19CS5612 - NETWORKS LABORATORY

LIST OF EXPERIMENTS :

1. Learn to use commands like tcpdump, netstat, ifconfig, nslookup and traceroute. Capture ping and traceroute PDUs using a network protocol analyzer and examine.
2. Write a HTTP web client program to download a web page using TCP sockets
3. Applications using TCP sockets like:
    - Echo client and echo server
    - Chat
    - File Transfer
4. Simulation of DNS using UDP sockets.
5. Write a code simulating ARP /RARP protocols.
6. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS.
7. Study of TCP/UDP performance using Simulation tool.
8. Simulation of Distance Vector/ Link State Routing algorithm.
9. Performance evaluation of Routing protocols using Simulation tool.
10. Simulation of error correction code (like CRC).
11. Implementation of Stop and Wait Protocol and Sliding Window Protocol

TABLE OF CONTENTS

S.NO.	DATE	EXPERIMENT TITLE	MARKS/ 10	SIGN.
1.		**Networking Commands**		
2.		**Write a HTTP web client program to download a web page using TCP sockets**		
3a.		**Socket program For Echo**		
3b.		**Client- Server application for Chat**		
3c.		**File Transfer in Client & Server**		
4.		**Simulation of DNS using UDP sockets.**		
5a.		**Write a code simulating ARP protocols.**		
5b.		**Write a code simulating RARP protocols.**		
6.		**Study of Network simulator and Simulation of Congestion Control Algorithms using NS.**		
7.		**Study of TCP/UDP performance using Simulation tool.**		
8a.		**Simulation of Distance Vector Routing algorithm.**		
8b.		**Simulation of Link State Routing algorithm.**		
9.		**Simulation of error correction code (like CRC).**		
10.		**Performance evaluation of Routing protocols using Simulation tool.**		

DATE:

1.NETWORKING COMMANDS

AIM:

To study the basic networking commands.

NETWORKING COMMANDS:

C:\>arp –a: ARP is short form of address resolution protocol, It will show the IP address of your computer along with the IP address and MAC address of your router.

C:\>hostname: This is the simplest of all TCP/IP commands. It simply displays the name of your computer.

C:\>ipconfig: The ipconfig command displays information about the host (the computer your sitting at)computer TCP/IP configuration.

C:\>ipconfig /all: This command displays detailed configuration information about your TCP/IP connection including Router, Gateway, DNS, DHCP, and type of Ethernet adapter in your system.

C:\>Ipconfig /renew: Using this command will renew all your IP addresses that you are currently (leasing) borrowing from the DHCP server. This command is a quick problem solver if you are having connection issues, but does not work if you have been configured with a static IP address.

C:\>Ipconifg /release: This command allows you to drop the IP lease from the DHCP server.

C:\>ipconfig /flushdns: This command is only needed if you're having trouble with your networks DNS configuration. The best time to use this command is after network configuration frustration sets in, and you really need the computer to reply with flushed.

C:\>nbtstat –a: This command helps solve problems with NetBIOS name resolution. (Nbt stands for NetBIOS over TCP/IP)

C:\>netdiag: Netdiag is a network testing utility that performs a variety of network diagnostic tests, allowing you to pinpoint problems in your network. Netdiag isn't installed by default, but can be installed from the Windows XP CD after saying no to the install. Navigate to the CD ROM drive letter and open the support\tools folder on the XP CD and click the setup.exe icon in the support\tools folder.

C:\>netstat: Netstat displays a variety of statistics about a computers active TCP/IP connections. This tool is most useful when you're having trouble with TCP/IP applications such as HTTP, and FTP.

C:\>nslookup: Nslookup is used for diagnosing DNS problems. If you can access a resource by specifying an IP address but not it's DNS you have a DNS problem.

C:\>pathping: Pathping is unique to Window's, and is basically a combination of the Ping and Tracert commands. Pathping traces the route to the destination address then launches a 25 second test of each router along the way, gathering statistics on the rate of data loss along each hop.

C:\>ping: Ping is the most basic TCP/IP command, and it's the same as placing a phone call to your best friend. You pick up your telephone and dial a number, expecting your best friend to reply with "Hello" on the other end. Computers make phone calls to each other over a network by using a Ping command. The Ping commands main purpose is to place a phone call to another computer on the network, and request an answer. Ping has 2 options it can use to place a phone call to another computer on the network. It can use the computers name or IP address.

C:\>route: The route command displays the computers routing table. A typical computer, with a single network interface, connected to a LAN, with a router is fairly simple and generally doesn't pose any network problems. But if you're having trouble accessing other computers on your network, you can use the route command to make sure the entries in the routing table are correct.

C:\>tracert: The tracert command displays a list of all the routers that a packet has to go through to get from the computer where tracert is run to any other computer on the internet.

Viva Questions and Answers:

1. What is the use of arp –a?

It will show the IP address of your computer along with the IP address and MAC address of your router.

2. What is the use of netstat?

Netstat displays a variety of statistics about a computers active TCP/IP connections. This tool is most useful when you're having trouble with TCP/IP applications such as HTTP, and FTP.

3. What is the use of ipconfig?

The ipconfig command displays information about the host (the computer your sitting at)computer TCP/IP configuration.

4. What is the use of ping?

The Ping commands main purpose is to place a phone call to another computer on the network, and request an answer. Ping has 2 options it can use to place a phone call to another computer on the network. It can use the computers name or IP address.

5. What is the use of tracert?

The tracert command displays a list of all the routers that a packet has to go through to get from the computer where tracert is run to any other computer on the internet.

RESULT:

Thus the above list of primitive has been studied.

DATE:

2.Write a HTTP web client program to download a web page using TCP sockets

AIM:

To Write a HTTP web client program to download a web page using TCP sockets.

ALGORITHM:

CLIENT SIDE:

1) Start the program.
2) Create a socket which binds the Ip address of server and the port address to acquire service.
3) After establishing connection send the url to server.
4) Open a file and store the received data into the file.
5) Close the socket.
6) End the program.

SERVER SIDE

1) Start the program.
2) Create a server socket to activate the port address.
3) Create a socket for the server socket which accepts the connection.
4) After establishing connection receive url from client.
5) Download the content of the url received and send the data to client.
6) Close the socket.
7) End the program.

PROGRAM

```
import javax.swing.*;
import  java.net.*;
import java.awt.image.*;
import javax.imageio.*;
import java.io.*;
import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;
public class HTTPClient
{
public static void main(String args[]) throws Exception
{
Socket  soc;
BufferedImage img = null;
soc=new
Socket("localhost",4000);
System.out.println("Client is running.");
try {
System.out.println("Reading image from disk.");
img = ImageIO.read(new File("digital-image-processing.jpg"));
ByteArrayOutputStream baos = new ByteArrayOutputStream();
ImageIO.write(img, "jpg", baos);
baos.flush();
byte[] bytes = baos.toByteArray(); baos.close();
System.out.println("Sending image to server.");
OutputStream out = soc.getOutputStream();
DataOutputStream dos = new DataOutputStream(out);
dos.writeInt(bytes.length);
dos.write(bytes, 0, bytes.length);
System.out.println("Image sent to server.");
dos.close();
out.close();
}
catch (Exception e)
{
System.out.println("Exception: " + e.getMessage());
soc.close();
}
soc.close();
}
}
```

SERVER PROGRAM

```
import java.net.*;
import java.io.*;
import java.awt.image.*;
import javax.imageio.*;
import javax.swing.*;
class HTTPServer
{
public static void main(String args[]) throws Exception
{
ServerSocket server=null;
Socket socket;
server=new ServerSocket(4000);
System.out.println("Server Waiting for image");
socket=server.accept();
System.out.println("Client connected.");
InputStream in = socket.getInputStream();
DataInputStream dis = new DataInputStream(in);
int len = dis.readInt();
System.out.println("Image Size: " + len/1024 + "KB");
byte[] data = new byte[len];
dis.readFully(data);
dis.close();
in.close();
InputStream ian = new ByteArrayInputStream(data);
BufferedImage bImage = ImageIO.read(ian);
JFrame f = new JFrame("Server");
ImageIcon icon = new ImageIcon(bImage);
JLabel l = new JLabel();
l.setIcon(icon);
f.add(l);
f.pack();
f.setVisible(true);
}
}
```

OUTPUT

SERVER OUTPUT

```
C:\Program Files\Java\jdk1.5.0\bin>javac HTTPServer.java
C:\Program Files\Java\jdk1.5.0\bin>java HTTPServer
Server Waiting for image
Client connected.
```

CLIENT OUTPUT

```
C:\Program Files\Java\jdk1.5.0\bin>javac HTTPClient.java
C:\Program Files\Java\jdk1.5.0\bin>java HTTPClient

Client is running.
Reading image from disk.
Image size :29kb
```

Viva Questions and Answers:

1. What is meant by socket?

A socket is a software object that acts as an end point establishing a bidirectional network communication link between a server-side and a client-side program.

2. Define HTTP.

HTTP means HyperText Transfer Protocol. HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

3. What is meant by client server communication?

Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

4. Define webpage.

A web page or webpage is a document commonly written in HTML (Hypertext Markup Language) that is accessible through the Internet or other networks using an Internet browser.

5. Define URL.

URL stands for Uniform Resource Locator, and is used to specify addresses on the World Wide Web. A URL is the fundamental network identification for any resource connected to the web (e.g., hypertext pages, images, and sound files). The protocol specifies how information from the link is transferred.

RESULT:

The webpage is successfully downloaded and the contents are displayed and verified.

DATE: 3. a. SOCKET PROGRAM FOR ECHO

Aim

To implement echo server and client in java using TCP sockets.

Algorithm :

Server

1. Create a server socket.
2. Wait for client to be connected.
3. Read text from the client
4. Echo the text back to the client.
5. Repeat steps 4-5 until 'bye' or 'null' is read.
6. Close the I/O streams
7. Close the server socket
8. Stop

Client

1. Create a socket and establish connection with the server
2. Get input from user.
3. If equal to bye or null, then go to step 7.
4. Send text to the server.
5. Display the text echoed by the server
6. Repeat steps 2-4
7. Close the I/O streams
8. Close the client socket
9. Stop

PROGRAM:

ECHO CLIENT

```
import java.net.*;
import java.io.*;
public class echoclient
{
public static void main(String[] args) throws IOException
 {
BufferedReader fromServer = null, fromUser = null;
 PrintWriter toServer = null;
Socket sock = null;
try {
 if (args.length == 0)
sock = new Socket(InetAddress.getLocalHost(), 4000);
else
sock = new Socket(InetAddress.getByName(args[0]), 4000);
fromServer = new BufferedReader(new InputStreamReader(sock.getInputStream()));
 fromUser = new BufferedReader(new InputStreamReader(System.in));
toServer = new PrintWriter(sock.getOutputStream(), true
String Usrmsg, Srvmsg;
System.out.println("Type \"bye\" to quit");
while (true)
{
System.out.print("Enter msg to server : ");
Usrmsg = fromUser.readLine();
if (Usrmsg==null || Usrmsg.equals("bye"))
{
toServer.println("bye");
break;
}
else
toServer.println(Usrmsg);
 Srvmsg = fromServer.readLine();
System.out.println(Srvmsg);
}
fromUser.close();
fromServer.close();
toServer.close();
sock.close();
}
catch (IOException ioe)
{
System.err.println(ioe);
}
```

Echo Server:

```
import java.net.*;
import  java.io.*;
public class echoserver
{
public static void main(String[] arg) throws IOException
 {
ServerSocket sock = null;
BufferedReader fromClient = null;
OutputStreamWriter toClient = null;
Socket client = null;
try {
sock = new ServerSocket(4000);
System.out.println("Server Ready");
client = sock.accept();
System.out.println("Client Connected");
FromClient = new BufferedReader(new InputStreamReader(client.getInputStream()));
toClient = new OutputStreamWriter(client.getOutputStream());
String line;
while (true)
{
line = fromClient.readLine();
if ( (line == null) || line.equals("bye"))
break;
System.out.println ("Client [ " + line + " ]");
toClient.write("Server [ "+ line +" ]\n");
toClient.flush();
 }
 fromClient.close();
toClient.close();
client.close();
sock.close();
System.out.println("Client Disconnected");
}
catch (IOException ioe)
{
System.err.println(ioe);
} } }
```

OUTPUT:

Server

```
$ javac echoserver.java
$ java tcpechoserver
Server Ready
Client Connected
Client [ hello ]
Client [ how are you ]
Client [ i am fine ]
Client [ ok ]
Client Disconnected
```

Client

```
$ javac echoclient.java
$ java tcpechoclient
Type "bye" to quit
Enter msg to server : hello
Server [ hello ]
Enter msg to server : how are you
Server [ how are you ]
 Enter msg to server : i am fine
Server [ i am fine ]
 Enter msg to server : ok
Server [ ok ]
Enter msg to server : bye
```

<u>Viva Questions and Answers:</u>

1. what is meant by socket programming?

Socket programs are used to communicate between various processes usually running on different systems. It is mostly used to create a client-server environment.

2. what is the use of java.net?

The java.net package provides 4 kinds of Sockets: Socket is a TCP client API, and will typically be used to connect to a remote host. ServerSocket is a TCP server API, and will typically accept connections from client sockets. DatagramSocket is a UDP endpoint API and is used to send and receive datagram packets.

3. Define echo.

Echo describes when data is sent to a computer or other networkdevices, and that information is sent back to verify the information was received. Echo can be a command used with operating systems, network devices, or a function of a device.

RESULT:

Thus the program for simulation of echo server was written & executed.

DATE:

3.B. CLIENT- SERVER APPLICATION FOR CHAT

Aim

To implement a chat server and client in java using TCP sockets.

Algorithm

Server

1. Create a server socket
2. Wait for client to be connected.
3. Read Client's message and display it
4. Get a message from user and send it to client
5. Repeat steps 3-4 until the client sends "end"
6. Close all streams
7. Close the server and client socket
8. Stop

Client

1. Create a client socket and establish connection with the server
2. Get a message from user and send it to server
3. Read server's response and display it
4. Repeat steps 2-3 until chat is terminated with "end" message
5. Close all input/output streams
6. Close the client socket
7. Stop

PROGRAM:

```
// TCP Chat Server--tcpchatserver.java
import java.io.*;
import java.net.*;
class tcpchatserver
{
public static void main(String args[])throws Exception
{
PrintWriter  toClient;
BufferedReader  fromUser,  fromClient;
 try
{
ServerSocket Srv = new ServerSocket(5555);
System.out.print("\nServer started\n");
Socket Clt = Srv.accept();
System.out.println("Client connected");
toClient = new PrintWriter(new BufferedWriter(new OutputStreamWriter(Clt.getOutputStream())), true);
fromClient = new BufferedReader(new InputStreamReader(Clt.getInputStream()));
fromUser = new BufferedReader(new InputStreamReader(System.in));
String CltMsg, SrvMsg;
while(true)
{
CltMsg= fromClient.readLine();
if(CltMsg.equals("end"))
break;
 else
{
 System.out.println("\nServer <<< " + CltMsg);
System.out.print("Message to Client : ");
SrvMsg = fromUser.readLine();
toClient.println(SrvMsg);
} }
System.out.println("\nClient Disconnected");
 fromClient.close();
toClient.close();
fromUser.close();
Clt.close();

Srv.close();
}
catch (Exception E)
{
System.out.println(E.getMessage());
 } } }
```

// TCP Chat Client--tcpchatclient.java

```
 import java.io.*;
 import java.net.*;
 class tcpchatclient
{
public static void main(String args[])throws Exception
 {
Socket Clt; PrintWriter toServer;
BufferedReader fromUser, fromServer;
try {
if (args.length > 1)
{
System.out.println("Usage: java hostipaddr");
 System.exit(-1); }
if (args.length == 0)
Clt = new Socket(InetAddress.getLocalHost(),5555);
 else
Clt = new Socket(InetAddress.getByName(args[0]), 5555);
toServer = new PrintWriter(new BufferedWriter(new OutputStreamWriter(Clt.getOutputStream())), true);
fromServer = new BufferedReader(new InputStreamReader(Clt.getInputStream()));
 fromUser = new BufferedReader(new InputStreamReader(System.in));
String CltMsg, SrvMsg;
System.out.println("Type \"end\" to Quit");
while (true) { System.out.print("\nMessage to Server : ");
CltMsg = fromUser.readLine();
toServer.println(CltMsg);
if (CltMsg.equals("end")) break;
 SrvMsg = fromServer.readLine();
System.out.println("Client <<< " + SrvMsg);
 } }
 catch(Exception E) {
System.out.println(E.getMessage());
} } }
```

OUT PUT :

SERVER

```
$ javac tcpchatserver.java
$ java tcpchatserver
Server started
Client connected
Server <<< hi
Message to Client : hello
 Server <<< how r u?
Message to Client : fine
Server <<< me too
Message to Client : bye
Client Disconnected
```

CLIENT

```
 $ javac tcpchatclient.java
$ java tcpchatclient
Type "end" to Quit
Message to Server : hi
Client <<< hello
Message to Server : how r u?
Client <<< fine
Message to Server : me too
 Client <<< bye
Message to Server : end
```

Viva Questions and Answers:

1. Define chat.

A real-time communication via keyboard between two or more users on a local network (LAN) or over the Internet

2. Define TCP.

TCP/IP stands for Transmission Control Protocol/Internet Protocol, which is a set of networking protocols that allows two or more computers to communicate. The Defense Data Network, part of the Department of Defense, developed TCP/IP, and it has been widely adopted as a networking standard.

3. what is socket API?

A socket API is an application programming interface (API), usually provided by the operating system, that allows application programs to control and use network sockets. Internet socket APIs are usually based on the Berkeley sockets standard.

RESULT:

Thus the above program a client-server application for chat using TCP / IP was executed and successfully.

DATE: 3.C. FILE TRANSFER IN CLIENT & SERVER

AIM:

To Perform File Transfer in Client & Server Using TCP/IP.

ALGORITHM:

CLIENT SIDE

1. Start.
2. Establish a connection between the Client and Server.
3. Socket ss=new Socket(InetAddress.getLocalHost(),1100);
4. Implement a client that can send two requests.
   i) To get a file from the server.
   ii) To put or send a file to the server.
5. After getting approval from the server ,the client either get file from the server or send
6. file to the server.

SERVER SIDE

1. Start.
2. Implement a server socket that listens to a particular port number.
3. Server reads the filename and sends the data stored in the file for the'get' request.
4. It reads the data from the input stream and writes it to a file in the server for the 'put' instruction.
5. Exit upon client's request.
6. Stop.

PROGRAM:

CLIENT SIDE

```
import java.net.*;
import java.io.*;
class MyClient {
  public static void main(String srgs[])throws IOException
  {
    Socket s=null;
    BufferedReader get=null;
    PrintWriter put=null;
    try
    {
      s=new Socket("127.0.0.1",8081);
      get=new BufferedReader(new InputStreamReader(s.getInputStream()));
      put=new PrintWriter(s.getOutputStream(),true);
    }
    catch(Exception e)
    {
      System.exit(0);
    }
    String u,f;
    System.out.println("Enter the file name to transfer from server:");
    DataInputStream dis=new DataInputStream(System.in);
    f=dis.readLine();
    put.println(f);
    File f1=new File("f:\\output");
    FileOutputStream  fs=new FileOutputStream(f1);
while((u=get.readLine())!=null)
      {
         byte jj[]=u.getBytes();
         fs.write(jj);
      }
      fs.close();
      System.out.println("File received");
      s.close();
   }
}
```

SERVER SIDE

```
import java.io.*;
import java.net.*;
public class FileServer {
   public static void main(String args[])throws IOException
   {
      ServerSocket ss=null;
      try
      {
         ss=new ServerSocket(8081);
      }
      catch(IOException e)
      {
         System.out.println("couldn't listen");
         System.exit(0);
      }
      Socket cs=null;
      try
      {
         cs=ss.accept();
         System.out.println("Connection established"+cs);
      }
      catch(Exception e)
      {
         System.out.println("Accept failed");
         System.exit(1);
      }
      PrintWriter put=new PrintWriter(cs.getOutputStream(),true);
      BufferedReader st=new BufferedReader(new InputStreamReader(cs.getInputStream()));
      String s=st.readLine();
      System.out.println("The requested file is : "+s);
      File f=new File(s);
      if(f.exists())
      {
         BufferedReader d=new BufferedReader(new FileReader(s));
         String line;
         while((line=d.readLine())!=null)
   {
            put.write(line);
            put.flush();
          }
          d.close();
          System.out.println("File transfered");
          cs.close();
          ss.close();
        }
    }
}
```

OUTPUT:

SERVER OUTPUT

```
C:\Program Files\Java\jdk1.6.0\bin>javac
FileServer.java
C:\Program Files\Java\jdk1.6.0\bin>java FileServer

Connection establishedSocket[addr=/127.0.0.1,port=50986,localport=8081]
The requested file is : network.txt
File transfered
```

CLIENT OUTPUT

```
C:\Program Files\Java\jdk1.6.0\bin>javac
MyClient.java C:\Program
Files\Java\jdk1.6.0\bin>java MyClient

Enter the file name to transfer from server: network.txt
File received
```

Viva Questions and Answers:

1. What is socket address?

A socket address is the combination of an IP address and a port number, much like one end of a telephone connection is the combination of a phone number and a particular extension. Based on this address, internet sockets deliver incoming data packets to the appropriate application process or thread.

2. What are the local socket address?

Local socket address: Local IP address and port number

3.Define File transfer.

File transfer is the process of copying or moving a file from one computer to another over a network or Internet connection. It enables sharing, transferring or transmitting a file or a logical data object between different users and/or computers both locally and remotely.

4. what is called as socket pairs?

Communicating local and remote sockets are called socket pairs.

RESULT:

Thus the File transfer Operation is done & executed successfully.

DATE: 4. Simulation of DNS using UDP sockets.

AIM:

To write a program to Simulation of DNS using UDP sockets..

ALGORITHM:

1. Start the program.
2. Get the frame size from the user
3. To create the frame based on the user request.
4. To send frames to server from the client side.
5. If your frames reach the server it will send ACK signal to client otherwise it will send NACK signal to client.
6. Stop the program

PROGRAM:

/ UDP DNS Server
Udpdnsserver

```
java import java.io.*;
import java.net.*;
 public class udpdnsserver
{
private static int indexOf(String[] array, String str)
 {
str = str.trim();
for (int i=0; i < array.length; i++)
{
if (array[i].equals(str)) return i;
 }
return -1;
 }
public static void main(String arg[])throws IOException
 {
String[] hosts = {"yahoo.com", "gmail.com","cricinfo.com", "facebook.com"};
String[] ip = {"68.180.206.184", "209.85.148.19","80.168.92.140", "69.63.189.16"};
System.out.println("Press Ctrl + C to Quit");
 while (true){
DatagramSocket serversocket=new DatagramSocket(1362);
byte[] senddata = new byte[1021];
 byte[] receivedata = new byte[1021];
 DatagramPacket recvpack = new DatagramPacket(receivedata, receivedata.length);
 serversocket.receive(recvpack);
String sen = new String(recvpack.getData()); InetAddress ipaddress =
 recvpack.getAddress(); int port = recvpack.getPort();
 String capsent;
 System.out.println("Request for host " + sen);
if(indexOf (hosts, sen) != -1) capsent = ip[indexOf
(hosts, sen)]; else capsent = "Host Not Found";
senddata = capsent.getBytes();
 DatagramPacket pack = new DatagramPacket (senddata, senddata.length,ipaddress,port);
serversocket.send(pack);
serversocket.close();
}
}
}
```

//UDP DNS Client –

```
 Udpdnsclient
.java import java.io.*; import java.net.*;
 public class udpdnsclient
 {
 public static void main(String args[])throws IOException
 {
 BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
 DatagramSocket clientsocket = new DatagramSocket();
 InetAddress ipaddress; if (args.length == 0)
 ipaddress = InetAddress.getLocalHost(); else
 ipaddress = InetAddress.getByName(args[0]); byte[] senddata =
 new byte[1024];
byte[] receivedata = new byte[1024];
int portaddr = 1362;
System.out.print("Enter the hostname : ");
String sentence = br.readLine();
Senddata = sentence.getBytes();
DatagramPacket pack = new DatagramPacket(senddata,senddata.length, ipaddress,portaddr);
 clientsocket.send(pack);
DatagramPacket recvpack =new DatagramPacket(receivedata,receivedata.length);
clientsocket.receive(recvpack);
String modified = new String(recvpack.getData());
System.out.println("IP Address: " + modified);
clientsocket.close();
}
}
```

OUTPUT :

Server

```
 javac udpdnsserver.java
 java udpdnsserver
 Press Ctrl + C to Quit Request for host yahoo.com
 Request for host cricinfo.com
 Request for host youtube.com
```

Client

```
 javac udpdnsclient.java
 java udpdnsclient
Enter the hostname : yahoo.com
IP Address: 68.180.206.184
java udpdnsclient
Enter the hostname : cricinfo.com
IP Address: 80.168.92.140
 java udpdnsclient
 Enter the hostname : youtube.com
IP Address: Host Not Found
```

Viva Questions and Answers:

1.Define DNS.

DNS. (Domain Name System) The Internet's system for converting alphabetic names into numeric IP addresses. For example, when a Web address (URL) is typed into a browser, DNSservers return the IP address of the Web server associated with that name.

2.Define UDP.

User Datagram Protocol (UDP) is part of the Internet Protocol suite used by programs running on different computers on a network. UDP is used to send short messages called datagrams but overall, it is an unreliable, connectionless protocol.

3. what is meant by MACaddress?

A MAC address is a hardware identification number that uniquely identifies each device on a network.

4. What is meant by IP address?

An Internet Protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. An IP address serves two principal functions: host or network interface identification and location addressing.

5. List the IP address class.

Class	1st Octet Decimal Range	Default Subnet Mask
A	1 – 126*	255.0.0.0
B	128 – 191	255.255.0.0
C	192 – 223	255.255.255.0
D	224 – 239	Reserved for Multicasting

RESULT:

Thus the above program a client-server application for chat using UDP was executed and successfully

DATE: 5.a.Write a code simulating ARP protocols.

AIM

To implement Address Resolution Protocol .

ALGORITHM

CLIENT SIDE

1. Establish a connection between the Client and Server. Socket ss=new Socket(InetAddress.getLocalHost(),1100);
2. Create instance output stream writer PrintWriter ps=new PrintWriter(s.getOutputStream(),true);
3. Get the IP Address to resolve its physical address.
4. Send the IPAddress to its output Stream.ps.println(ip);
5. Print the Physical Address received from the server.

SERVER SIDE

1. Accept the connection request by the client. ServerSocket ss=new ServerSocket(2000);Socket s=ss.accept();
2. Get the IPaddress from its inputstream. BufferedReader br1=new BufferedReader(newInputStreamReader(s.getInputStream())); ip=br1.readLine();
3. During runtime execute the processRuntime r=Runtime.getRuntime(); Process p=r.exec("arp -a "+ip);
4. Send the Physical Address to the client.

PROGRAM

ARP CLIENT

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

class ArpClient
{
public static void main(String args[])throws IOException
{
  try
{
Socket ss=new Socket(InetAddress.getLocalHost(),1100);
PrintStream ps=new PrintStream(ss.getOutputStream());
BufferedReader br=new BufferedReader(newInputStreamReader(System.in));
                String ip;
                System.out.println("Enter the IPADDRESS:");
```

```
                    ip=br.readLine();
ps.println(ip);
String str,data;
BufferedReader br2=new
BufferedReader(newInputStreamReader(ss.getInputStream()));
System.out.println("ARP From Server::"); do
{
str=br2.readLine();
System.out.println(str);
}
while(!(str.equalsIgnoreCase("end")));
}
catch(IOException e)
{
System.out.println("Error"+e);

}}}
```

ARP SERVER

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

class ArpServer
{
public static void main(String args[])throws IOException
{
try
{
ServerSocket ss=new ServerSocket(1100);
Socket s=ss.accept();
PrintStream ps=new PrintStream(s.getOutputStream());
BufferedReader br1=new BufferedReader(newInputStreamReader(s.getInputStream()));
String ip;
 ip=br1.readLine();
Runtime r=Runtime.getRuntime();
Process p=r.exec("arp -a "+ip);
BufferedReader br2=new BufferedReader(newInputStreamReader(p.getInputStream()));
String str;
while((str=br2.readLine())!=null)
{
ps.println(str);
}}
catch(IOException e)
{
System.out.println("Error"+e); }}}
```

OUTPUT

```
C:\Networking Programs>java ArpServer
C:\Networking Programs>java ArpClient
Enter the IPADDRESS:
192.168.11.58
ARP From Server::
Interface: 192.168.11.57 on Interface 0x1000003
Internet Address Physical Address     Type
192.168.11.58    00-14-85-67-11-84 dynamic
```

Viva Questions and Answers:

1. Define ARP.

The address resolution protocol (**arp**) is a protocol used by the Internet Protocol (IP) [RFC826], specifically IPv4, to map IP network addresses to the hardware addresses used by a data link protocol.

2. Define IPv4.

IPv4 (**Internet Protocol Version 4**) is the fourth revision of the Internet Protocol (IP) used to to identify devices on a network through an addressing system.

RESULT

Thus the implementation of ARP is done & executed successfully.

DATE: 5.b.Write a code simulating RARP protocols.

AIM:

To write a java program for simulating RARP protocols.

ALGORITHM:

CLIENT

1. Start the program
2. using datagram sockets UDP function is established.

2.Get the MAC address to be converted into IP address.
3.Send this MAC address to server.
4.Server returns the IP address to client.

SERVER

1. Start the program.
2. Server maintains the table in which IP and corresponding MAC addresses are stored.
3. Read the MAC address which is send by the client.
4. Map the IP address with its MAC address and return the IP address to client.

CLIENT:

```
import java.io.*; import
java.net.*; import
java.util.*; class
Clientrarp12
{
        public static void main(String args[])
        {
        try
        {
                DatagramSocket client=new DatagramSocket();
                InetAddress addr=InetAddress.getByName("127.0.0.1");
                byte[] sendbyte=new byte[1024];
                byte[] receivebyte=new byte[1024];
                BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
                System.out.println("Enter the Physical address (MAC):");
                String str=in.readLine();
                sendbyte=str.getBytes();
          DatagramPacket sender=newDatagramPacket(sendbyte,sendbyte.length,addr,1309);
                client.send(sender);
             DatagramPacket receiver=new DatagramPacket(receivebyte,receivebyte.length);
client.receive(receiver);
                String s=new String(receiver.getData()); System.out.println("The
                Logical Address is(IP): "+s.trim()); client.close();
        }


        catch(Exception e)
        {
```

```
        System.out.println(e);
    }
    }
}
```

SERVER:

```
import java.io.*; import
java.net.*; import
java.util.*; class
Serverrarp12
{
    public static void main(String args[])
    {
    try
    {
        DatagramSocket server=new DatagramSocket(1309);
        while(true)
        {
        byte[] sendbyte=new byte[1024];
        byte[] receivebyte=new byte[1024];
         DatagramPacket receiver=new DatagramPacket(receivebyte,receivebyte.length);
            server.receive(receiver);
            String str=new String(receiver.getData()); String
            s=str.trim();
            InetAddress addr=receiver.getAddress();
            int port=receiver.getPort();
            String ip[]={"165.165.80.80","165.165.79.1"};
            String
            mac[]={"6A:08:AA:C2","8A:BC:E3:FA"};
            for(int i=0;i<ip.length;i++)
            {
                if(s.equals(mac[i]))
                {
             sendbyte=ip[i].getBytes();
DatagramPacket  sender=newDatagramPacket(sendbyte,sendbyte.length,addr,port);

                    server.send(sender);
                    break;
                }
            }
            break;
        }
    }
    catch(Exception e)
    {
        System.out.println(e);
```

```
          }
          }
}
```

OUTPUT:

```
I:\ex>java Serverrarp12
I:\ex>java Clientrarp12
Enter the Physical address (MAC):
6A:08:AA:C2
The Logical Address is(IP): 165.165.80.80
```

Viva Questions and Answers:

1. Define RARP.
RARP (Reverse Address Resolution Protocol) is a protocol by which a physical machine in a local area network can request to learn its IP address from a gateway server's Address Resolution Protocol (ARP)
table .

2. What is the use of RARP?
RARP is used for the opposite purpose; namely, to convert an ethernet address to an IP address. Its linux-network-administration purpose is to enable diskless machines, such as X workstations, to find out their IP address at boot time.

RESULT:

Thus the implementation of RARP is done & executed successfully.

DATE: 6. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS.

OBJECTIVE:

To study of network simulator (ns) and simulation of congestion control algorithms using ns.

SYSTEM REQUIREMENTS:

PC: Pentium or higher

One LAN card onboard or on PCI slot with 10/100Mbps speed.
128MB RAM

500MB free space on Hard
drive CD ROM drive

Serial port, LPT port & USB port installed on system Operating
System: Windows 2000 or higher

THEORY:

LTS-01 Local area network / wireless local area network trainer system:

It is designed to help students understand the basic concepts, modes of operation and protocols involved in networking. The trainer has integrated hardware flow control on panel board for better understanding of different types of LAN topologies involved in networking. The trainer system is provided with windows-based user friendly software with analysis of protocols, different layers, network and measurement of error rate and throughput.

Students can easily do connections in different topologies and can learn actual data transfer either through hardware or through simulated network concept. Facility is provided into system software to introduce errors into packets being sent and analyze the effect of error on different protocols and hence find the effect on through put graph as well.

Trainer and its various types of experimentation using this system. This system works into server-client base. For any topology user has to select one server and select the network type whether it is LAN or WLAN. To understand the topology concept user can connect two or more clients to hardware. Depending on the topology selected user will have option to select protocols for the selected topology. Upon selection of protocol user can then create network of connected computers.

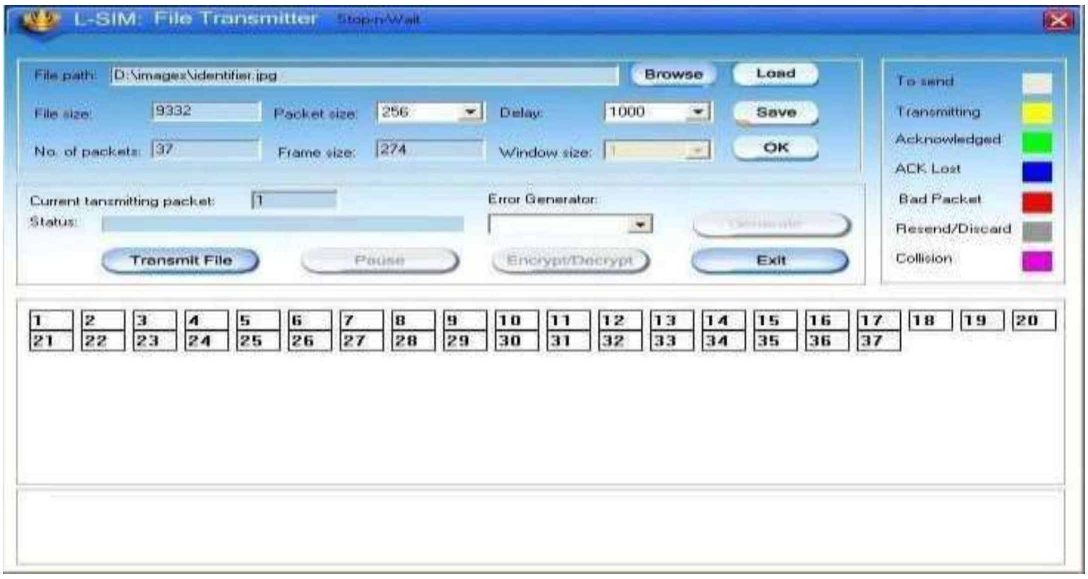
In any network which is created by user server can send or can communicate with any of the clients however clients can communicate only with server, no client to client communication is possible. Transmitter port protocol & network analysis can be done after communication is over between server and clients. Throughput v/s Packet size graph can be plotted for which at least two file transfers should be carried out. This plot can be printed to attach in the lab exercise sheet.

For the LAN network LAN cards must be installed prior to start work on this trainer. For wire less LAN USB ports should be available on the computers which are to be used for experimentation.In WLAN wireless access cards gets connected to computer USB ports and access point gets connected to hardware device.

L-SIM LAN Protocol Simulator & Analyzer Software:

It is designed to teach the basic concepts, topologies & various protocols involved in networking. The software is provided with analysis of protocols, different layers, network and measurement of error rate and throughput. Facility is provided to introduce errors into packets beingsent and analyze the effect of error on different protocols and hence find the effect on throughput graph as well. Software is supported with neat operating instruction manual and online help.

MODEL WINDOW DIAGRAM FOR L-SIM



L-SIM: File Transmitter Stop-n-Wait

File path: D:\images\identifier.jpg

File size: 9332

Packet size: 256

Delay: 1000

No. of packets: 37

Frame size: 274

Window size: 1

Current tanzmitting packet: 1

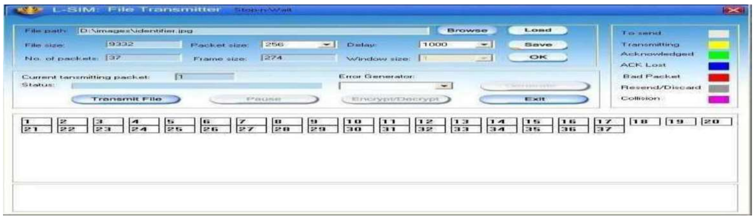
Status:

Error Generator:

N-S IM Network simulation software:

It is developed to provide basic understanding and implementation of various advanced concepts in networking. The software provides an opportunity to understand network fundamentals through animations & simulations. The simulation provides for network experimentation with various LAN and WAN protocols, network devices, routers, encryption, decryption, file transfer, error insertion and analysis of error rate and throughput etc. This software covers Ethernet LAN, wireless LAN and router. All networking theory is explained using simulation and animation.

MODEL WINDOW DIAGRAM FOR N-SIM



L-SIM: File Transmitter Stop-n-Wait

File path: D:\images\identifier.jpg

File size: 9332 Packet size: 256 Delay: 1000

No. of packets: 37 Frame size: 274 Window size: 1

Current tansmitting packet: 1

Status:

Error Generator:

Rapid advances in computer & communication technologies have resulted in the increasing merger of these two fields. The lines have blurred among computing, switching & digital transmission equipment; and the same digital techniques are being used for data, audio & video transmission. Merging & evolving technologies, coupled with increasing demands for efficient & timely collection, processing & dissemination of information, have led to the development of integrated systems that transmit & process all types of data.

These intgrated systems are broadly divided as follows

- **DATA COMMUNICATION** dealing with transmission, transmission media, signal decoding, interfacing, data link control & multiplexing

- **NETWORKING** deals with the technology & architecture of communication network

- **COMMUNICATION PROTOCOLS** which covers the architecture as well as analysis of individual protocols at various layers depending on the hardware & software Network laboratory is designed & developed considering the curriculum offered by Anna University. Trainers offered under network laboratory are designed for students at all level to study and understand all the concepts of data communication, data transfer using serial and parallel ports, Ethernet and wireless LAN with complete protocol understanding and actual hands on with hardware & software with ease.

Network laboratory consists of DCT-03 Data communication trainer kit, LTS- 01 LAN / Wireless LAN training system, L-SIM LAN / WLAN protocol simulator and analyzer software & N-SIM Network simulation software.

The DCT-03: Data communication trainer is a unique trainer kit for the development of exercises and theoretical-experimental courses to understand the basic concept and working of modes and protocols in serial and parallel communication.

The trainer kit consists of functional blocks for serial and parallel communication system.

The trainer kit is highly innovative from a technological as well as an educational point of view. The trainer kit is used as “basic unit” to examine all the peculiar operating standards of serial and parallel communication system. The only external equipments required are two Computers with serial and parallel communication ports and an Oscilloscope. Utmost care has been laid in the design and quality control of all circuits, to ensure the repeatability of the results of the experiments.

Data communication is a term referred when the sender and receiver are digital devices, which communicate with each other by means of binary information. The objective of this trainer kit is to clear the various aspects of the data communications which comprise of

- The information source or sender.
- The medium for carrying information.
- The information receiver.
- The communication protocols, which ensure proper transfer of data.

With an increasing demand in information exchange the field of data communication technique is emerging as the only solution, to satisfy the various needs of today's communication sector and to achieve very high bandwidth along with highest accuracy. The communication media is shifting from analog signal transfer towards digital communication.

With PC becoming the biggest storage devices in digital form, it becomes the main source and destination for information exchange. With rapid growth in both the communication technologies as well as computer hardware and software technologies, these two fields are merged to form a data communication network. Now the digital data is used for data, voice and image transmission.

Depending upon the application the communication link can be of point to point communication between two devices or a multipoint communication between at least 3 devices and data transfer can be serial or in parallel form.

Viva Questions and Answers:

1. Define NS.

A Network Simulator is an object oriented, event driven discrete simulator written in C++, with an object oriented tool command language (OTCL) interpreter as a front end.

2. What protocols does NS support?

A lot! Almost all variants of TCP, several forms of multicast, wired networking, several ad hoc routing protocols and propagation models (but not cellular phones), data diffusion, satellite, and other stuff. See the documentation (described above) for details, or download ns and look.

3. Define data communication.

Data communication is a term referred when the sender and receiver are digital devices, which communicate with each other by means of binary information.

4. List of Network Simulators:

Ns2 (Network Simulator 2).
Ns3 (Network Simulator 3).
OPNET.
OMNeT++.
NetSim.

REAL.
QualNet.
J-Sim.

5. Define OPNET.

OPNET Network Simulator. OPNET Network simulator is a tool to simulate the behavior and performance of any type of network. The main difference Opnet Network Simulator comparing to other simulators lies in its power and versatility. IT Guru provides pre-built models of protocols and devices.

RESULT:

The study of network simulator (ns) and simulation of congestion control algorithms using ns is executed and verified.

DATE: 7. Study of TCP/UDP performance using Simulation tool.

AIM:

To Study of TCP/UDP performance using Simulation tool.

TOOLS USED:

Opnet Simulator

INTRODUCTION:

The transport layer protocols provide connection- oriented sessions and reliable data delivery services. This paper seeks to reflect a comparative analysis between the two transport layer protocols, which are TCP/IP and UDP/IP, as well to observe the effect of using these two protocolsin a client server network. The similarities and differences between TCP and UDP over the Internet are also presented in our work. We implement a network structure using Opnet Modeler and finally, based on the practical results obtained we present the conclusions-showing the difference between these two protocols and how they work.

The transport layer is not just another layer. It is the heart of the whole protocol hierarchy. Its task is to provide reliable, cost-effective data transport from the source machine to the destination machine, independently of the physical network or networks currently in use.

TCP and UDP are transport layer components that provide the connection point through which applications access network services. TCP and UDP use IP, which is a lower-layer best effort delivery service. IP encapsulates TCP packets and UDP datagrams and delivers this information across router-connected internet works.

The ultimate goal of the transport layer is to provide efficient, reliable, and cost-effective service to its users, normally processes in the application layer. To achieve this goal, the transport layer makes use of the services provided by the network layer. Without the transport layer, the whole concept of layered protocols would make little sense e.g. The Transport Layer prepares applications data for transport over the network and processes network data to be used by applications. It is responsible for the end-to-end transfer of data over the network and is the four of the OSI model. The Transport layer meets a number of functions:

- enabling the applications to communicate over the network at the same time when using a single device;
- ensure that all amount of data is receive by the correct application;
- responsible for fragmentation and reassembly;
- develop mechanism for handling errors.

<u>Comparison Between TCP And UDP</u>

Service	**TCP**	**UDP**
Flow controls	The receiver can signal the sender to slow down.	ACKs, which are used in TCP to control packet flow, are not returned.
Connection setup	It takes time, but with TCP reliability is ensured.	No connection is required.
Guaranteed message delivery	Returns acknowledgments.	UDP does not return ACKs, the receiver can’t signal that packets have been successfully delivered.
Congestion controls	Network devices can take advantage of TCP ACK to control the behavior of sender.	If ACK, are missing, the network cannot signal congestion to the sender.

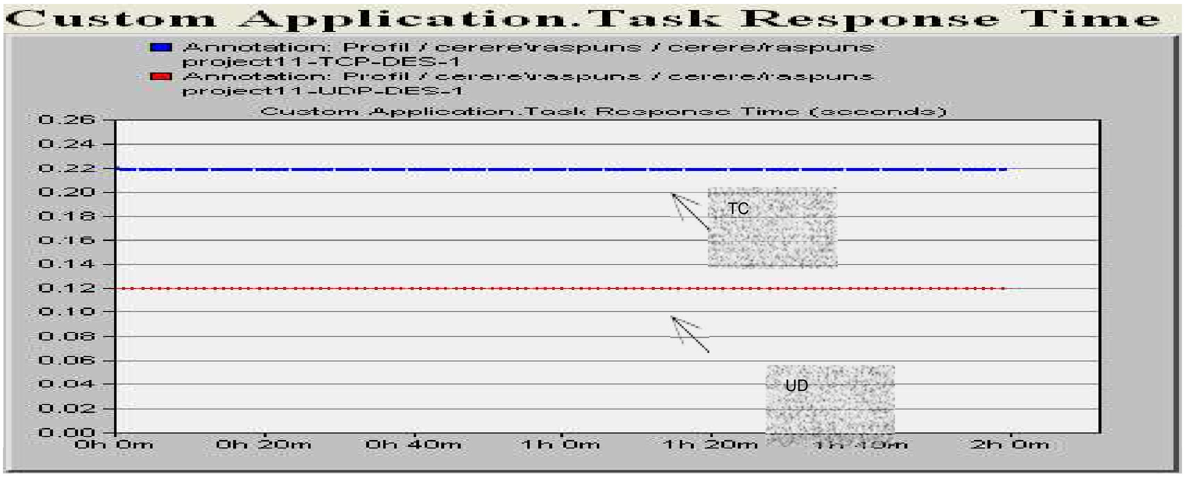
A big difference between TCP and UDP is the congestion control algorithm. For the TCP, congestion algorithm prevents the sender from overrunning the network capacity, while TCP can adapt the sender's rate with the network capacity and attempt to avoid potential congestions problems.

User Datagram Protocol (UDP**)**, another transport protocol in IP networks, is described e.g. The User Datagram Protocol (UDP) provides an unreliable connectionless delivery service using IP to transport messages between machines e.g. [5]. It uses IP to carry messages, but adds the ability to distinguish among multiple destinations within a given host computer. Is a connectionless protocol which doesn't provide flow control, reliability or error recovery and the retransmissions of data in case of errors must be ordered by other protocols. UDP is designed for applications that do not have to recompose the data segment that arrives from the sender. In another way, application-level protocols are directly responsible for the security of data transmitted.

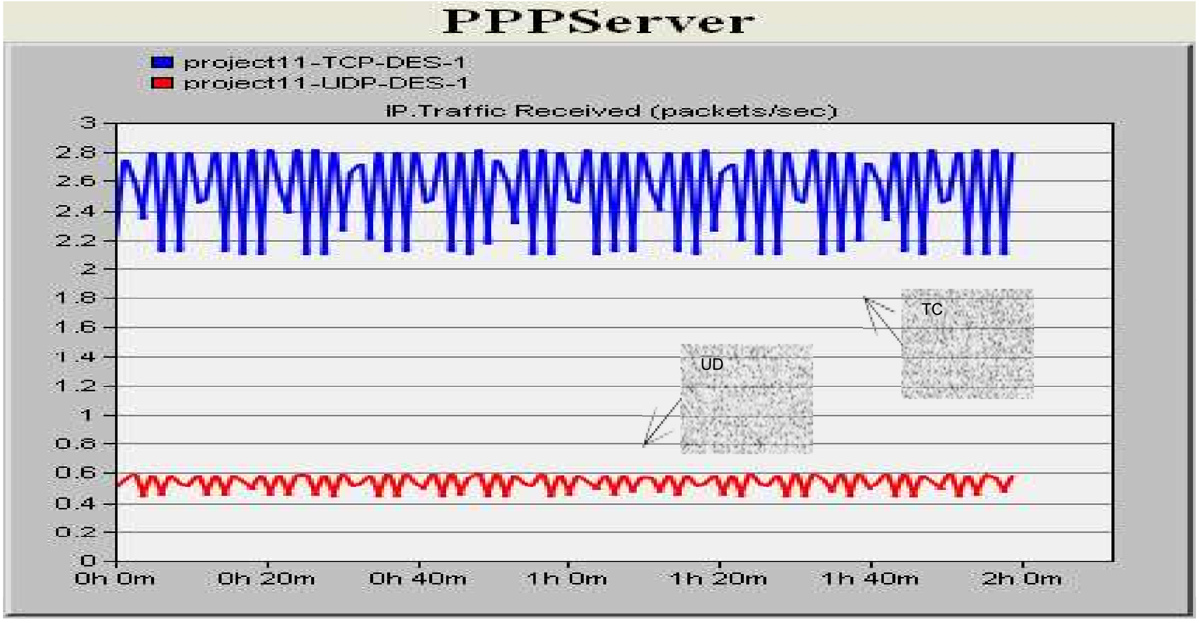
Difference from the TCP is that there is no mechanism for error detections. If applications that use UDP doesn't have their own mechanism for information retrieval can lose those data and be forced to retransmitted again. On the other side this applications are not slow down by the confirmation process and the memory will be available for work much faster.

SIMULATION RESULTS:

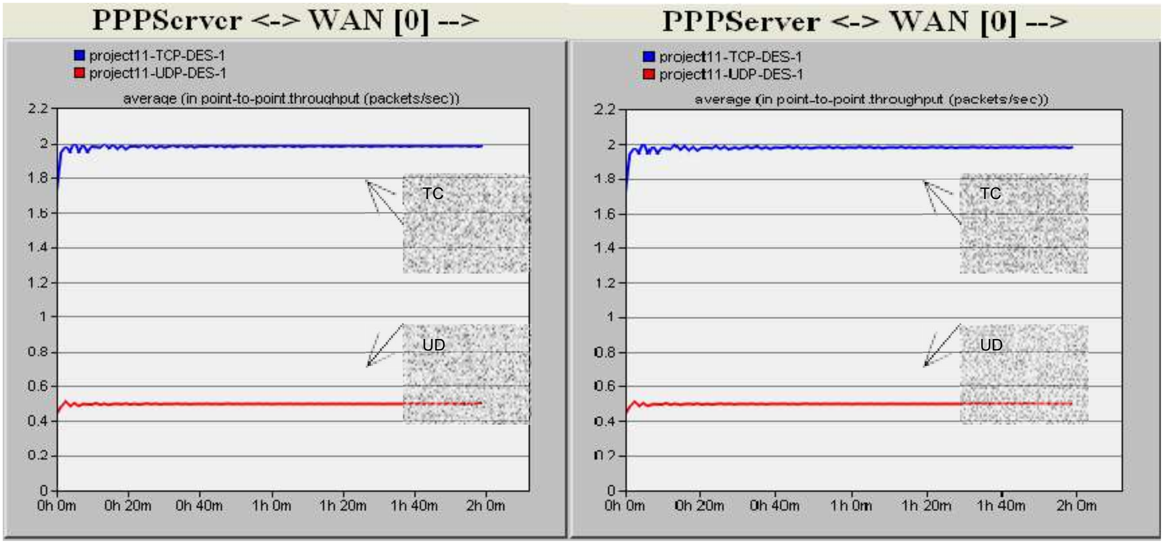
The simulation time is set for two hours data transfer between LAN network and the server with no packet latency and packet discard ratio of 0% while packets traverse thru the WAN. The task response time, in seconds, Fig. 1, shows how long the application need to be completed. The time when using TCP to complete the task is greater that the one using UDP. When using TCP, source and destination need to perform a three-way handshake before starting sending data and all amount of data need to be acknowledge by the destination when it is receive, so is taking more time than UDP, which doesn't perform this tasks.



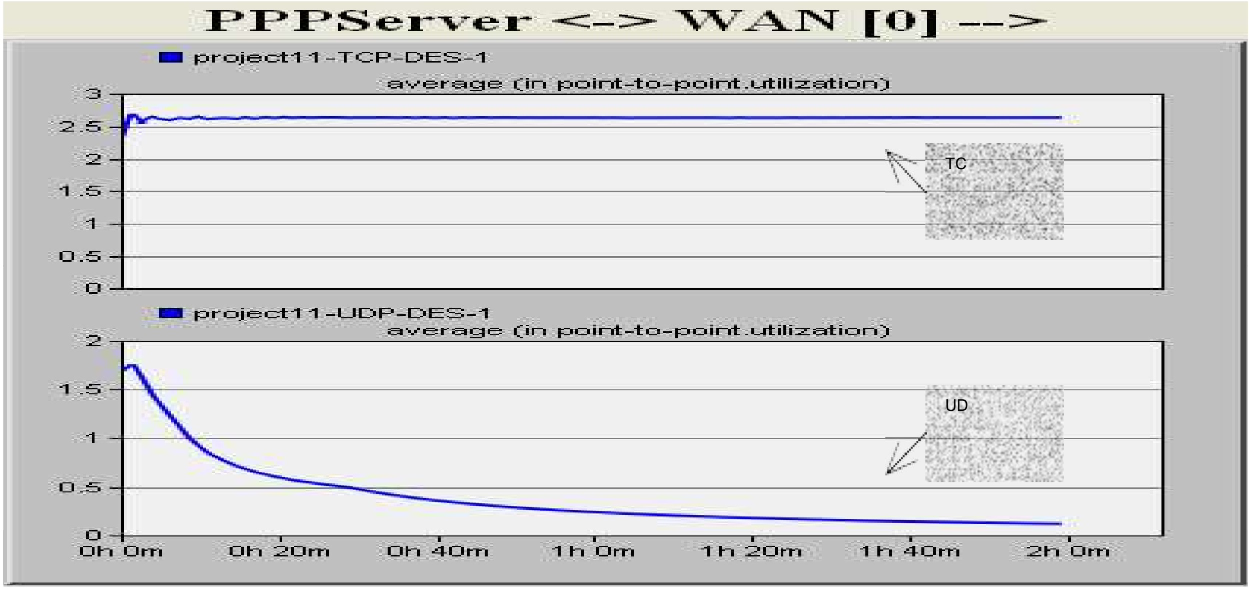
1. Response time for TCP and UDP



2. Traffic received (packets/sec) for the server



3. Traffic/Link utilization from the WAN to the server



Link utilization with a 0.5% packets discard ratio

The main difference between these two protocols is that TCP provides reliability and congestion control services, while UDP is orientated to improve performance.

The most important and common thing that TCP and UDP are using is the ability to set a host-to-host communication channel, so the packets will be delivered between processes running ontwo different computers. UDP is the right choice for application where reliability is not a must but the speed and performance is. Instead, TCP, even if it takes more time for the processes, has additional functions like same order delivery, reliability and flow control. As future work, we planto conduct several studies regarding packets routing in computer networks to improve the fairnessof data transmissions using different network protocols.

RESULT:

Thus the TCP/UDP performance has been simulated successfully using OPNET.

DATE: 8.a Simulation of Distance Vector Routing algorithm.

AIM:

To implement the Distance – Vector Routing Algorithm

APPARATUS REQUIRED:

1. VI-RTSIM software.
2. Personal computer.

THEORY:

Distance Vector Algorithm:

- A Distance vector routing, each router periodically share its knowledge about the entire network with it's neighbors.
- The three keys to under this algorithm are
    1. Knowledge about the whole network.
    2. Routing only to neighbor.
    3. Information sharing at regular intervals.

Knowledge about the whole work:

- Each router shares its knowledge about entire network. It sends all of its collected knowledge about the network to its neighbors.

Routing only to neighbor:

- Each router periodically sends its knowledge about the network only to those routers to which it has direct links. It sends whatever knowledge it has.

Information sharing at regular intervals:

- The every 30 seconds, each router sends its information about the whole network to its neighbors.

Sharing Information:

- LAN's are connected by router, represented by the assuming A, B, C, D, E and F.
- Distance vector routing simplifies the routing process by assuming a lost of one unit for every link.
- The efficiency of transmission is a function only of the number of links required to reach a destination. In this, the cost on hop count.

Routing Table:

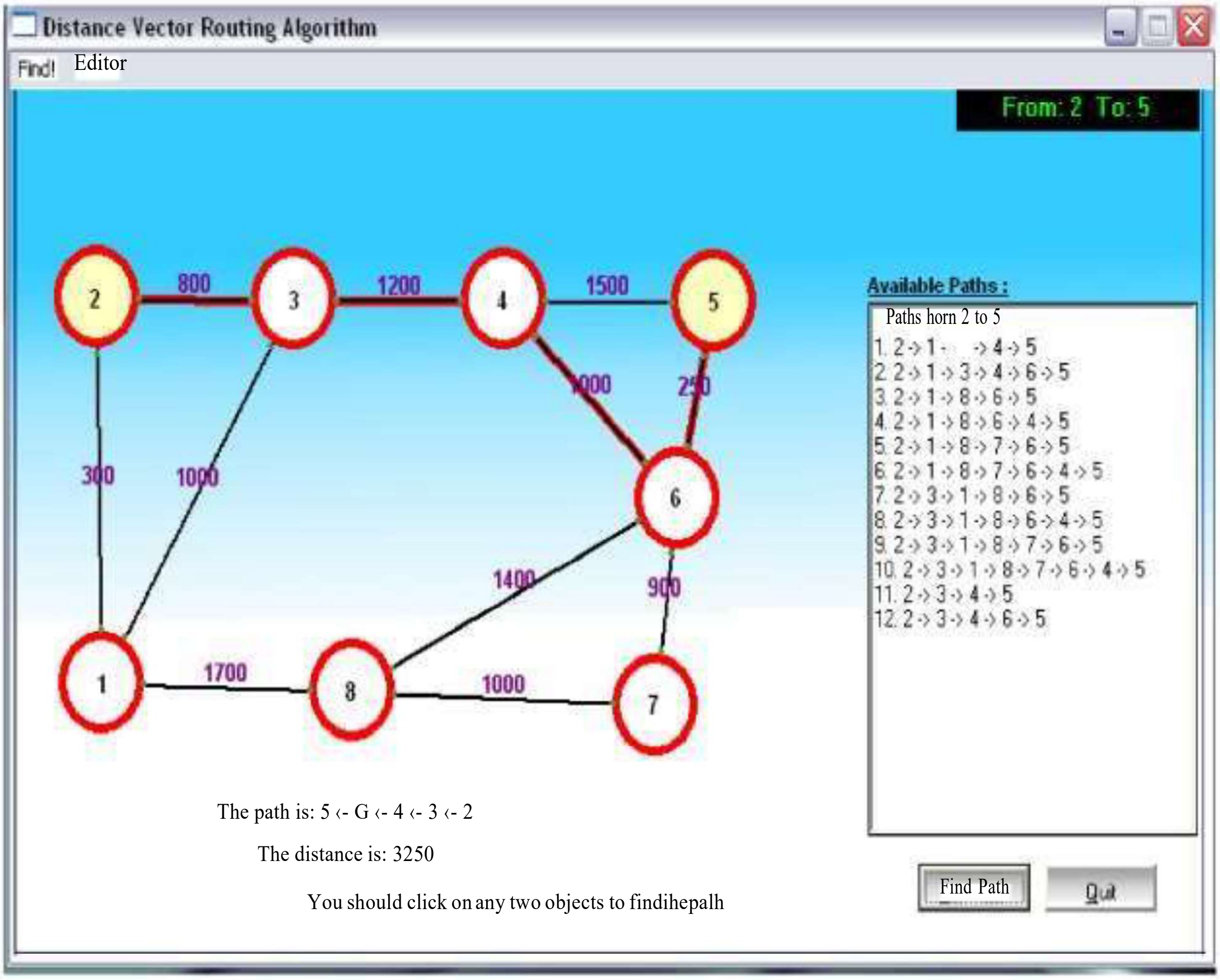
- Each router gets its initial knowledge about the internet work and how it uses shared information to update that knowledge.
- The routing table has e columns network lost router ID.
- The first block is final destination of packet.
- The second block is no of hop count.
- The third block is that to which a packet delivers must.

Updating algorithm:

- Updating algorithm requires that the router first has one hop to the hop count field for each advertised router.
- The router should apply the below rules to each router, if the advertised destination is not in routing table
- If next hop field is same, router should replace the entry in the table with advertised one.
- If next hop field is same, router should replace the entry in the table with advertised one.
- . If next hop field is not the same, advertised hop count is smaller than the one in the table, the router should replace the entry in the table with new one.
- IF advertised hop count is not smaller, the router should do no routing.

PROCEDURE

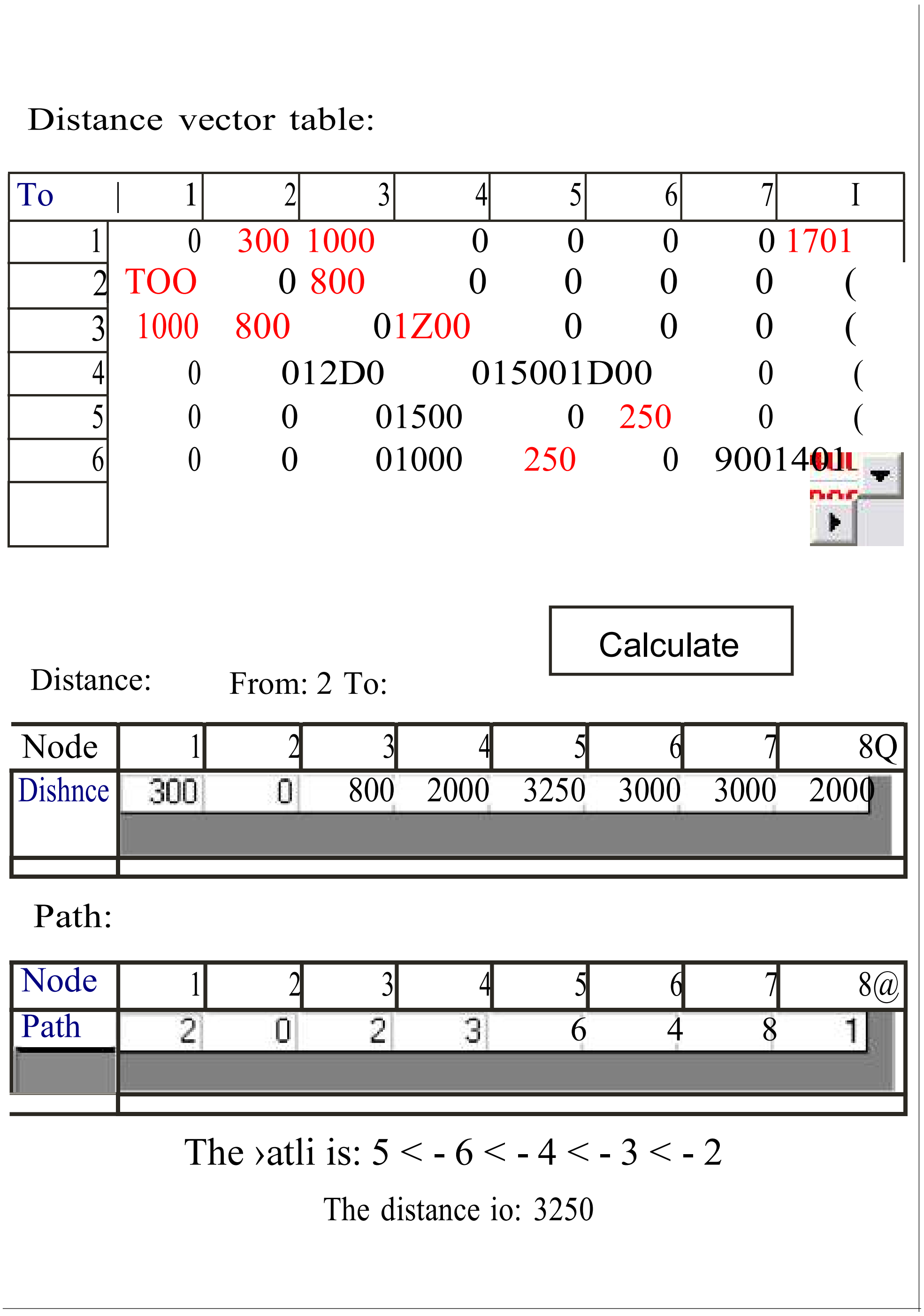
1. Open VI-RTSIM software from desktop
2. Click the Simulation menu bar
3. Select the “Distance – Vector Routing Algorithm” option from Routing algorithm menu bar.
4. Network with routers connected through link is drawn by using option in editor(add router, join link, delete router, delete link, Add caption to link, add caption to router)
5. Select any two nodes to find the shortest distance between them.
6. Click the Find path Button to run the program.
7. Now the shortest paths between the two nodes are calculated.



Available Paths :

```
Paths horn 2 to 5
1. 2 -> 1 -     -> 4 -> 5
2. 2 -> 1 -> 3 -> 4 -> 6 -> 5
3. 2 -> 1 -> 8 -> 6 -> 5
4. 2 -> 1 -> 8 -> 6 -> 4 -> 5
5. 2 -> 1 -> 8 -> 7 -> 6 -> 5
6. 2 -> 1 -> 8 -> 7 -> 6 -> 4 -> 5
7. 2 -> 3 -> 1 -> 8 -> 6 -> 5
8. 2 -> 3 -> 1 -> 8 -> 6 -> 4 -> 5
9. 2 -> 3 -> 1 -> 8 -> 7 -> 6 -> 5
10. 2 -> 3 -> 1 -> 8 -> 7 -> 6 -> 4 -> 5
11. 2 -> 3 -> 4 -> 5
12. 2 -> 3 -> 4 -> 6 -> 5
```

```
The path is: 5 <- G <- 4 <- 3 <- 2
The distance is: 3250
You should click on any two objects to findihepalh
```



Distance vector table:

To |	1	2	3	4	5	6	7	I
1	0	300	1000	0	0	0	0	1701
2	TOO	0	800	0	0	0	0	(
3	1000	800		01Z00	0	0	0	(
4	0		012D0			015001D00	0	(
5	0	0		01500	0	250	0	(
6	0	0		01000	250	0	9001401	

Distance: From: 2 To:

Node	1	2	3	4	5	6	7	8Q
Dishnce	300	0	800	2000	3250	3000	3000	2000

Path:

Node	1	2	3	4	5	6	7	8@
Path	2	0	2	3	6	4	8	1

The ›atli is: 5 < - 6 < - 4 < - 3 < - 2

The distance io: 3250

Viva questions and answers:

1. **what is distance vector routing protocol?**

In distance vector routing the least cost route between any two nodes is the route with minimum distance. In this each node maintains a vector table of minimum distances to every node. The table at each node also guides the packets to the desired node by showing the next stop in the route.

2. **What is advantage in distance vector routing protocol?**

In this each node shares its routing table with its immediate neighbors periodically and when there is a change.

3. **What are the routing protocols?**

- Intra domain routing protocols
- Interdomain routing protocols

4. **What is intradomain routing protocols?**

Routing inside an autonomous system is referred to as intradomain routing protocols.

5. **What is RIP?**

RIP(routing information protocol) is an intradomain routing protocol used inside an autonomous system. it is very simple protocol based on distance vector routing.

RESULT:

Thus Distance Vector routing algorithm has been implemented and shortest-path has been circulated.

DATE: 8.b Simulation of Link State Routing Algorithm.

AIM:

To implement the Link State Routing Algorithm

APPARATUS REQUIRED:

1. VI-RTSIM software.
2. Personal computer.

THEORY:

Link State Vector Algorithm:

- In Link state routing, each router share its information of its neighbors with every other router in the inter-network.

Knowledge about the neighborhood:

- Instead of sending its entire routing table, a router sends information about its neighborhood only.

To all router:

- Each router send this information to every other router on the internetworking, not just to its neighbors.
- If s does so by a process called “flooding” it means that a router sends its information.

Information sharing when there is a Change:

- Each router sends out information about the neighbors when there is a change.

Information sharing:

- Link state routing process use the same internet work as distance vector algorithm.
- Here each other sends its knowledge about is neighbors to every other router in the internet work.
- Cost is applied only by routers and not by any other station on a network, if cost was added by every station, instead of by routers alone, it would accumulate unpredictably.
- Cost is applied as a packet leaves the router rather then as if enters. Most networks are broadcast networks. When a packet is in network every station, including the router, can pick itup, we cannot assign any cost to a packet.

Link state packet:

- ❖ When a router floods the network with information about its neighborhood, it is said to be advertising. The basis of this advertising is a short packet called a link state packet (LSP).

Advertiser	Network	Cost	Neighbor

Getting information about neighbors:

- ❖ A router gets its information about its neighbors by periodically sending them a short greeting packet.
- ❖ If the neighbor responds to the greeting as expected, it is assumed to be alive and functioning.

Initialization:

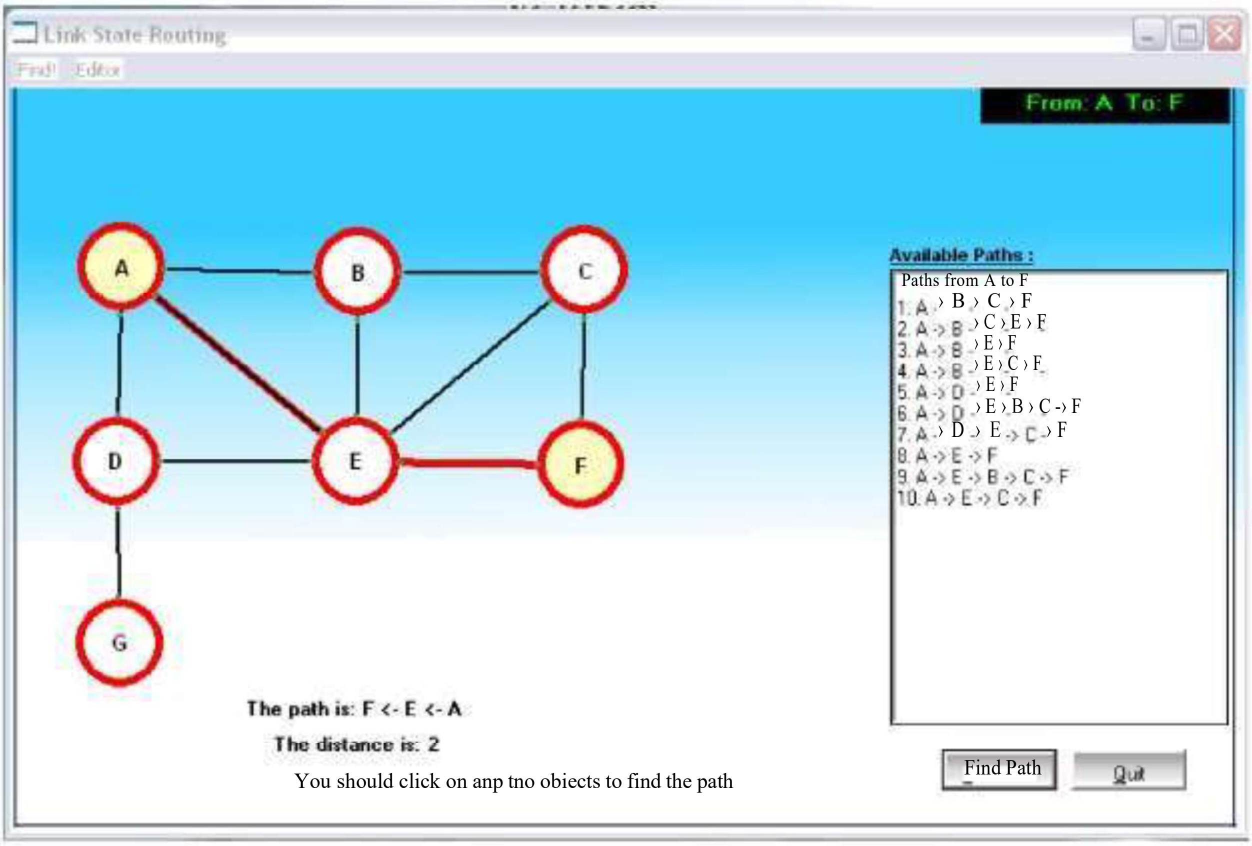
- ❖ Imagine that all routers in our sample internet work come up at the same time.
- ❖ Each router sends a greeting packet to its neighbors to find out the state of each link.

Link – State Database:

- ❖ Every router every LSP and puts the information into a link-state database.
- ❖ Because every router receives the same LSPs every router builds the same database.
- ❖ It stores this database on its disk and uses it to calculate its routing table. If a router is added to be deleted from the system, the whole database must be shared for fast updating.

PROCEDURE

1. Open VI-RTSIM software from desktop
2. Click the Simulation menu bar
3. Select the "Link State Routing Algorithm" option from Routing algorithm menu bar.
4. Network with routers connected through link is drawn by using option in editor(add router, join link, delete router, delete link, Add caption to link, add caption to router)
5. Select any two nodes to find the shortest distance between them.
6. Click the Find path Button to run the program.
7. Now the shortest paths between the two nodes using link state routing algorithm was calculated.



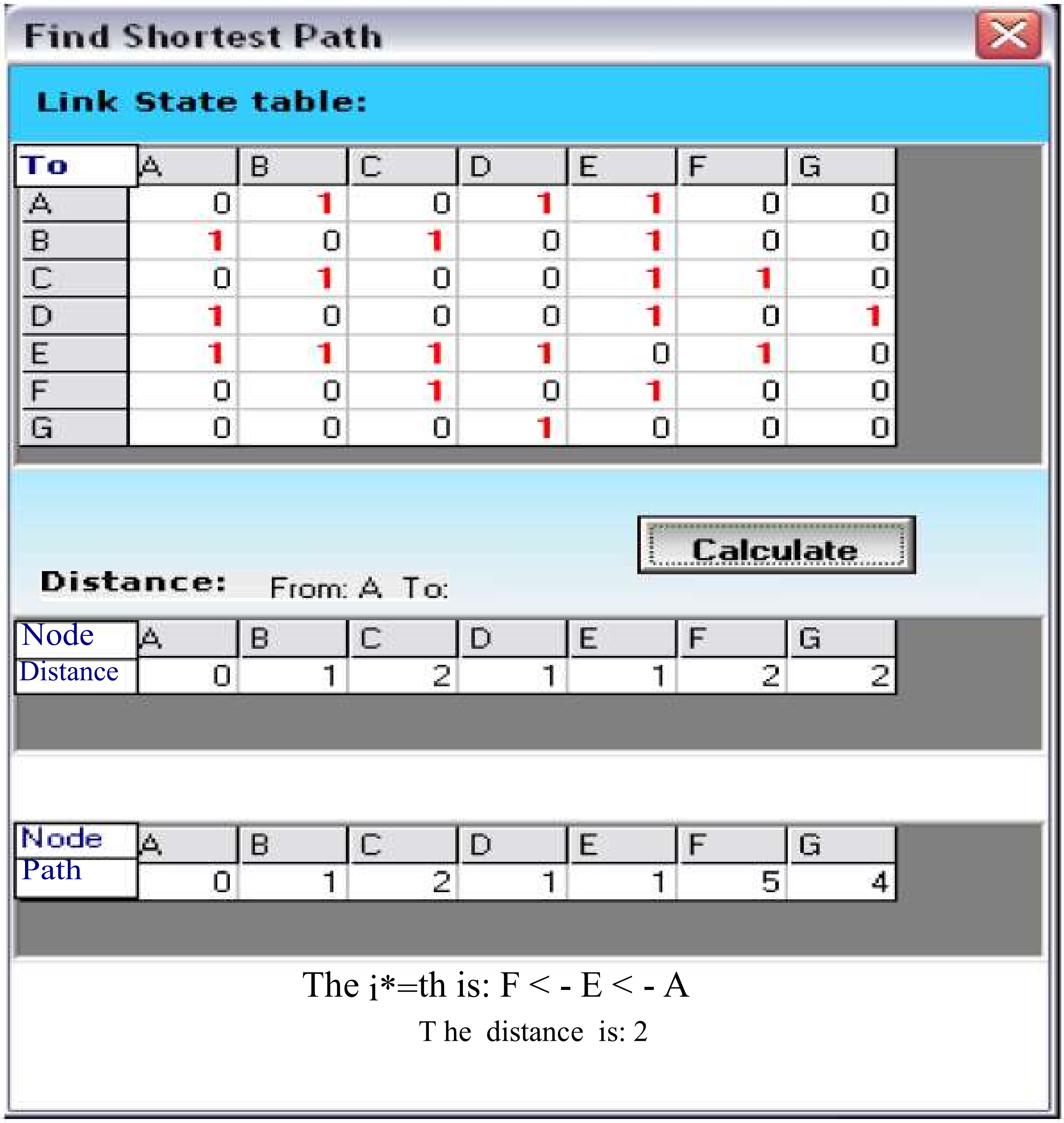
From: A To: F

Paths from A to F
1. A -> B -> C -> F
2. A -> B -> C -> E -> F
3. A -> B -> E -> F
4. A -> B -> E -> C -> F
5. A -> D -> E -> F
6. A -> D -> E -> B -> C -> F
7. A -> D -> E -> C -> F
8. A -> E -> F
9. A -> E -> B -> C -> F
10. A -> E -> C -> F

The path is: F <- E <- A

The distance is: 2

You should click on anp tno obiects to find the path



Find Shortest Path

Link State table:

To	A	B	C	D	E	F	G
A	0	1	0	1	1	0	0
B	1	0	1	0	1	0	0
C	0	1	0	0	1	1	0
D	1	0	0	0	1	0	1
E	1	1	1	1	0	1	0
F	0	0	1	0	1	0	0
G	0	0	0	1	0	0	0

Distance: From: A To:

Node	A	B	C	D	E	F	G
Distance	0	1	2	1	1	2	2

Node	A	B	C	D	E	F	G
Path	0	1	2	1	1	5	4

The ¡*=th is: F < - E < - A

T he distance is: 2

Viva questions and answers:

1. What is link state routing protocol?

It has a different concept from that of distance vector routing. In this each node in the domain has the entire topology of the domain-the list of nodes and links, how they are connected including the type, cost and condition of the links.

2. What is other name of link state routing protocol?

Dijkstra's algorithm.

3. What is advantage of link state routing protocol? The link state

packet can carry a large amount of information. It has topology

4. How the routing table is builded?

By formation of a shortest path tree for each node. Calculation of a

routing table based on the shortest path tree. Creation of the

states of the links by each node.

5. What is periodic update?

A node sends its routing table, normally every 30's in a periodic update. The period depends on the protocol that is using distance vector routing.

RESULT:

Thus Link-State routing algorithm has been implemented and shortest-path has been circulated.

DATE: 9. Simulation of error correction code (like CRC)

AIM:

To implement and check the error detection/error correction techniques in networks using a c program.

APPARATUS REQUIRED:

1. Pc-ino
2. C/c++compiler

THEORY:

Error Detection

- Bit errors occur in frames due to electrical interference or thermal noise.
- Detecting errors is one part of the problem; correcting errors is the other.
- What happens when an error is detected?
- Two basic approaches:
  - Notify the sender that message is corrupt so the sender can retransmit it; (most often used in every day applications)
  - Use an error-correcting code to reconstruct the correct message

Transmission Errors

· External electromagnetic signals can cause incorrect delivery of data

- · Data can be received incorrectly
- · Data can be lost
- · Unwanted data can be generated

· Any of these problems are called *transmission errors*

Error Detection

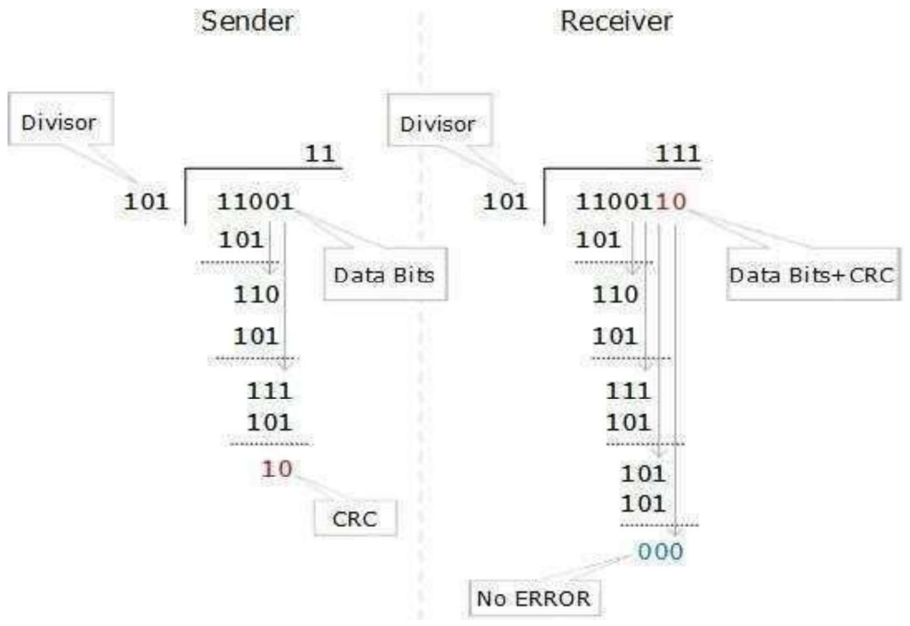
- Detecting Transmission Errors: basic idea is to add redundant information to a frame that can determine if errors have been introduced.

Error Correction or Error Detection?

- When error is detected, frame is discarded and resent, using bandwidth and causing latency, waiting for its arrival.
- Error correction requires additional bit to be sent with every frame.
- Correction is useful when
  - 1) errors are probable or
  - 2) the cost of retransmission is too high

<u>Cyclic Redundancy Check (CRC)</u>

CRC is a different approach to detect if the received frame contains valid data. This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder. Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a codeword. The sender transmits data bits as code words.



At the other end, the receiver performs division operation on codewords using the same CRC divisor. If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.

PROCEDURE:

- Start the process.
- Give the data which is the message.
- Compile and run the program.
- Enter the received hamming code.
- The error is corrected codeword.

PROGRAM FOR CODE GENERATION FOR ERROR DETECTION AND CORRECTION

```
import java.util.Scanner;
class CRC
{
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
int m,g[],n,d[],z[],r[],msb,i,j,k;

System.out.print("ENTER NUMBER OF DATA BITS:\n");
n=sc.nextInt();
System.out.print("ENTER NUMBER OF GENERATOR BITS:\n");
m=sc.nextInt();
d=new int[n+m];
g=new int[m];
System.out.print("ENTER DATA BITS:\n");

for(i=0;i<n;i++)
d[i]=sc.nextInt();
System.out.print("ENTER GENERATOR BITS:\n");
for(j=0;j<m;j++)
g[j]=sc.nextInt();

for(i=0;i<m-1;i++)
d[n+i]=0;
r=new int[m+n];

for(i=0;i<m;i++)
r[i]=d[i];
z=new int[m];

for(i=0;i<m;i++)
z[i]=0;

for(i=0;i<n;i++)
{
k=0;
msb=r[i];
for(j=i;j<m+i;j++)
{
if(msb==0)
r[j]=xor(r[j],z[k]);
else
r[j]=xor(r[j],g[k]);
k++;
}
r[m+i]=d[m+i];
}

System.out.print("\nTHE CODE BITS ADDED ARE:");
for(i=n;i<n+m-1;i++)
{
d[i]=r[i];
System.out.print(d[i]);
}
   System.out.println("");

System.out.print("\nTHE CODE DATA IS:");
for(i=0;i<n+m-1;i++)
{
System.out.print(d[i]);
}
   System.out.println("");
}

public static int xor(int x,int y){
if(x==y)
return(0);
else
return(1);
}

}
```

OUTPUT:

```
F:\java>javac CRC.java

F:\java>java CRC


ENTER NUMBER OF DATA BITS:
8
ENTER NUMBER OF GENERATOR BITS:
5
ENTER DATA BITS:
1
1
0
0
1
0
1
0
ENTER GENERATOR BITS:
1
0
1
0
1

THE CODE BITS ADDED ARE:0111

THE CODE DATA IS:110010100111
```

Viva questions and answers:

1. Define CRC.

A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

2. Define error detection.

Error detection is the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver. Error correction is thedetection of errors and reconstruction of the original, error-free data.3.Defne error correcting.

4. List error detecting methods.

There are three main techniques for detecting errors in frames: Parity Check, Checksum and Cyclic Redundancy Check (CRC).

5. Define parity check.

A parity check is the process that ensures accurate data transmission between nodes during communication. A parity bit is appended to the original data bits to create an even or odd bit number; the number of bits with value one.

RESULT:

Thus the error detection/error correction techniques were implemented successfully.

Implementation of Stop and Wait Protocol and Sliding Window Protocol.

Simulation of Sliding Window Protocol

```
#include<stdio.h>
 #include<stdlib.h>
 #include<math.h>
 int k,time,win=2,i2=0,frame=0,a[20],b[20],i,j,s,r,ack,c,d;
 int send(int,int);
 int receive();
 int checsum(int *);
 main()
 {
 int i1=0,j1=0,c1;
 printf("Enter the frame size\n");
 scanf("%d",&frame);
 printf("Enter the window size\n");
 scanf("%d",&win);
 j1=win;
 for(i=0;i<frame;i++)
 {
 a[i]=rand();
 }
 k=1;
 while(i1<frame)
 {
 if((frame-i1)<win)
 j1=frame-i1;
 printf("\n\ntransmit the window no %d\n\n",k);
 c1=send(i1,i1+j1);
 ack=receive(i1,i1+j1,c1);
 if (ack!=0)
 {
```

```
printf("\n\n1.Selective window\n");
printf("2.Go back N\n");
scanf("%d",&ack);
switch(ack)
{
case 1:
printf("\n\n\t Selective window \t\nEnter the faulty frame no\n");
scanf("%d",&i2);
printf("\n\n Retransmit the frame %d \n",i2);
send(i2,i2+1);
break;
case 2:
printf("\n\n\t Go back n\t\n\n");
printf("\nRetransmit the frames from %d to %d\n",i1,i1+j1);
send(i1,i1+j1);
break;
}
}
 i1=i1+win;
 k++;
 }
 }
 int send(c,d)
 {
 int t1;
 for(i=c;i<d;i++)
 {
 b[i]=a[i];
 printf("frame %d is sent\n",i);
 }
 s=checsum(&a[c]);
 return(s);  }
 int receive(c,d,c2)
 int c2;
 {
 r=checsum(&b[c]);
 if(c2==r)
 {
 return(0);
 }
else
return(1);
}

int checsum(int *c)
{
int sum=0;
for(i=0;i<win;i++)
sum=sum^(*c);
return sum;
}
```

Output:

```
Enter the frame size
50
Enter the window size
5
transmit the window no 1
frame 0 is sent
frame 1 is sent
frame 2 is sent
frame 3 is sent
frame 4 is sent

1.selective window
2.Go back N
1
    selective window
enter the faculty frame no
15
retransmit the frames from 15
frame 15 is sent

transmit the window no 2
frame 5 is sent
frame 6 is sent
frame 7 is sent
frame 8 is sent
frame 9 is sent

1.selective window
2.Go back N
```

Implementation of Stop and Wait Protocol and Sliding Window Protocol.

Simulation of Sliding Window Protocol

```
#include<stdio.h>
 #include<stdlib.h>
 #include<math.h>
 int k,time,win=2,i2=0,frame=0,a[20],b[20],i,j,s,r,ack,c,d;
 int send(int,int);
 int receive();
 int checsum(int *);
 main()
 {
 int i1=0,j1=0,c1;
 printf("Enter the frame size\n");
 scanf("%d",&frame);
 printf("Enter the window size\n");
 scanf("%d",&win);
 j1=win;
 for(i=0;i<frame;i++)
 {
 a[i]=rand();
 }
 k=1;
 while(i1<frame)
 {
 if((frame-i1)<win)
 j1=frame-i1;
 printf("\n\ntransmit the window no %d\n\n",k);
 c1=send(i1,i1+j1);
 ack=receive(i1,i1+j1,c1);
 if (ack!=0)
 {
```

```
printf("\n\n1.Selective window\n");
printf("2.Go back N\n");
scanf("%d",&ack);
switch(ack)
{
case 1:
printf("\n\n\t Selective window \t\nEnter the faulty frame no\n");
scanf("%d",&i2);
printf("\n\n Retransmit the frame %d \n",i2);
send(i2,i2+1);
break;
case 2:
printf("\n\n\t Go back n\t\n\n");
printf("\nRetransmit the frames from %d to %d\n",i1,i1+j1);
send(i1,i1+j1);
break;
}
}
 i1=i1+win;
 k++;
 }
 }
 int send(c,d)
 {
 int t1;
 for(i=c;i<d;i++)
 {
 b[i]=a[i];
 printf("frame %d is sent\n",i);
 }
 s=checsum(&a[c]);
 return(s);  }
 int receive(c,d,c2)
 int c2;
 {
 r=checsum(&b[c]);
 if(c2==r)
 {
 return(0);
 }
else
return(1);
}

int checsum(int *c)
{
int sum=0;
for(i=0;i<win;i++)
sum=sum^(*c);
return sum;
}
```

Output:

```
Enter the frame size
50
Enter the window size
5
transmit the window no 1
frame 0 is sent
frame 1 is sent
frame 2 is sent
frame 3 is sent
frame 4 is sent

1.selective window
2.Go back N
1
   selective window
enter the faculty frame no
15
retransmit the frames from 15
frame 15 is sent

transmit the window no 2
frame 5 is sent
frame 6 is sent
frame 7 is sent
frame 8 is sent
frame 9 is sent

1.selective window
2.Go back N
```