratory manual on **COMPUTER NETWORKS LABORATORY-BCS502** with two primary objectives: To make the students comfortable with basic principles of problem solving and to train them in evolving as an efficient COMPUTER NETWORKS by strengthening their CN programming abilities.

This material provides students an exposure to problem solving approaches and solution to number of problems using Java and NS-3 programming. The problems discussed in this manual comprises of an algorithm, programming solution, alternative logic and extensive test cases. Viva questions, frequently appeared examination questions and practicing programming problems constitute an indispensable part of this material.

Our profound and sincere efforts will be fruitful only when students acquire the extensive knowledge by reading this manual and apply the concepts learnt apart from the requirements specified in COMPUTER NETWORKS LABORATORY-BCS502 as prescribed by VTU, Belagavi.

Departments of CSE

TABLE OF CONTENTS

ACKNOWLEDGMENT

A material of this scope would not have been possible without the contribution of many people. We express our sincere gratitude to RNS Group of Companies for their magnanimous support in all our endeavors.

We are grateful to Dr.Ramesh Babu H S, Principal, RNSIT, Dr. Kavitha C, HOD, CSE for extending their constant encouragement and support.

Our heartfelt thanks to Prof. Devaraju B M and Prof. Chethana H R for their unparalleled contribution throughout the preparation of this comprehensive manual. We also acknowledge our colleagues for their timely suggestions and unconditional support.

Department of CSE

INTRODUCTION TO NS3

NS3 Simulator Basics

- NS-3 is a network simulator
- Developed for network research and education
- Developed afterns-2
- ns-3 is written in C++
- Bindings in Python
- ns-3 uses the waf build system

Waf is a build automation tool designed to assist in the automatic compilation and installation of computer software. It is written inPython.

Waf features:

- Portable to Unix and non-Unix systems
- Lightweight
- Offers a Turing-complete programming language
- Support for standard targets: configure, build, clean, install, and uninstall
- Parallel builds
- Colored output and progress bar display
- Scripts are Python modules
- XML script front-end and a dedicated, easy-to-parse "IDE output" mode to ease the interaction with integrated development environments
- Modular configuration scheme with customizable command-line parsing
- Daemon mode for background recompilation
- Find source files intelligently (glob()-like) to ease script maintenance
- Support for global object cache to avoid unnecessary recompilations
- Support for unit tests run on programs at the end of builds

Waf supports:

- AC/C++preprocessor for computing dependencies simulation programs are C++executables or python scripts

Features

- It is a discrete event simulator
- Modular design / Opensource
- Actively developed (ContrastNS-2)
- Developed in C++. Python binding available.
- Live visualizer
- Logging facility for debugging

- Tracing facility for getting output
- Can be connected to a real network
- Direct Code Execution(DCE)

How to install ns3?
Download tarball from www.nsnam.orgthe recent release in your directory. Go to that directory and untar the tarball.

tar xvfj ns3.tar

It creates the directory for ns3 change to it.

cd ns3

For compiling and installing

./build.py --enable-examples --enable-tests

How to run ns3 script?
To test the installation copy one example available in the distribution to scratch directory and build and run the same using the commands below:

cd ns3.26
cp examples/tutorial/first.cc scratch/first.cc
./waf --run scratch/first

Steps in writing scripts

- Include necessary files
- Use appropriate namespace
- Set simulation time resolution(Optional)
- Enable logging for different modules(Optional)
- Create nodes
- Create net devices with MAC and PHY
- Attach Net devices to nodes and set interconnections
- Install protocol stack In nodes
- Set network address for interfaces
- Setup routing
- Install applications in nodes
- Setup tracing(Optional)
- Set application start and stop time
- Set simulation start time(Optional)
- Run simulation
- Release resources at end of simulation

Tutorial : First.cc

Simple point to point (Wired network) link between server and client is established here. This program is in your NS3 repository.(example/tutorial/first.cc)

Note : To know about NS3, you must have the base knowledge in c++ and OOPS concept.

1. ModuleIncludes

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"

Each of the ns-3 include files is placed in a directory called ns3 (under the build directory) during the build process to help avoid include file name collisions. The ns3/core-module.h file corresponds to the ns-3 module you will find in the directorysrc/core in your downloaded release distribution. If you list this directory you will find a large number of header files. When you do a build, Waf will place public header files in an ns3 directory under the appropriate build/debug or build/optimized directory depending on your configuration. Waf will also automatically generate a module include file to load all of the public header files.

2. NS3Namespace

using namespace ns3;

The ns-3 project is implemented in a C++ namespace called ns3. This groupsall ns-3-related declarations in a scope outside the global namespace, which we hope will help with integration with other code. The C++ using statement introduces the ns-3 namespace into the current (global) declarative region. This is a fancy way of saying that after this declaration, you will not have to type ns3:: scope resolution operator before all of the ns-3 code in order to use it. If you are unfamiliar with namespaces, please consult almost any C++ tutorial and compare the ns3 namespace and usage here with instances of the std namespace and the using namespace std; statements you will often find in discussions of cout andstreams.

3. Set simulation timeresolution

*int main (int argc, char *argv[])*

This is just the declaration of the main function of your program (script). Just as in any C++ program, you need to define a main function that will be the first function run. There is nothing at all special here. Your ns3 script is just a C++program.

The next line sets the time resolution to one nanosecond, which happens to be the default value:

Time::SetResolution (Time::NS);

The resolution is the smallest time value that can be represented (as well as the smallest represent able difference between two time values). You can change the resolution exactly once. The mechanism enabling this flexibility is somewhat memory hungry, so once the resolution has been set explicitly we release the memory, preventing further updates. (If you don't set the resolution explicitly, it will default to one nanosecond, and the memory will be released when the simulation starts.)

4. Enable logging for differentmodules

NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");

this line declares a logging component called FirstScriptExample that allows you to enable and disable console message logging by reference to thename.

5. Createnodes

NodeContainer

nodes;nodes.Create (2);

The NodeContainer topology helper provides a convenient way to create, manage and access any Node objects that we create in order to run a simulation. The first line above just declares a NodeContainer which we call nodes. The second line calls the Create method on the nodes object and asks the container to create twonodes.

6. Create net devices with MAC andPHY

PointToPointHelperpointToPoint;

It instantiates a PointToPointHelper object on the stack. From a high-level perspective the next line,

pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));

Above line tells the PointToPointHelper object to use the value “5Mbps” (five megabits per second) as the “DataRate” when it creates a PointToPointNetDevice object. From a more detailed perspective, the string “DataRate” corresponds to what we call an Attribute of the PointToPointNetDevice.

pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

It tells the PointToPointHelper to use the value “2ms” (two milliseconds) as the value of the transmission delay of every point to point channel it subsequentlycreates.

7. Attach Net devices to nodes and setinterconnections

NetDeviceContainerdevices;

devices = pointToPoint.Install (nodes);

The first line declares the device container mentioned above and the second does the heavy lifting. The Install method of the PointToPointHelper takes a NodeContainer as a parameter. Internally, a NetDeviceContainer is created. For each node in the NodeContainer (there must be exactly two for a point- to-point link) a PointToPointNetDevice is created and saved in the devicecontainer.

A PointToPointChannel is created and the two PointToPointNetDevices are attached. When objects are created by the PointToPointHelper, the Attributes previously set in the helper are used to initialize the corresponding Attributes in the created objects. After executing the pointToPoint.Install (nodes) call we will have two nodes, each with an installed point-to-point net device and a single point-to-point channel

between them. Both devices will be configured to transmit data at five megabits per second over the channel which has a two millisecond transmissiondelay.

8. Install protocol stack innodes

The only user-visible API is to set the base IP address and network mask to use when performing the actual address allocation (which is done at a lower level inside the helper).

Ipv4AddressHelper address;

address.SetBase ("10.1.1.0", "255.255.255.0");

It declares an address helper object and tell it that it should begin allocating IP addresses from thenetwork 10.1.1.0 using the mask 255.255.255.0 to define the allocatable bits. By default the addresses allocated will start at one and increase monotonically, so the first address allocated from this base will be 10.1.1.1, followed by 10.1.1.2, etc. The low level ns3 system actually remembers all of the IP addresses allocated and will generate a fatal error if you accidentally cause the same address to be generated twice (which is a very hard to debug error, by theway).

9. Set network address forinterfaces

Ipv4InterfaceContainer interfaces = address.Assign (devices);

It performs the actual address assignment. In ns-3 we make the association between an IP address and a device using an Ipv4Interface object. Just as we sometimes need a list of net devices created by a helper for future reference we sometimes need a list of Ipv4Interface objects. The Ipv4InterfaceContainer provides this functionality. Now we have a point-to-point network built, with stacks installed and IP addresses assigned. What we need at this point are applications to generatetraffic.

10. Setuprouting

UdpEchoServerHelperechoServer (9);

ApplicationContainerserverApps = echoServer.Install (nodes.Get (1));

The first line of code in the above snippet declares the UdpEchoServerHelper. As usual, this isn't the application itself, it is an object used to help us create the actual applications. One of our conventions is to place required Attributes in the helper constructor. In this case, the helper can't do anything useful unless it is provided with a port number that the client also knows about. Rather than just picking one and hoping it all works out, we require the port number as a parameter to the constructor. The constructor, in turn, simply does a SetAttribute with the passed value. If you want, you can set the "Port" Attribute to another value later usingSetAttribute.

Similar to many other helper objects, the UdpEchoServerHelper object has an Install method. It is the execution of this method that actually causes the underlying echo server application to be instantiated and attached to a node. Interestingly, the Install method takes a NodeContainter as a parameter just as the other

Install methods we have seen. This is actually what is passed to the method even though it doesn't look so in this case. There is a C++ implicit conversion at work here that takes the result of nodes.Get (1) (which returns a smart pointer to a node object — Ptr<Node>) and uses that in a constructor for an unnamed NodeContainer that is then passed to Install. If you are ever at a loss to find a particular method signature in C++ code that compiles and runs just fine, look for these kinds ofimplicit conversions.

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (10.0));

Above lines cause the echo server application to Start (enable itself) at one second into the simulation and to Stop (disable itself) at ten seconds into the simulation. By virtue of the fact that we have declared a simulation event (the application stop event) to be executed at ten seconds, the simulation will last at least tenseconds.

11. Install applications innodes

UdpEchoClientHelperechoClient (interfaces.GetAddress (1), 9);

echoClient.SetAttribute ("MaxPackets", UintegerValue (1));

echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));

echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

For the echo client, however, we need to set five different Attributes. The first two Attributes are set during construction of the UdpEchoClientHelper. We pass parameters that are used (internally to the helper) to set the "RemoteAddress" and "RemotePort" Attributes in accordance with our convention to make required Attributes parameters in the helperconstructors.

The zeroth interface in the interfaces container is going to correspond to the IP address of the zeroth node in the nodes container. The first interface in the interfaces container corresponds to the IP address of the first node in the nodes container. So, in the first line of code (from above), we are creating the helper and telling it so set the remote address of the client to be the IP address assigned to the node on which the server resides. We also tell it to arrange to send packets to portnine.

The "MaxPackets" Attribute tells the client the maximum number of packets we allow it to send during thesimulation.

The "Interval" Attribute tells the client how long to wait between packets, and the "PacketSize" Attribute tells the client how large its packet payloads should be. With this particular combination of Attributes, we are telling the client to send one 1024-bytepacket.

12. Set application start and stoptime

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (10.0));

First it will run the event at 1.0 seconds, which will enable the echo server application (this event may, in turn, schedule many other events). Then it will run the event scheduled for t=2.0 seconds which will start the echo client application. Again, this event may schedule many more events. The start event implementation in the echo client application will begin the data transfer phase of the simulation by sending a packet to theserver.

13. Runsimulation

Simulator::Run ();

When Simulator::Run is called, the system will begin looking through the list of scheduled events and executing them. Eventually, since we only send one packet (recall the MaxPackets Attribute was set to one), the chain of events triggered by that single client echo request will taper off and the simulation will go idle. Once this happens, the remaining events will be the Stop events for the server and the client. When these events are executed, there are no further events to process and Simulator::Run returns. The simulation is thencomplete.

14. Release resources at end ofsimulation

All that remains is to clean up. This is done by calling the global function Simulator::Destroy. As the helper functions (or low level ns-3 code) executed, they arranged it so that hooks were inserted in the simulator to destroy all of the objects that were created. You did not have to keep track of any of these objects yourself — all you had to do was to call Simulator::Destroy and exit. The ns-3 system took care of the hard part for you. The remaining lines of our first ns-3 script, first.cc, do just that:

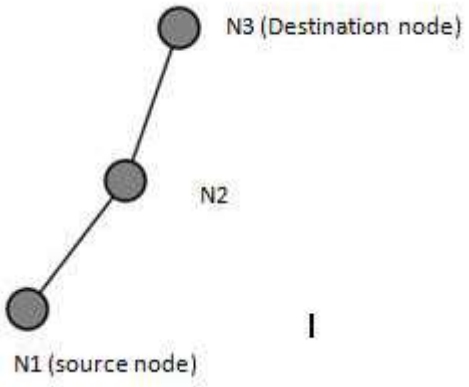
Simulator::Destroy ();

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 packetsdropped.

Overview:

Point-to-point network topology is a simple topology that displays the network of exactly two hosts (computers, servers, switches or routers) connected with a cable. Point-to-point topology is widely used in the computer networking and computer architecture. It is also used in the telecommunications systems when we speak about the communication connection of two nodes or endpoints.

- Add 3 nodes
- Establish link between the nodes using TCP/UDP
- Show the number of packets dropped in pyviz and trace route.



Networktopology

```
      10.1.1.0       10.1.2.0
n0--------------n1.......... n2
point-to-point
```

Inthisprogramwehavecreated3pointtopointnodesn0,n1,n2.Node n0 has IP address 10.1.1.1 and n3 has 10.1.2.2. Node n1 has 2 interfaces(10.1.1.2and10.1.2.1). OnOffHelper application is used to generatethetrafficatsourcenoden0.Packetsmovefromn0ton2via n1. Acknowledgment is sent from n2 to n0 via n1. Details of the flow(Number of packets sent, received and dropped) can be verified by using tracemetrics(lab1.trfile).

Program

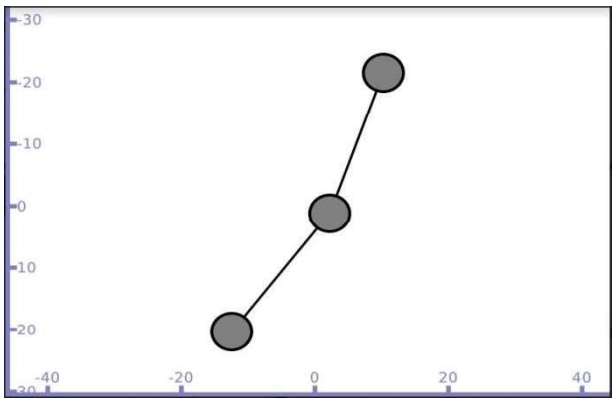
```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include"ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include"ns3/traffic-control-module.h"

using namespace ns3;
```

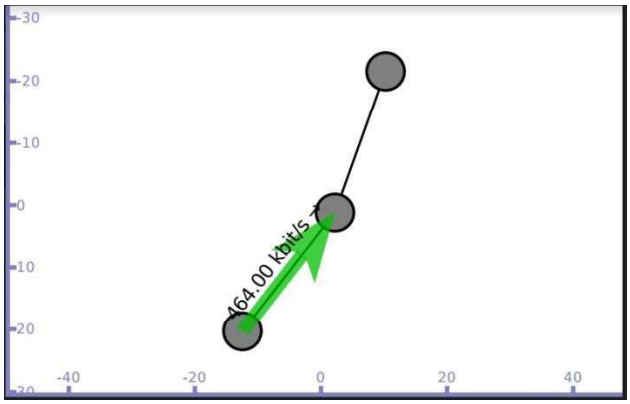
```
NS_LOG_COMPONENT_DEFINE("Lab-Program-1");

int main (int argc, char *argv[])
{
std::string socketType= "ns3::TcpSocketFactory";;

CommandLinecmd;
cmd.Parse (argc, argv);

NodeContainernodes;
nodes.Create(3);           //3 point-to-point nodes arecreated

InternetStackHelperstack;
stack.Install(nodes);     //TCP-IP layer functionality configured on allnodes

//Bandwidth and delay set for the point-to-point channel. Vary these
parametersto   //seethevariationinnumberofpacketssent/received/dropped.
PointToPointHelperp2p1;
p2p1.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
p2p1.SetChannelAttribute ("Delay", StringValue ("1ms"));

//Set the base address for the first network(nodes n0 andn1)
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");

NetDeviceContainerdevices;
devices = p2p1.Install (nodes.Get (0), nodes.Get (1));
Ipv4InterfaceContainerinterfaces=address.Assign(devices);

//Setthebaseaddressforthesecondnetwork(nodesn1andn2) devices =
p2p1.Install (nodes.Get (1), nodes.Get (2)); address.SetBase
("10.1.2.0","255.255.255.0");
interfaces = address.Assign (devices);

//RateErrorModel allows us to introduce errors into a Channel at a given rate.
//Vary the error rate value to see the variation in number of packets dropped
Ptr<RateErrorModel>em=CreateObject<RateErrorModel>();
em->SetAttribute ("ErrorRate", DoubleValue(0.00002));
devices.Get (1)->SetAttribute ("ReceiveErrorModel", PointerValue (em));

//create routing table at all nodes
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

uint32_t payloadSize = 1448;
OnOffHelperonoff(socketType, Ipv4Address::GetAny ());

//Generate traffic by using OnOff application
onoff.SetAttribute("OnTime",   StringValue
("ns3::ConstantRandomVariable[Constant=1]"));
onoff.SetAttribute ("OffTime", StringValue
("ns3::ConstantRandomVariable[Constant=0]"));
onoff.SetAttribute ("PacketSize", UintegerValue (payloadSize));
onoff.SetAttribute("DataRate",StringValue("50Mbps"));//bit/s

uint16_t port = 7;
//Install receiver (for packetsink) on node 2
AddresslocalAddress1(InetSocketAddress(Ipv4Address::GetAny(),port));
PacketSinkHelper packetSinkHelper1 (socketType, localAddress1);
```

```
ApplicationContainersinkApp1=packetSinkHelper1.Install(nodes.Get(2));
sinkApp1.Start (Seconds(0.0));
sinkApp1.Stop (Seconds (10));

//Install sender app on node 0
ApplicationContainerapps;
AddressValueremoteAddress(InetSocketAddress(interfaces.GetAddress(1),
port));
onoff.SetAttribute ("Remote",remoteAddress);
apps.Add (onoff.Install (nodes.Get (0)));
apps.Start (Seconds (1.0));
apps.Stop (Seconds (10));

Simulator::Stop (Seconds(10));

AsciiTraceHelper ascii;
p2p1.EnableAsciiAll (ascii.CreateFileStream ("lab1.tr"));

//Run the simulator
Simulator::Run ();
Simulator::Destroy ();
return 0;
}
```

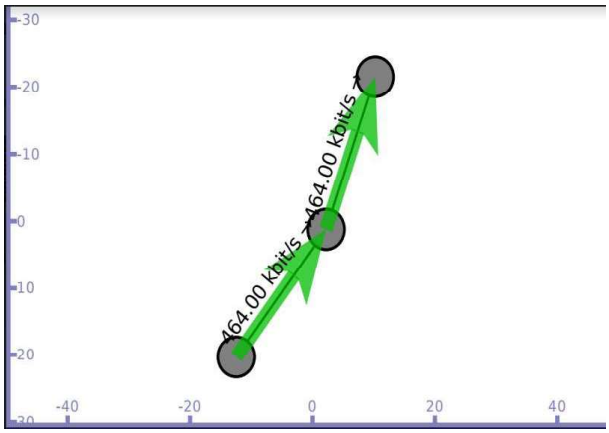
./waf - - run scratch/Program1 - -vis

Output

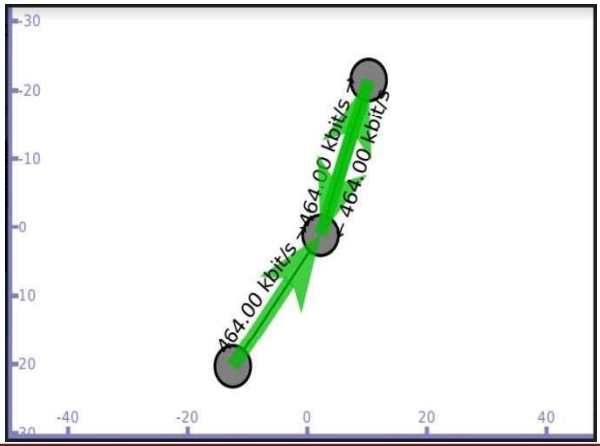




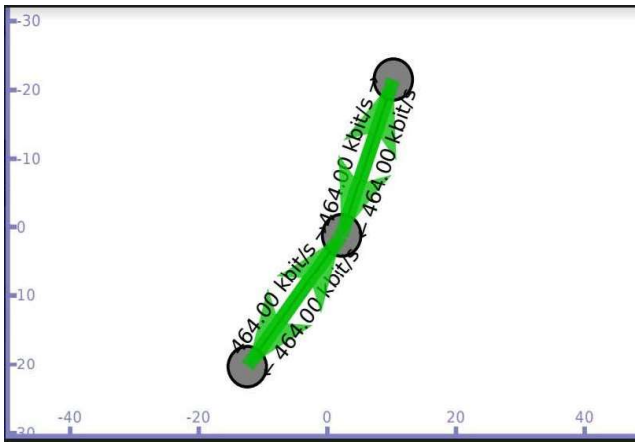
Packetsentfromn0ton1andthenton2

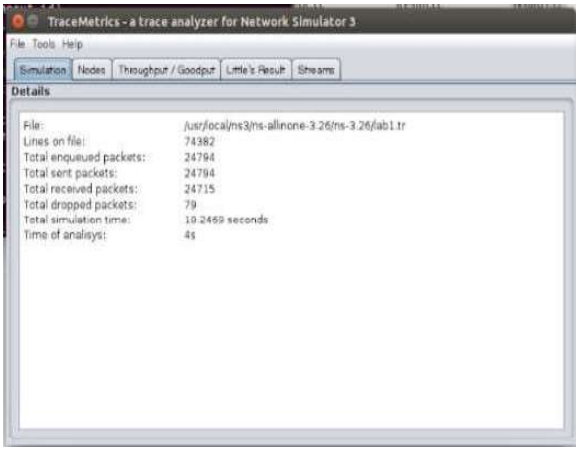


Acknowledgment sent fromn2



Flow details on trace file lab1.tr





2. Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due tocongestion.

Overview:

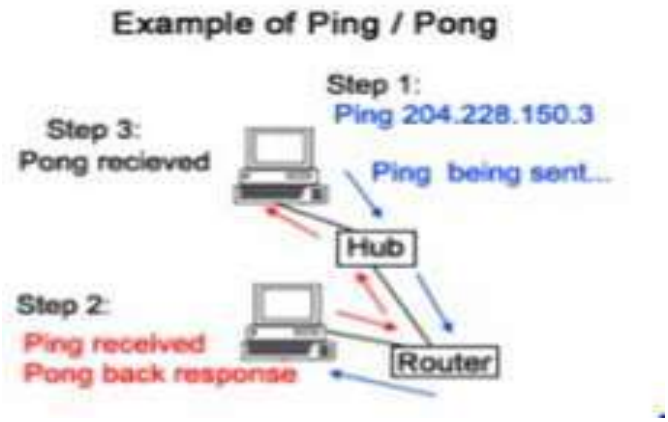
Ping - Packet Internet Groper

What is Ping?

- A computer network utility to determine whether a specific IP address is accessible
- Measures the round-trip time for messages sent from the originating host to a destination computer and back

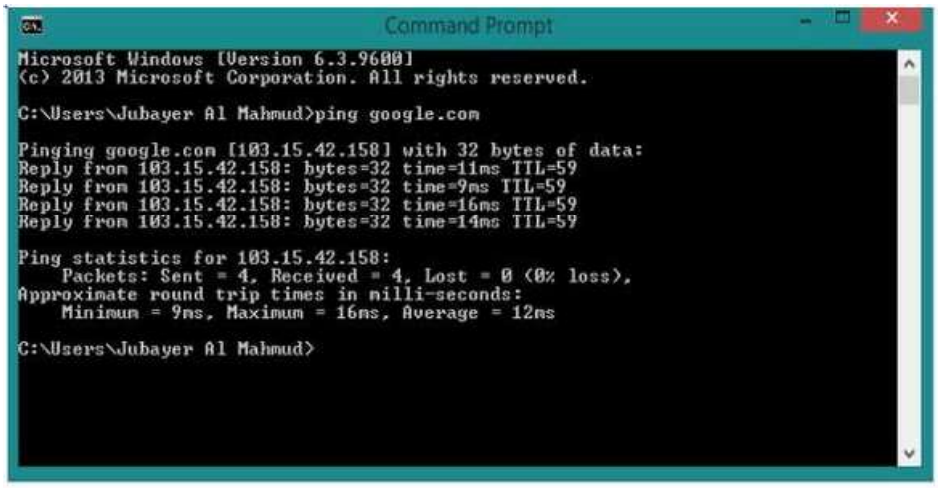
How ping works?

- Operates by sending ICMP echo request (ping) packets to the target host and waiting for an ICMP echo reply (pong)
- It measures the round-trip time from transmission to reception, reporting errors and packet loss

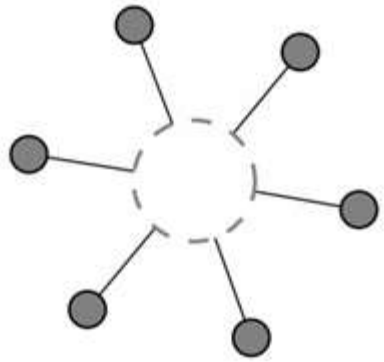


What does Ping test shows?

- a statistical summary of the response packets received
- Also includes minimum, maximum and mean RTT



Topology



Networktopology

```
n0    n1   n2   n3  n4   n5
|     |    |    |   |    |
============================
CSMA channel with base IP 10.1.1.0
```

Inthisprogramwehavecreated6CSMAnodesn0,n1,n2,n3,n4andn5 withIPaddresses10.1.1.1,10.1.1.2,10.1.1.3,10.1.1.4,10.1.1.5and 10.1.1.6 respectively.
Nodes n0 and n1 ping node n2, we can visualize the ping messages transferredbetweenthenodes.Datatransferisalsosimulatedbetween thenodesn0andn2usingUdpSocketFactorytogeneratetraffic.

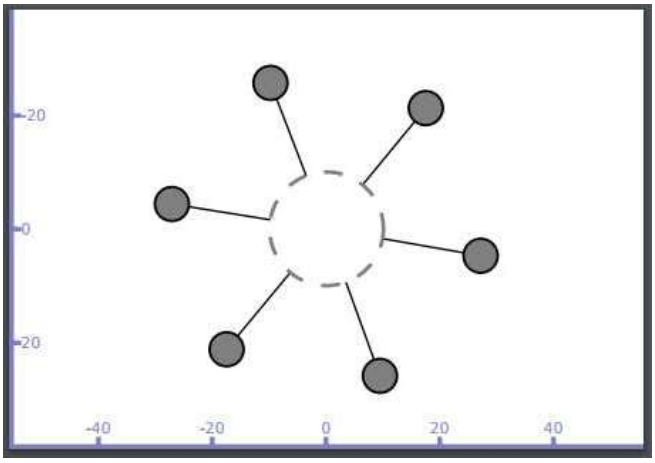
Program

```
#include <iostream>
#include"ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/csma-module.h"
#include "ns3/applications-module.h"
#include"ns3/internet-apps-module.h"
#include"ns3/internet-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE("Lab-Program-2");

static void PingRtt (std::string context, Time rtt)
{
std::cout<< context <<""<<rtt<< std::endl;
}

int main (int argc, char *argv[])
{
CommandLine cmd;
cmd.Parse (argc, argv);

// Here, we will explicitly create six nodes.
NS_LOG_INFO ("Createnodes.");
NodeContainer c;
```

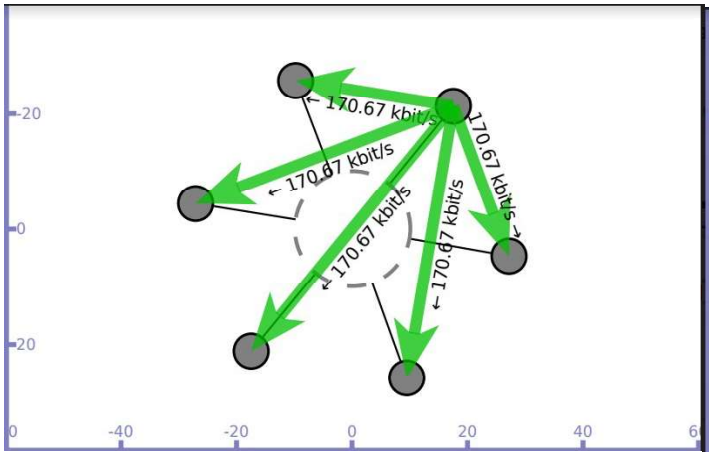
```
c.Create (6);

// connect all our nodes to a shared channel.
NS_LOG_INFO ("BuildTopology.");
CsmaHelpercsma;
csma.SetChannelAttribute("DataRate",DataRateValue(DataRate(10000)));
csma.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (0.2)));
NetDeviceContainerdevs= csma.Install(c);

// add an ip stack to allnodes.
NS_LOG_INFO ("Add ip stack.");
InternetStackHelperipStack;
ipStack.Install (c);

// assign ip addresses

NS_LOG_INFO ("Assign ipaddresses.");
Ipv4AddressHelper ip;
ip.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer addresses = ip.Assign (devs);

NS_LOG_INFO ("Create Sink.");

//CreateanOnOffapplicationtosendUDPdatagramsfromnodezerotonode1. NS_LOG_INFO
("CreateApplications.");
uint16_t port=9;     // Discard port (RFC863)

OnOffHelperonoff("ns3::UdpSocketFactory",
Address (InetSocketAddress (addresses.GetAddress (2), port)));
onoff.SetConstantRate (DataRate ("500Mb/s"));

ApplicationContainerapp = onoff.Install (c.Get (0));
// Start the application
app.Start (Seconds(6.0));
app.Stop (Seconds(10.0));

// Create an optional packet sink to receive these packets
PacketSinkHelpersink ("ns3::UdpSocketFactory",
Address (InetSocketAddress (Ipv4Address::GetAny (), port)));
app = sink.Install (c.Get (2));
app.Start (Seconds (0.0));

NS_LOG_INFO ("Create pinger");
V4PingHelperping=V4PingHelper(addresses.GetAddress(2));
NodeContainerpingers;
pingers.Add (c.Get(0));
pingers.Add (c.Get(1));

ApplicationContainerapps;
apps = ping.Install(pingers);
apps.Start (Seconds (1.0));
apps.Stop (Seconds (5.0));

// finally, print the ping rtts.
Config::Connect ("/NodeList/*/ApplicationList/*/$ns3::V4Ping/Rtt",
MakeCallback (&PingRtt));

NS_LOG_INFO ("RunSimulation.");

AsciiTraceHelperascii;
csma.EnableAsciiAll (ascii.CreateFileStream ("ping1.tr"));
```

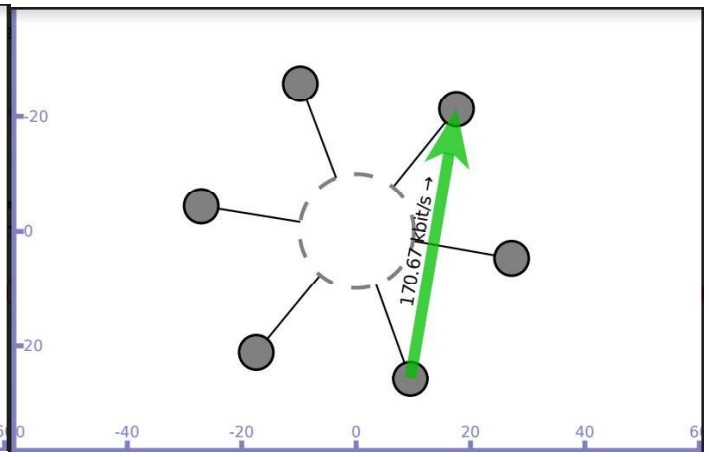
```
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
}
```

./waf - - run scratch/Program2 - -vis

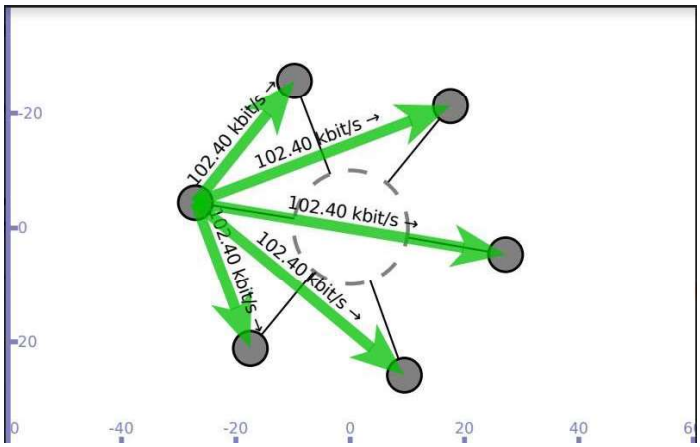


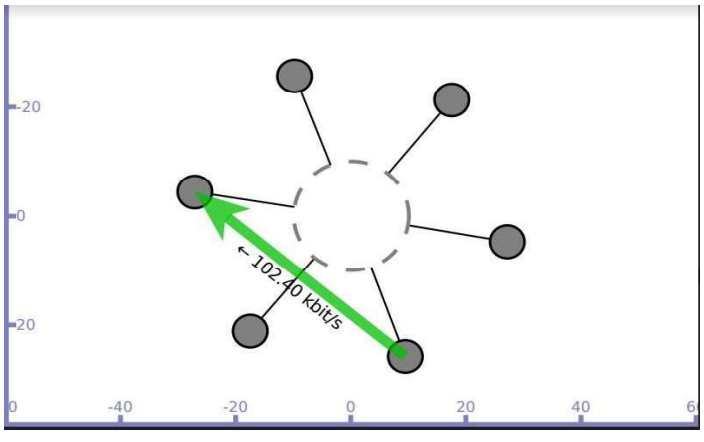
Noden1sendspingmessageton2(Broadcastmessageisgenerated)andonlyn2respondston1



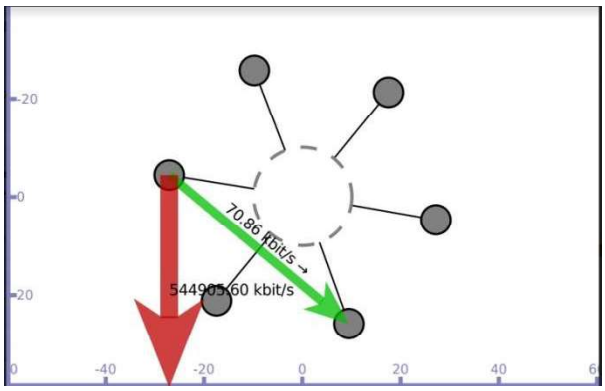


Noden0sendspingmessageton2(Broadcastmessageisgenerated)andonlyn2respondston0

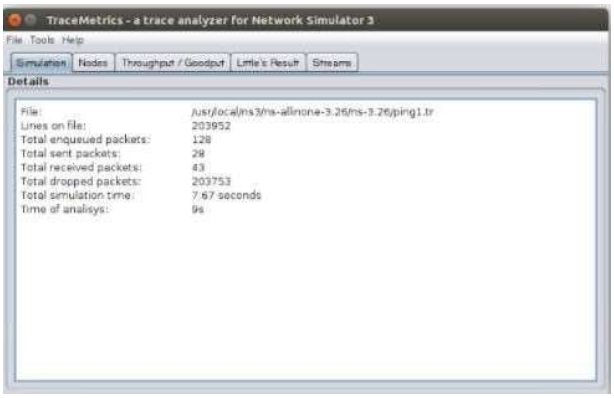




Datatransfersimulatedbetweennodesn0andn2



Trace file(ping1.tr)generated



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

OVERVIEW

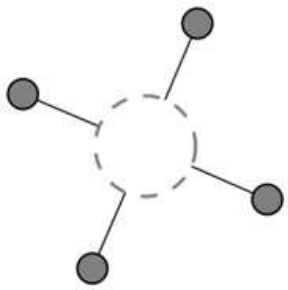
Network congestion in data **networking** is the reduced quality of service that occurs when a **network** node or link is carrying more data than it can handle. Typical effects include queuing delay, packet loss or the blocking of new connections.

TCP Slow Start is part of the congestion control algorithms put in place by TCP to help control the amount of data flowing through to a network. This helps regulate the case where too much data is sent to a network and the network is incapable of processing that amount of data, thus resulting in network congestion.

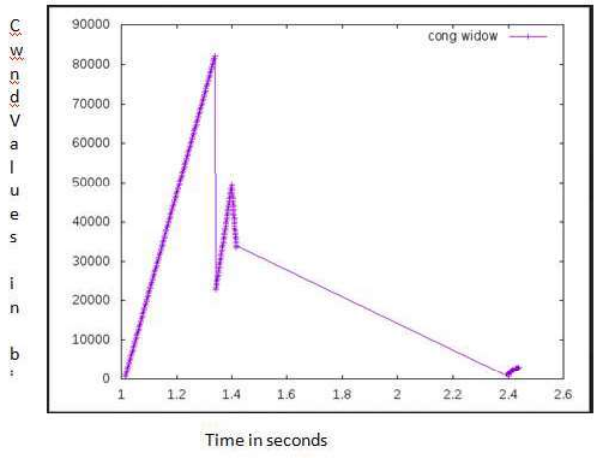
When transmission of data from sender to receiver begins in a network, there may be unknown conditions as to what the network can handle. Slow Start helps to mitigate the pitfalls of this unknown by implementing the following functionality.

1. A sender begins transmissions to a receiver by slowly probing the network with a packet that contains its initial congestion window (cwnd).
2. The client receives the packet and replies with its maximum buffer size, also known as the receiver's advertised window (rwnd).
3. If the sender receives an acknowledgement (ACK) from the client, it then doubles the amount of packets to send to the client.
4. Step 3 is repeated until the sender no longer receives ACK from the receiver which means either congestion is detected, or the client's window limit has been reached.

Topology



Congestion graph



Networktopology

```
n0    n1   n2   n3
|     |    |    |
====================== CSMA channel with base IP10.1.1.0
                         Source node-n0      sink node -n1
```

Inthisprogramwehavecreated4CSMAnodesn0,n1,n2andn3withIP addresses10.1.1.1,10.1.1.2,10.1.1.3and10.1.1.4respectively.Data transmissionissimulatedbetweennodesn0andn1.Oncethecwndvalues are generated, they are exported to .dat file and congestion graph is plot usinggnuplot.

Program

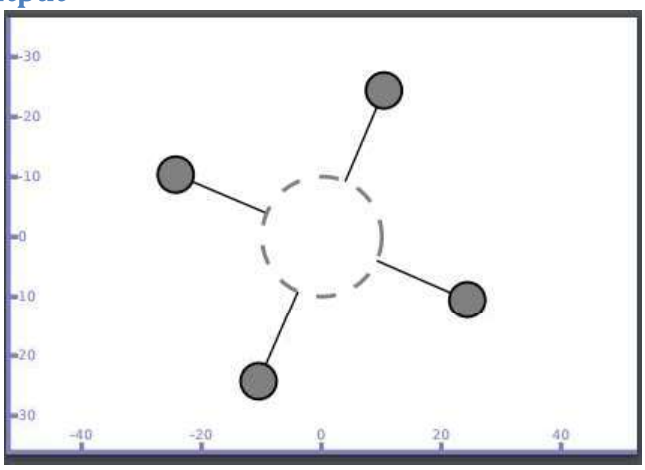
```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include"ns3/applications-module.h"
#include <iostream>
#include "ns3/csma-module.h"
#include "ns3/network-application-helper.h"

using namespacens3;

NS_LOG_COMPONENT_DEFINE ("3rd LabProgram");

using namespacens3;

NS_LOG_COMPONENT_DEFINE ("3rd Lab Program");

int
main (int argc, char *argv[])
{
  CommandLinecmd;
  cmd.Parse (argc,argv);

  NS_LOG_INFO ("Createnodes.");
  NodeContainernodes;
  nodes.Create (4); //4 csma nodes are created

  CsmaHelpercsma;
  csma.SetChannelAttribute ("DataRate", StringValue ("5Mbps"));
  csma.SetChannelAttribute("Delay",TimeValue(MilliSeconds(0.0001)));

  NetDeviceContainer devices;
  devices = csma.Install(nodes);

//RateErrorModel allows us to introduce errors into a Channel at a given rate.

  Ptr<RateErrorModel>em = CreateObject<RateErrorModel> ();
```

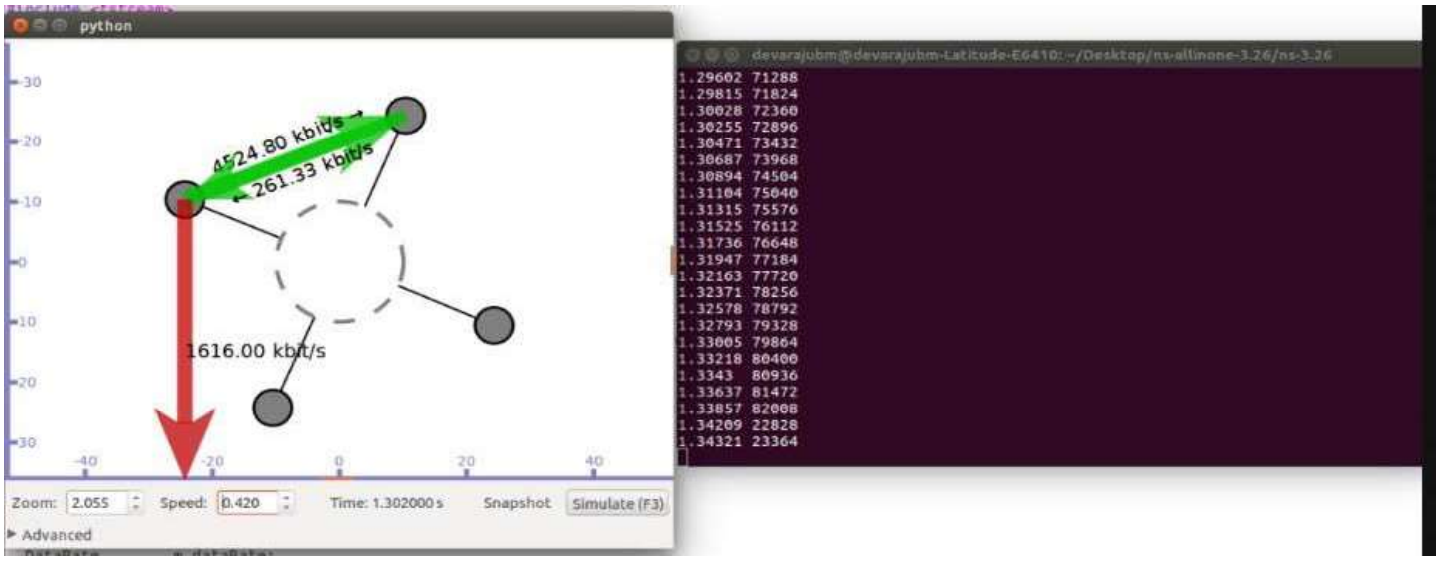
```
  em->SetAttribute ("ErrorRate", DoubleValue (0.00001));
  devices.Get (1)->SetAttribute ("ReceiveErrorModel", PointerValue (em));

  InternetStackHelperstack;
  stack.Install (nodes);

  Ipv4AddressHelper address;
  address.SetBase ("10.1.1.0", "255.255.255.0");
  Ipv4InterfaceContainerinterfaces=address.Assign(devices);

  uint16_t sinkPort =8080;

  AddresssinkAddress(InetSocketAddress(interfaces.GetAddress(1),
sinkPort));

//PacketSink Application is used on the destination node to receiveTCP
//connections and data. Creates objects in an abstract way and associates
//type-id along with the object.

  PacketSinkHelperpacketSinkHelper("ns3::TcpSocketFactory",
InetSocketAddress (Ipv4Address::GetAny (), sinkPort));

//Install sinkapp on node 1

  ApplicationContainersinkApps=packetSinkHelper.Install(nodes.Get(1));
  sinkApps.Start (Seconds(0.));
  sinkApps.Stop (Seconds(20.));

//next two lines of code will create the socket and connect the trace source.

  Ptr<Socket>ns3TcpSocket=Socket::CreateSocket(nodes.Get(0),
TcpSocketFactory::GetTypeId());
  ns3TcpSocket->TraceConnectWithoutContext("CongestionWindow",MakeCallback
(&CwndChange));

//creates an Object of type NetworkApplication (Class present innetwork-
//application-helper.h)

  Ptr<NetworkApplication> app = CreateObject<NetworkApplication> ();

//Next statement tells the Application what Socket to use, what address to
//connect to, how much data to send at each send event, how many send events
//to generate and the rate at which to produce data from thoseevents.

  app->Setup(ns3TcpSocket,sinkAddress,1040,1000,DataRate("50Mbps"));
  nodes.Get (0)->AddApplication(app);
  app->SetStartTime (Seconds(1.));
  app->SetStopTime (Seconds(20.));

//Displays packet drop msg

  devices.Get (1)->TraceConnectWithoutContext("PhyRxDrop", MakeCallback
(&RxDrop));

  AsciiTraceHelperascii;
  csma.EnableAsciiAll(ascii.CreateFileStream("3lan.tr"));
  csma.EnablePcapAll (std::string ("3lan"), true);
```

```
  Simulator::Stop (Seconds(20));
  Simulator::Run ();
  Simulator::Destroy ();

  return 0;
}
```

./waf - - run scratch/Program3 - -vis

Output





Redirect the output to a file called cwnd.dat

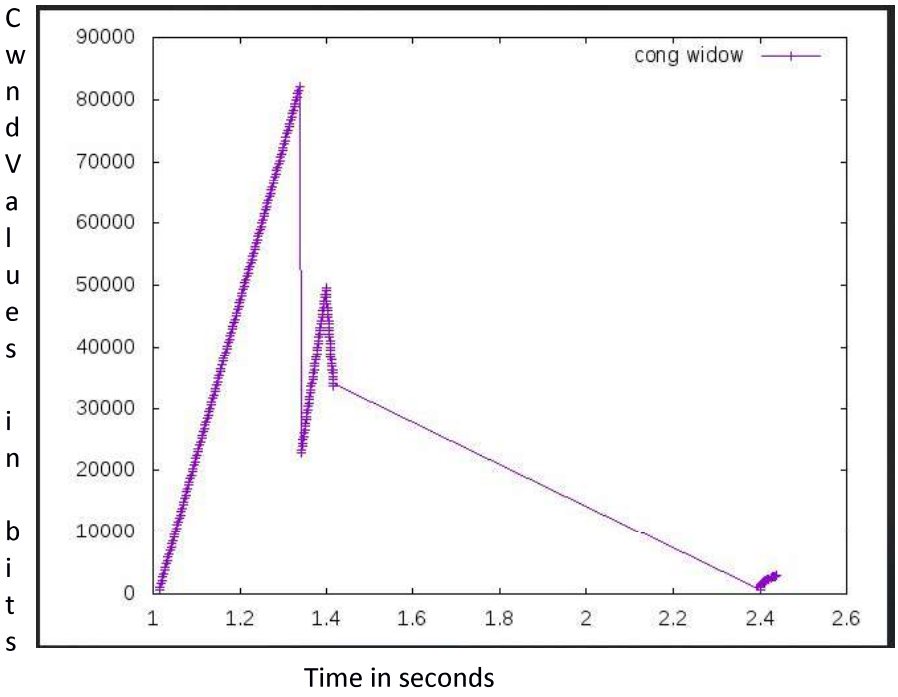
./waf --run scratch/Program3 > cwnd.dat 2>&1

Now run gnuplot

gnuplot> set terminal png size 640,480

gnuplot> set output "cwnd.png"

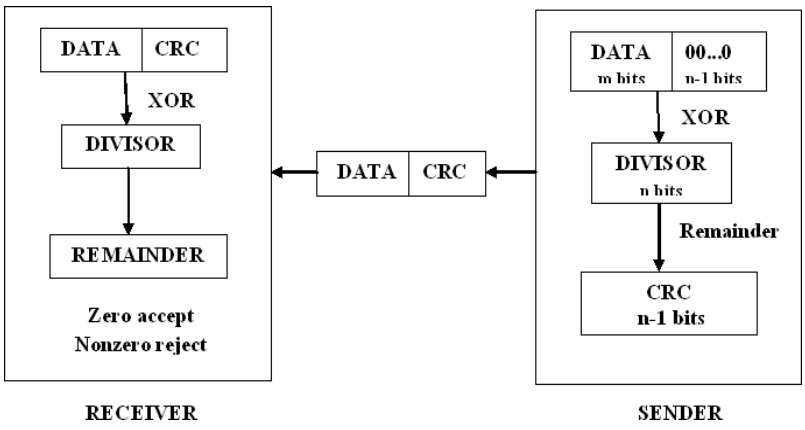
gnuplot> plot "cwnd.dat" using 1:2 title 'Congestion Window' with linespointsgnuplot> exit


Cwnd Values in bits
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
cong widow
Time in seconds

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

The cyclic redundancy check (CRC) is a technique used to detect errors in digital data. CRC is a hash function that detects accidental changes to raw computer data commonly used in digital telecommunications networks and storage devices such as hard disk drives. This technique was invented by W. Wesley Peterson in 1961 and further developed by the CCITT (International Telegraph and Telephone Consultative Committee). Cyclic redundancy checks are quite simple to implement in hardware and can be easily analyzed mathematically. It is one of the better techniques in detecting common transmission errors.

As explained above cyclic redundancy check is also applied to storage devices like hard disks. In this case, check bits are allocated to each block in the hard disk. When a corrupt or incomplete file is read by the computer, the cyclic redundancy error is reported. This could be from another storage device or from CD/DVDs. The common reasons for errors include system crashes, incomplete or corrupt files, or files with lots of bugs.



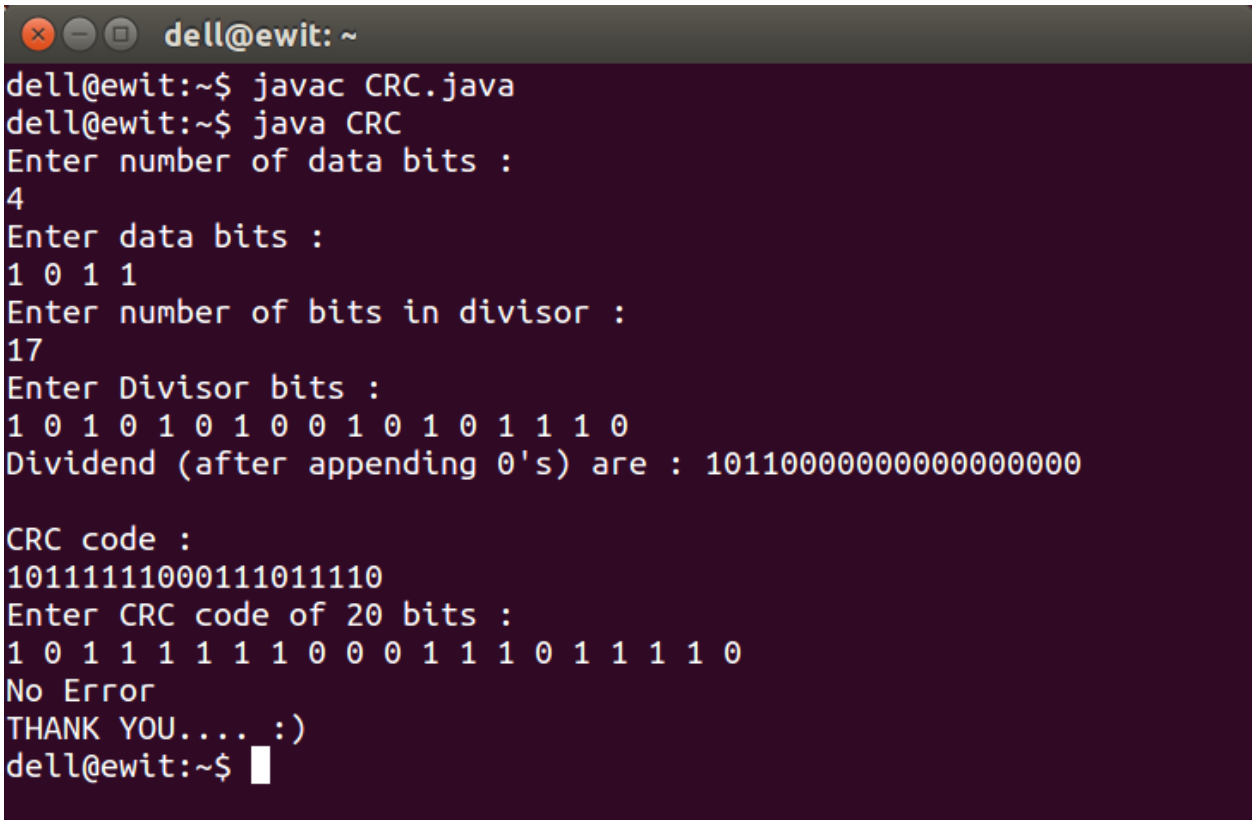
Steps:

- First, a string ofn-1 0sisappendedtothe data unit.
- Thenumberof0sisonelessthanthenumberofbitsinthedivisor which is n bits.
- Then the newly elongated data unit is divided by the divisor using a process called binary division (XOR).
- TheremainderisCRC.TheCRCreplacestheappended0sattheend of the data unit.
- The data unit arrives at the receiver first, followed by the CRC.
- The receiver treats whole string as the data unit and divides it by the same divisor that was used to find the CRC remainder.
- If the remainder is 0 then the data unit is error free. Otherwise it having some error and it must be discarded.

```
import java.io.*;
import java.util.*;
class CRC
{
	public static void main(String a[]) throws IOException
	{
		Scanner sc=new Scanner(System.in);
		int[] message;
		int[] gen;
		int[] app_message;
		int[] rem;
		int[] trans_message;
		int message_bits,gen_bits,total_bits;

		System.out.println("Enter no bits in mwssage:");
		message_bits=sc.nextInt();

		message=new int [message_bits];
		System.out.println("\nEnter message bits:");
		for(int i=0; i<message_bits;i++)
			message[i]= sc.nextInt();

		System.out.println("\nEnter number of bits in gen:");
		gen_bits= sc.nextInt();

		gen=new int[gen_bits];
		System.out.println("\nEnter gen bits:");
		for(int i=0;i<gen_bits;i++)
			gen[i]= sc.nextInt();

		total_bits=message_bits+gen_bits-1;

		app_message=new int[total_bits];
		rem=new int[total_bits];
		trans_message=new int[total_bits];

		for(int i=0;i<message.length;i++)
			app_message[i]=message[i];

		System.out.println("\nMessage bits are:");
		for(int i=0;i<message_bits;i++)
			System.out.print("\t"+message[i]);

		System.out.println("\nGenerators bits are:");
		for(int i=0;i<gen_bits;i++)
			System.out.print("\t"+gen[i]);
```

```
		System.out.println("\nAppended message is:");
		for(int i=0;i<app_message.length;i++)
			System.out.print("\t"+app_message[i]);

		for(int j=0;j<app_message.length;j++)
			rem[j]=app_message[j];

		rem=computecrc(app_message,gen,rem);

		for(int i=0;i<app_message.length;i++)
			trans_message[i]=(app_message[i]^rem[i]);

		System.out.println("\nTransmitted message from the transmitter is:");
		for(int i=0;i<trans_message.length;i++)
			System.out.print("\t"+trans_message[i]);

		System.out.println("\nEnter received message of"+total_bits+"bits at receiver end:");
		for(int i=0;i<trans_message.length;i++)
			trans_message[i]= sc.nextInt();

		System.out.println("\nReceived message is:");
		for(int i=0;i<trans_message.length;i++)
			System.out.print("\t"+trans_message[i]);

		for(int j=0;j<trans_message.length;j++)
			rem[j]=trans_message[j];

		rem=computecrc(trans_message,gen,rem);
		for(int i=0;i<rem.length;i++)
		{
			if(rem[i]!=0)
			{
				System.out.println("\nThere is error in the received message");
				break;
			}
			if(i==rem.length-1)
				System.out.println("\nThere is no erron in the received mesage!!");
		}
	}
	static int[] computecrc(int app_message[],int gen[], int rem[])
	{
		int current=0;
		while(true)
		{
			for(int i=0;i<gen.length;i++)
				rem[current+i]=(rem[current+i]^gen[i]);
			while(rem[current]==0 && current!=rem.length-1)
```

```
                    current++;
                if((rem.length-current)<gen.length)
                    break;
            }
            return rem;
        }
}
```



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

The sliding window protocol is a well-known technique that plays a significant role in ensuring reliable and orderly data exchange between a sender and a receiver. In this section, we will delve into the concept of the sliding window protocol and demonstrate how to implement it using Java.

Understanding the Sliding Window Protocol

The sliding window protocol is a communication protocol used to manage the flow control and reliability of data transmission over a network. It allows the sender to transmit a specified number of packets, known as the window size, without waiting for an acknowledgment from the receiver for each packet. This approach enhances efficiency by minimizing the communication overhead.

The protocol employs two primary components: the sender's sliding window and the receiver's sliding window. The sender's window keeps track of the packets that have been sent but not yet acknowledged, while the receiver's window tracks the expected sequence of packets to receive. As acknowledgments are received, both windows slide forward, allowing for the continuous flow of data.

Implementation of the Sliding Window Protocol

To implement the sliding window protocol in Java, we will create a simplified example of a sender and a receiver using sockets for communication. We will assume a reliable connection, so the focus will be on the sliding window mechanism.

Server Program

```
import java.io.*;
import java.net.*;
import java.util.Random;
public class SlidingWindowServer {
private static final int PORT = 12345;
private static final int WINDOW_SIZE = 4;
private static final double PACKET_LOSS_RATE = 0.1; // 10% packet loss
public static void main(String[] args) {
try (ServerSocket serverSocket = new ServerSocket(PORT)) {
System.out.println("Server is running and waiting for connections...");
try (Socket clientSocket = serverSocket.accept();
BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true)) {
System.out.println("Client connected.");
String line;
int base = 0;
while ((line = in.readLine()) != null) {
if (simulatePacketLoss()) {
System.out.println("Packet " + line + " lost during transmission.");
} else {
System.out.println("Received packet: " + line);
// Simulate acknowledgment
if (base < WINDOW_SIZE) {
out.println("ACK: " + base);
base++;
}
}
}
```

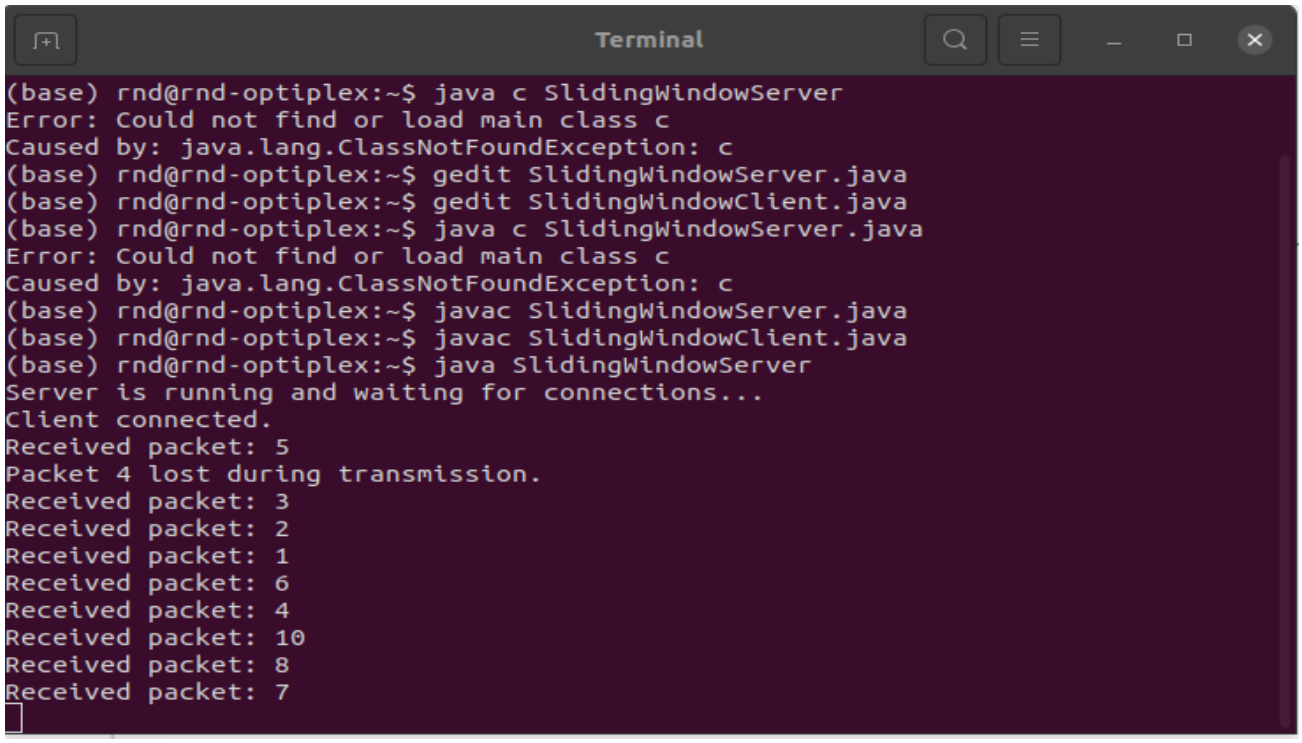
```
}
} catch (IOException e) {
e.printStackTrace();
}
}
private static boolean simulatePacketLoss() {
Random random = new Random();
return random.nextDouble() < PACKET_LOSS_RATE;
}
}
```

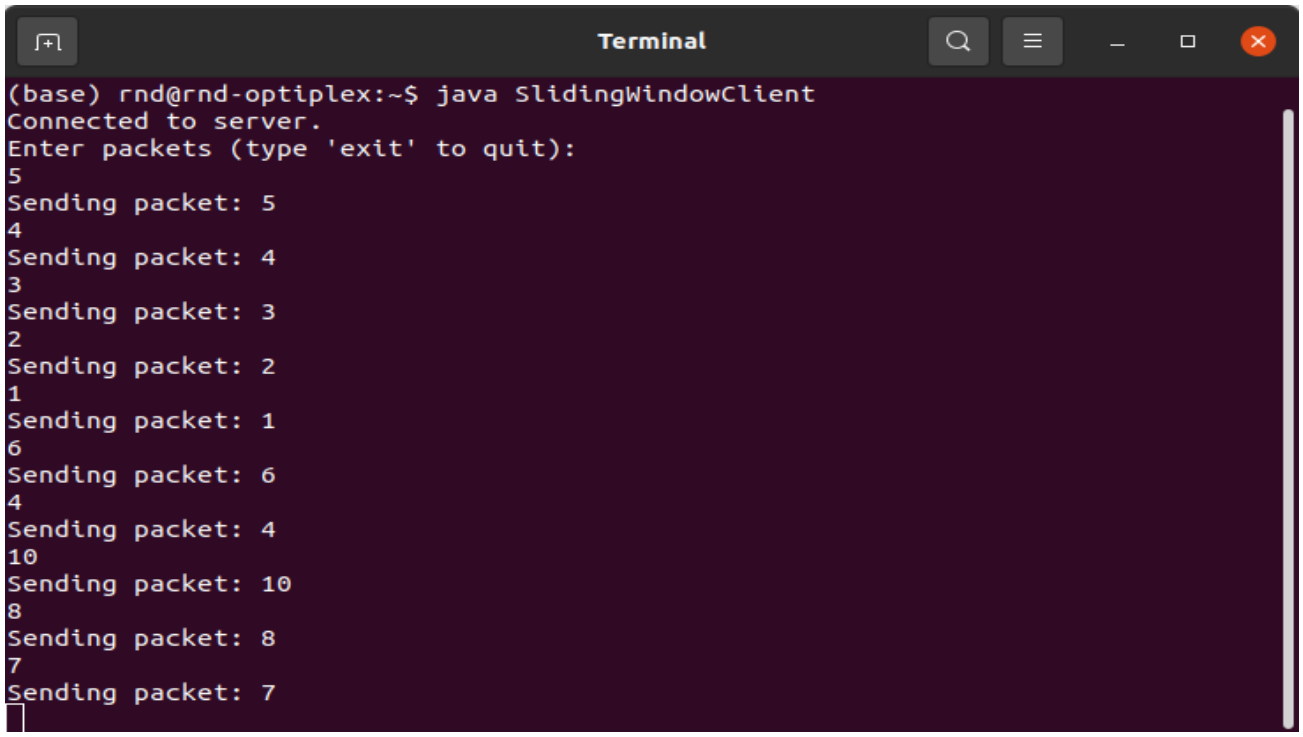
Client Program

```
import java.io.*;
import java.net.*;
import java.util.Scanner;
import java.util.Random;
public class SlidingWindowClient {
private static final String SERVER_ADDRESS = "localhost";
private static final int PORT = 12345;
private static final int WINDOW_SIZE = 4;
private static final double PACKET_LOSS_RATE = 0.1; // 10% packet loss
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
try (Socket socket = new Socket(SERVER_ADDRESS, PORT);
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()))) {
System.out.println("Connected to server.");
System.out.println("Enter packets (type 'exit' to quit):");
int base = 0;
```

```
while (true) {
if (base >= WINDOW_SIZE) {
// Wait for acknowledgment
String ack = in.readLine();
if (ack != null) {
System.out.println("Received " + ack);
base++;
}
}
String packet = scanner.nextLine();
if (packet.equalsIgnoreCase("exit")) {
break;
}
if (simulatePacketLoss()) {
System.out.println("Packet " + packet + " lost during transmission.");
} else {
System.out.println("Sending packet: " + packet);
out.println(packet);
}
}
} catch (IOException e) {
e.printStackTrace();
}
}
private static boolean simulatePacketLoss() {
Random random = new Random();
return random.nextDouble() < PACKET_LOSS_RATE;
}
}
```

Snapshot:


Terminal
(base) rnd@rnd-optiplex:~$ java c SlidingWindowServer
Error: Could not find or load main class c
Caused by: java.lang.ClassNotFoundException: c
(base) rnd@rnd-optiplex:~$ gedit SlidingWindowServer.java
(base) rnd@rnd-optiplex:~$ gedit SlidingWindowClient.java
(base) rnd@rnd-optiplex:~$ java c SlidingWindowServer.java
Error: Could not find or load main class c
Caused by: java.lang.ClassNotFoundException: c
(base) rnd@rnd-optiplex:~$ javac SlidingWindowServer.java
(base) rnd@rnd-optiplex:~$ javac SlidingWindowClient.java
(base) rnd@rnd-optiplex:~$ java SlidingWindowServer
Server is running and waiting for connections...
Client connected.
Received packet: 5
Packet 4 lost during transmission.
Received packet: 3
Received packet: 2
Received packet: 1
Received packet: 6
Received packet: 4
Received packet: 10
Received packet: 8
Received packet: 7


Terminal
(base) rnd@rnd-optiplex:~$ java SlidingWindowClient
Connected to server.
Enter packets (type 'exit' to quit):
5
Sending packet: 5
4
Sending packet: 4
3
Sending packet: 3
2
Sending packet: 2
1
Sending packet: 1
6
Sending packet: 6
4
Sending packet: 4
10
Sending packet: 10
8
Sending packet: 8
7
Sending packet: 7

6. Develop a program to find the shortest path between vertices using bellman-ford and path vector routing algorithm.

Steps:

Input : Graph and a source vertex *src* .

Output: Shortest distance to all vertices from *src*. If there is a negative weight cycle, then shortest distances are not calculated, negative weight cycle is reported.

1. This step initializes distances from source to all vertices as in finite and distance to source itself as 0. Create an array dist[] of size |V| with all values as infinite except dist[src] where src is source vertex.

2. This step calculates shortest distances. Do following |V|-1 times where |V| is the number of vertices in given graph.

a) Do following for each edge u-v.

If dist[v]>dist[u]+weigh to f edge uv, then update
list[v]. dist[v] = dist[u] + weight of edge uv

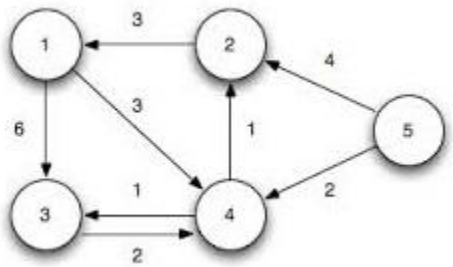
3. Thisstepreportsifthereisanegativeweightcycleingraph.Dofollowingfor each edge u-v.

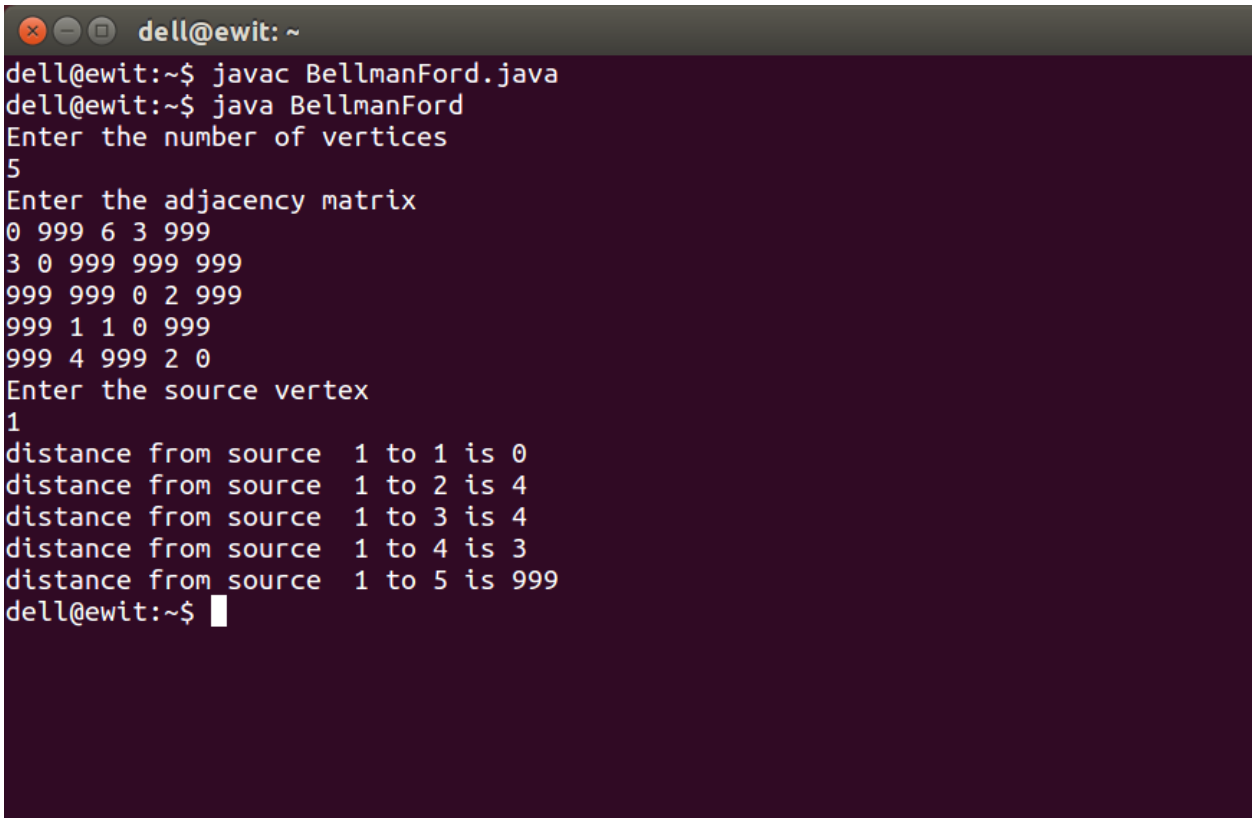
If dist[v] >dist[u] + weight of edge uv, then
"Graph contains negative weight cycle".

The idea of step 3 is, step 2 guarantees shortest distances if graph doesn't contain negative weight cycle. If we iterate through all edges one more time and get a shorter path for any vertex, then there is a negative weight cycle.

```
import java.util.Scanner;
public class BellmanFord
{
	private int D[];
	private int NoV;
	public static final int MAX_VALUE = 999;

	public BellmanFord(int NoV)
	{
		this.NoV = NoV;
		D = new int[NoV + 1];
	}
	public void BellmanFordEvaluation(int source, int A[][])
	{
		for (int node = 1; node <= NoV; node++)
		{
			D[node] = MAX_VALUE;
		}
		D[source] = 0;
		for (int node = 1; node <= NoV - 1; node++)
		{
			for (int i = 1; i <= NoV; i++)
			{
				for (int j = 1; j <= NoV; 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 <= NoV; i++)
		{
			for (int j = 1; j <= NoV; j++)
			{
				if (A[i][j] != MAX_VALUE)
				{
					if (D[j] > D[i]+ A[i][j])
					{
						System.out.println("The Graph contains negative
egde cycle");
						return;
					}
```

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

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

                System.out.println("Enter the number of vertices");
                NoV = scanner.nextInt();
                int A[][] = new int[NoV + 1][NoV + 1];

                System.out.println("Enter the adjacency matrix");
                for (int i = 1; i <= NoV; i++)
                {
                        for (int j = 1; j <= NoV; j++)
                        {
                                A[i][j] = scanner.nextInt();
                        }
                }

                System.out.println("Enter the source vertex");
                source = scanner.nextInt();
                BellmanFord bellmanford = new BellmanFord(NoV);
                bellmanford.BellmanFordEvaluation(source, A);
                scanner.close();
        }
}
```




dell@ewit: ~
dell@ewit:~$ javac BellmanFord.java
dell@ewit:~$ java BellmanFord
Enter the number of vertices
5
Enter the adjacency matrix
0 999 6 3 999
3 0 999 999 999
999 999 0 2 999
999 1 1 0 999
999 4 999 2 0
Enter the source vertex
1
distance from source 1 to 1 is 0
distance from source 1 to 2 is 4
distance from source 1 to 3 is 4
distance from source 1 to 4 is 3
distance from source 1 to 5 is 999
dell@ewit:~$

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.

Methods and description

α. **Public Server Socket(int port)throws IOException:** Attempts to create a server socket bound to the specified port. An exception occurs if the port is already bound by another application.

β. **public Socket accept() throws IOException:** Waits for an incoming client. This method blocks until either a client connects to the server on the specified port or the socket times out, assuming that the time-out value has been set using the set So Timeout() method. Otherwise, this method blocks indefinitely.

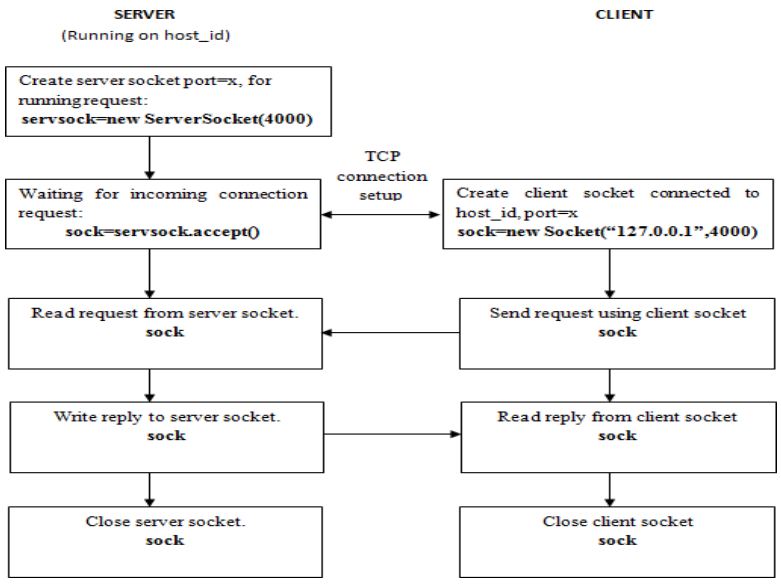
χ. **public Socket(Inet Address host, int port) throws IOException:** This method attempts to connect to the specified server at the specified port. If this constructor does not throw an exception, the connection is successful and the client is connected to the server.

δ. **Public Inet Address get Inet Address()**: This method returns the address of the other computer that this socket is connected to.

ε. **Public int getPort()**:Returns the port the socket is bound to on there mote machine.

ϕ. **Public InputStreamgetInputStream() throws IOException**: Returns the input stream of the socket. The input stream is connected to the output stream of the remote socket.

γ. **Public OutputStreamgetOutputStream() throws IOException**: Returns the output stream of the socket. The output stream is connected to the input stream of the remote socket.



// TCP Server

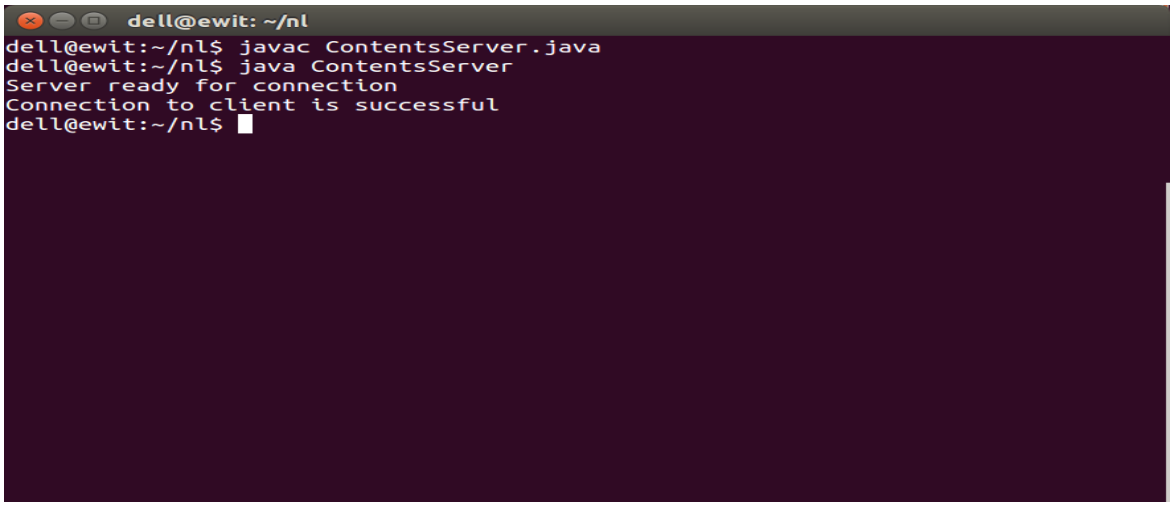
```
import java.net.*;
import java.io.*;
public class TCPS
{
public static void main(String[] args) throws Exception
{
ServerSocket sersock=new ServerSocket(4000);
System.out.println("Server ready for connection");
Socket sock=sersock.accept();
System.out.println("Connection Is successful and waiting 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();
}
}
```

//TCP Client

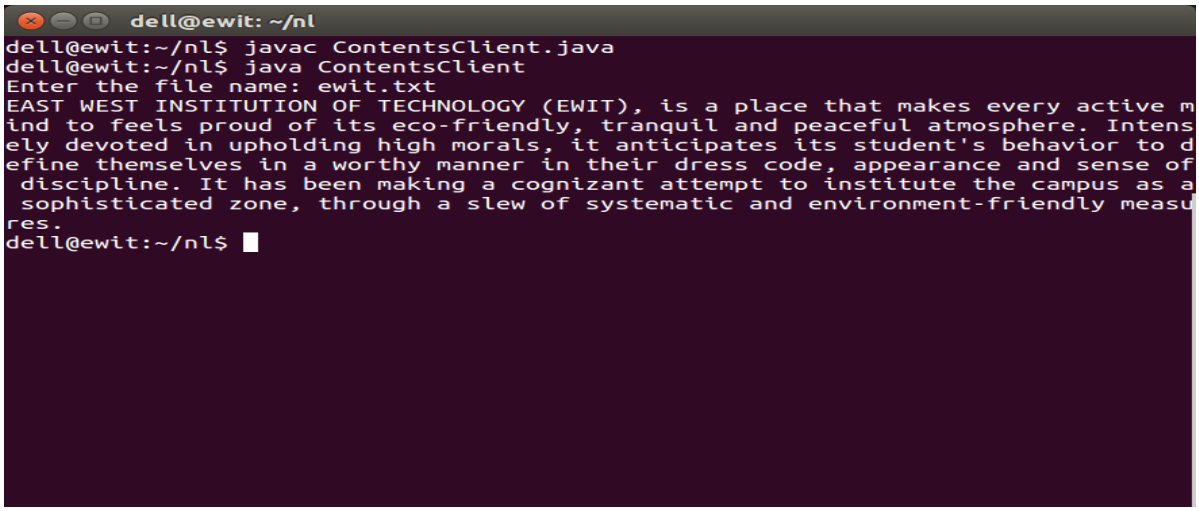
```
import java.net.*;
import java.io.*;
public class TCPC
{
public static void main(String[] args) throws Exception
{
Socket sock=new Socket("127.0.01",4000);
System.out.println("Enter the filename");
BufferedReader keyRead=new BufferedReader(new InputStreamReader(System.in));
String fname=keyRead.readLine();
```

```
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();
}
}
```

TCPServer:



TCPClient:



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

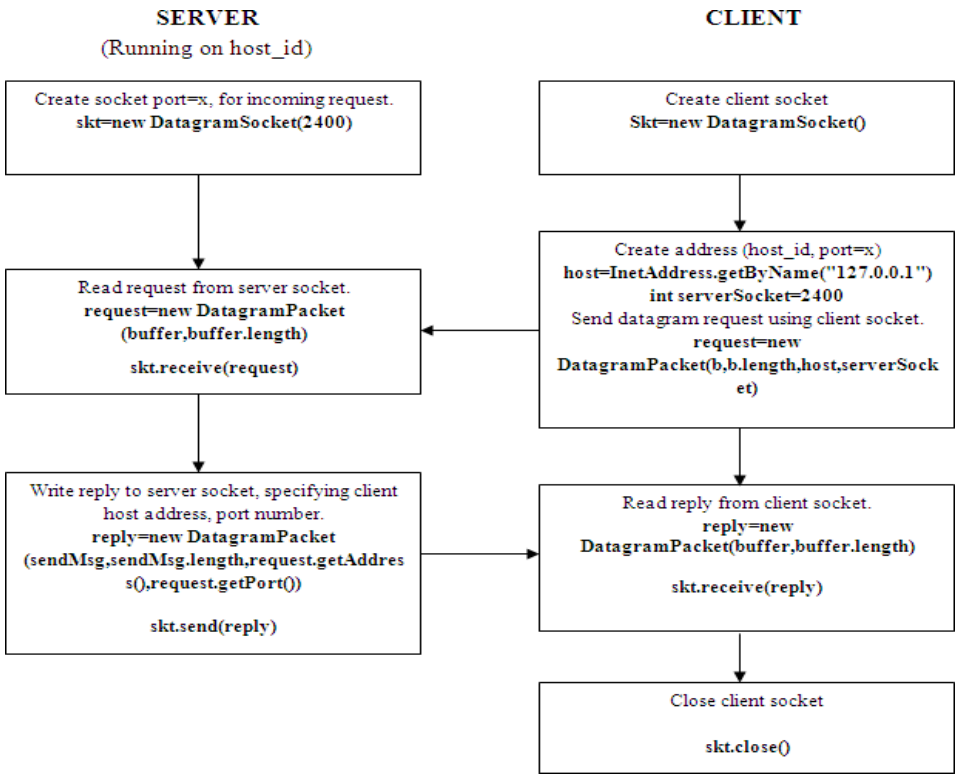


Fig 8.1:UDP client/server communication flow.

Methods and description

α. **DatagramSocket(intport) throws SocketEeption:** it creates a datagram socket and binds it with the given Port Number.

β. **DatagramPacket(byte[]buffer,intlength):** it creates a datagram packet. This constructor is used to receive the packets.

χ. **DatagramPacket(byte[] buffer, int length, InetAddress address, int port):**it creates a datagram packet. This constructor is used to send the packets.

UDP Server Source code:

```
import java.io.*;
importjava.net.*;
importjava.util.*;
 public class udps
{
publicstaticvoid main(String[]args)
{
DatagramSocketskt=null;
Scannersc=newScanner(System.in); try
{
skt=newDatagramSocket(2400
); byte[] buffer=new
byte[1000]; while(true)
{
DatagramPacketrequest=newDatagramPacket(buffer,buffer.length);
skt.receive(request);
Stringmessage=sc.nextLine();
byte[]sendMsg=message.getByte
s(); DatagramPacket reply=new
DatagramPacket(sendMsg,sendMsg.length,request.getAddress(),request.getPort());
skt.send(reply);
}
}
catch(Exceptionex)
{
}
}
}
```

UDP Client Source code:

```
import java.io.*;
importjava.net.*;
publicclassudpc
{
        publicstaticvoid main(String[]args)
        {
DatagramSocketskt;
try
    {
        skt=new DatagramSocket();
          Stringmsg="textmessage";
        byte[]b=msg.getBytes();
        InetAddresshost=InetAddress.getByName("127.0.0.1");
        intserverSocket=2400;
        DatagramPacketrequest=newDatagramPacket(b,b.length,host,serverSocket);
        skt.send(request);
        byte[]buffer=newbyte[1000];
        DatagramPacketreply=newDatagramPacket(buffer,buffer.length);
skt.receive(reply);
System.out.println("clientrecieved:"+newString(reply.getData()));
skt.close();
    }
catch(Exceptionex){}
}
}
```

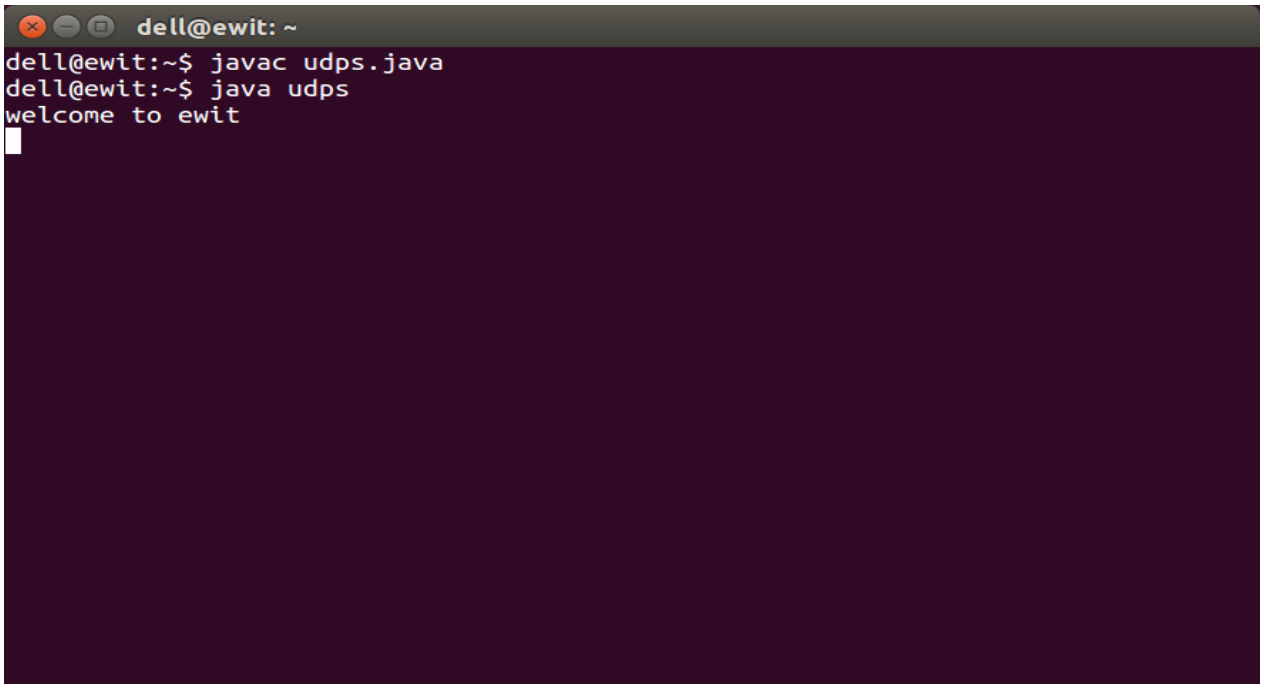


Fig 8.2: UDP server

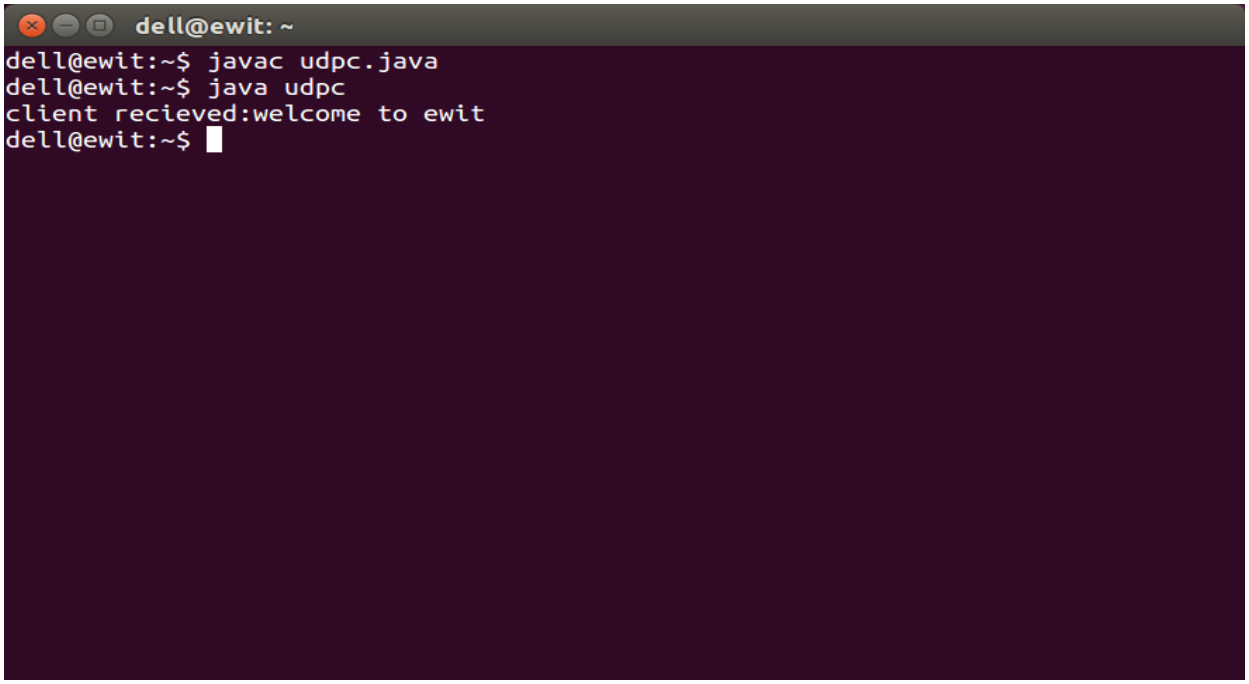


Fig 8.2: UDP Client

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

RSA is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. Asymmetric means that there are two different keys (one is public key and another is private key). This is also called public key cryptography, because one of the m 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.

Following are the steps of RSA algorithm.

Key Generation

1. Generate two large prime numbers **p** and **q**, such that **p!=q**.
2. Let **n=p*q.**
3. Let **t=(p-1)*(q-1)**
4. Choose a small numbere ,co-prime tot, with
   GCD(t,e)=1and **1<e<t.**
5. Find d, such that
   d*emodt=1. Publish **e** and
   n as **public key**. Keep **d**
   and **t** as **secret key**.

Encryption

Cipher=(Message)^{e}modn

Decryption

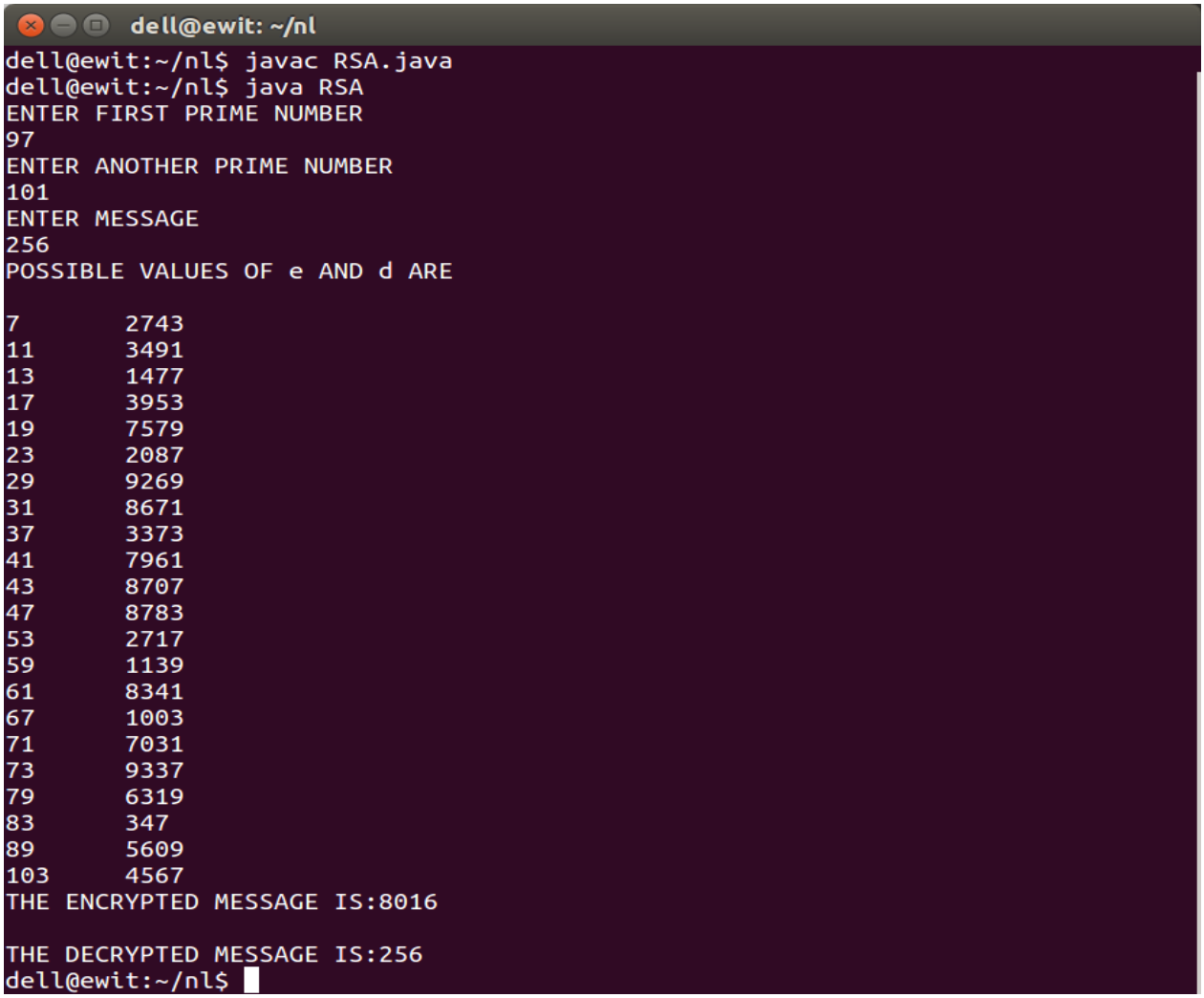
Message=(Cipher)^{d}modn

```
importjava.util.Scanner;
public class RSA
{
        publicstaticintp,q,n,t,flag,msg,m,temp;
        public staticint e[]=new int[100];
         public staticint d[]=new int[100];
        public static int prime( intpr)
        {
                inti;

                Doublea=(Math.sqrt(p
                r)); m=a.intValue();

                for(i=2;i<=m;i++)
                {
                        if(pr%i==0)
                                return0;
                }

                return1;

        }
        publicstaticvoidce()
        {
                intk=0;

                for(inti=2;i<t;i++)
                {
                        if(t%i==0)
                                conti

                        nue;

                        flag=prime(i

                        );

                        if(flag==1&&i!=p&&i!=q)
                        {

                                e[k]=i;
                                flag=cd(e[k]);
                                if(flag>0)
                                {
                                        d[k]=fl
                                        ag;
                                        k++;
```

```
                        }
                        if(k==99)
                    break;
                    }
            }
}

publicstaticintcd( intx)
{
        intk=1;

        while(true)
        {
                k=k+t;

                if(k%x==0)
                        return(k/x);
        }

}

publicstaticvoid encrypt()
{

        intpt,ct,key=e[0],k;

        pt=msg;
        k=1;

        for(intj=0;j<key;j++)
        {
                k=k*pt;
                k=k%n;
        }

        ct=k;
        temp
        =ct;
        System.out.println("\nTHEENCRYPTEDMESSAGEIS:"+ct);

}

Publicstatic void decrypt()
{
        intpt,ct,key=d[0],k;
        ct=temp;
```

```
k=1;

                for(intj=0;j<key;j++)
                {
                        k=k
                        *ct;
                        k=k
                        %n;
                }

                pt=k;

               System.out.println("\nTHEDECRYPTEDMESSAGEIS:"+pt);
        }

        publicstaticvoidmain(Stringargs[])
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("ENTERFIRSTPRIMENUMBER"
                ); p=sc.nextInt();
                flag=prime(p);
                if(flag==0)
                {
                        System.out.println("WRONGINPU
                        T"); System.exit(1);
                }
                System.out.println("ENTERANOTHERPRIMENUMBER");
                q=sc.nextInt();
                flag=prime(q);
                if(flag==0||p==q)
                {
                        System.out.println("WRONGINPU
                        T"); System.exit(1);
                }

                System.out.println("ENTERMESSAG
                E"); msg=sc.nextInt();
                n=p*q;
                t=(p-1)*(q-1);

                ce();

                System.out.println("POSSIBLEVALUESOFeANDdARE");
                for (int i=0;i< m-1;i++)
                        System.out.printf("\n%d\t%d",e[i],d[i]);

                encrypt();
```

```
                decrypt();
            }
    }
```


dell@ewit: ~/nl
dell@ewit:~/nl$ javac RSA.java
dell@ewit:~/nl$ java RSA
ENTER FIRST PRIME NUMBER
97
ENTER ANOTHER PRIME NUMBER
101
ENTER MESSAGE
256
POSSIBLE VALUES OF e AND d ARE
7 2743
11 3491
13 1477
17 3953
19 7579
23 2087
29 9269
31 8671
37 3373
41 7961
43 8707
47 8783
53 2717
59 1139
61 8341
67 1003
71 7031
73 9337
79 6319
83 347
89 5609
103 4567
THE ENCRYPTED MESSAGE IS:8016
THE DECRYPTED MESSAGE IS:256
dell@ewit:~/nl$

10. Develop a program for congestion control using leaky bucket algorithm.

To understand this concept first we have to know little about traffic shaping.

Traffic Shaping: This is a mechanism to control the amount and the rate of the traffic sent to the network.

Two techniques can shape traffic:

1. Leaky Bucket
2. Token Bucket.

Suppose we have a bucket in which we are pouring water in a random order but we have to get water in a fixed rate, for this we will make a hole at the bottom of the bucket. It will ensure that watercomingoutisinsomefixedrate.Andalsoifbucketisfullwewillstoppouringinit. The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate.

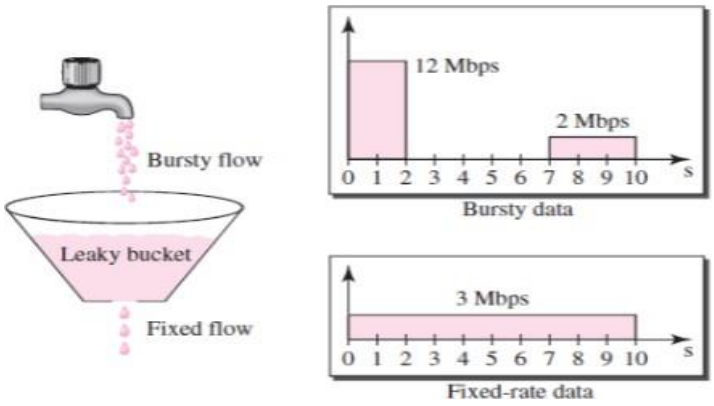


Fig 10.1: Leaky bucket scenario

In the above figure, we assume that the network has committed a bandwidth of 3 Mbps for a host. The use of the leaky bucket shapes the input traffic to make it conform tot his commitment. In Figure the host sends a burst of data at a rate of 12 Mbps for 2 s, for a total of 24 M bits of data. The host is silent for 5 s and then sends data at a rate of 2 Mbps for 3 s, for a total of 6 M bits of data. In all, the host has sent 30 M bits of data in 10 s. The leaky bucket smoothes the traffic by sending out data at a rate of 3Mbps during the same10s.Without the leaky bucket, the beginning burst may have hurt the network by consuming more bandwidth than is set aside for this host. We can also see that the leaky bucket may prevent congestion.

```
import java.util.Random;
import java.util.Scanner;
public class Leaky
{
public static int bsize=0,packet,tgen,j=1;
public static String stop;
public static final int bmax=1024;
public static final int orate=100;
public static final int delay=1500;
public static Random r=new Random();
public static Random t=new Random();
public class generating extends Thread
{
public void run()
{
while(stop==null)
{
tgen=t.nextInt(3000);
packet=r.nextInt(512);
if(bsize+packet<bmax)
{
bsize=bsize+packet;
System.out.printf("%13d%10d%15d%20d\n",j++,packet,bsize,bmax-bsize);
}
else
System.out.println("Bucket OverFlow, "+packet+" size of packet discarded");
try{Thread.sleep(tgen);}catch(Exception e){};
}
}
}
public class leaking extends Thread
{
public void run()
{
while(true)
{
if(bsize>0 && bsize-orate>0) //output packet rate is 100bytes
{
bsize=bsize-orate;
System.out.printf("%38d%20d%15d\n",bsize,(bmax-bsize),orate);
```

```
}
else
{
System.out.printf("%38d%20d%15d\n",0,bmax,bsize);
bsize=0;
if(stop!=null)
return;}
try{Thread.sleep(delay);}catch(Exception e){};
}
}
}
//@SuppressWarnings("resource")
public static void main(String[] args)
{
Leaky le=new Leaky();
Scanner in=new Scanner(System.in);
generating g=le.new generating();
leaking l=le.new leaking();
System.out.println("Started");
System.out.println("Output Rate is:"+orate+"\nAnd it is flowing at interval:"+
((float)delay/1000)+"sec");
System.out.println("Enter any key to stop input");
System.out.printf("Packet number | Input Packet | Bucket filled | Remaining space|Output rate");
System.out.println();
g.start();
try{Thread.sleep(10);}catch(Exception e){};
l.start();
stop=in.next();
}
}
```

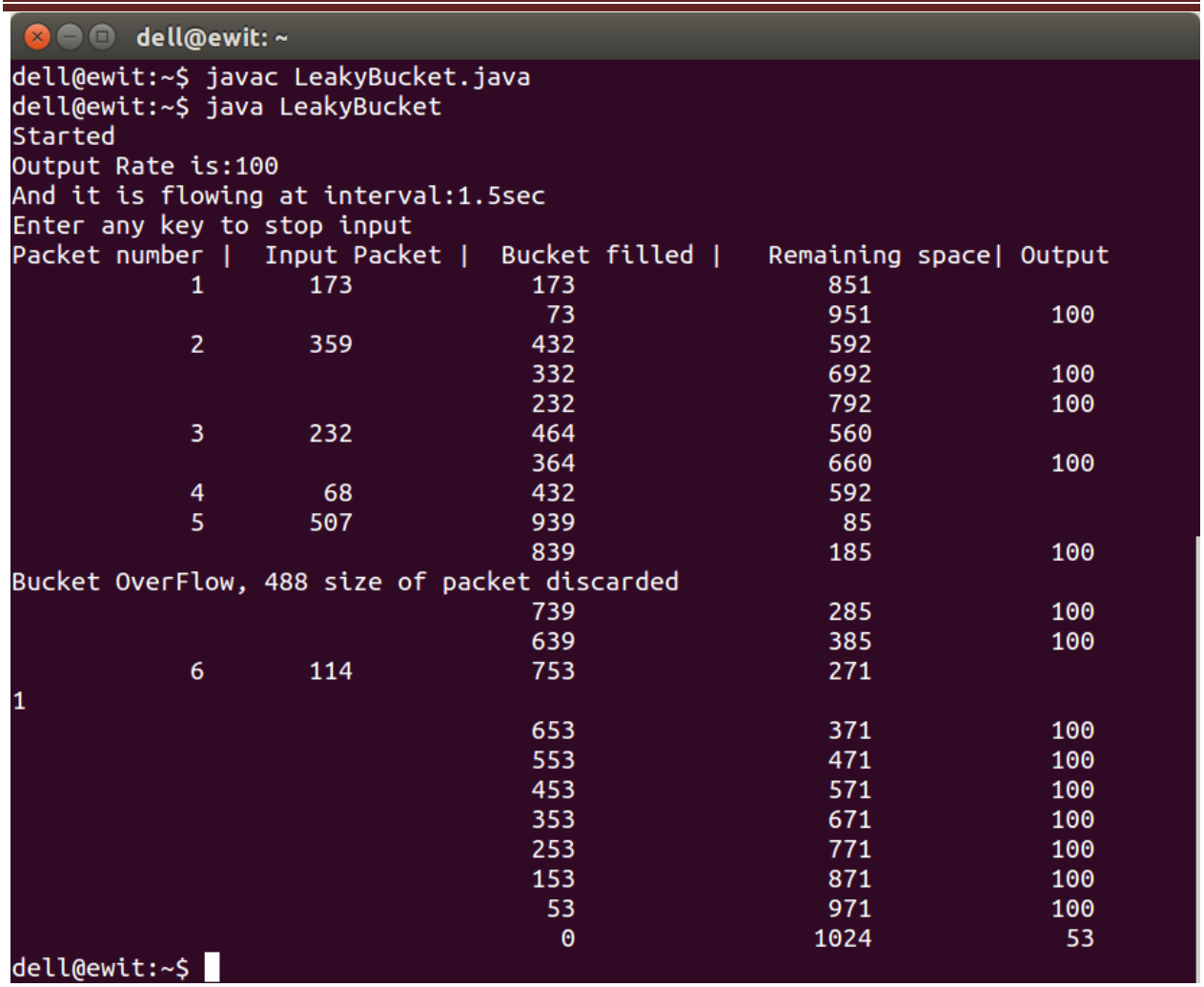


Fig 10.2: Congestion control using Leaky bucket algorithm.

VIVA QUESTION AND ANSWER

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

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

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

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

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

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

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

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

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

10) What is the maximum length allowed for a UTP cable?
A single segment of UTP cable has an allowable length of 90 to 100 meters. This limitation can be overcome by using repeaters and switches.

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

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

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

14) Briefly describe NAT.

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

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

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

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

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

17) What is RIP?

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

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

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

19) What is NIC?

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

20) What is WAN?

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

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

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

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

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

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

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

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

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

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

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

26) What does 10Base-T mean?

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

27) What is a private IP address?

Private IP addresses are assigned for use on intranets. These addresses are used for internal networks and are not routable on external public networks. These ensures that no conflicts are

present among internal networks while at the same time the same range of private IP addresses are reusable for multiple intranets since they do not "see" each other.

28) What is NOS?

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

29) What is DoS?

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

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

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

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

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

32) What is the advantage of address sharing?

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

33) What are MAC addresses?

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

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

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

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

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

36) What is the main purpose of OSPF?

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

37) What are firewalls?

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

38) Describe star topology

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

39) What are gateways?

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

40) What is the disadvantage of a star topology?

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

41) What is SLIP?

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

42) Give some examples of private network addresses.

10.0.0.0 with a subnet mask of 255.0.0.0
172.16.0.0 with subnet mask of 255.240.0.0
192.168.0.0 with subnet mask of 255.255.0.0

43) What is tracert?

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

44) What are the functions of a network administrator?

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

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

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

46) What is Hybrid Network?

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

47) What is DHCP?

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

48) What is the main job of the ARP?

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

49) What is TCP/IP?

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

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

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

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

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

52) What is the use of a default gateway?Default gateways provide means for the local networks to connect to the external network. The default gateway for connecting to the external network is usually the address of the external router port.

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

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

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

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

55) What is netstat?

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

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

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

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

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

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

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

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

59) What is ICMP?

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

60) What is Ping?

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

61) What is peer to peer?

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

62) What is DNS?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

70) What is ipconfig?

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

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

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

72) What is client/server?

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

73) Describe networking.

Networking refers to the inter connection between computers and peripherals for data communication.

Networking can be done using wired cabling or through wireless link.

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

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

75) Explain clustering support

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

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

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

77) Describe Ethernet.

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

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

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

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

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

80) What is SMTP?

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

81) What is multicast routing?

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

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

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

83) How are IP addresses arranged and displayed?

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

84) Explain the importance of authentication.

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

85) What do mean by tunnel mode?

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

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

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

87) What is one advantage of mesh topology?

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

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

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

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

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

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

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

91) Explain profile in terms of networking concept?

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

92) What is sneakernet?

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

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

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

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

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

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

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

96) What is one basic requirement for establishing VLANs?

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

97) What is IPv6?

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

98) What is RSA algorithm?

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

99) What is mesh topology?

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