```
set ns [new Simulator]

set nf [open star.nam w]
$ns namtrace-all $nf

proc finish {} {
global ns nf
$ns flush-trace
close $nf
exec nam star.nam &
exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 10ms DropTail
$ns duplex-link $n3 $n2 10Mb 10ms DropTail

$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

$ns at 1.0 "finish"

$ns run
```

```
set ns [new Simulator]

$ns rtproto DV

set nf [open prac3.nam w]
$ns namtrace-all $nf

proc finish {} {
global ns nf
$ns flush-trace
close $nf
exec nam prac3.nam &
exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 10ms DropTail
$ns duplex-link $n2 $n3 10Mb 10ms DropTail
$ns duplex-link $n3 $n0 10Mb 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 right
$ns duplex-link-op $n3 $n0 orient right

set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n2 $sink
$ns connect $tcp $sink

set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set packet_size_ 1000
$ftp set rate_ 1mb

$ns at 1.0 "$ftp start"
$ns rtmodel-at 2.0 down $n1 $n2
```

```
$ns rtmodel-at 3.0 up $n1 $n2
$ns at 4.0 "$ftp stop"

$ns at 5.0 "finish"
$ns run



set ns [new Simulator]

$ns color 1 Blue
$ns color 2 Red

set nf [open prac4.nam w]
$ns namtrace-all $nf

proc finish {} {
global ns nf
$ns flush-trace
close $nf
exec nam prac4.nam &
exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 20Mb 10ms DropTail
$ns duplex-link $n1 $n2 20Mb 10ms DropTail
$ns duplex-link $n3 $n2 5Mb 10ms DropTail

$ns queue-limit $n2 $n3 5

$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

$ns duplex-link-op $n2 $n3 queuePos 1

set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
```

```
$ns attach-agent $n3 $sink
$ns connect $tcp $sink

$tcp set fid_ 1

set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set rate_ 1mb

set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null

$udp set fid_ 2

set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb

$ns at 1.0 "$ftp start"
$ns at 1.5 "$cbr start"
$ns at 2.0 "$ftp stop"
$ns at 3.0 "$cbr stop"

$ns at 5.0 "finish"

$ns run
```

```
set ns [new Simulator]

set nf [open prac1.nam w]
$ns namtrace-all $nf

proc finish {} {
global ns nf
$ns flush-trace
```

```
close $nf
exec nam prac1.nam &
exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 10ms DropTail
$ns duplex-link $n3 $n2 10Mb 10ms DropTail

$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink

set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set rate_ 1mb

$ns at 1.0 "$ftp start"
$ns at 2.0 "$ftp stop"

$ns at 5.0 "finish"

$ns run



set ns [new Simulator]

set nf [open prac2.nam w]
$ns namtrace-all $nf

proc finish {} {
```

```
global ns nf
$ns flush-trace
close $nf
exec nam prac2.nam &
exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 10ms DropTail
$ns duplex-link $n3 $n2 10Mb 10ms DropTail

$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $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_ 1mb

$ns at 1.0 "$cbr start"
$ns at 2.0 "$cbr stop"


$ns at 3.0 "finish"

$ns run
```

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

```
public class TCPClient {
    public static void main(String args[]){
        int serverport = 8080;
        String serveraddress="localhost";
        try{
            Socket socket = new Socket(serveraddress, serverport);
            System.out.println("Connected to Server");

            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new
PrintWriter(socket.getOutputStream(),true);

            BufferedReader  console = new BufferedReader(new
InputStreamReader(System.in));
            String msg;
            while(true){
                System.out.print("Enter message (or 'quit' to exit): ");
                msg = console.readLine();
                if(msg == null || msg.equalsIgnoreCase("quit")){
                    break;
                }
                out.println(msg);
                String response= in.readLine();
                System.out.println("Server responce :"+ response);


            }

            in.close();
            out.close();
            socket.close();

        }
        catch(IOException e){
            e.printStackTrace();

        }
        finally{
        }


    }

}
```

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

public class TCPServer{
    public static void main (String args []){
        int port = 8080;
        try{
            ServerSocket serverSocket = new ServerSocket(port);
            System.out.println("Connected to port"+ port);
            Socket clientsocket= serverSocket.accept();
            System.out.println("Client Connected");

            BufferedReader in = new BufferedReader(new
InputStreamReader(clientsocket.getInputStream()));
            PrintWriter out = new
PrintWriter(clientsocket.getOutputStream(),true);

            String msg;
            while((msg = in.readLine()) != null){
                System.out.println("Recived: "+ msg);
                out.println("msg recived from client");
            }
            in.close();
            out.close();
            clientsocket.close();
            serverSocket.close();

        }
        catch(IOException e){
            e.printStackTrace();
        }
        finally{

        }
    }
}
```

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

public class UDPClient {
    public static void main(String[] args) {
        try {
            DatagramSocket clientSocket = new DatagramSocket();

            BufferedReader inputReader = new BufferedReader(new
InputStreamReader(System.in));
            InetAddress serverAddress =
InetAddress.getByName("127.0.0.1");
            int serverPort = 12345;

            while (true) {
                System.out.print("Enter message to send to server (type
'exit' to exit): ");
                String sendData = inputReader.readLine();

                byte[] sendBuffer = sendData.getBytes();

                DatagramPacket sendPacket = new DatagramPacket(sendBuffer,
sendBuffer.length, serverAddress, serverPort);

                clientSocket.send(sendPacket);

                if (sendData.equalsIgnoreCase("exit")) {
                    System.out.println("Closing client...");
                    break;
                }
                byte[] receiveBuffer = new byte[1024];
                DatagramPacket receivePacket = new
DatagramPacket(receiveBuffer, receiveBuffer.length);
                clientSocket.receive(receivePacket);

                String receivedData = new String(receivePacket.getData(),
0, receivePacket.getLength());
                System.out.println("Received from " +
receivePacket.getAddress().getHostAddress() + ": " + receivedData);
            }

            clientSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

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

public class UDPServer{
    public static void main(String args []){
        try{
        DatagramSocket serverSocket = new DatagramSocket(12345);
        System.out.println("Udp server listening ....");
        byte [] reciverbuffer = new byte [1024];
        while(true){
            DatagramPacket receivePacket = new
DatagramPacket(reciverbuffer,reciverbuffer.length);
            serverSocket.receive(receivePacket);
            String receiveData = new
String(receivePacket.getData(),0,receivePacket.getLength());
            System.out.println("Received from client :" + receiveData);
            if (receiveData.equalsIgnoreCase("exit")) {
                System.out.println("Client has exited. Closing
server...");
                break;
            }
            String responceData = "Server:" + receiveData;
            byte [] sendBuffer = responceData.getBytes();
            DatagramPacket sendPacket = new
DatagramPacket(sendBuffer,sendBuffer.length,receivePacket.getAddress(),rec
eivePacket.getPort());
            serverSocket.send(sendPacket);
        }
        serverSocket.close();
        }
        catch (IOException e){
            e.printStackTrace();
        }
    }
}
```