

Commands:

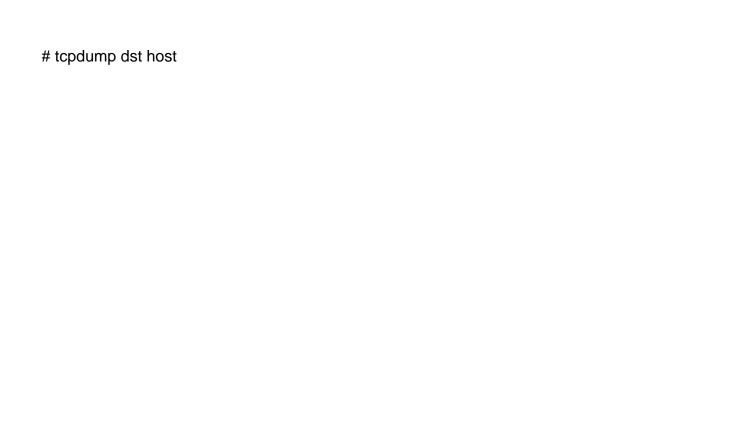
Tcpdump:

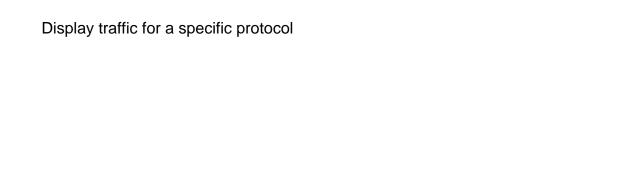
Display traffic between 2 hosts:



isplay traffic from a source or destination host only:	

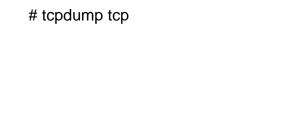
To display traffic from only a source (src) or destination (dst) host: # tcpdump src host	





Provide the protocol as an argument to display only traffic for a specific protocol, for example tcp, udp, ic	m

For example to display traffic only for the tcp traffic :	





To filter based on a source or destination port:







Netstat provides information and statistics about protocols in use and current TCP/IP network connections

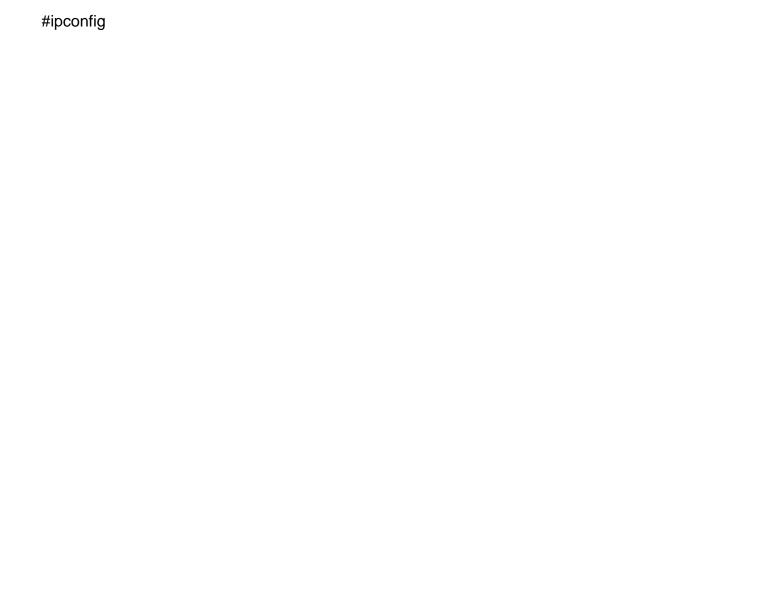




rompt. This utility allows you to get the IP address information of a Windows computer.	

Using ipconfig

					<b>-</b>	
From the comm	and prompt, type	e ipconfig to rur	the utility with	default options.	The output of th	e default co









Trace route:



Traceroute is a network diagnostic tool used to track the pathway taken by a packet on an IP network from

The response time of each hop is calculated. To guarantee accuracy, ea	ch hop is queried multiple times (u



Proceeding in this way,	, traceroute uses th	e returned ICMP	Time Exceeded r	messages to build	a list of rout

With the tracert command shown above, we're asking tracert to show us the path from the local com-	nputer a

www.google.com. #tracert google.com

Ping:		

The ping command sends an echo request to a host available on the network. Using this command, you	ou

Ca

Ex.No: 2 Write a HTTP web client program to download a web page using TCP sockets





import java.awt.image.BufferedImage; import java.io.ByteArrayOutputStream; import java.io.File;



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.









dos.write(bytes, 0, bytes.length); System.out.println("Image sent to server. "); dos.close();							



}			



{			

System.out.println("Exception: " + e.getMessage());



}			



}			

}			





import java.awt.image.\*; import javax.imageio.\*; import javax.swing.\*; class Server

{			

public static void main(String args[]) throws Exception

{			



server=new ServerSocket(4000); System.out.println("Server Waiting for image");



DataInputStream dis = new DataInputStream(in); int len = dis.readInt();



dis.close();		

in.close();

InputStream ian = new ByteArrayInputStream(data); BufferedImage bImage = ImageIO.read(ian)	; JFrame

ImageIcon icon = new ImageIcon(bImage); JLabel I = new JLabel();





}			

}			



When you run the client code, following output screen would appear on client side.

Ex.No: 3 Applications using TCP sockets like: Echo client and echo server, Chat and File Transfer



EchoServer.java import java.net.\*; import java.io.\*; public class EServer

{			



{			

ServerSocket s=null; String line; DataInputStream is; PrintStream ps; Socket c=null;

{			



}			

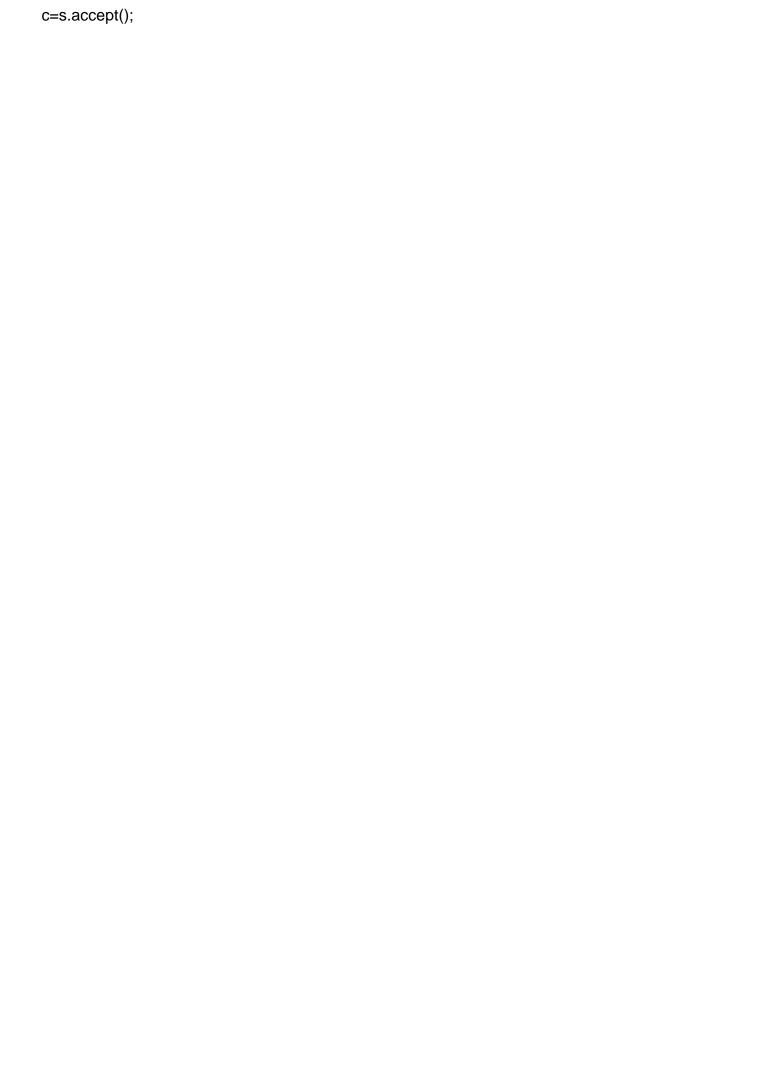


{			

}			

{			







ps=new PrintStream(c.getOutputStream()); while(true)

{			



}			

}			



{			



}			

}			

}			

EClient.java import java.net.\*; import java.io.\*; public class EClient



{			



DataInputStream is,is1; PrintStream os;

{			

InetAddress ia = InetAddress.getLocalHost(); c=new Socket(ia,9000);

}			



{			

}			

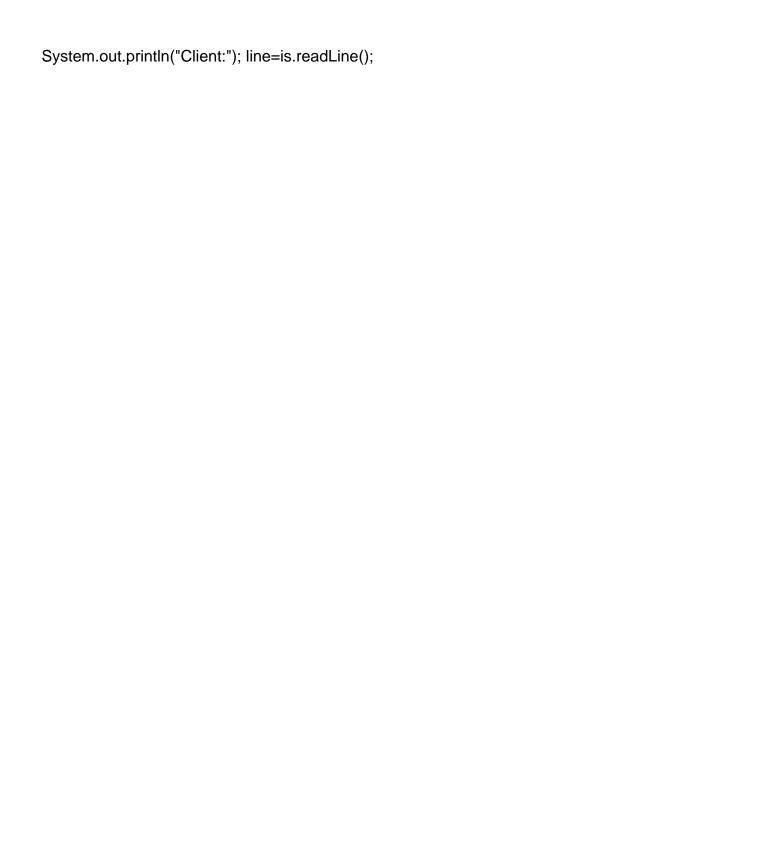
{			



 $os=new\ PrintStream(c.getOutputStream());\ is=new\ DataInputStream(System.in);$ 

is1=new DataInputStream(c.getInputStream()); while(true)

{			





System.out.println("Server:" + is1.readLine());

}			

}			



{			



}			



C:\Program Files\Java\jdk1.5.0\bin>java	c EServer.java C:\Prograr	m Files\Java\jdk1.5.0\bin>java	a EServer C



C:\Program Files\Java\jd	lk1.5.0∖bin>javac EC	lient.java C:\Progra	am Files\Java\jdk1.5	5.0\bin>java EClie	nt Clie

Server:Hai Server Client: Hello Server:Hello Client:end Server:end Client:ds





UDPserver.java import java.io.\*; import java.net.\*; class UDPserver

{			

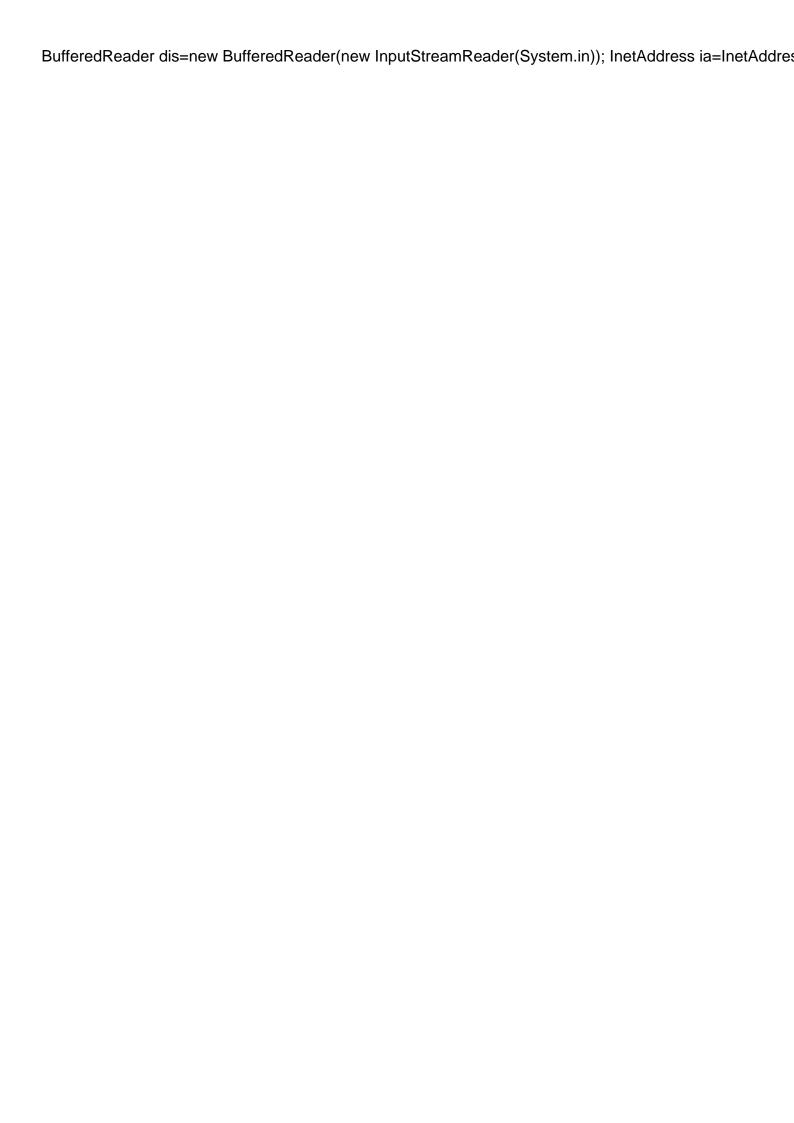


public static byte buffer[]=new byte[1024]; public static int clientport=789,serverport=790;

public static void main(String args[])throws Exception

{			

ds=new DatagramSocket(clientport); System.out.println("press ctrl+c to quit the program");	





{			

DatagramPacket p=new DatagramPacket(buffer,buffer.length); ds.receive(p);



String str=dis.readLine(); if(str.equals("end"))





}			

}			

}			

UDPclient.java import java .io.\*; import java.net.\*; class UDPclient

{			



public static int clientport=789,serverport=790;

public static void main(String args[])throws Exception

{			

byte buffer[]=new byte[1024]; ds=new DatagramSocket(serverport);



InetAddress ia=InetAddress.getLocalHost(); while(true)

{			

System.out.println("Client:"); String str=dis.readLine(); if(str.equals("end"))





String psx=new String(p.getData(),0,p.getLength()); System.out.println("Server:" + psx);

}			

}			

}			





C:\Program Fi	iles\Java\jdk1.5	.0∖bin>javac U	DPserver.java	C:\Program F	Files∖Java∖jdk1	.5.0\bin>java	UDPse

press ctrl+c to quit the program Client:Hai Server

Server: Hello Client Client: How are You Server: I am Fine



C:\Program Files\	Java\jdk1.5.0\bin>	javac UDPclien	t.java C:∖Prograi	m Files∖Java∖jdk1.	5.0\bin>java UDP	clie

Client:Hai Server Server:Hello Clie Client:How are You Server:I am Fine Client:end

## FILE TRANSFER







public class FileServer

{			

public static void main(String[] args) throws Exception

{			



ServerSocket ssock = new ServerSocket(5000); Socket socket = ssock.accept();



InetAddress IA = InetAddress.getByName("localhost");



File file = new File("e:\\Bookmarks.html"); FileInputStream fis = new FileInputStream(file);

BufferedInputStream bis = new BufferedInputStream(fis); //Get socket's output stream

OutputStream os = socket.getOutputStream(); //Read File Contents into contents array

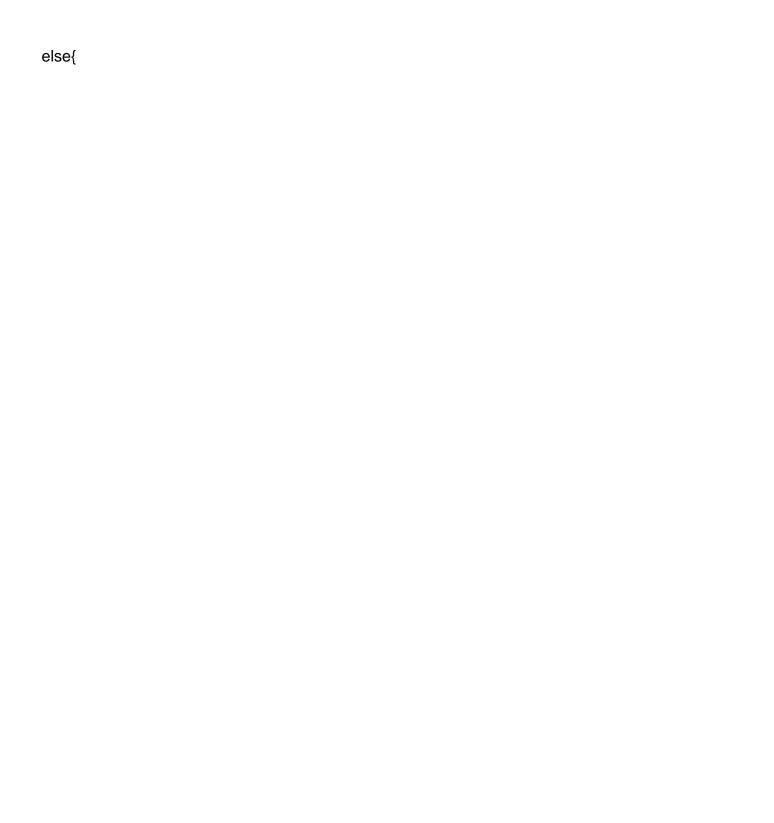


long fileLength = file.length(); long current = 0;

long start = System.nanoTime(); while(current!=fileLength){

int size = 10000;

if(fileLength - current >= size) current += size;



size = (int)(fileLength - current); current = fileLength;

}			





System.out.print("Sending file ... "+(current\*100)/fileLength+"% complete!");

}			

os.flush();







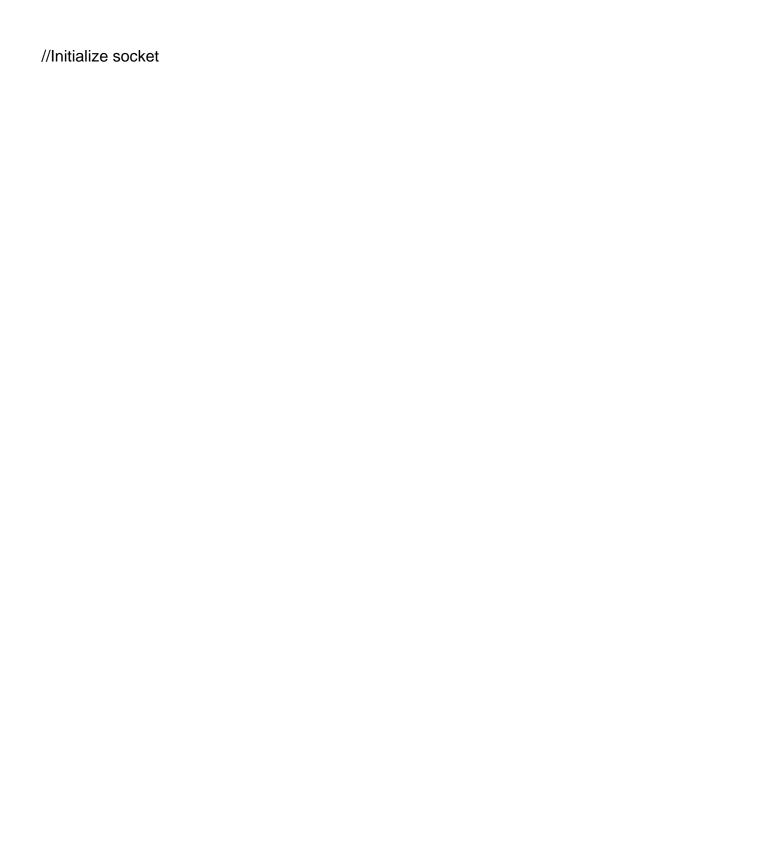
File Client:		

mport java.io.BufferedOutputStream; import java.io.FileOutputStream; import java.io.InputStrea	m;

















while((bytesRead=is.read(contents))!=-1) bos.write(contents, 0, bytesRead);





}			

}			





E:\nwlab>client E:\nwlab>java FileClient File saved successfully!



Ex.No: 4 Simulation of DNS using UDP Sockets



**DNS Server** 





{			

private static int indexOf(String[] array, String str)

{			

str = str.trim();

for (int i=0; i < array.length; 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",

System.out.println("Press Ctrl + C to Quit"); while (true)

{			



byte[] receivedata = new byte[1021];





InetAddress ipaddress = recvpack.getAddress(); int port = recvpack.getPort();



System.out.println("Request for host " + sen); if(indexOf (hosts, sen) != -1)

capsent = ip[indexOf (hosts, sen)];



capsent = "Host Not Found";



DatagramPacket pack	x = new DatagramPacke	et (senddata, s	senddata.length,ipa	address,port); serv	ersocket.s



UDP DNS Client java import java.io.\*; import java.net.\*;



{			

public static void main(String args[])throws IOException

{			



InetAddress ipaddress; if (args.length == 0)







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,







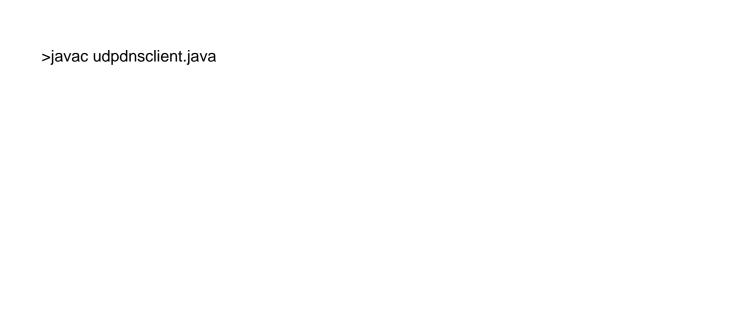




Press Ctrl + C to Quit Request for host yahoo.com Request for host cricinfo.com

Request for host youtube.com







Enter the hostname : yahoo.com IP Address: 68.180.206.184



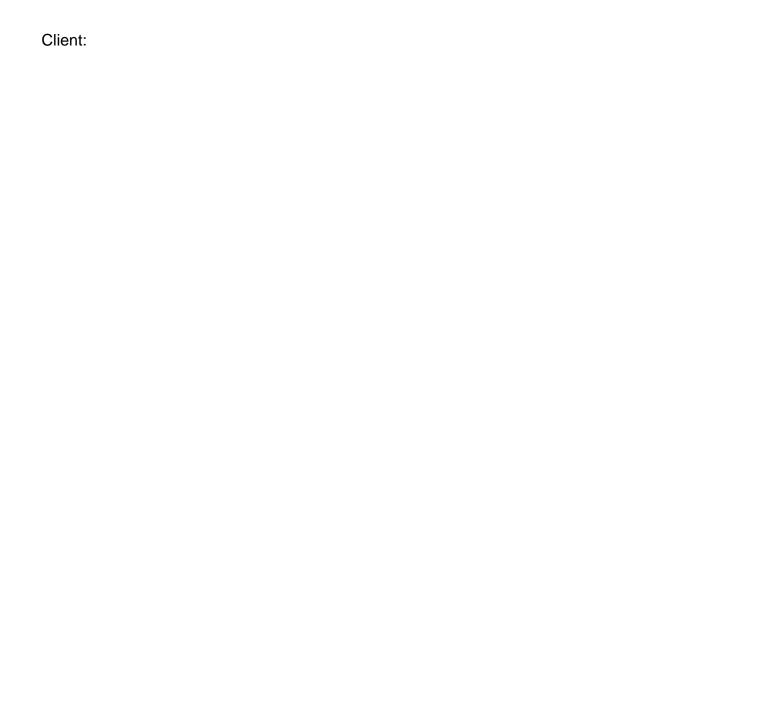
Enter the hostname : cricinfo.com IP Address: 80.168.92.140



Enter the hostname : youtube.com IP Address: Host Not Found

Ex.No:5 Write a code simulating ARP /RARP protocols





import java.io.\*; import java.net.\*; import java.util.\*; class Clientarp

{			



{			

{			





String str1=in.readLine(); dout.writeBytes(str1+'\n'; String str=din.readLine();

System.out.println("The Physical Address is: "+str); clsct.close();

}			



{			



}			

Server:

import java.io.\*; import java.net.\*; import java.util.\*; class Serverarp

{			



{			

ServerSocket obj=new ServerSocket(139); Socket obj1=obj.accept(); while(true)

{			



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++)

{			



{			



}			

}			



}			

}			



{			



}			

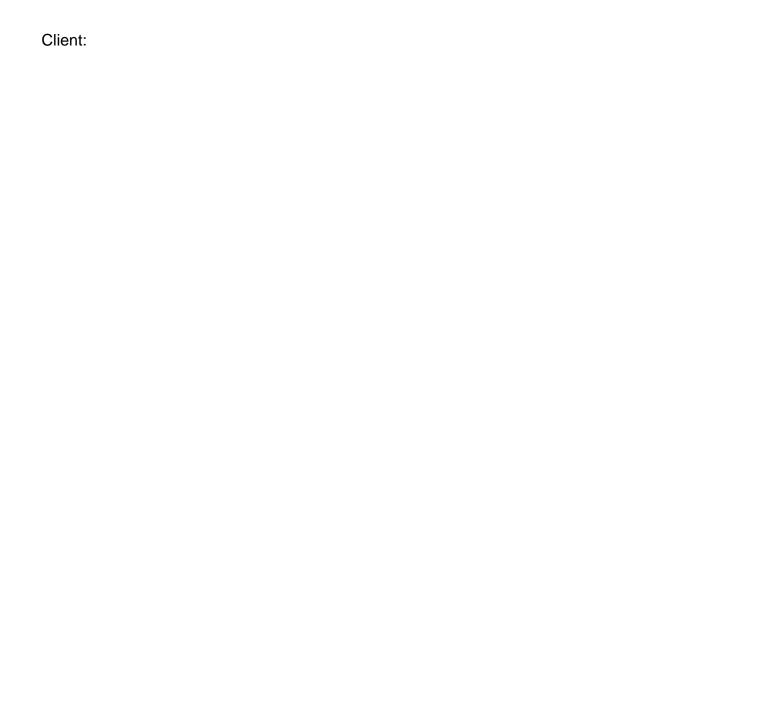




The Physical Address is: 6A:08:AA:C2

Program for Reverse Address Resolution Protocol (RARP) using UDP





import java.io.\*; import java.net.\*; import java.util.\*; class Clientrarp12

{			



{			

{			



InetAddress addr=InetAddress.getByName("127.0.0.1");



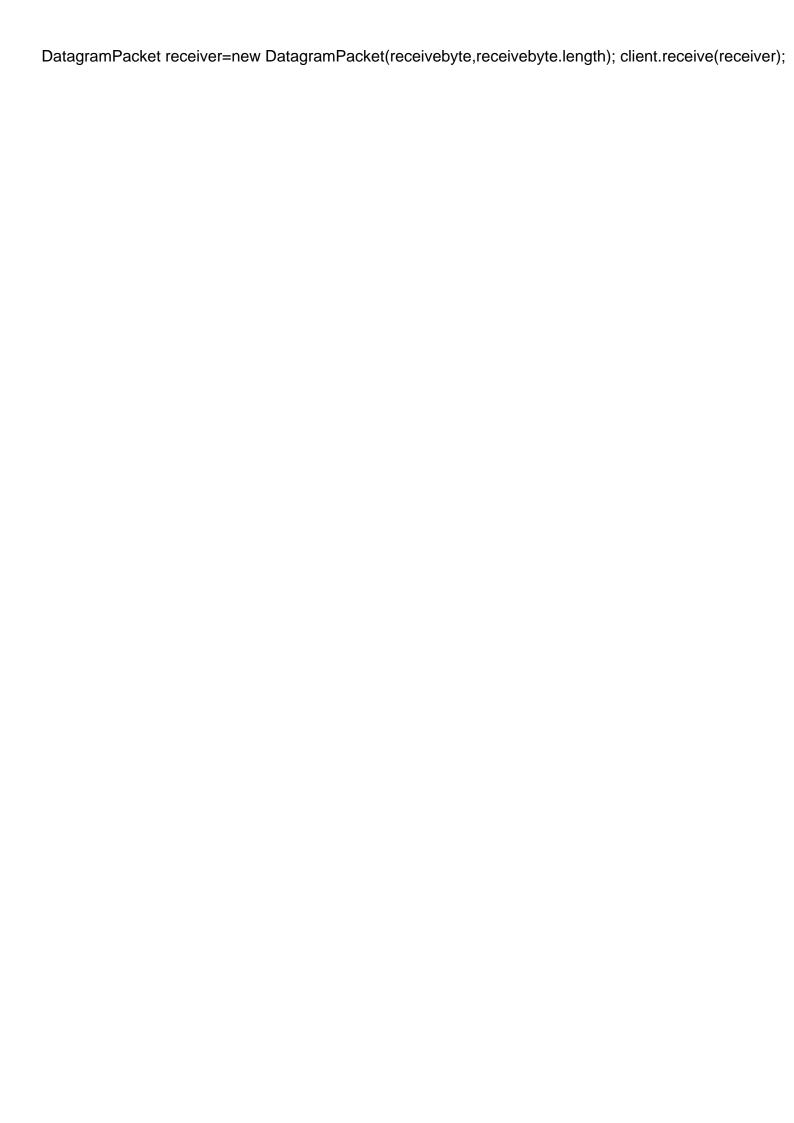














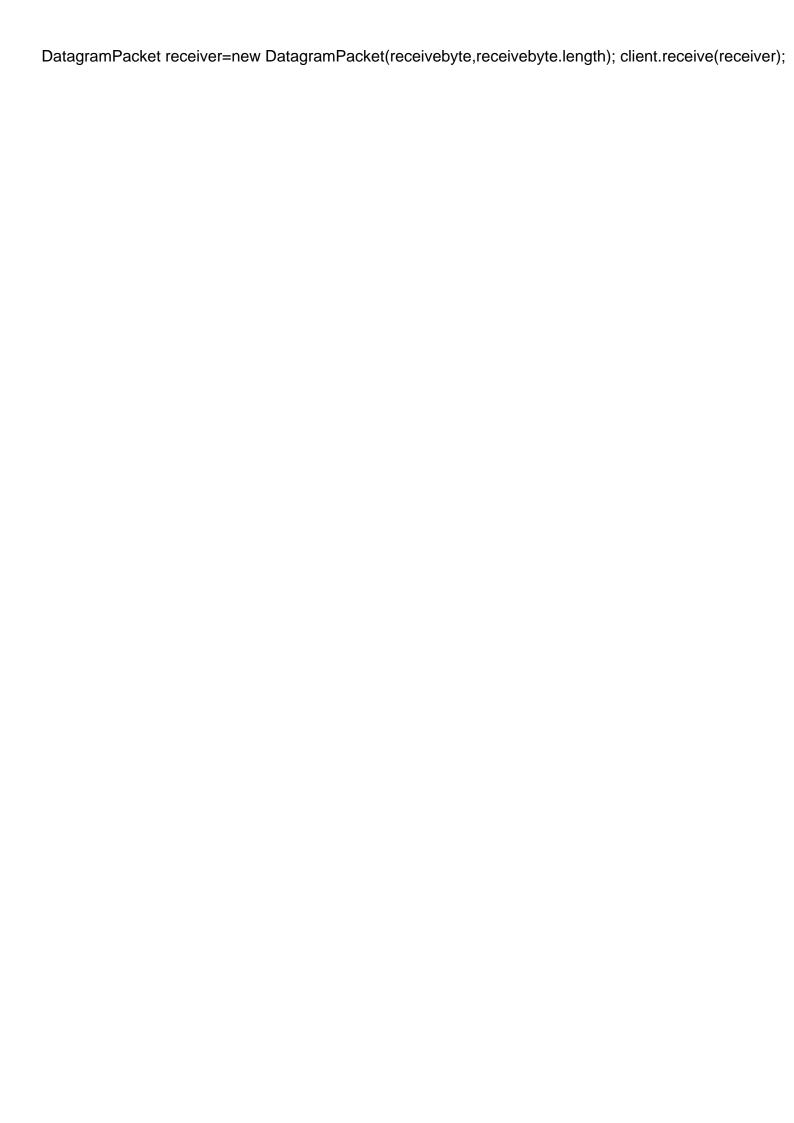


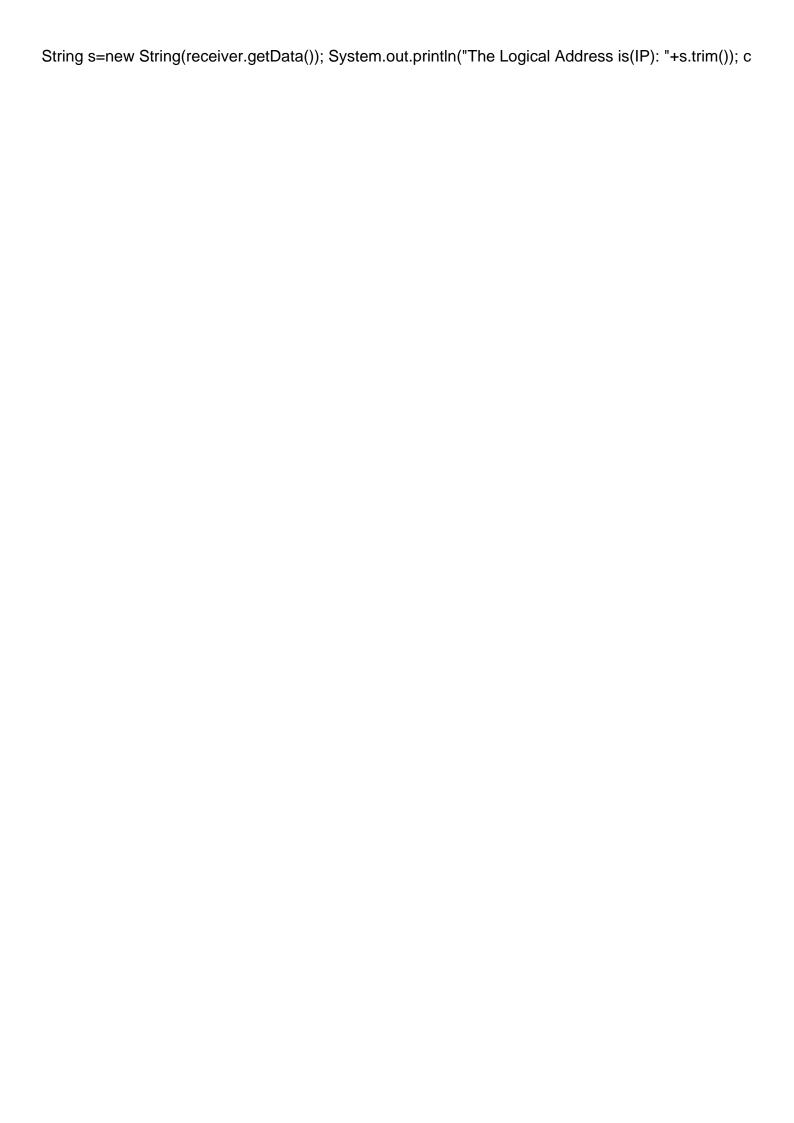












}			



{			



Server:

import java.io.\*; import java.net.\*; import java.util.\*; class Serverrarp12

{			



{			

DatagramSocket server=new DatagramSocket(1309); while(true)

{			

byte[] sendbyte=new byte[1024]; byte[] receivebyte=new byte[1024];



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++)

{			



{			

sendbyte=ip[i].getBytes(); DatagramPacket sender = new







{			



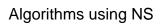


I:\ex>java Serverrarp12 I:\ex>java Clientrarp12

Enter the Physical address (MAC): 6A:08:AA:C2

The Logical Address is(IP): 165.165.80.8

Ex.No: 6 Study of Network simulator (NS) and Simulation of Congestion Control



Program:

include <wifi\_lte/wifi\_lte\_rtable.h> struct r\_hist\_entry \*elm, \*elm2;

int num\_later = 1;

 $elm = STAILQ\_FIRST(\&r\_hist\_);$ 

while (elm != NULL && num\_later <= num\_dup\_acks\_){ num\_later;

 $elm = STAILQ\_NEXT(elm, \, linfo\_); \\$ 

}			

if (elm != NULL){

 $elm = findDataPacketInRecvHistory(STAILQ\_NEXT(elm,linfo\_));\\$ 

if (elm != NULL){

elm2 = STAILQ\_NEXT(elm, linfo\_); while(elm2 != NULL){

if (elm2->seq\_num\_ < seq\_num && elm2->t\_recv\_ <



STAILQ\_REMOVE(&r\_hist\_,elm2,r\_hist\_entry,linfo\_); delete elm2;





 $elm2 = STAILQ\_NEXT(elm, linfo\_);$ 

}			

}			

}			

}			

void DCCPTFRCAge	nt::removeAcksRe	cvHistory(){ stru	ct r_hist_entry	*elm1 = STAILQ	_FIRST(&r_hist_)	,

int num\_later = 1;

while (elm1 != NULL && num\_later <= num\_dup\_acks\_){ num\_later;

 $elm1 = STAILQ\_NEXT(elm1, linfo\_); \\$ 

}			

if(elm1 == NULL)

return;

 $elm2 = STAILQ\_NEXT(elm1, linfo\_); \ while(elm2 != NULL)\{$ 

 $if (elm2->type\_ == DCCP\_ACK) \{ \ STAILQ\_REMOVE (\&r\_hist\_,elm2,r\_hist\_entry,linfo\_); \ delete \ elm2; \\$ 



elm1 = elm2;

}			

 $elm2 = STAILQ\_NEXT(elm1, linfo\_); \\$ 

}			

}			

inline r\_hist\_entry

*DCCPTFRCAgent::findDataPacketInRecvHistory(r_his	st_entry *start){    while(start != NULL &8	start->type

start = STAILQ\_NEXT(start,linfo\_); return start;

}			

Ex.No: 7 Study of TCP/UDP performance using Simulation tool.





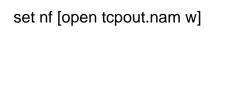
\$ns color 0 Blue

\$ns color 1 Red

\$ns color 2 Yellow set n0 [\$	ons node] set n1 [\$ns	node] set n2 [\$ns node	] set n3 [\$ns node]

set f [open tcpout.tr w]

\$ns trace-all \$f



\$ns namtrace-all \$nf

\$ns duplex-link \$n0 \$n2 5Mb 2ms DropTail

\$ns duplex-link \$n1 \$n2 5Mb 2ms DropTail

\$ns duplex-link \$n2 \$n3 1.5Mb 10ms DropTail

\$ns duplex-link-op \$n0 \$n2 orient right-up

\$ns duplex-link-op \$n1 \$n2 orient right-down

\$ns duplex-link-op \$n2 \$n3 orient right

\$ns duplex-link-op \$n2 \$n3 queuePos 0.5 set tcp [new Agent/TCP]

set sink [new Agent/TCPSink]

\$ns attach-agent \$n1 \$tcp

\$ns attach-agent \$n3 \$sink

\$ns connect \$tcp \$sink

set ftp [new Application/FTP]

\$ftp attach-agent \$tcp

\$ns at 1.2 "\$ftp start"

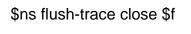
\$ns at 1.35 "\$ns detach-agent \$n1 \$tcp; \$ns detach-agent \$n3 \$sink"

\$ns at 3.0 "finish" proc finish {} {

}			







close \$nf



exec xgraph tcpout.tr -geometry 600x800 & exec nam tcpout.nam &

**UDP** Performance

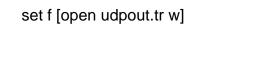




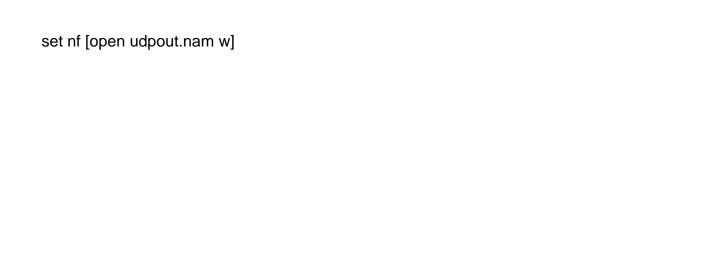
\$ns color 0 Blue

\$ns color 1 Red

\$ns color 2 Yellow set n0 [\$	ons node] set n1 [\$ns	node] set n2 [\$ns node	] set n3 [\$ns node]



\$ns trace-all \$f



\$ns namtrace-all \$nf

\$ns duplex-link \$n0 \$n2 5Mb 2ms DropTail

\$ns duplex-link \$n1 \$n2 5Mb 2ms DropTail

\$ns duplex-link \$n2 \$n3 1.5Mb 10ms DropTail

\$ns duplex-link-op \$n0 \$n2 orient right-up

\$ns duplex-link-op \$n1 \$n2 orient right-down

\$ns duplex-link-op \$n2 \$n3 orient right

\$ns duplex-link-op \$n2 \$n3 queuePos 0.5 set udp0 [new Agent/UDP]

\$ns attach-agent \$n0 \$udp0

set cbr0 [new Application/Traffic/CBR]

\$cbr0 attach-agent \$udp0 s

\$ns attach-agent \$n1 \$null0 set null1 [new Agent/Null]

\$ns attach-agent \$n1 \$null1

\$ns connect \$udp0 \$null0

\$ns connect \$udp1 \$null1

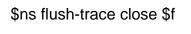
\$ns at 1.0 "\$cbr0 start"

\$ns at 1.1 "\$cbr1 start"

puts [\$cbr0 set packetSize\_] puts [\$cbr0 set interval\_]

\$ns at 3.0 "finish" proc finish {} {

global ns f nf



close \$nf

puts "Running nam.." exec nam udpout.nam & exit 0

}			





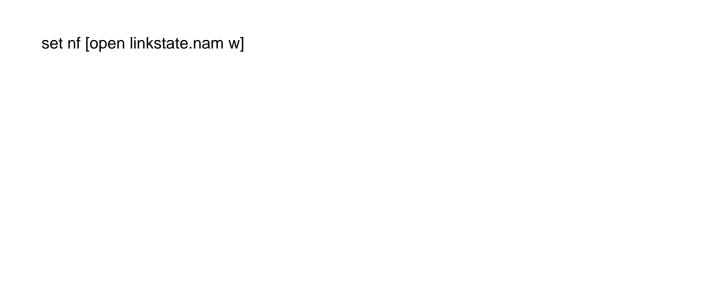
Ex.No: 8 Simulation of Distance Vector/ Link State Routing algorithm.

LINK STATE ROUTING PROTOCOL



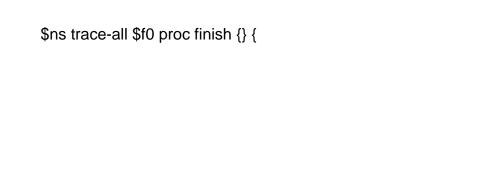


\$ns rtproto LS



\$ns namtrace-all \$nf

set f0 [open linkstate.tr w]



global ns f0 nf

\$ns flush-trace close \$f0 close \$nf

exec nam linkstate.nam & exit 0

}			

for {set i 0} {i <7} {incr i} { set n(i) [n = node]

}			

for {set i 0} {\$i <7} {incr i} {

 $n \leq n \leq n$ 

}			

set udp0 [new Agent/UDP]

\$ns attach-agent \$n(0) \$udp0

set cbr0 [new Application/Traffic/CBR]

\$cbr0 set packetSize\_ 500

\$cbr0 attach-agent \$udp0 set null0 [new Agent/Null]

\$ns attach-agent \$n(3) \$null0

\$ns connect \$udp0 \$null0

\$ns at 0.5 "\$cbr0 start"

\$ns rtmodel-at 1.0 down \$n(1) \$n(2)

\$ns at 5.0 "finish"



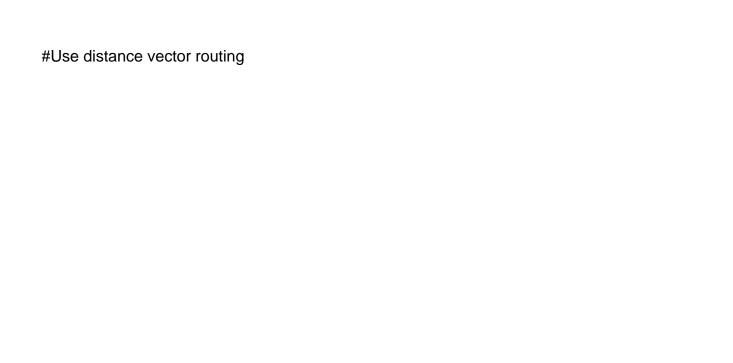


DISTANCE VECTOR ROUTING ALGORITHM



#Distance vector routing protocol ? distvect.tcl #Create a simulator object





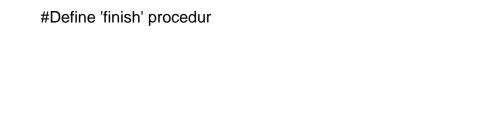




\$ns namtrace-all \$nf # Open tracefile

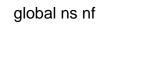


\$ns trace-all \$nt



proc finish {}

{			





#Execute nam on the trace file exec nam -a out.nam &

}			

# Create 8 nodes set n1 [\$ns node] set n2 [\$ns node] set n3 [\$ns node] set n4 [\$ns node] set n5 [\$ns node]	Эt



\$ns duplex-link \$n1 \$n2 1Mb 10ms DropTail

\$ns duplex-link \$n2 \$n3 1Mb 10ms DropTail

\$ns duplex-link \$n3 \$n4 1Mb 10ms DropTail

\$ns duplex-link \$n4 \$n5 1Mb 10ms DropTail

\$ns duplex-link \$n5 \$n6 1Mb 10ms DropTail

\$ns duplex-link \$n6 \$n7 1Mb 10ms DropTail

\$ns duplex-link \$n7 \$n8 1Mb 10ms DropTail

\$ns duplex-link \$n8 \$n1 1Mb 10ms DropTail # specify layout as a octagon

\$ns duplex-link-op \$n1 \$n2 orient left-up

\$ns duplex-link-op \$n2 \$n3 orient up

\$ns duplex-link-op \$n3 \$n4 orient right-up

\$ns duplex-link-op \$n4 \$n5 orient right

\$ns duplex-link-op \$n5 \$n6 orient right-down

\$ns duplex-link-op \$n6 \$n7 orient down

\$ns duplex-link-op \$n7 \$n8 orient left-down



\$ns attach-agent \$n1 \$udp0

#Create a CBR traffic source and attach it to udp0

\$cbr0 set packetSize\_ 500

\$cbr0 attach-agent \$udp0



\$ns attach-agent \$n4 \$null0



\$ns connect \$udp0 \$null0



\$ns at 0.0 "\$n1 label Source"

\$ns at 0.0 "\$n4 label Destination"

\$ns at 0.5 "\$cbr0 start"

\$ns rtmodel-at 1.0 down \$n3 \$n4

\$ns rtmodel-at 2.0 up \$n3 \$n4

\$ns at 4.5 "\$cbr0 stop"

#Call the finish procedure after 5 seconds of simulation time

\$ns at 5.0 "finish" #Run the simulation





Ex.No:9 Performance Evaluation of Routing protocols using Simulation tool.					







\$ns color 1 Blue

\$ns color 2 Red #Open the Trace file

set file1 [open out.tr w]

\$ns trace-all \$file1





{			

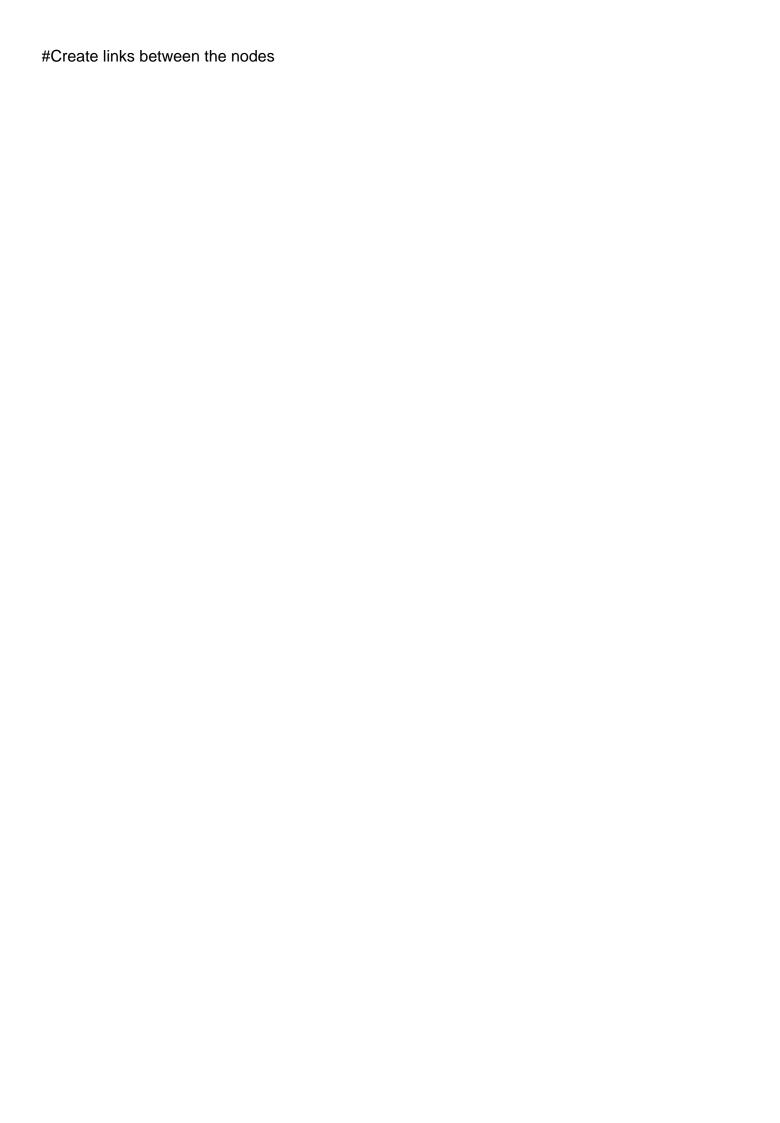
global ns file1 file2

\$ns flush-trace close \$file1 close \$file2

exec nam out.nam &







\$ns duplex-link \$n0 \$n1 0.3Mb 10ms DropTail

\$ns duplex-link \$n1 \$n2 0.3Mb 10ms DropTail

\$ns duplex-link \$n2 \$n3 0.3Mb 10ms DropTail

\$ns duplex-link \$n1 \$n4 0.3Mb 10ms DropTail

\$ns duplex-link \$n3 \$n5 0.5Mb 10ms DropTail

\$ns duplex-link \$n4 \$n5 0.5Mb 10ms DropTail



\$ns duplex-link-op \$n0 \$n1 orient right

\$ns duplex-link-op \$n1 \$n2 orient right

\$ns duplex-link-op \$n2 \$n3 orient up

\$ns duplex-link-op \$n1 \$n4 orient up-left

\$ns duplex-link-op \$n3 \$n5 orient left-up

\$ns duplex-link-op \$n4 \$n5 orient right-up

#Setup a TCP connection

set tcp [new Agent/TCP/Newreno]

\$ns attach-agent \$n0 \$tcp

set sink [new Agent/TCPSink/DelAck]

\$ns attach-agent \$n5 \$sink

\$ns connect \$tcp \$sink

#Setup a FTP over TCP connection set ftp [new Application/FTP]

\$ftp attach-agent \$tcp

\$ftp set type\_FTP

\$ns rtmodel-at 4.5 up \$n1 \$n4

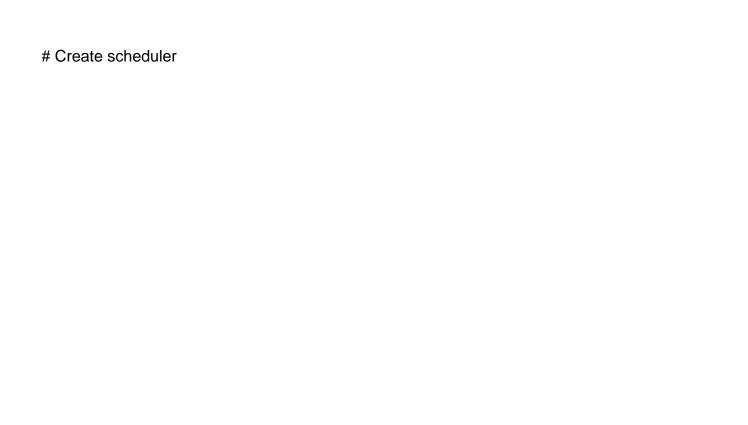
\$ns at 0.1 "\$ftp start"

\$ns at 6.0 "finish"



MULTICASTING ROUTING PROTOCOL





#Create an event scheduler wit multicast turned on set ns [new Simulator -multicast on]	

#\$ns multicast #Turn on Tracing

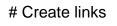


\$ns trace-all \$tf

# Turn on nam Tracing set fd [open mcast.nam w]

\$ns namtrace-all \$fd # Create nodes





\$ns duplex-link \$n0 \$n2 1.5Mb 10ms DropTail

\$ns duplex-link \$n1 \$n2 1.5Mb 10ms DropTail

\$ns duplex-link \$n2 \$n3 1.5Mb 10ms DropTail

\$ns duplex-link \$n3 \$n4 1.5Mb 10ms DropTail

\$ns duplex-link \$n3 \$n7 1.5Mb 10ms DropTail

\$ns duplex-link \$n4 \$n5 1.5Mb 10ms DropTail

\$ns duplex-link \$n4 \$n6 1.5Mb 10ms DropTail

# Routing protocol: say distance vector #Protocols: CtrMcast, DM, ST, BST set mproto DM



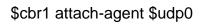


\$ns attach-agent \$n0 \$udp0

\$udp0 set dst\_addr\_ \$group1

\$udp0 set dst\_port\_ 0

set cbr1 [new Application/Traffic/CBR]

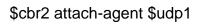


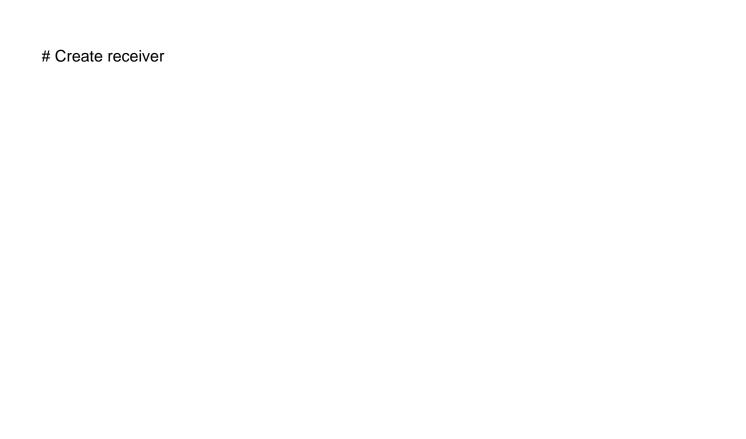
# Transport agent for the traffic source set udp1 [new Agent/UDP]				

\$ns attach-agent \$n1 \$udp1

\$udp1 set dst\_addr\_ \$group2

set cbr2 [new Application/Traffic/CBR]





set rcvr1 [new Agent/Null]

\$ns attach-agent \$n5 \$rcvr1

\$ns at 1.0 "\$n5 join-group \$rcvr1 \$group1" set rcvr2 [new Agent/Null]

\$ns attach-agent \$n6 \$rcvr2

\$ns at 1.5 "\$n6 join-group \$rcvr2 \$group1" set rcvr3 [new Agent/Null]

\$ns attach-agent \$n7 \$rcvr3

\$ns at 2.0 "\$n7 join-group \$rcvr3 \$group1" set rcvr4 [new Agent/Null]

\$ns attach-agent \$n5 \$rcvr1

\$ns at 2.5 "\$n5 join-group \$rcvr4 \$group2" set rcvr5 [new Agent/Null]

\$ns attach-agent \$n6 \$rcvr2

\$ns at 3.0 "\$n6 join-group \$rcvr5 \$group2" set rcvr6 [new Agent/Null]

\$ns attach-agent \$n7 \$rcvr3

\$ns at 3.5 "\$n7 join-group \$rcvr6 \$group2"

\$ns at 4.0 "\$n5 leave-group \$rcvr1 \$group1"

\$ns at 4.5 "\$n6 leave-group \$rcvr2 \$group1"

\$ns at 5.0 "\$n7 leave-group \$rcvr3 \$group1"

\$ns at 5.5 "\$n5 leave-group \$rcvr4 \$group2"

\$ns at 6.0 "\$n6 leave-group \$rcvr5 \$group2"

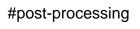
\$ns at 6.5 "\$n7 leave-group \$rcvr6 \$group2" # Schedule events

\$ns at 0.5 "\$cbr1 start"

\$ns at 9.5 "\$cbr1 stop"

\$ns at 0.5 "\$cbr2 start"

\$ns at 9.5 "\$cbr2 stop"



\$ns at 10.0 "finish" proc finish {}

{			

global ns tf

\$ns flush-trace close \$tf

exec nam mcast.nam & exit 0

}			





\$ns color 10 red

\$ns color 11 green

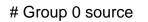
\$ns color 30 purple

\$ns color 31 green

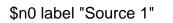
# Manual layout: order of the link is significant! #\$ns duplex-link-op \$n0 \$n1 orient right	



#\$ns duplex-link-op \$n2 \$n3 queuePos 0.5

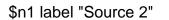


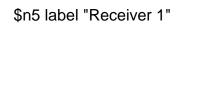




# Group 1 source

















#\$n2 add-mark m0 red #\$n2 delete-mark m0"



\$ns set-animation-rate 3.0ms



Ex.No:10 Simulation of ErrorDetection Code (like CRC)



import java.io.\*; class crc\_gen

{			

public static void main(String args[]) throws IOException {













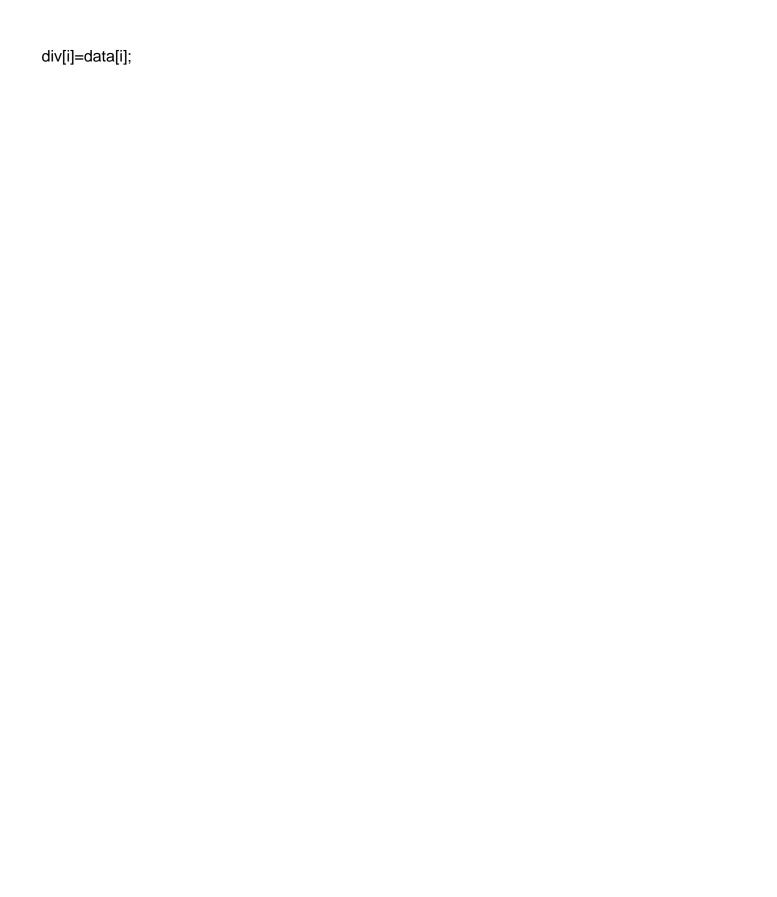


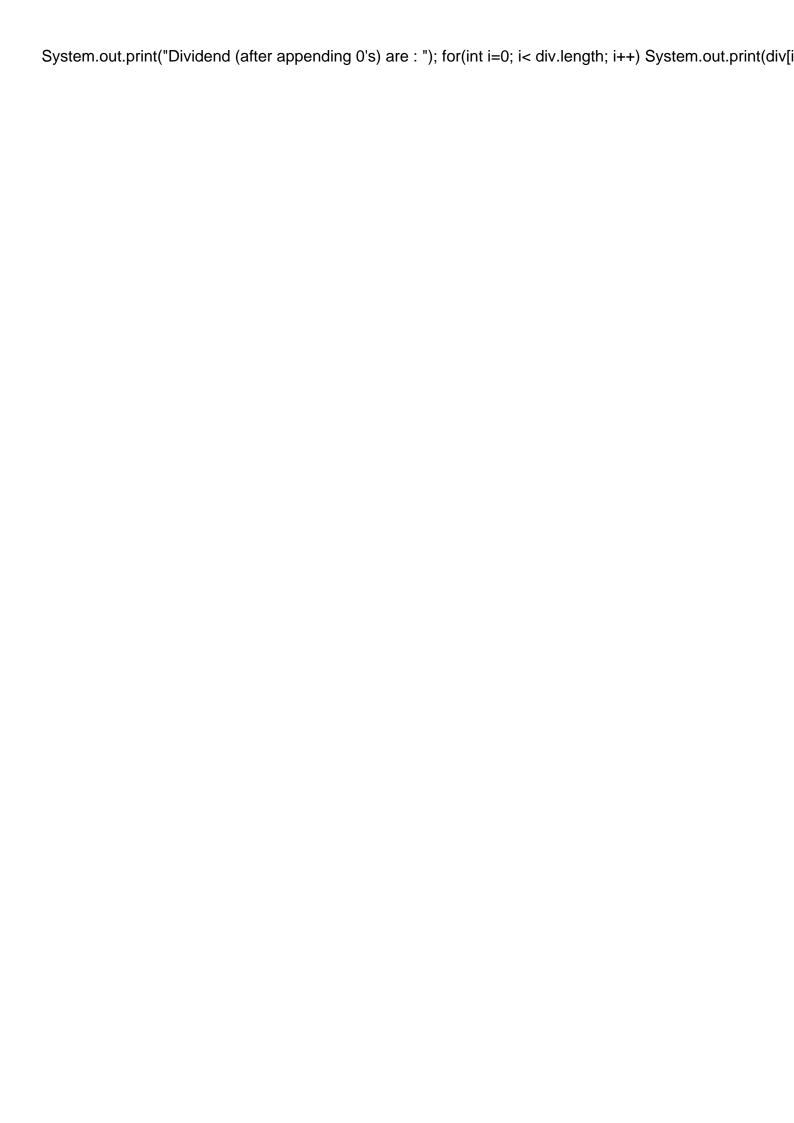
for(int i=0; i<	divisor_bits; i+	+) divisor[i]=Ir	nteger.parsel	nt(br.readLir	ne()); System	.out.print("Da	ta bits are : ");

System.out.print("divisor bits are : "); for(int i=0; i< divisor\_bits; i++) System.out.print(divisor[i]); Sys \*/ tot\_length=data\_bits+divisor\_bits-1; div=new int[tot\_length];

rem=new int[tot\_length]; crc=new int[tot\_length];

/\* CRC GENERATION \*/ for(int i=0;i<data.length;i++)







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

}			

rem=divide(div, divisor, rem); for(int i=0;i<div.length;i++)

{			



}			



/\* ERROR DETECTION \*/ System.out.println();





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

}			

rem=divide(crc, divisor, rem); for(int i=0; i< rem.length; i++)

{			

if(rem[i]!=0)

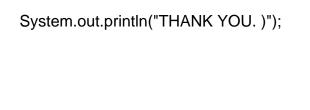
{			

System.out.println("Error"); break;

}			

if(i==rem.length-1) System.out.println("No Error");

}			



}			



{			

int cur=0; while(true)

{			





}			

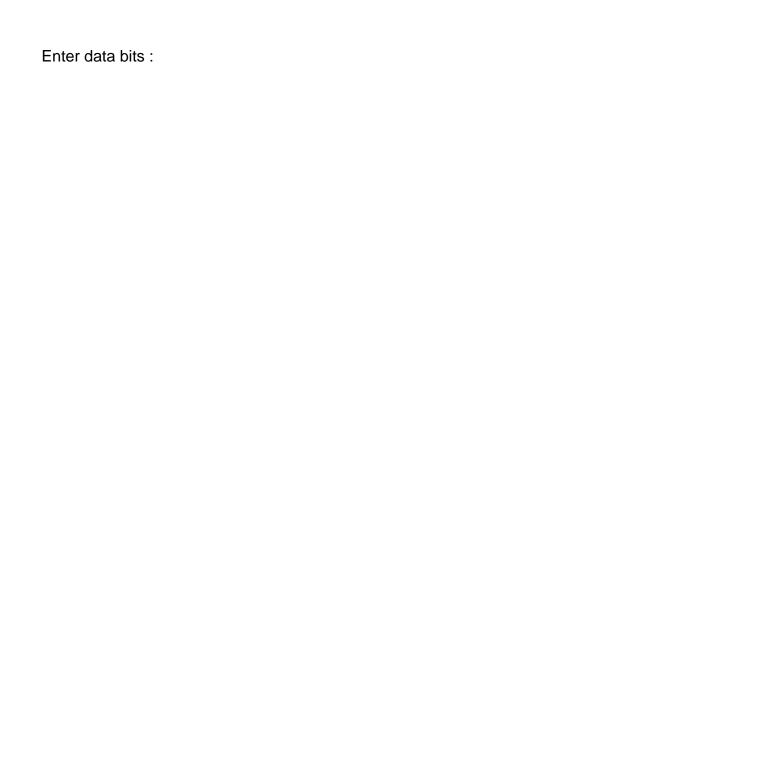
return rem;	

}			

}			











Dividend (after appending 0's) are :					

CARRIER SENSE MULTIPLE ACCESS





\$ns color 1 blue

\$ns color 2 red

set fi1 [open out.tr w]

set winfile [open WinFile w]

\$ns trace-all \$fi1

set fi2 [open out.nam w]



{			

\$ns flush-trace close \$fi1 close \$fi2

exec nam out.nam & exit 0

}			

set n0 [\$ns node] set n1 [\$ns node] set n2 [\$ns node]

set n3 [\$ns node] set n4 [\$ns node] set n5 [\$ns node]





\$ns duplex-link \$n0 \$n2 2Mb 10ms DropTail

\$ns duplex-link \$n1 \$n2 2Mb 10ms DropTail

\$ns simplex-link \$n2 \$n3 0.3Mb 100ms DropTail

\$ns simplex-link \$n3 \$n2 0.3Mb 100ms DropTail



\$ns attach-agent \$n0 \$tcp

set sink [new Agent/TCPSink/DelAck]

\$ns attach-agent \$n4 \$sink

\$ns connect \$tcp \$sink

\$tcp set window\_ 8000

set ftp [new Application/FTP]

\$ftp set type\_FTP

set udp [new Agent/UDP]

\$ns attach-agent \$n1 \$udp set null [new Agent/Null]

\$ns attach-agent \$n5 \$null

\$ns connect \$udp \$null

set cbr [new Application/Traffic/CBR]

\$cbr attach-agent \$udp

\$cbr set type\_ CBR

\$cbr set packet\_size\_ 1000

\$cbr set rate\_ 0.01mb

\$cbr set random\_ false

\$ns at 0.1 "\$cbr start"

\$ns at 1.0 "\$ftp start"

\$ns at 24.0 "\$ftp stop"

\$ns at 24.5 "\$cbr stop"



{			

global ns set time 0.1



set cwnd [\$tcpSource set cwnd_] set w	nd [\$tcpSource set	window_] puts \$file "\$	now \$cwnd"

\$ns at [expr \$now+\$time] "plotwindow \$tcpSource \$file"

}			

\$ns at 1.0 "plotwindow \$tcp \$winfile"

\$ns at 5 "\$ns trace-annotate \"packet drop\""

\$ns at 125.0 "finish"



