Ex.No: 1 DATE:	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 yoursitting 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 inyour 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 DHCPserver.

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. (Nbtstands 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.

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

Thus the above list of primitive has been studied successfully.

Ex.No: 2 DATE:	**Write a HTTP web client program to download a web page usingTCP 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

CLIENT

```
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 Client{
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

```
import java.net.*;
import java.io.*;
import java.awt.image.*;
import javax.imageio.*;
import javax.swing.*;
class Server
{
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

CLIENT

CONNECTION ESTABLISHED

Server Waiting For image

SERVER

CONNECTION ACCEPTED

Client Connected.

Image Size:29KB

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

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

Ex.No: 3 a DATE:	Socket Program for Echo Client and EchoServer

AIM:

To write a socket program for implementation of echo.

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 a data to server.
4. Receive and print the same data from server.
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 the data from client.
5. Print and send the same data to client.
6. Close the socket.
7. End the program.

PROGRAM:

ECHO CLIENT

```
import java.io.*;
import java.net.*;
public class eclient
{
public static void main(String args[])
{
Socket c=null;
String line;
```

```
DataInputStream is,is1;
PrintStream os;
try
{
c=new Socket("localhost",8080);
}
catch(IOException e)
{
System.out.println(e);
}
try
{
os=new PrintStream(c.getOutputStream());
is=new DataInputStream(System.in);
is1=new DataInputStream(c.getInputStream());
do
{
System.out.println("client");
line=is.readLine();
os.println(line);
if(!line.equals("exit"))
System.out.println("server:"+is1.readLine());
}while(!line.equals("exit"));
}
catch(IOException e)
{
System.out.println("socket closed");
}
}
}
```

Echo Server:

```
import java.io.*;
import java.net.*;
import java.lang.*;
public class eserver
{
public static void main(String args[])throws IOException
{
ServerSocket s=null;
String line;
DataInputStream is;
PrintStream ps;
Socket c=null;
try
{
s=new ServerSocket(8080);
}
catch(IOException e)
{
System.out.println(e);
}
try
{
c=s.accept();
is=new DataInputStream(c.getInputStream());
ps=new PrintStream(c.getOutputStream());
while(true)
{
line=is.readLine();
System.out.println("msg received and sent back to client");
ps.println(line);
}
```

```
}
catch(IOException e)
{
System.out.println(e);
}
}
}
```

OUTPUT:

CLIENT

Enter the IP address 127.0.0.1

CONNECTION ESTABLISHED

Enter the data CSE

Client received CSE

SERVER

CONNECTION ACCEPTED

Server received CSE

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

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

Ex.No: 3 b DATE:	**Socket Program for Client and Server Application for Chat**

AIM:

To write a client-server application for chat using TCP

ALGORITHM:

CLIENT

1. Start the program
2. Include necessary package in java
3. To create a socket in client to server.
4. The client establishes a connection to the server.
5. The client accept the connection and to send the data from client to server.
6. The client communicates the server to send the end of the message
7. Stop the program.

SERVER

1. Start the program
2. Include necessary package in java
3. To create a socket in server to client
4. The server establishes a connection to the client.
5. The server accept the connection and to send the data from server to client and vice versa
6. The server communicates the client to send the end of the message.
7. Stop the program.

PROGRAM:

TCPserver1.java

```
import  java.net.*;
 import java.io.*;
public class TCPserver1
{
 public static void main(String arg[])
{
ServerSocket
```

```
s=null;
String line;
DataInputStream is=null,is1=null;
PrintStream os=null;
Socket c=null;
try
{
s=new ServerSocket(9999);
}
catch(IOException e)
{
System.out.println(e);
} try
{
c=s.accept();
is=new
DataInputStream(c.getInputStream());
is1=new DataInputStream(System.in);
os=new  PrintStream(c.getOutputStream());
do
{
line=is.readLine();
System.out.println("Client:"+line)
;System.out.println("Server:");
line=is1.readLine();
os.println(line);
}
while(line.equalsIgnoreCase("quit")==false)
is.close();
os.close();
}
catch(IOException e)
{
System.out.println(e);

}
```

```
}
}
```

TCPclient1.java

```
import java.net.*;
import java.io.*;
 public class TCPclient1
{
 public static void main(String arg[])
{
Socket
c=null;String
line;
DataInputStream
is,is1;PrintStream os;
try
{
c=new Socket("10.0.200.36",9999);
}
catch(IOException e)
{
System.out.println(e);
}
try
{
 os=new PrintStream(c.getOutputStream());
 is=new DataInputStream(System.in);
is1=new DataInputStream(c.getInputStream());
do
{
System.out.println("Client:");
line=is.readLine();
os.println(line);
System.out.println("Server:" + is1.readLine());
}
while(line.equalsIgnoreCase("quit")==false);
s1.close();
os.close();
}
catch(IOException e)
{
System.out.println("Socket Closed!Message Passing is over");
}

}
```

OUTPUT :

SERVER

C:\Program Files\Java\jdk1.5.0\bin>javac TCPserver1.java Note:
TCPserver1.java uses or overrides a deprecated API. Note:
Recompile with -deprecation for details.
C:\ProgramFiles\Java\jdk1.5.0\bin>java
TCPserver1

Client: Hai Server

Server:Hai Client

Client: How are you

Server:Fine
Client: quit
Server:quit

CLIENT

C:\ProgramFiles\Java\jdk1.5.0\bin>javac
TCPclient1.javaNote: TCPclient1.java uses or overrides a deprecated
API.Note: Recompile with -deprecation for details.
C:\ProgramFiles\Java\jdk1.5.0\bin>java TCPclient1

Client:Hai Server
Server: Hai Client
Client:How are
youServer: Fine
Client:quit
Server: quit

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

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

Ex.No: 4 DATE:	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

```
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

```
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

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

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

Ex.No: 5 DATE:	**Use a tool like Wireshark to capture packets and examine the packets**

AIM:

To Use Wireshark to understand the operation of TCP/IP layers:

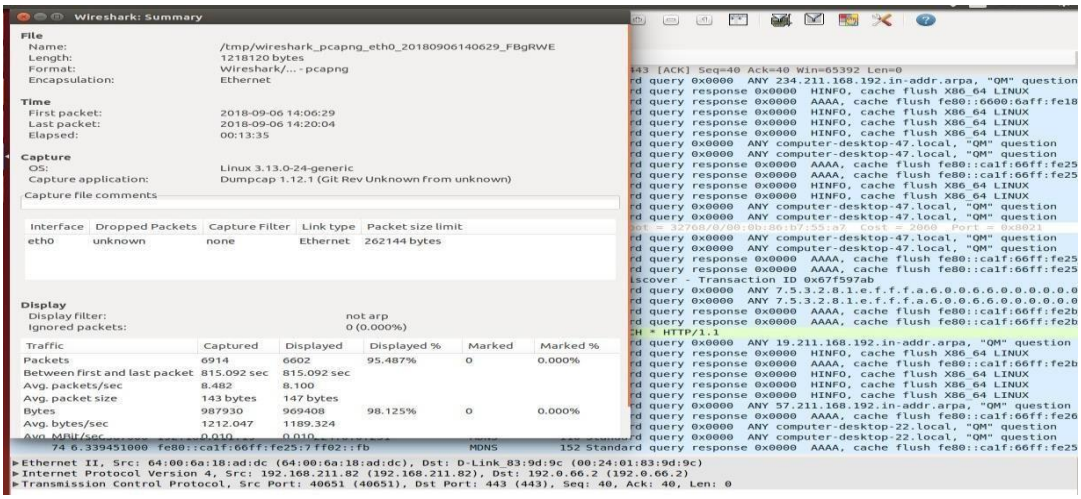
- Ethernet layer : Frame header , Frame size etc.
- Data Link Layer : MAC address,ARP (IP and MAC address binding)
- Network Layer : IP Packet (header, fragmentation), ICMP (Query and Echo)
- Transport Layer : TCP ports, TCP handshake segments etc.
- Application Layer : DHCP , FTP , HTTP header formats.

ALGORITHM:

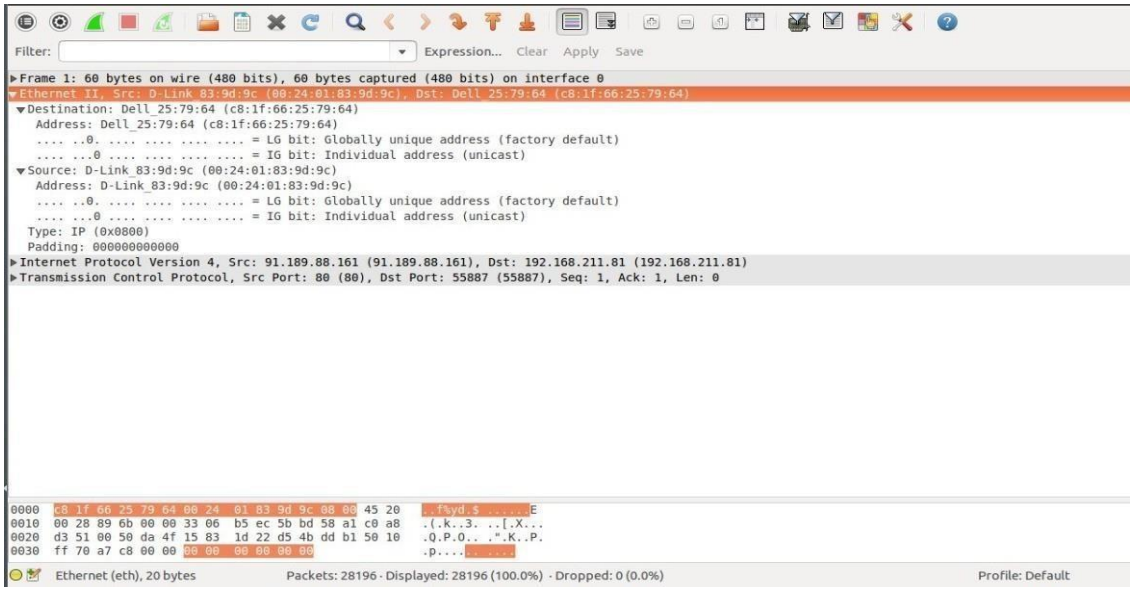
- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.

OUTPUT:

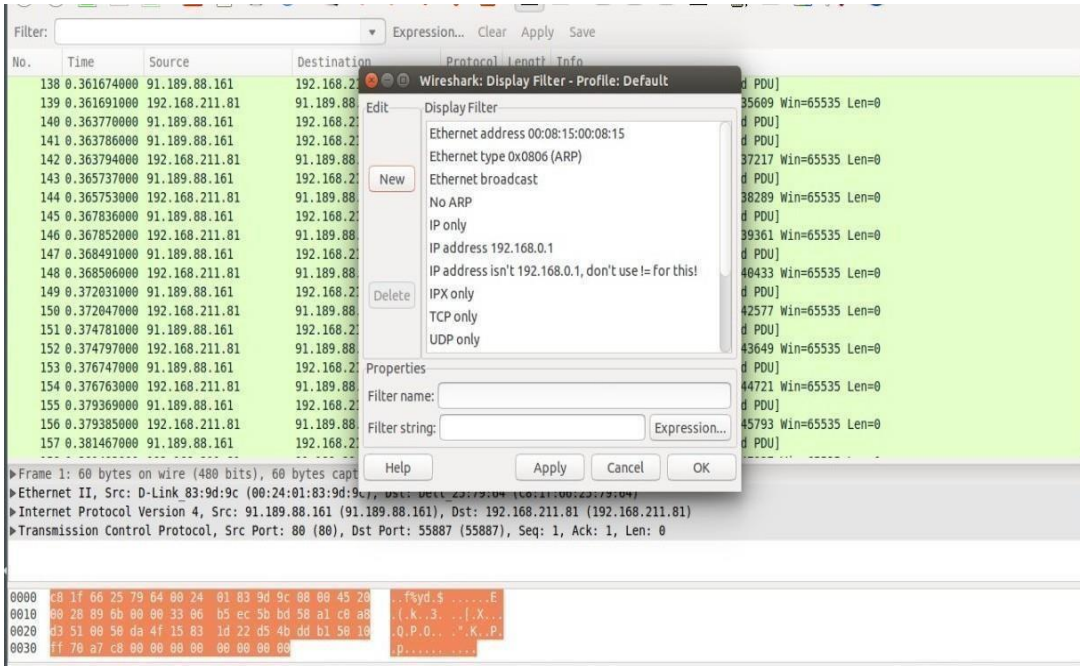
Wireshark Summary



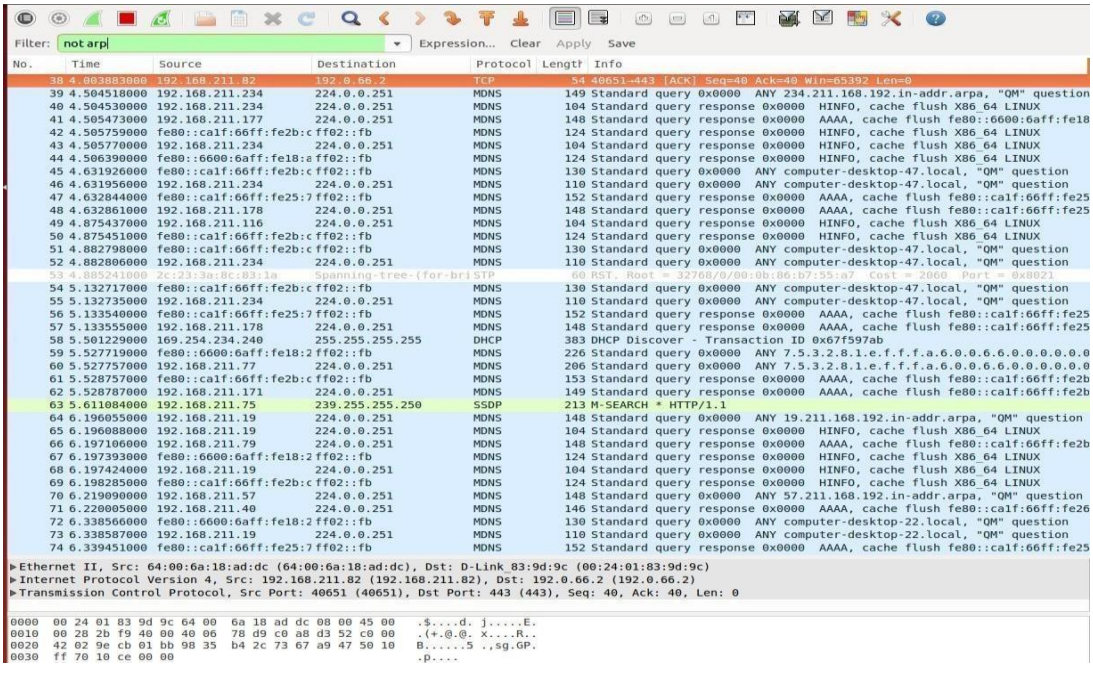
Wireshark : Display Filter



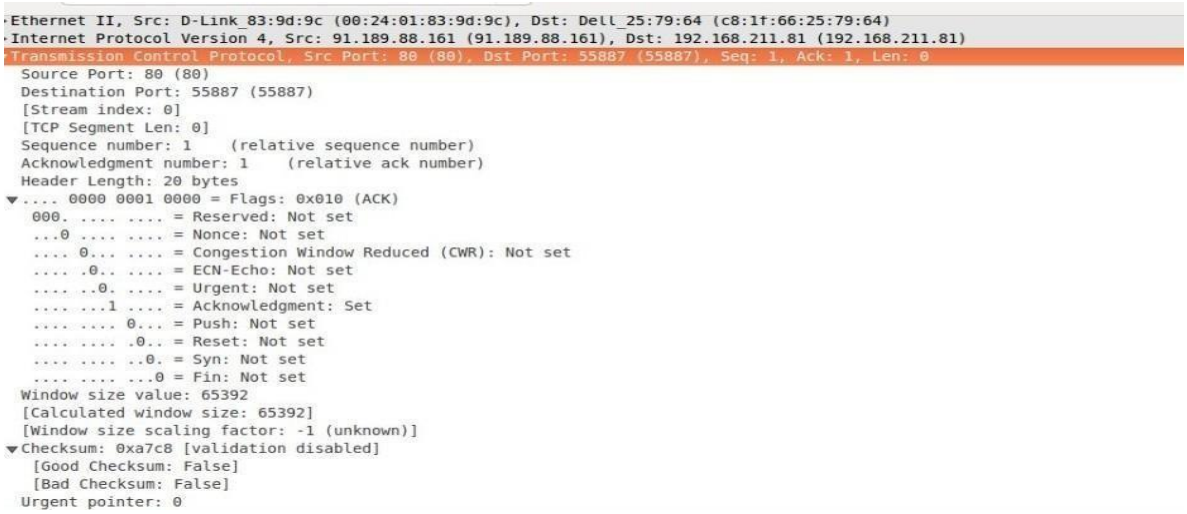
Ethernet Layer :



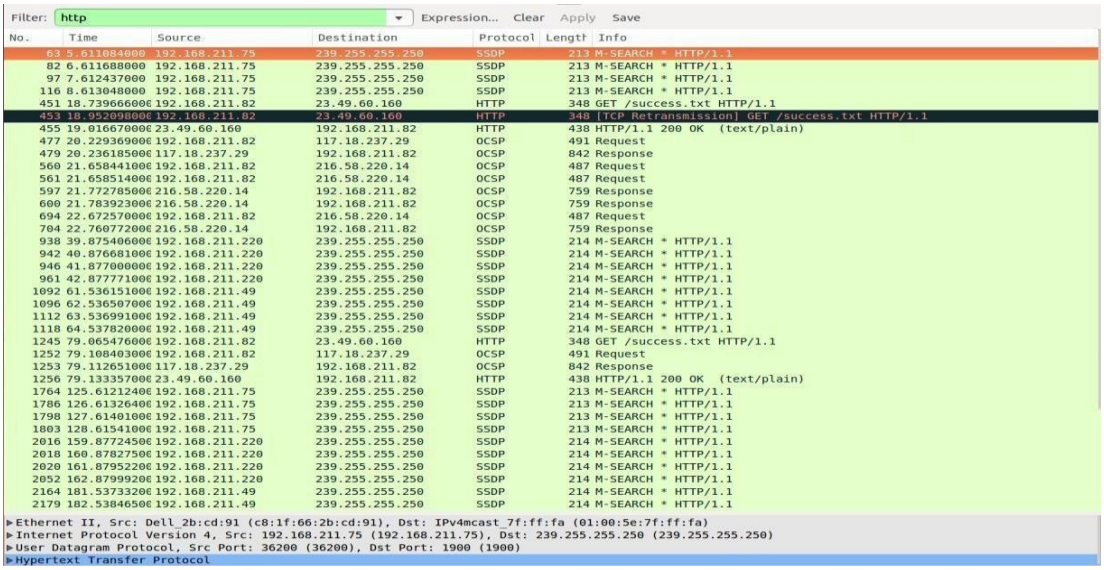
Data Link Layer :



Network Layer :



Transport Layer :



Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

Thus the above Wireshark operation of TCP/IP layers was successfully Executed.

Ex.No: 6 a DATE:	Write a code simulating ARP protocols

AIM

To implement Address Resolution Protocol.

ALGORITHM

CLIENT SIDE

1. Establish a connection between the Client and Server.
   Socketss=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 process
4. Runtime r=Runtime.getRuntime();
5. Process p=r.exec("arp -a "+ip);
6. 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
```

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT

Thus the implementation of ARP is done & executed successfully.

Ex.No: 6 b DATE:	**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. Get the MAC address to be converted into IP address.Send this MAC address to server.
3. 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.

PROGRAM

CLIENT:

```
import java.io.*;
 importjava.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("TheLogical 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()); Strings=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
```

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

Thus the implementation of RARP is done and executed successfully.

Ex.No: 7 DATE:	**Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS.**

AIM:

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

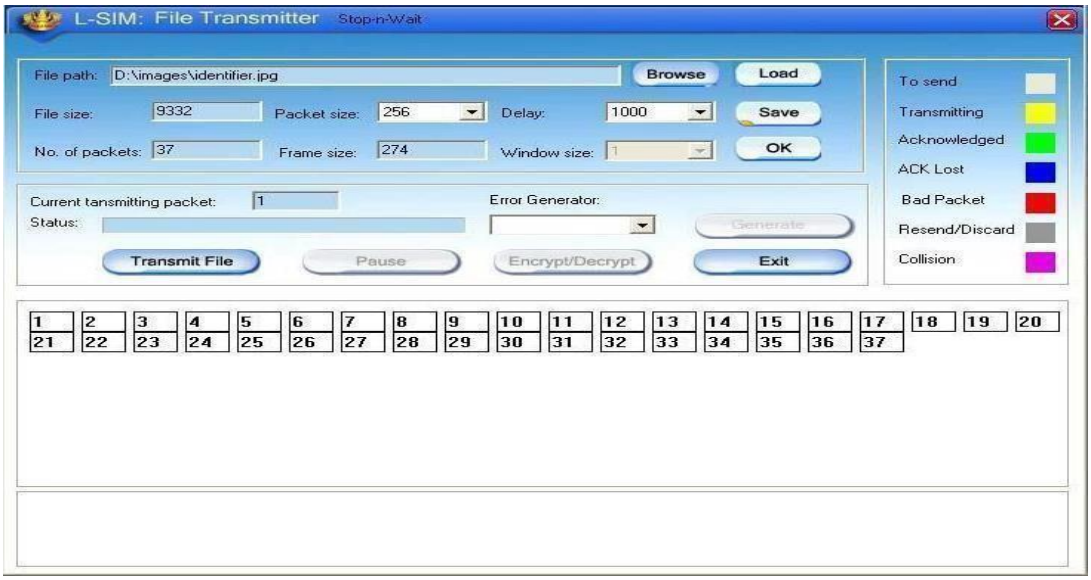
In any network which is created by user server can send or can communicate with any of the clients however clients can communicate only with server, no client to client communication is possible. Transmitter port protocol & network analysis can be done after communication is over between server and clients. 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 being sent 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.

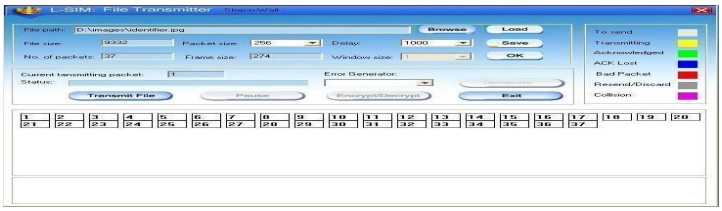
MODEL WINDOW DIAGRAM FOR L-SIM



N-SIM 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



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 integrated 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.

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

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

Ex.No: 8 DATE:	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 protocols in 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.

Comparison Between TCP And UDP

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.

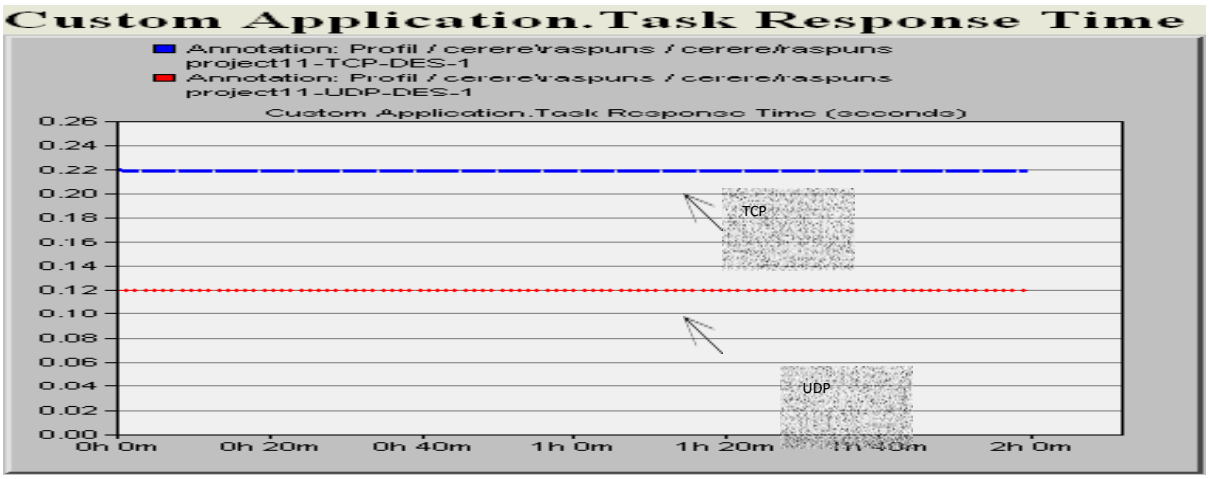
User Datagram Protocol (UDP)**,** another transport protocol in IP networks, is described e.g. The User Datagram Protocol (UDP) provides an unreliable connectionless delivery service using IP to transport messages between machines e.g. [5]. It uses IP to carry messages, but adds the ability to distinguish 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.

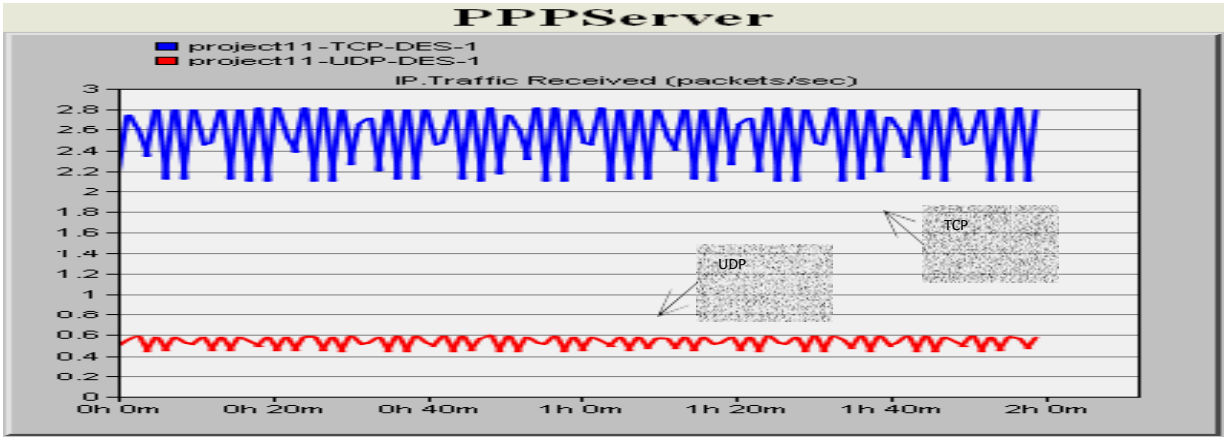
SIMULATION RESULTS:

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

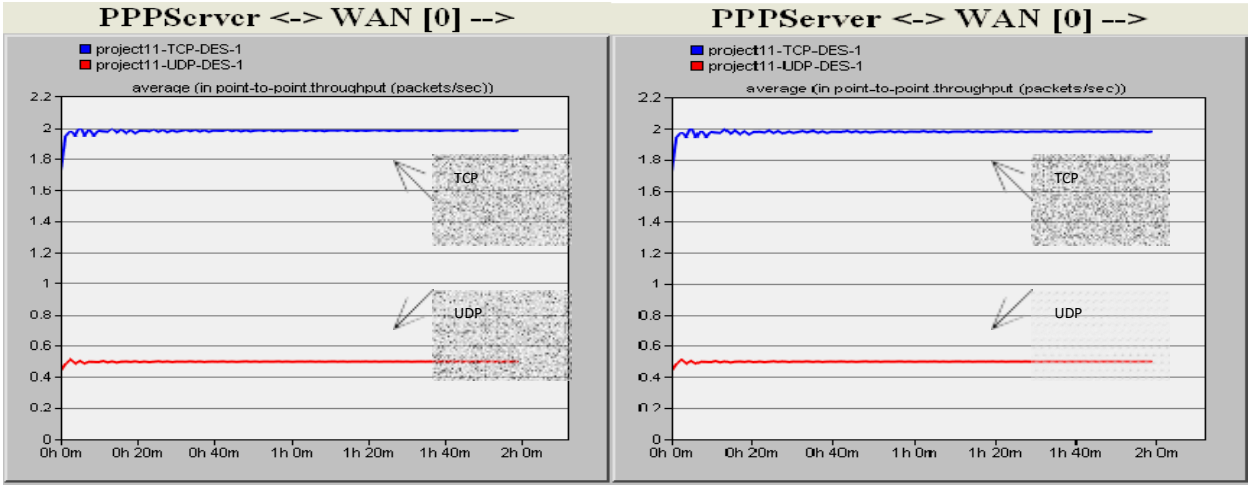
and destination need to perform a three-way handshake before starting sending data and all amount of data need to be acknowledge by the destination when it is receive, so is taking more time than UDP, which doesn't perform this tasks.



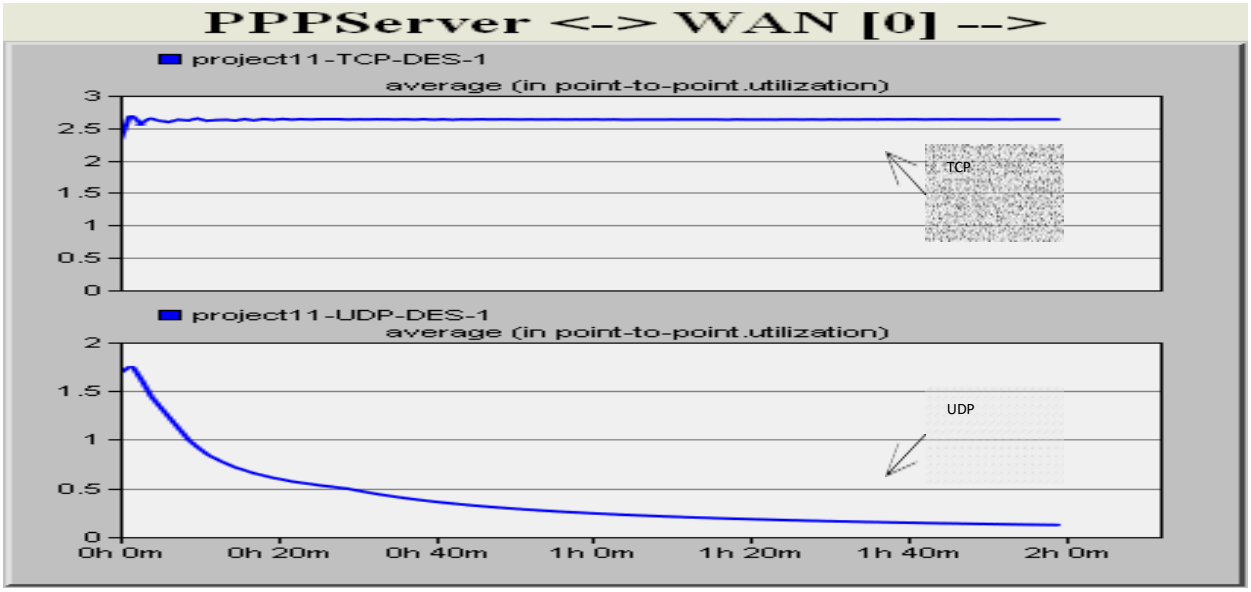
1. Response time for TCP and UDP



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 on two 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 plan to conduct several studies regarding packets routing in computer networks to improve the fairness of data transmissions using different network protocols.

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

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

Ex.No: 9a DATE:	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.

- Traffic/Link utilization from the WAN to the server

Routing Table:

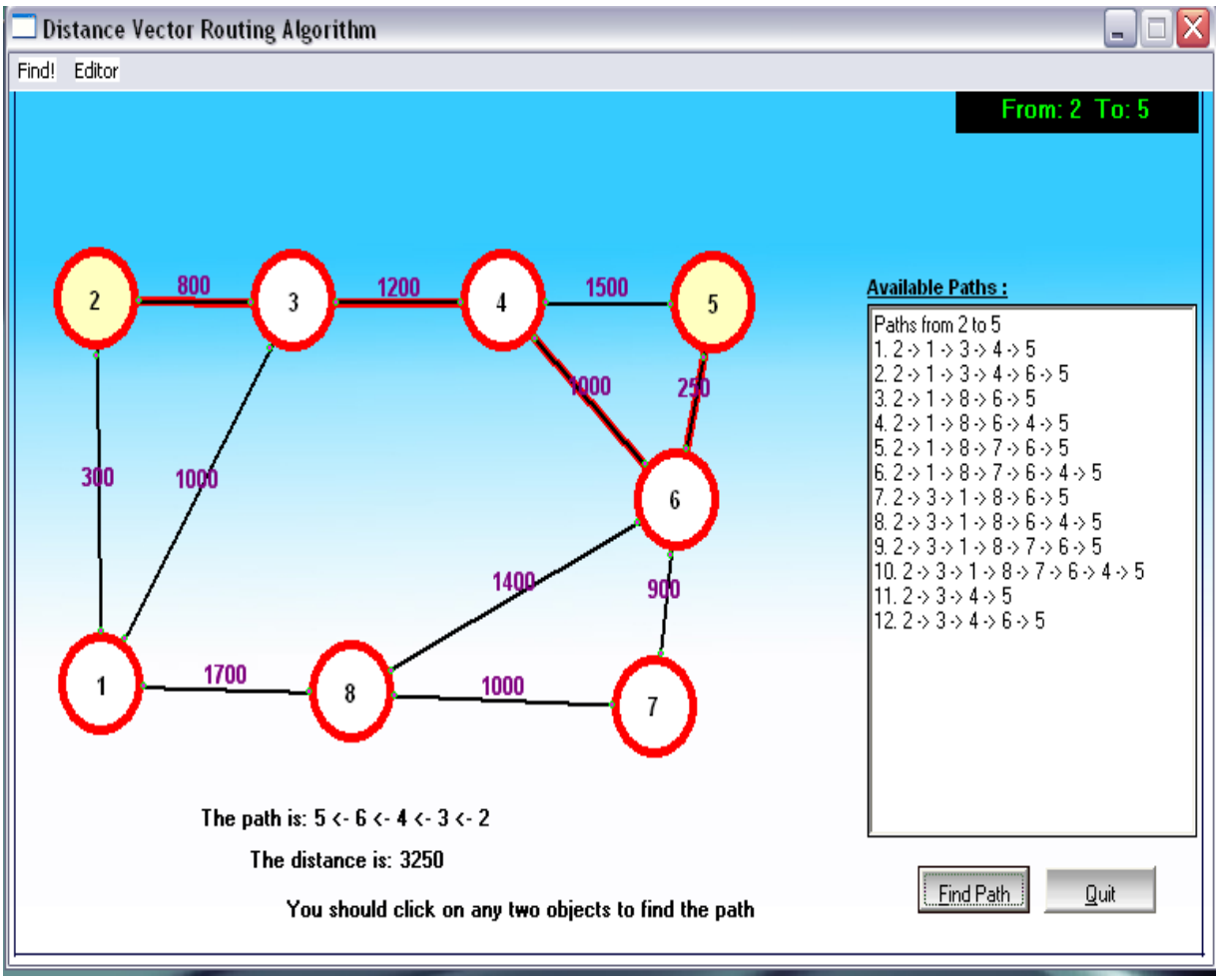
- Each router gets its initial knowledge about the internet work and how it uses shared information to update that knowledge.
- The routing table has e columns network lost router ID.
- The first block is final destination of packet.
- The second block is no of hop count.
- The third block is that to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

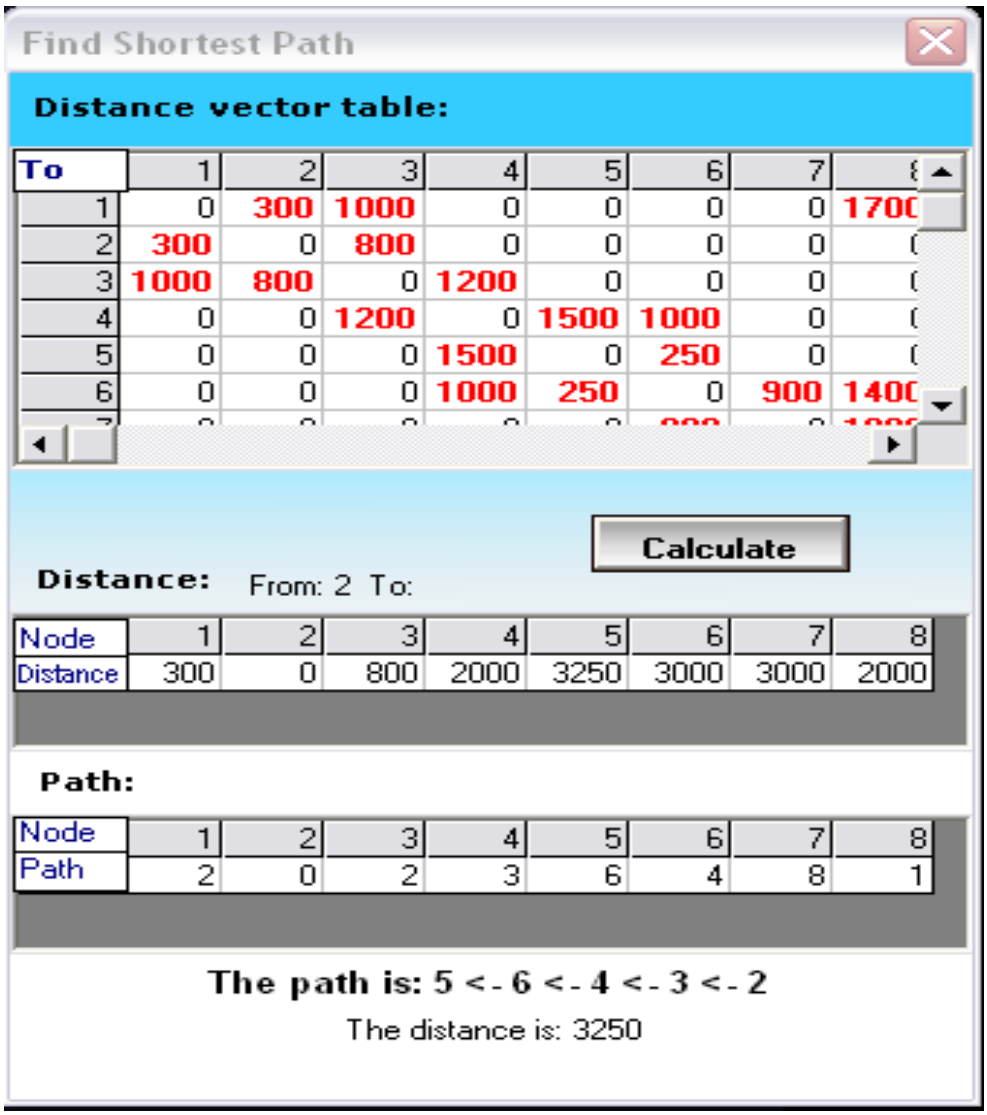
Updating algorithm:

- 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.


Distance Vector Routing Algorithm
Find!
Editor
From: 2 To: 5
2
3
4
5
6
1
8
7
800
1200
1500
1000
250
300
1000
1400
900
1700
1000
Available Paths :
Paths from 2 to 5
1. 2 -> 1 -> 3 -> 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 <- 6 <- 4 <- 3 <- 2
The distance is: 3250
You should click on any two objects to find the path
Find Path
Quit



Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

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

Ex.No: 9b DATE:	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 it up, 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:

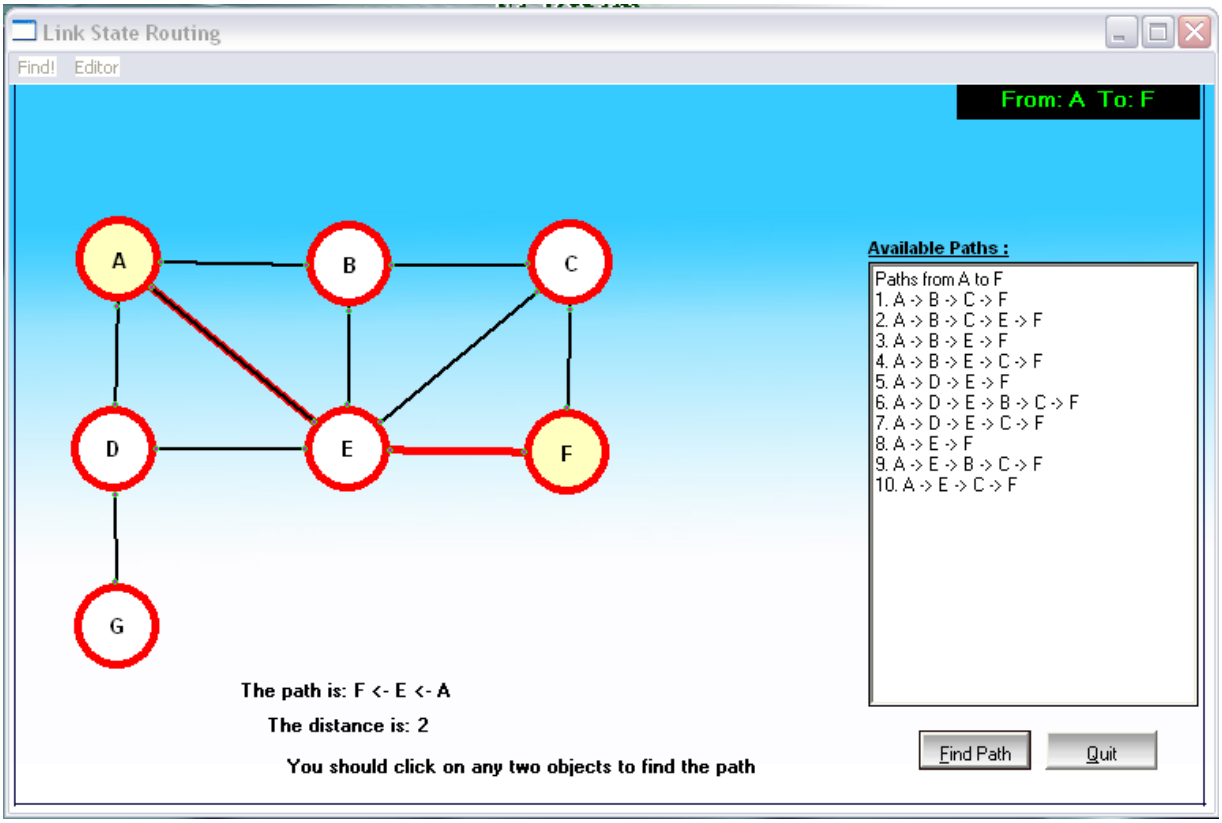
- Imagine that all routers in our sample internet work come up at the same time.
- Each router sends a greeting packet to its neighbors to find out the state of each link.

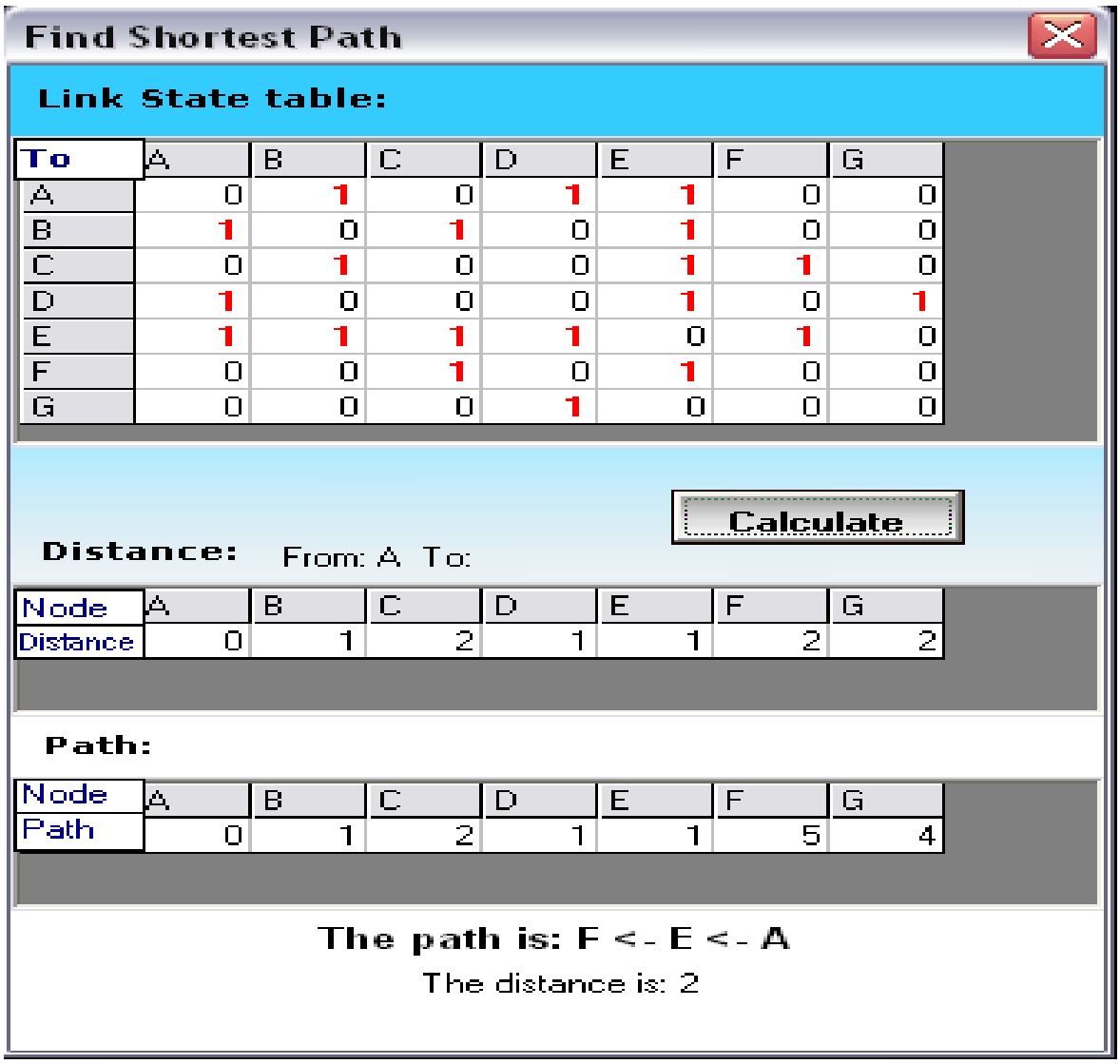
Link – State Database:

- 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.


Link State Routing
Find!
Editor
From: A To: F
A
B
C
D
E
F
G
Available Paths :
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 any two objects to find the path
Find Path
Quit



Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

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

Ex.No: 10 DATE:	**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?

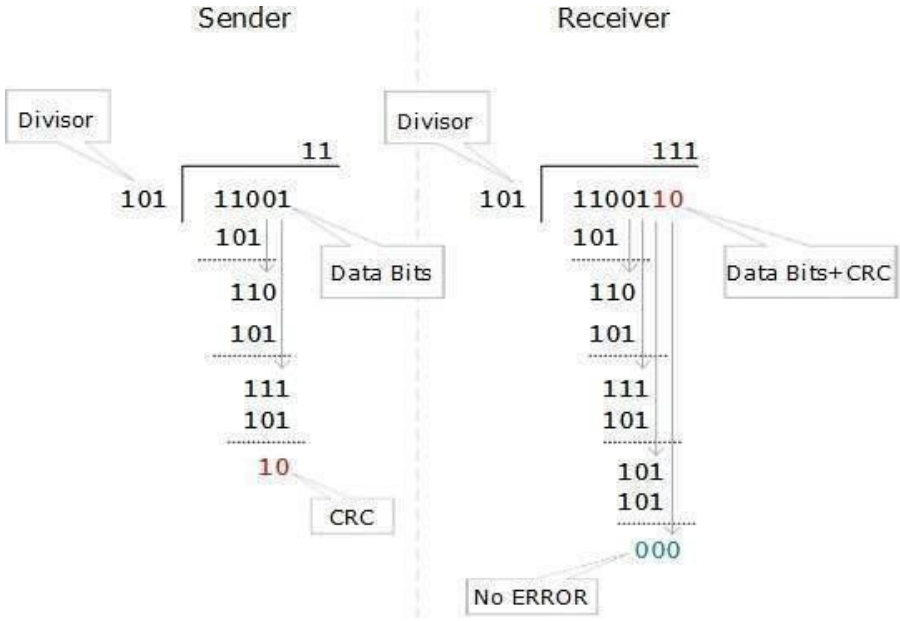
- When error is detected, frame is discarded and resent, using bandwidth and causing latency, waiting for its arrival.
- Error correction requires additional bit to be sent with every frame.
- Correction is useful when

  1) errors are probable or

  2) the cost of retransmission is too high

Cyclic Redundancy Check (CRC)

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

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
int main()
{
int i,j,k,count,err-pos=0,flag=0;
char w[20],cw[20],data[20];
printf("enter data as binary bit stream(7 bits):\n");
scanf("%s",data);
for(i=1,j=0,k=0;i<12;i++)
{
if(i==(int)pow(2,j))
{
dw[i]='?;
j++;
}
else
{
dw[i]=data[k];
k++;
}
}
for(i=0;j<4;i++)
{
count=0;
for(j=(int)pow(2,i);j<12;j<12;j+=(int)pow(2,i
))
{
for (k=0;k<(int)pow(2,i);k++)
{
if(dw[j]=='1')count++;j++;
}
}
if(count%2==0)
dw[(int)pow(2,i)]='
0';else
dw[(int)pow(2,i)]='
1';
}
printf("in code word
is\n\n");for(i=1;i<12;i++)
printf("%c",dw[i]);
printf("\n\n enter the received hamming
```

```
code\n\n");scanf("%s",cw);
for(i=12;i>0;i--)
cw[i]=cw[i-1];
for(i=0;i<4;i+
+)
{
count=0;
for(j=(int)pow(2,i);j<12;j+=(int)pow(2,i)
)
{
for(k=0;k<(int)pow(2,i);k++)
{

if(cw[j]=='1')count++;j++;
}

}
if (count%2!=0)
err-pos=err-post+(int)pow(2,i);
}
if(err-pos==0)
printf("\n\n there is no error in received code word
\n");else
{
if(cw[err-pos]==dw[err-pos])
{
printf("\n\n there are 2 or more errors in received
code……\n\n"); printf("sorry…! hamming code cannot correct 2
or more errors….\n");flag=1;
}
else
printf("in there is an error in bit position %d of received code word
\n",err-pos);if(flag==0)
{
cw[err-pos]=(cw[err-pos]=='1')?'0':'1';

printf("\n\n corrected code word is
\n\n");for(i=1;i<12;i++)
printf("%c",cw[i]);
}
}
printf("\n\n");
}
```

OUTPUT:

Enter data as binary bit stream(7 bits):

1110110

Code word
is
1110110011
0
Enter the received hamming
code10101100110

There is an error in bit position 2 of received code word corrected code word is
11101100110
Enter data as binary bit stream(7
bits)11101110
Code word
is
1110110011
0
Enter the received hamming
code00101100110
There are 2 or more error in received
code…Sorry…!

Description	Max.Marks	Obtained Marks
Observation & Viva Voce	10	
Record	5	
Total	15	

RESULT:

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