JYOTHY INSTITUTE OF TECHNOLOGY

Tataguni, Off Kanakapura Road, Bengaluru – 560 082

APPROVED BY AICTE & AFFILIATED TO VTU, BELAGAVI

Department of
Computer Science & Engineering

Accredited by NBA, New Delhi

Lab Manual

COMPUTER NETWORK LABORATORY

18CSL57

Prepared By:

Dr. Swathi K
Associate Professor
Dept. of CSE, JIT,
Bengaluru

Institution Vision & Mission

Vision:

"To be an Institution of Excellence in Engineering education, Innovation and Research and work towards evolving great leaders for the country's future and meeting global needs."

Mission:

The Institution aims at providing a vibrant, intellectually and emotionally rich teaching learning environment with the State of the Art Infrastructure and recognizing and nurturing the potential of each individual to evolve into ones own self and contribute to the welfare of all.

Department Vision & Mission

Vision:

"To be a center of excellence in Computer Science and Engineering education, focus on research, innovation and entrepreneurial skill development with professional competency."

Mission:

M1: To provide state of the art ICT infrastructure and innovative, research-oriented teaching-learning environment and motivation for self-learning & problem-solving abilities by recruiting committed faculty.

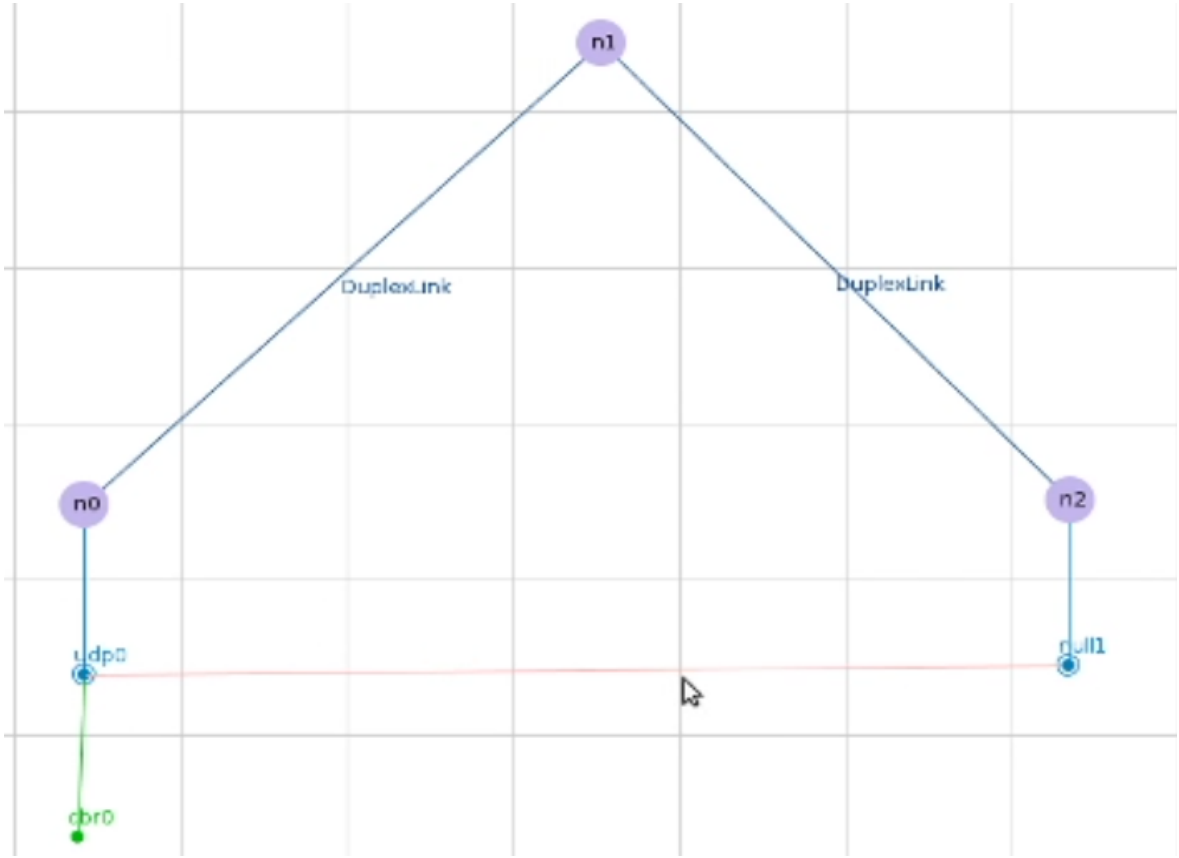
M2: To encourage Industry Institute Interaction & multi-disciplinary approach for problem-solving and adapt to ever-changing global IT trends.

M3: To imbibe awareness of societal responsibility and leadership qualities with professional competency and ethics.

PART –A PROGRAMS

1. Implement three node point-to-point networks with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.

Topology



Procedure:

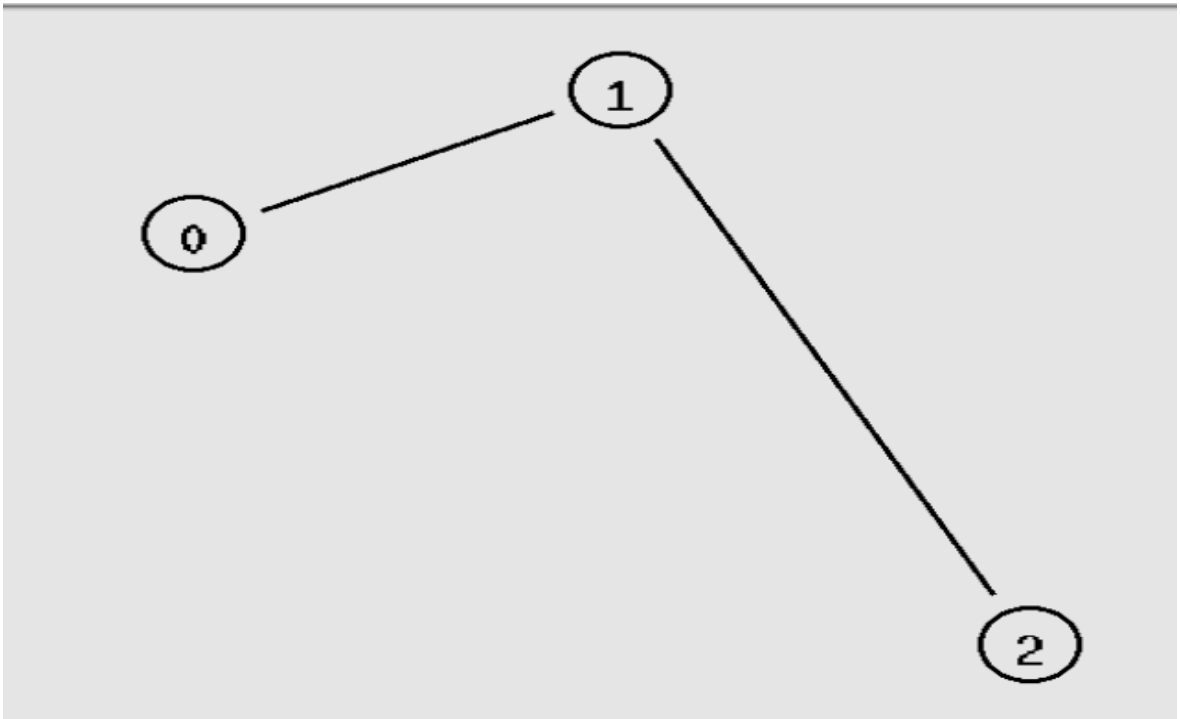
1. Open nsg2.1.jar
2. Select new wired scenario
3. Place 3 nodes and connect them with duplex link as shown in topology above.
4. Click agent, select agent type udp and connect udp to node n0
5. Click agent, select agent type null and connect null to node n2
6. Connect null from udp to null
7. Click application, select application type cbr and connect cbr to udp
8. Save tcl script and open terminal
9. Run the simulation program by the command ns filename.tcl
   i) Here "ns" indicates network simulator. We get the topology shown in the snapshot

below.

ii) Now press the play button in the simulation window and simulation will begin.

10. Create awk file and after simulation is completed run awk file to see the output,
    awk -f filename.awk out.tr

Simulation Topology



Result:

Total number of packets sent: 250
Total number of packets received: 250
Total number of packets dropped: 0

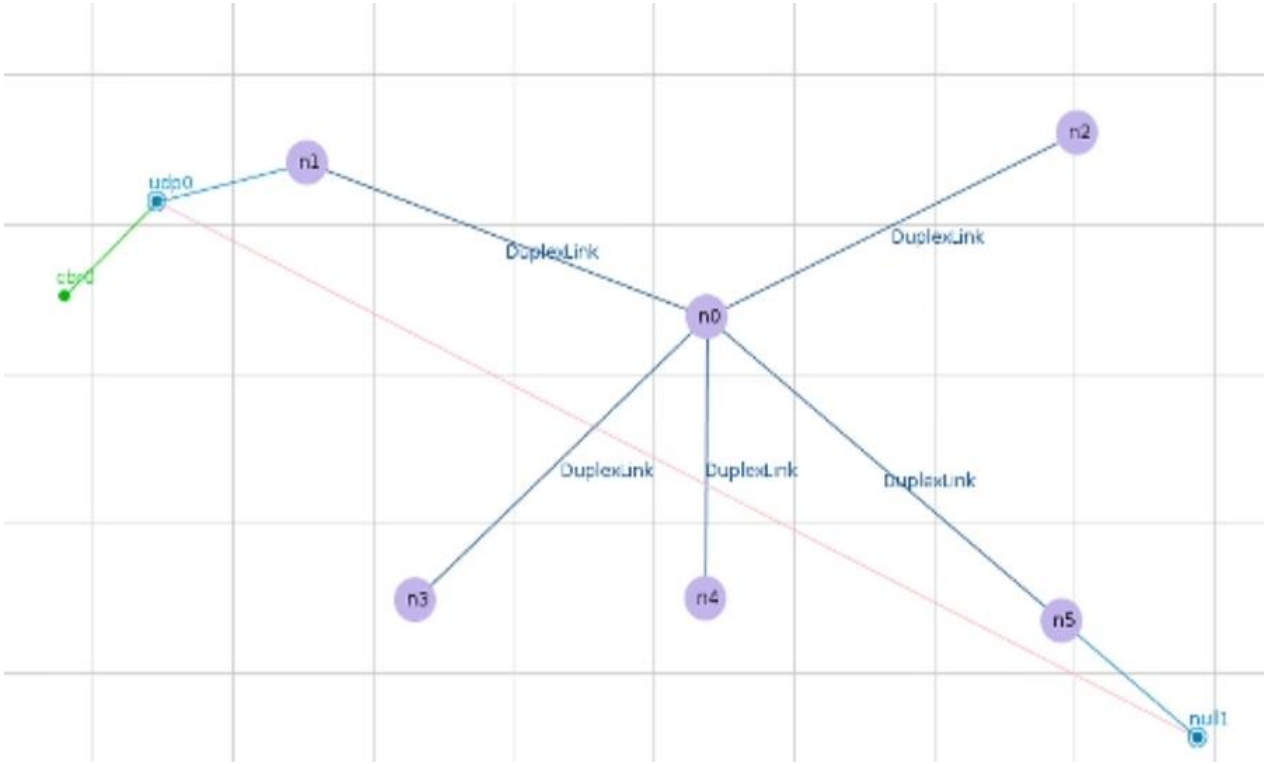
Conclusion: Increase in the bandwidth and queue size, decreases the number of packets dropped.

AWK Script:

```
BEGIN{
 dcount = 0;
 rcount = 0;
 }
 {
 event = $1; if(event ==
 "d")
 {
 dcount++;
}
 if(event == "r")
 {
 rcount++;
 }
 } END
 {
 printf("The no.of packets dropped : %d\n ",dcount); printf("The no.of packets recieved
 : %d\n ",rcount);
 }
```

2. Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets droppeddue to congestion.

Topology:



Procedure:

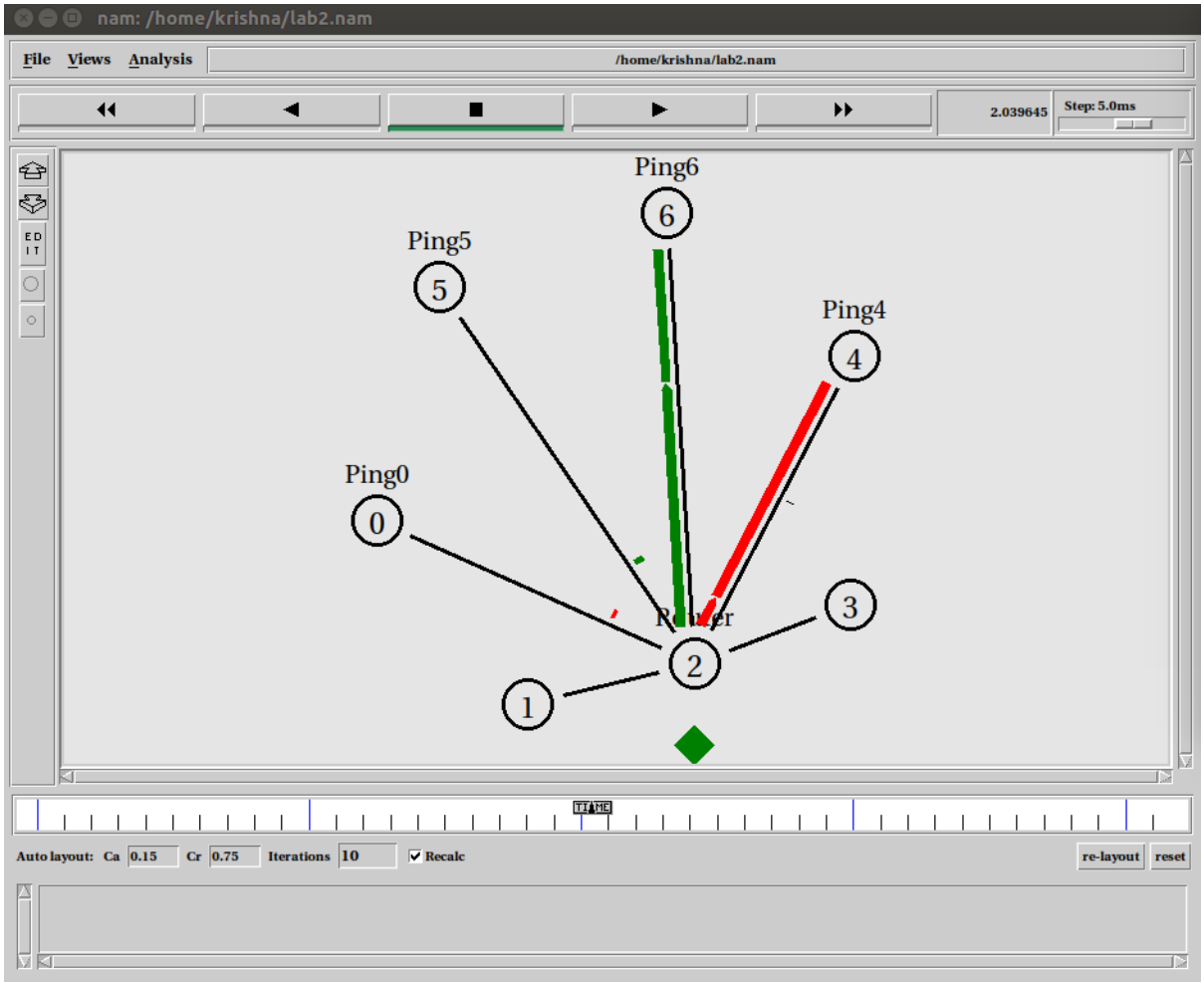
1. Open nsg2.1.jar
2. Select new wired scenario
3. Place 5 nodes and connect them with duplex link as shown in topology above.
4. Click agent, select agent type udp and connect udp to node n1
5. Click agent, select agent type null and connect null to node n5
6. Connect null from udp to null
7. Click application, select application type cbr and connect cbr to udp
8. Save tcl script and open terminal
9. Run the simulation program by the command ns filename.tcl

iii) Here "ns" indicates network simulator. We get the topology shown in the snapshot below.

iv) Now press the play button in the simulation window and simulation will begin.

10. Create awk file and after simulation is completed run awk file to see the output,
    awk -f filename.awk out.tr

Simulation Topology



Result:

Total number of packets received: 335

Conclusion: While ping message transmission , there is no packet drop due to congestion

AWK Script:

```
BEGIN{
 dcount = 0;
 rcount = 0;
 }
 {
 event = $1; if(event ==
 "d")
 {
 dcount++;
}
 if(event == "r")
 {
 rcount++;
 }
 } END
 {
 printf("The no.of packets dropped : %d\n ",dcount); printf("The no.of packets recieved
 : %d\n ",rcount);
 }
```

Add TCL Snippet

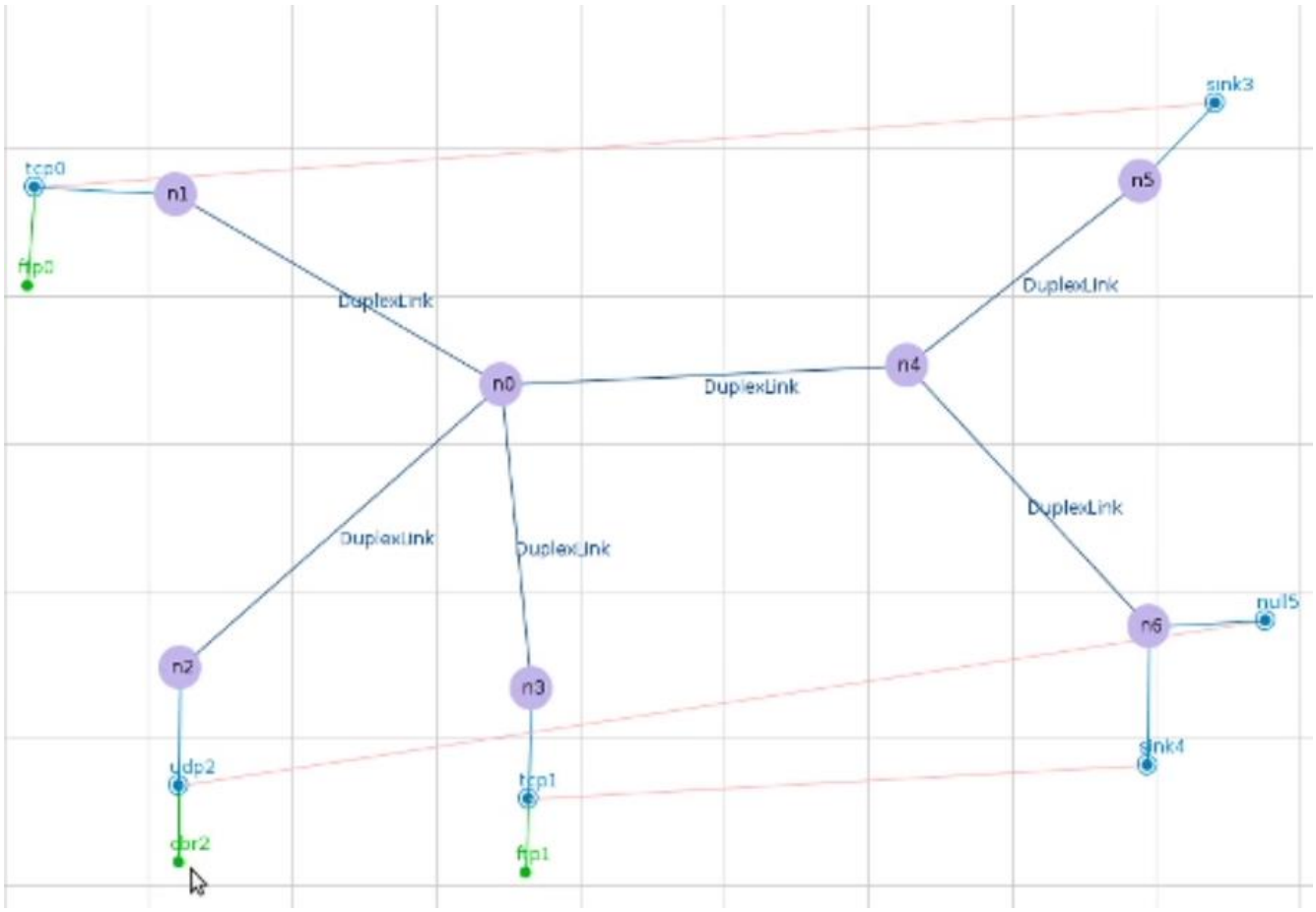
//agent definition

```
Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts "node[$node_ id] received Ping answer from $from with roundtriptime $rtt ms"}
```

```
set p0 [new Agent/Ping]
$ns attach-agent $n1 $p0
set p1 [new Agent/Ping]
$ns attach-agent $n4 $p1
$ns connect $p0 $p1

set i 0.1
while {$i<39} {
$ns at $i "$p1 send"
$ns at $i "$p0 send"
set i [expr $i + 0.1]
}
```

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

Topology



Procedure:

1. Open nsg2.1.jar
2. Select new wired scenario
3. Place 7 nodes and connect them with duplex link as shown in topology above.
4. Click agent, select agent type tcp and connect tcp to node n1
5. Click agent, select agent type tcp and connect tcp to node n3
6. Click agent, select agent type udp and connect udp to node 2
7. Click agent, select agent type tcpsink and connect tcpsink to node n5
8. Click agent, select agent type tcpsink and connect tcpsink to node n6
9. Click agent, select agent type null and connect null to node n6
10. Connect null from tcp0 to tcpsink3
11. Connect null from tcp1 to tcpsink4

12. Connect null from udp to null
13. Click application, select application type ftp and connect ftp0 to tcp0
14. Click application, select application type ftp and connect ftp1 to tcp1
15. Click application, select application type cbr and connect cbr2 to udp2
16. Save tcl script and open terminal
17. Run the simulation program by the command ns filename.tcl
    v) Here "ns" indicates network simulator. We get the topology shown in the snapshot below.
    vi) Now press the play button in the simulation window and simulation will begin.
18. Output Xgraph will be generated.

Add TCL Snippet

//before agent definition

```
set file1 [ open cw1.out w ]
set file2 [ open cw2.out w ]
```

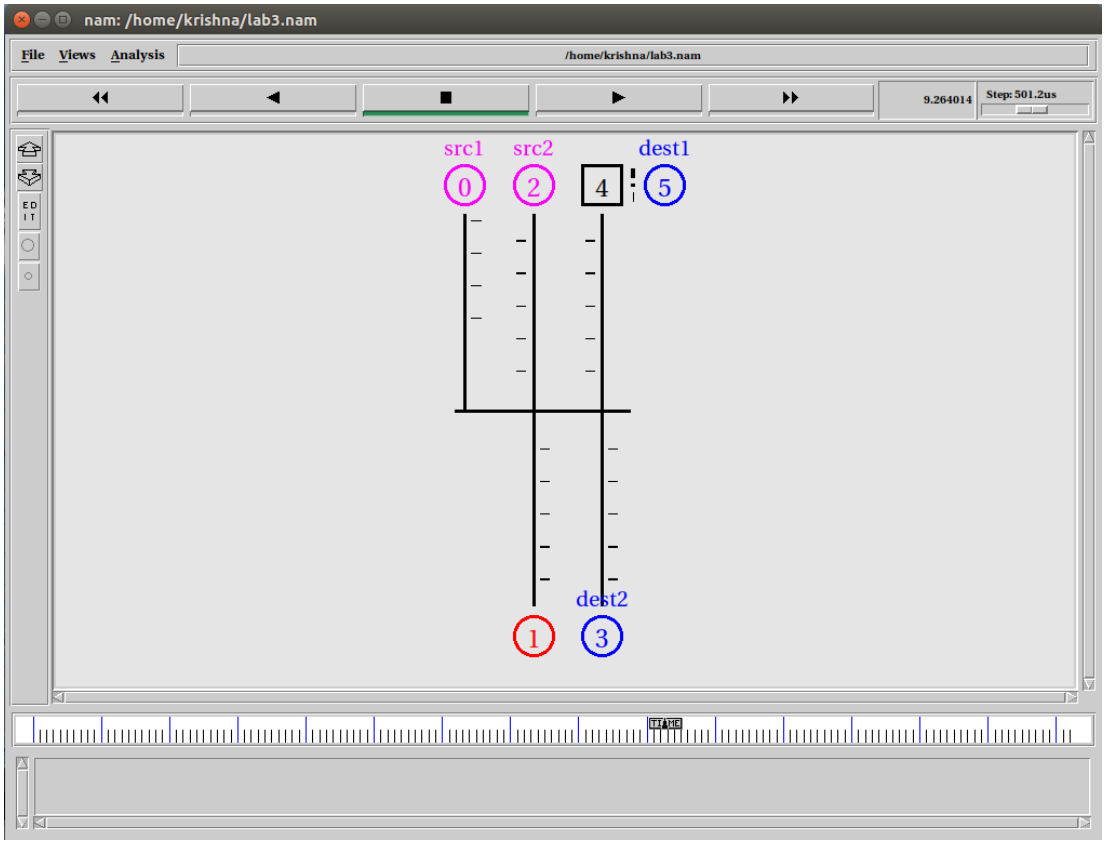
//before termination check names of ftp udp and tcp

```
proc record { } {
global tcp0 tcp1 file1 file2

set ns [ Simulator instance ]
set time 0.5
set cw1 [ $tcp0 set cwnd_ ]
set cw2 [ $tcp1 set cwnd_ ]
set now [ $ns now ]
puts $file1 "$now $cw1"
puts $file2 "$now $cw2"
$ns at [ expr $now + $time ] "record"
}

$ns at 1.0 "record"
$ns at 15.0 "$cbr2 start"
$ns at 75.0 "$cbr2 stop"
$ns at 5.0 "$ftp3 start"
$ns at 80.0 "$ftp3 stop"
$ns at 10.0 "$ftp0 start"
$ns at 35.0 "$ftp0 stop"
```

//before exit and inside termination

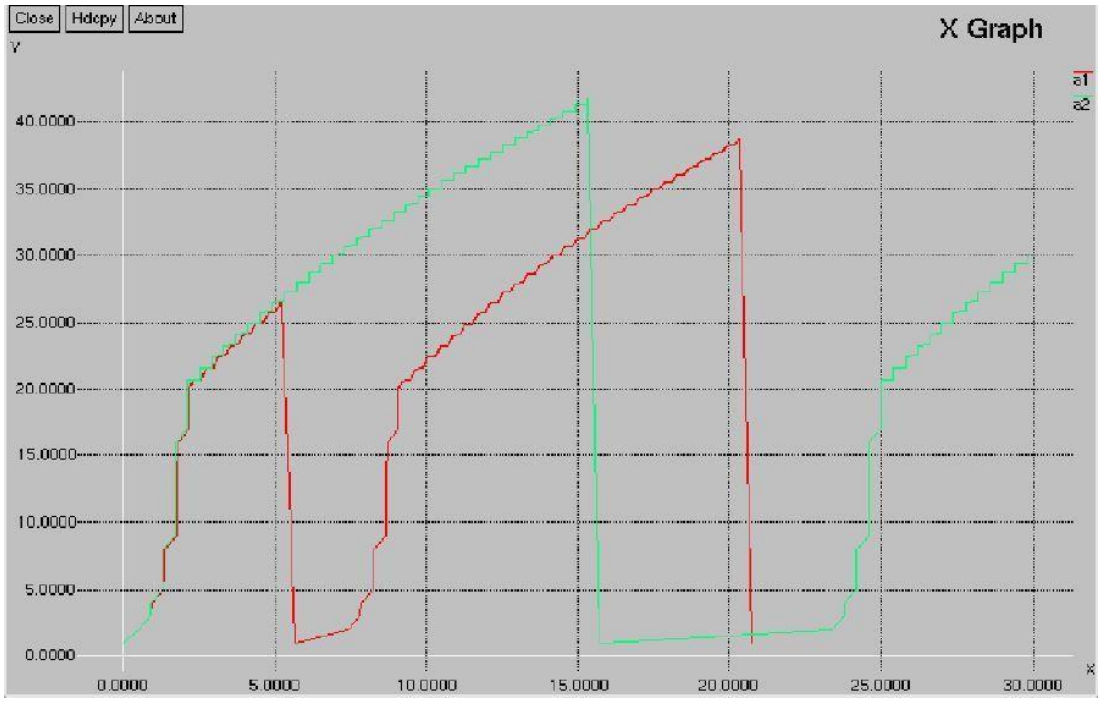
```
exec /home/jit/Desktop/xg/bin/xgraph cw1.out &
exec /home/jit/Desktop/xg/bin/xgraph cw2.out &
```

Simulation Topology



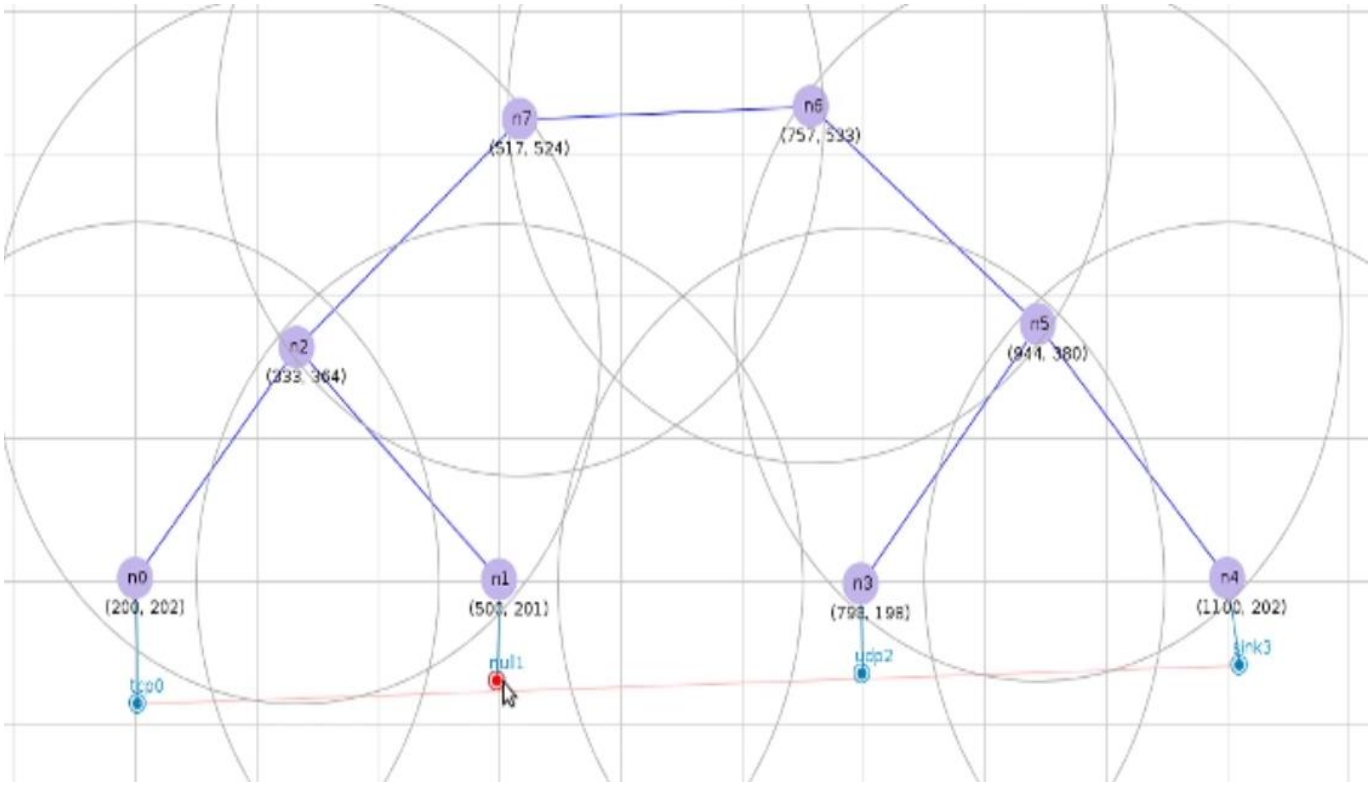
Output :

Xgraph



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

Topology



Procedure:

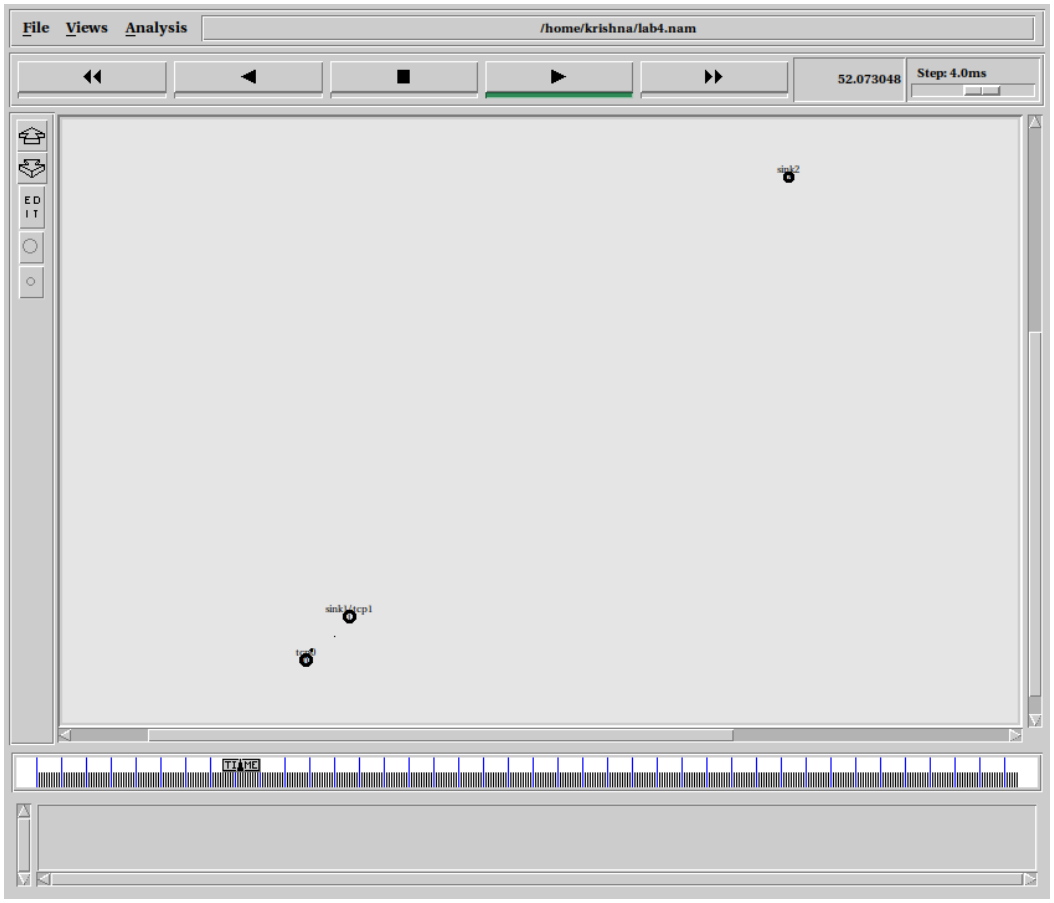
1. Open nsg2.1.jar
2. Select new wireless scenario
3. Place nodes as shown in topology above.
4. Click agent, select agent type tcp and connect tcp0 to node n0
5. Click agent, select agent type null and connect null to node n1
6. Click agent, select agent type udp and connect udp2 to node n3
7. Click agent, select agent type tcpsink and connect tcpsink3 to node n4
8. Click application, select application type ftp and connect ftp to tcp0
9. Click application, select application type cbr and connect cbr to udp2
10. Save tcl script and open terminal
11. Run the simulation program by the command ns filename.tcl

vii) Here "ns" indicates network simulator. We get the topology shown in the snapshot below.

viii) Now press the play button in the simulation window and simulation will begin.

12. Create awk file and after simulation is completed run awk file to see the output,
    awk -f filename.awk out.tr

Simulation Topology:



Result:

Total no. of packets sent by node 0: 225
Total no. of packets received by node 4: 180
Total no. of packets sent by node 3: 355
Total no. of packets received by node 1: 250

Conclusion: Point to Point throughput will be greater than broadcast throughput

AWK SCRIPT

```
BEGIN{
tcp0packetsent=0
tcp0packetrec=0
tcp1packetsent=0
tcp1packetrec=0
}
{
if($1=="s" && $3=="_0_" && $4=="AGT")
{
tcp0packetsent++;
}
if($1=="r" && $3=="_4_" && $4=="AGT")
{
tcp0packetrec++;
}
if($1=="s" && $3=="_3_" && $4=="AGT")
{
tcp0packetsent++;
}
if($1=="r" && $3=="_1_" && $4=="AGT")
{
tcp0packetrec++;
}}
END{
```

```
printf("\n Total no. of packets sent by node 0:%d\n",tcp0packetsent++);
printf("\n Total no. of packets received by node 4:%d\n",tcp0packetrec++);
printf("\n Total no. of packets sent by node 3:%d\n",tcp1packetsent++);
printf("\n Total no. of packets received by node 1:%d\n",tcp1packetrec++);
}
```

//Changes to do in TCL SCRIPT

Replace DSDV routing protocol with AODV

Change 10 to 50

PART - B

PART-B

Program 1:

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

```
import java.io.*;
class Crc
{
   public static void main(String args[]) throws IOException
  {

    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    int[ ] data;
    int[  ]div;
    int[ ]divisor;
    int[ ]rem;
    int[ ]crc;
    int data_bits, divisor_bits, tot_length;

    System.out.println("Enter number of data bits : ");
    data_bits=Integer.parseInt(br.readLine());
    data=new int[data_bits];

    System.out.println("Enter data bits : ");
    for(int i=0; i<data_bits; i++)
    data[i]=Integer.parseInt(br.readLine());

    System.out.println("Enter number of bits in divisor : ");
    divisor_bits=Integer.parseInt(br.readLine());
    divisor=new int[divisor_bits];
    System.out.println("Enter Divisor bits : ");
    for(int i=0; i<divisor_bits; i++)
```

```
divisor[i]=Integer.parseInt(br.readLine());
tot_length=data_bits+divisor_bits-1; div=new
int[tot_length];
        rem=new int[tot_length];
        crc=new int[tot_length];

        for(int i=0;i<data.length;i++)
        div[i]=data[i];

        System.out.print("Dividend (after appending 0's) are : ");
        for(int i=0; i<div.length; i++)
        System.out.print(div[i]);
        System.out.println();
        for(int j=0; j<div.length; j++)
        {
                rem[j] = div[j];
        }

      rem=divide(div, divisor, rem);
      for(int i=0;i<div.length;i++)
      {
              crc[i]=(div[i]^rem[i]);
      }
      System.out.println();
      System.out.println("CRC code : ");
      for(int i=0;i<crc.length;i++)
      System.out.print(crc[i]);
      System.out.println();
```

```
        System.out.println("Enter CRC code of "+tot_length+" bits : ");
        for(int i=0; i<crc.length; i++)
        crc[i]=Integer.parseInt(br.readLine());

  for(int j=0; j<crc.length; j++)
  {
    rem[j] = crc[j];
  }
   rem=divide(crc, divisor, rem);
   for(int i=0; i<rem.length; i++)
   {
         if(rem[i]!=0)
        {
              System.out.println("Error");
              break;
        }
       if(i==rem.length-1)
       System.out.println("No  Error");
   }
 }

static int[] divide(int div[],int divisor[], int rem[])
{
    int cur=0;
    while(true)
   {
      for(int i=0;i<divisor.length;i++)
      rem[cur+i]=(rem[cur+i]^divisor[i]);
      while(rem[cur]==0 && cur!=rem.length-1)
      cur++;

     if((rem.length-cur)<divisor.length)
```

```
    break;
  }
    return rem;
} }
```

Output:

```
krishna@ubuntu:~$ javac Crc.java
krishna@ubuntu:~$ java Crc
Enter number of data bits :
7
Enter data bits :
1
0
1
1
0
0
1
Enter number of bits in divisor :
3
Enter Divisor bits :
1
0
1
Dividend (after appending 0's) are : 101100100

CRC code :
101100111
Enter CRC code of 9 bits :
1
0
1
1
0
0
1
1
1
No Error
```

Create/ Compile/ Run

- gedit filename.java → Create
- javac filename.java → Compile
- java filename → Run

Program 2:

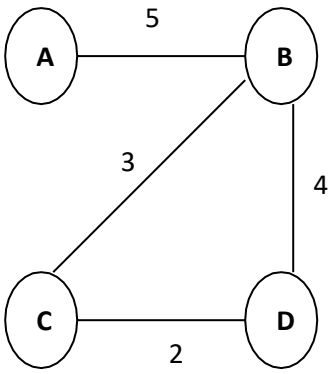
Shortest path between vertices using bellman-ford algorithm

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

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

public void BellmanFordEvaluation(int source,int adjacencymatrix[][])
{
for(int node=1;node<=numberofvertices;node++)
{
        distances[node] = MAX_VALUE;
}
distances[source]=0;

for(int node =1;node<=numberofvertices-1;node++)
{
for (int sourcenode =1 ;sourcenode<=numberofvertices;sourcenode++)
{
for(int destinationnode=1;destinationnode<=numberofvertices;destinationnode++)
{
if (adjacencymatrix[sourcenode][destinationnode]!=MAX_VALUE)
{
if(distances[destinationnode]>distances[sourcenode]+adjacencymatrix[sourcenode][destinationnode])
{
distances[destinationnode]=distances[sourcenode]+adjacencymatrix[sourcenode][destinationnode];
}
```

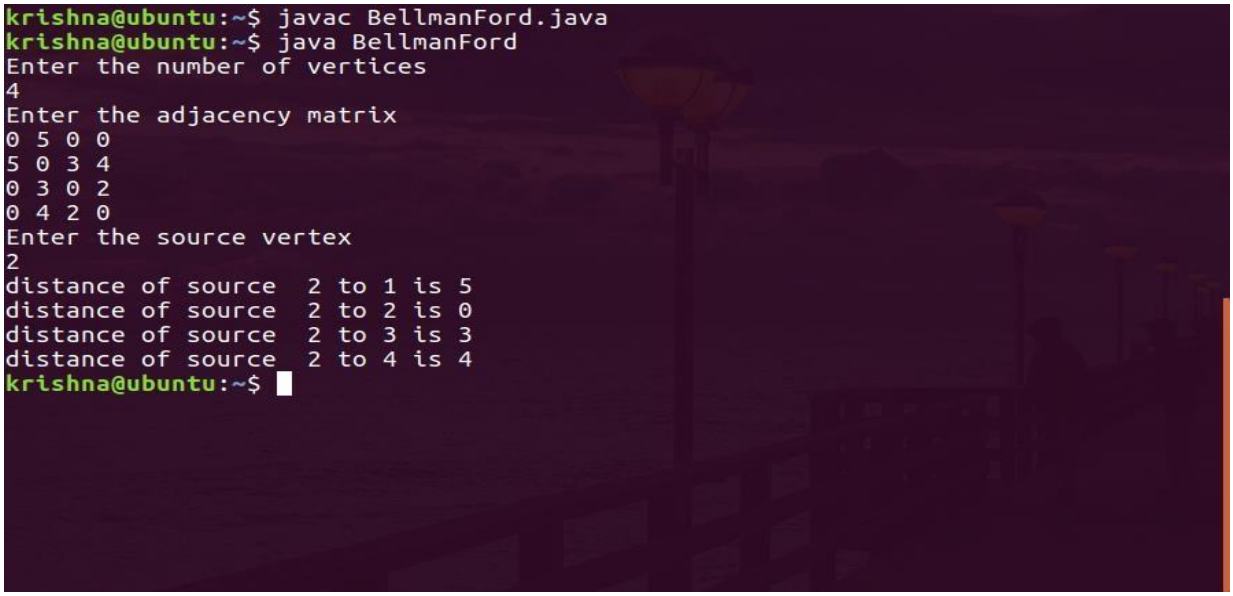
```
}
}
}
}
for (int vertex=1;vertex<=numberofvertices;vertex++)
{
System.out.println("distance of source"+source+"to"+vertex+"is"+distances[vertex]);
}
}

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

```
System.out.println("enter the source vertex");
source=scanner.nextInt();
BellmanFord bellmanford=new BellmanFord(numberofvertices);
bellmanford.BellmanFordEvaluation(source,adjacencymatrix);
scanner.close();
}
}
```

Input Graph:



Output:


krishna@ubuntu:~$ javac BellmanFord.java
krishna@ubuntu:~$ java BellmanFord
Enter the number of vertices
4
Enter the adjacency matrix
0 5 0 0
5 0 3 4
0 3 0 2
0 4 2 0
Enter the source vertex
2
distance of source 2 to 1 is 5
distance of source 2 to 2 is 0
distance of source 2 to 3 is 3
distance of source 2 to 4 is 4
krishna@ubuntu:~$

Program 3:

TCP/IP sockets, 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.

Client Program

```
import java.net.*;
import java.io.*;
public class ContentsClient
{
  public static void main( String args[ ] ) throws Exception {
    Socket sock = new Socket("127.0.0.1", 4000);
     // reading the file name from keyboard. Uses input
    stream      System.out.print("Enter the file name");
    BufferedReader keyRead = new BufferedReader(new InputStreamReader(System.in));
    String fname = keyRead.readLine();


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


    // receiving the contents from server. Uses input stream
    InputStream istream = sock.getInputStream();
    BufferedReader socketRead = new BufferedReader(new InputStreamReader(istream));


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

```
    keyRead.close();
 }
}
```

Server Program

```
import java.net.*;
import java.io.*;
public class ContentsServer
{
  public static void main(String args[]) throws Exception {

    // establishing the connection with the server
    ServerSocket sersock = new ServerSocket(4000);
    System.out.println("Server ready for
    connection");
    Socket sock = sersock.accept(); // binding with port: 4000
    System.out.println("Connection is successful and wating for client request");

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

     // reading file contents
    BufferedReader contentRead = new BufferedReader(new FileReader(fname) );

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

    String str;
    while((str = contentRead.readLine()) != null) // reading line-by-line from file
```

```
    {

        pwrite.println(str); // sending each line to client
    }
     sock.close(); sersock.close(); // closing network sockets
    pwrite.close(); fileRead.close(); contentRead.close();
  }
}
```

Output:

In the server side terminal

javac server.java

java server

>Server ready for connection

>Connection is successful and waiting for client request

In the client side terminal

javac client.java

java client

Enter the file name: swathi.txt

>swathi k, Dept. of CSE, JIT

Program 4:

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

Sender.java

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

Receiver.java

```
import java.io.*;
import java.net.*;
public class Receiver
{
   public static void main(String[] args) throws IOException
  {
    DatagramSocket socket = new DatagramSocket(4444);
    byte[] buf = new byte[256];
    DatagramPacket packet = new DatagramPacket(buf, buf.length);
    System.out.println("Waiting ...");
    socket.receive(packet);
```

```
    String s = new String(packet.getData(), 0, packet.getLength());
    System.out.println(packet.getAddress().getHostName() + ": " + s);
  }
}
```

- Compile the program.
- Start the receiver by running "java Receiver".
- Assuming that the receiver is running on a host with IP address 127.0.0.1
  Start the sender by running:

java Sender 127.0.0.1 "My String"

- The receiver program should now display the string "My String".
- Repeat this exercise, with the difference, that you run the sender and receiver on two different hosts.

Output:

```
krishna@ubuntu:~$ javac Sender.java
krishna@ubuntu:~$ java Sender 127.0.0.1 "Hello Ubuntu"
krishna@ubuntu:~$
```

```
krishna@ubuntu:~$ javac Receiver.java
krishna@ubuntu:~$ java Receiver
Waiting ...
localhost: Hello Ubuntu
krishna@ubuntu:~$
```

Program 5:

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

```
import java.math.BigInteger;
import java.util.*;
class rsa
{
 public static void main(String args[])
 {
    Scanner ip=new Scanner(System.in);
    int p,q,n,e=1,j;
    int d=1,i1;
    int t1,t2;
    int pt[]= new int[10];
    int ct[]=  new  int[10];
    int rt[]=  new  int[10];
    int temp[]= new int[10];
    String i=new String();

    System.out.println("Enter the two prime numbers:");
    p=ip.nextInt();
    q=ip.nextInt();

    System.out.println("Enter the message to be sent");
    i=ip.next();
    i1=i.length();
    n=p*q;
    t1=p-1;
    t2=q-1;



      System.out.println("\n ------------------------");
```

```
System.out.println("Sender Side:");
while((t1*t2)%e==0)
{
    e++;
}
System.out.println("Public Key(e)= "+e);
System.out.println("----------------------------");
for(j=0;j<i1;j++)
{
    pt[j]=(i.charAt(j))-96;
//      System.out.println("Plain Text= "+pt[j]);
    ct[j]=((int)Math.pow(pt[j],e))%n;
    System.out.println("Cipher Text= "+ct[j]);
}
    System.out.println("\nTransmitted Message:");
    for(j=0;j<i1;j++)
    {
        temp[j]=ct[j]+96;
        System.out.print((char)temp[j]);
    }

    System.out.println("\n\n -------------------------");
    System.out.println("Receiver Side:");
    while((d*e)%(t1*t2)!=1)
    {
       d++;
    }

System.out.println("Private Key(d)= "+d);
System.out.println("----------------------------");

for(j=0;j<i1;j++)
{
```

```
        //System.out.println("cipher Text= "+ct[j]);
          BigInteger very_big_no = BigInteger.valueOf(ct[j]);
          very_big_no = very_big_no.pow(d);
          very_big_no = very_big_no.mod(BigInteger.valueOf(n));
          rt[j] =  very_big_no.intValue();
          System.out.println("Plain Text= "+rt[j]);
    }
          System.out.println("\n------------------------");
          System.out.println("Decrypted Message:");
          for(j=0;j<i1;j++)
         {
            rt[j]=rt[j]+96;
          System.out.print((char)rt[j]);
         }
          System.out.println("\n------------------------");
          ip.close();
  }
}
```

Output:

```
java rsa
Enter the two prime numbers:
5
11
Enter the message to be sent
global

---------------------
Sender Side:
Public Key(e)= 3
---------------------
Cipher Text= 13
Cipher Text= 23
Cipher Text= 20
Cipher Text= 8
Cipher Text= 1
Cipher Text= 23
```

```
Transmitted Message:
mwthaw

--------------------
Receiver Side:
Private Key(d)= 27
--------------------
Plain Text= 7
Plain Text= 12
Plain Text= 15
Plain Text= 2
Plain Text= 1
Plain Text= 12

--------------------
Decrypted Message:
global
--------------------
```

Program 6:

Write a program for Congestion control using leaky bucket algorithm.

```
Import java.util.Scanner;

public class LeakyBucket {

  Public static void main(String[] args) throws InterruptedException {

  Int n, incoming, outgoing, store=0, bucketsize;
  Scanner scan = new Scanner(system.in);
  System.out.println("Enter bucket size, outgoing rate, number of inputs and
incoming size);
  bucketsize = scan.nextInt();
  outgoing = scan.nextInt();
  n = scan.nextInt();
  Incoming = scan.nextInt();
  while(n!=0)
   {
      System.out.println("Incoming size is " + incoming);
       if(incoming <= (bucketsize-store))
        {
           store+= incoming;
           System.out.println("Bucket buffer size is " + store +" out of " +
bucketzise);
         }
         else
         {
           System.out.println("Packet loss : " + (incoming-(bucketsize-store)));
           store=bucketsize;
           System.out.println("Bucket buffer size is " + store +" out of " +
bucketzise);
          }
          store-=outgoing;
           System.out.println("After outgoing: " + store + " packets left out of "
+ bucketsize + " in buffer);
           n—;
          Thread.sleep(3000);
    }
       scan.close();
}
}
```

Output:

jit@jit-optiplex-comp31:~$ gedit LeakyBucket.java

jit@jit-optiplex-comp31:~$ javac LeakyBucket.java

jit@jit-optiplex-comp31:~$ java LeakyBucket

Enter bucket size, outgoing rate, no. of inputs and incoming size

5
3
2
4

Incoming size is 4
Bucket buffer size is 4 out of 5
After outgoing : 1 packet left out of 5 in buffer

Incoming size is 4
Bucket buffer size is 5 out of 5
After outgoing : 2 packets left out of 5 in buffer

Output (when bucket size is less than incoming size)

jit@jit-optiplex-comp31:~$ java LeakyBucket

Enter bucket size, outgoing rate, no. of inputs and incoming size
2
1
1
4

Incoming size is 4
Packet loss : 2
Bucket buffer size is 2 out of 2
After outgoing: 1 packets left out of 2 in buffer

VIVA QUESTIONS

VIVA QUESTIONS

1. Explain What is Network?

A network is a set of devices connected by physical media links. A network is recursively is a connection of two or more nodes by a physical link or two or more networks connected by one or more nodes.

2. What is a Link?

At the lowest level, a network can consist of two or more computers directly connected by some physical medium such as coaxial cable or optical fiber. Such a physical medium is called as Link.

3. What is a node?

A network can consist of two or more computers directly connected by some physical medium such as coaxial cable or optical fiber. Such a physical medium is called as Links and the computer it connects is called as Nodes.

4. What is a gateway or Router?

A node that is connected to two or more networks is commonly called as router or Gateway. It generally forwards message from one network to another.

5. What is point-point link?

If the physical links are limited to a pair of nodes it is said to be point-point link.

6. What is Multiple Access?

If the physical links are shared by more than two nodes, it is said to be Multiple Access.

7. What are the advantages of Distributed Processing?

a. Security/Encapsulation

b. Distributed database

c. Faster Problem solving

d. Security through redundancy

e. Collaborative Processing

8. What are the criteria necessary for an effective and efficient network?

a. Performance

It can be measured in many ways, including transmit time and response time. b. Reliability

It is measured by frequency of failure, the time it takes a link to recover from a failure, and the networks robustness.

c. Security

Security issues includes protecting data from unauthorized access and virues.

9. Name the factors that affect the performance of the network?

a. Number of Users

b. Type of transmission medium

c. Hardware

d. Software

10. Name the factors that affect the reliability of the network?

a. Frequency of failure

b. Recovery time of a network after a failure

11. Name the factors that affect the security of the network?

a. Unauthorized Access

b. Viruses

12. What is Protocol?

A protocol is a set of rules that govern all aspects of information communication.

13. What are the key elements of protocols?

The key elements of protocols are

a. Syntax

It refers to the structure or format of the data that is the order in which they are presented.

b. Semantics

It refers to the meaning of each section of bits.

c. Timing

Timing refers to two characteristics: When data should be sent and how fast they can be sent.

14. What are the key design issues of a computer Network?

a. Connectivity

b. Cost-effective Resource Sharing

c. Support for common Services

d. Performance

15. Define Bandwidth and Latency?

Network performance is measured in Bandwidth (throughput) and Latency (Delay). Bandwidth of a network is given by the number of bits that can be transmitted over the network in a certain period of time. Latency corresponds to how long it t5akes a message to travel from one end off a network to the other. It is strictly measured in terms of time.

16. Define Routing?

The process of determining systematically hoe to forward messages toward the destination nodes based on its address is called routing.

17. What is a peer-peer process?

The processes on each machine that communicate at a given layer are called peer-peer process.

18. When a switch is said to be congested?

It is possible that a switch receives packets faster than the shared link can accommodate and stores in its memory, for an extended period of time, then the switch will eventually run out of buffer space,and some packets will have to be dropped and in this state is said to congested state.

19. What is semantic gap?

Defining a useful channel involves both understanding the applications requirements and recognizing the limitations of the underlying technology. The gap between what applications expects and what the underlying technology can provide is called semantic gap.

20. What is Round Trip Time?

The duration of time it takes to send a message from one end of a network to the other and back, is called RTT.

21. Define the terms Unicasting, Multiccasting and Broadcasting?

If the message is sent from a source to a single destination node, it is called Unicasting.

If the message is sent to some subset of other nodes, it is called Multicasting.

If the message is sent to all the m nodes in the network it is called Broadcasting.

22. What is Multiplexing?

Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link.

23. Name the categories of Multiplexing?

a. Frequency Division Multiplexing (FDM)

b. Time Division Multiplexing (TDM)

i. Synchronous TDM

ii. ASynchronous TDM Or Statistical TDM.

c. Wave Division Multiplexing (WDM)

24. What is FDM?

FDM is an analog technique that can be applied when the bandwidth of a link is greater than the combined bandwidths of the signals to be transmitted.

25. What is WDM?

WDM is conceptually the same as FDM, except that the multiplexing and demultiplexing involve light signals transmitted through fiber optics channel.

26. What is TDM?

TDM is a digital process that can be applied when the data rate capacity of the transmission medium is greater than the data rate required by the sending and receiving devices.

27. What is Synchronous TDM?

In STDM, the multiplexer allocates exactly the same time slot to each device at all times, whether or not a device has anything to transmit.

28. List the layers of OSI

a. Physical Layer

b. Data Link Layer

c. Network Layer

d. Transport Layer

e. Session Layer

f. Presentation Layer

g. Application Layer

29. Which layers are network support layers?

a. Physical Layer

b. Data link Layer and

c. Network Layers

30. Which layers are user support layers?

a. Session Layer

b. Presentation Layer and

c. Application Layer

31. Which layer links the network support layers and user support layers?

The Transport layer links the network support layers and user support layers.

32. What are the concerns of the Physical Layer?

Physical layer coordinates the functions required to transmit a bit stream over a physical medium.

a. Physical characteristics of interfaces and media

b. Representation of bits

c. Data rate

d. Synchronization of bits

e. Line configuration

f. Physical topology

g. Transmission mode

33. What are the responsibilities of Data Link Layer?

The Data Link Layer transforms the physical layer, a raw transmission facility, to a reliable link and is responsible for node-node delivery.

a. Framing

b. Physical Addressing

c. Flow Control

d. Error Control

e. Access Control

34. What are the responsibilities of Network Layer?

The Network Layer is responsible for the source-to-destination delivery of packet possibly across multiple networks (links).

a. Logical Addressing

b. Routing

35. What are the responsibilities of Transport Layer?

The Transport Layer is responsible for source-to-destination delivery of the entire message.

a. Service-point Addressing

b. Segmentation and reassembly

c. Connection Control

d. Flow Control

e. Error Control

36. What are the responsibilities of Session Layer?

The Session layer is the network dialog Controller. It establishes, maintains and synchronizes the interaction between the communicating systems.

a. Dialog control

b. Synchronization

37. What are the responsibilities of Presentation Layer?

The Presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.

a. Translation

b. Encryption

c. Compression

38. What are the responsibilities of Application Layer?

The Application Layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as e-mail, shared database management and other types of distributed information services.

a. Network virtual Terminal

b. File transfer, access and Management (FTAM)

c. Mail services

d. Directory Services

39. What are the two classes of hardware building blocks?

Nodes and Links.

40. What are the different link types used to build a computer network?

a. Cables

b. Leased Lines

c. Last-Mile Links

d. Wireless Links

41. What are the categories of Transmission media?

a. Guided Media

i. Twisted Pair cable

1. Shielded TP

2. Unshielded TP

ii. Coaxial Cable

iii. Fiber-optic cable

b. Unguided Media

i. Terrestrial microwave

ii. Satellite Communication

42. What are the types of errors?

a. Single-Bit error

In a single-bit error, only one bit in the data unit has changed

b. Burst Error

A Burst error means that two or more bits in the data have changed.

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

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

a. Vertical Redundancy Check (VRC)

b. Longitudinal Redundancy Check (VRC)

c. Cyclic Redundancy Check (VRC)

d. Checksum

44. What is Redundancy?

The concept of including extra information in the transmission solely for the purpose of comparison. This technique is called redundancy.

45. What is VRC?

It is the most common and least expensive mechanism for Error Detection. In VRC, a parity bit is added to every data unit so that the total number of 1s becomes even for even parity. It can detect all single-bit errors. It can detect burst errors only if the total number of errors in each data unit is odd.

46. What is LRC?

In LRC, a block of bits is divided into rows and a redundant row of bits is added to the whole block. It can detect burst errors. If two bits in one data unit are damaged and bits in exactly the same positions in another data unit are also damaged, the LRC checker will not detect an error. In LRC a redundant data unit follows n data units.

47. What is CRC?

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

48. What is Checksum?

Checksum is used by the higher layer protocols (TCP/IP) for error detection

49. List the steps involved in creating the checksum.

a. Divide the data into sections

b. Add the sections together using 1s complement arithmetic

c. Take the complement of the final sum, this is the checksum.

50. What are the Data link protocols?

Data link protocols are sets of specifications used to implement the data link layer. The categories of Data Link protocols are

Asynchronous Protocols

Synchronous Protocols

a. Character Oriented Protocols

b. Bit Oriented protocols

51. Compare Error Detection and Error Correction:

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

52. What is Forward Error Correction?

Forward error correction is the process in which the receiver tries to guess the message by using redundant bits.

53. Define Retransmission?

Re transmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Re sending is repeated until a message arrives that the receiver believes is error-freed.

54. What are Data Words?

In block coding, we divide our message into blocks, each of k bits, called datawords. The block coding process is one-to-one. The same dataword is always encoded as the same codeword.

55. What are Code Words?

r redundant bits are added to each block to make the length n = k + r. The resulting n-bit blocks are called codewords. 2n 2k codewords that are not used. These codewords are invalid or illegal.

56. What is a Linear Block Code?

A linear block code is a code in which the exclusive OR (addition modulo-2) of two valid codewords creates another valid codeword.

57. What are Cyclic Codes?

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

58. Define Encoder?

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

59. Define Decoder?

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

60. What is Framing?

Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination

address defines where the packet has to go and the sender address helps the recipient acknowledge the receipt.

61. What is Fixed Size Framing?

In fixed-size framing, there is no need for defining the boundaries of the frames. The size itself can be used as a delimiter.

62. Define Character Stuffing?

In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

63. What is Bit Stuffing?

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

64. What is Flow Control?

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

65. What is Error Control?

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

66. What Automatic Repeat Request (ARQ)?

Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term error control refers primarily to methods of error detection and retransmission. Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).

67. What is Stop-and-Wait Protocol?

In Stop and wait protocol, sender sends one frame, waits until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame.

68. What is Stop-and-Wait Automatic Repeat Request?

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

69. What is usage of Sequence Number in Relaible Transmission?

The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame. Since we want to minimize the frame size, the smallest range that provides unambiguous communication. The sequence numbers can wrap around.

70. What is Pipelining ?

In networking and in other areas, a task is often begun before the previous task has ended. This is known as pipelining.

71. What is Sliding Window?

The sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver. In other words, he sender and receiver need to deal with only part of the possible sequence numbers.

72. What is Piggy Backing?

A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

73. What are the two types of transmission technology available?

(i) Broadcast and (ii) point-to-point

74. What is subnet?

A generic term for section of a large networks usually separated by a bridge or router.

75. Difference between the communication and transmission.

Transmission is a physical movement of information and concern issues like bit polarity, synchronisation, clock etc.

Communication means the meaning full exchange of information between two communication media.

76. What are the possible ways of data exchange?

(i) Simplex (ii) Half-duplex (iii) Full-duplex.

77. What is SAP?

Series of interface points that allow other computers to communicate with the other layers of network protocol stack.

78. What is meant by Router ?

A router is an electronic device that interconnects two ormore computer networks, and selectively interchangespackets of data between them . A router is a networkingdevice whose software and hardware are customized to thetasks of routing and forwarding information.

79. What is meant by Hubs ?

A common connection point for devices in a network. Hubsare commonly used to connect segments of a LAN. A hubcontains multiple ports. When a packet arrives at one port, itis copied to the other ports so that all segments of the LANcan see all packets.

80. What is meant by Bridges ?

A network bridge connects multiple network segments at the data link layer (layer 2) of the OSI model.Bridges come in three basic types:

Local bridges: Directly connect local area networks(LANs)

Remote bridges: Can be used to create a wide areanetwork (WAN) link between LANs. Remote bridges, where the connecting link is slower than the endnetworks, largely have been replaced with routers.

Wireless bridges: Can be used to join LANs orconnect remote stations to LANs.

81. What do u mean by NIC (Network Interface Card) ?

A network card, network adapter, or NIC (network interfacecard) is a piece of computer hardware designed to allowcomputers to communicate over a computer network. Itprovides physical access to a networking medium and oftenprovides a low-level addressing system through the use of MAC addresses.

82. What do u mean by Repeater ?

Arepeateris anelectronicdevice that receives a signal, cleans it of unnecessary noise, regenerates it, andretransmitsit at a higher power level, or to the other side of anobstruction, so that the signal can cover longer distanceswithout degradation

83. Definitions of Firewall ?

Firewalls are the most important aspect of the networkand its security in today's era. Due to maximization of attackson the networks from various groups stealing data, denyingservices etc the firewall is playing a vital roles in computernetworks.

84. What is the Difference between HUB and SWITCH ?

HUB-it is a network device that provide a centralconnection on for cables from work station , server etc.the hubtake incoming single one port and provide every port so thatis main reason for collision.switch.it also provide central connection to the work station.itprovide the unique cast.it is better than a hub.HUB HUb (1)work on physical layer in osi model(2) Hub is half dupllex(3) collision detection is on in hub(4) Hub broadcast of transmitted message.Switch (1) switch work in data link layer in osi model(2) switch work half and full duplex(3) collision detection is off (4) switch transmitted messagee unicast and some timebroadcast.

85. Difference between Physical Address andLogical Address ?

A Physical address is a 48-bit flat address burned into theROM of the NIC card which is a Layer1 device of the OSImodel. This iis divided into 24-bit vendor code and 24-bitserial address. This is unique for each system and cannot bechanged.A Logical address is a 32- bit address assigned to each systemin a network. This works in Layer-3 of OSI Model. Thiswould be generally the IP address.

86. What is MAC Address ?

A unique48 Bit address assigned to each network card. Itsalso called as physical address.

87. What is PING Utility ?

PING stands Packet Internet Gopher. This is a utility forensuring connectivity between computers . ICMP protocolworks behind this utility. Under it , sending node sendspackets to destination node and reply is received if there isproper communication between two.Ping command use to check the Destination host or routerconnetivity to use the icmp and echo packet.

88. What do u mean by Gateway ?

They relay packets among networks that have differentprotocols (e.g. between a LAN and a WAN). They accept apacket formatted for one protocol and convert it to a packetformatted for another protocol before forwarding it. Theyoperate in all seven layers of the OSI model