

## PART-A

**1a) Design a program in which sender module should count the no of bytes in the frame and receive module should display each frame received.**

### Sender.java

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

public class Sender{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of frames to be sent: ");
        int n = sc.nextInt();
        try{
            Socket s = new Socket("localhost",7002);
            DataOutputStream dataout = new
DataOutputStream(s.getOutputStream());
            sc = new Scanner(System.in);
            for(int i=0;i<n;i++)
            {
                System.out.println("Enter the frame to send: ");
                String data = sc.nextLine();
                int len = data.length();
                System.out.println("Sending frame "+data+" with
"+len+" bytes.");
                dataout.writeUTF(data);
                dataout.flush();
            }
            dataout.close();
            s.close();
            sc.close();
        }

        catch(Exception e){
            System.out.println("Error Occured: " +e);
        }
    }
}
```

```
}
```

## **Receiver.java**

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

public class Receiver{
    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of frames to be recieved: ");
        int n = sc.nextInt();
        try{
            ServerSocket ss = new ServerSocket(7002);
            Socket s = ss.accept();
            DataInputStream datain = new
DataInputStream(s.getInputStream());
            String finaldata = new String();
            finaldata = "";
            for(int i=0;i<n;i++)
            {
                String data = datain.readUTF();
                System.out.println("Recieved frame:"+data);
                finaldata = finaldata.concat(data+ " ");
            }
            System.out.println("Final Recieved Data: "+finaldata);
            datain.close();
            ss.close();
            sc.close();
        }
        catch(Exception e){
            System.out.println("Error Occured: "+ e);
        }
    }
}
```

```
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p1a>javac Receiver.java
```

```
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p1a>java Receiver
```

```
Enter number of frames to be recieved:
```

```
3
```

```
Recieved frame:ABC
```

```
Recieved frame:DEF
```

```
Recieved frame:GHI
```

```
Final Recieved Data: ABC DEF GHI
```

```
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p1a>_
```

```
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p1a>javac Sender.java
```

```
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p1a>java Sender
```

```
Enter number of frames to be sent:
```

```
3
```

```
Enter the frame to send:
```

```
ABC
```

```
Sending frame ABC with 3 bytes.
```

```
Enter the frame to send:
```

```
DEF
```

```
Sending frame DEF with 3 bytes.
```

```
Enter the frame to send:
```

```
GHI
```

```
Sending frame GHI with 3 bytes.
```

```
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p1a>
```

## 2. Design a program to implement bit stuffing, encoding and decoding concept in data link layer.

### Sender.java

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

```
import java.util.*;
```

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

```
public class Sender{
```

```
    public static void main(String[] args){
```

```
        try{
```

```
            Socket s = new Socket("localhost",6589);
```

```
            DataOutputStream dataout = new
```

```
DataOutputStream(s.getOutputStream());
```

```
            Scanner sc = new Scanner(System.in);
```

```
            System.out.println("Enter data: ");
```

```
            String data = sc.nextLine();
```

```
            int count = 0;
```

```
            String str = "";
```

```
            for(int i=0;i<data.length();i++){
```

```
                char ch = data.charAt(i);
```

```
                if(ch == '1'){
```

```
                    count++;
```

```
                    if(count < 5)
```

```
                        str += ch;
```

```
                }else{
```

```

        str = str+ch+'0';
        count = 0;
    }
}
else
{
    str += ch;
    count = 0;
}
}
System.out.println("Flag is 10110101");
str = "10110101"+str+"10110101";

System.out.println("Data stuffed in client: "+str);
dataout.writeUTF(str);
dataout.flush();
dataout.close();
s.close();
sc.close();
}

catch(Exception e){
    System.out.println("Error Occured: " +e);
}

}
}

```

## **Receiver.java**

```

import java.io.*;
import java.util.*;
import java.net.*;

public class Receiver{
    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);
        try{
            ServerSocket ss = new ServerSocket(6589);
            Socket s = ss.accept();
            DataInputStream datain = new DataInputStream(s.getInputStream());
            String data = datain.readUTF();
            int count = 0;
            System.out.println("Data stuffed from sender is: "+data);
            System.out.println("Unstuffed Data is: ");
            for(int i=8;i<data.length()-8;i++)
            {
                char ch = data.charAt(i);

```

```

        if(ch == '1'){
            System.out.print(ch);
            count ++;
            if(count == 5){
                i++;
                count = 0;
            }
        }
        else
        {
            System.out.print(ch);
            count = 0;
        }
    }
    System.out.println();
    datain.close();
    ss.close();
    sc.close();
}
catch(Exception e){
    System.out.println("Error Occured: "+ e);
}
}
}

```

<pre> E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\P1B&gt;javac Receiver.java  E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\P1B&gt;java Receiver Data stuffed from sender is: 10110101111100011111000010110101 Unstuffed Data is: 11100011111000  E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\P1B&gt; </pre>	<pre> E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\P1B&gt;javac Sender.java  E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\P1B&gt;java Sender Enter data: 11100011111000 Flag is 10110101 Data stuffed in client: 10110101111100011111000010110101  E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\P1B&gt; </pre>
--	---

## 2. Design and implement CRC error detection method used in data link layer

```

import java.io.*;
import java.util.*;
class CRC
{

```

```

public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter Generator");
    String gen = sc.nextLine();
    System.out.println("Enter Data: ");
    String data = sc.nextLine();
    String code = data;
    while(code.length() < (data.length()+gen.length() - 1))
        code = code + '0';
    code = data + div(code, gen);
    System.out.println("Transmitted Code Word is: "+code);
    System.out.println("Please Enter Received code word: ");
    String rec = sc.nextLine();
    if(Integer.parseInt(div(rec,gen)) == 0)
        System.out.println("The received code word contains no
errors");
    else
        System.out.println("The recieved code word contains errors");
}

```

```

static String div(String num1, String num2)
{
    int n = num2.length();
    String result = num1.substring(0,n);
    String rem = "";
    for(int i=0;i<num2.length();i++){
        if(result.charAt(i) == num2.charAt(i))
            rem += '0';
        else
            rem += '1';
    }

    while(n < num1.length())
    {
        if(rem.charAt(0) == '0')
        {
            rem = rem.substring(1,rem.length());
            rem = rem + String.valueOf(num1.charAt(n));
            n++;
        }
        result = rem;
        rem = "";
        for(int i=0;i<num2.length();i++){
            if(result.charAt(i) == num2.charAt(i))
                rem += '0';
            else

```

```

        rem += '1';
    }

    }
    return rem.substring(1,rem.length());
}
}

```

```

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p2>java CRC
Enter Generator
1010
Enter Data:
1010000
Transmitted Code Word is: 1010000010
Please Enter Received code word:
1010000011
The recieved coode word contains errors

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p2>java CRC
Enter Generator
1010
Enter Data:
1010000
Transmitted Code Word is: 1010000010
Please Enter Received code word:
1010000010
The received code word contains no errors

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p2>_

```

**3a. Design a program to implement client server model (TCP) using socket programming.**

#### TCPServer.java

```

import java.io.*;
import java.util.*;
import java.nio.file.*;
import java.nio.charset.*;
import java.net.*;

```

```

public class TCPServer

```

```

{
    public static void main(String args[]) throws IOException
    {
        ServerSocket ss = new ServerSocket(9889);
        System.out.println("Server waiting for connections from client..");
        Socket s = ss.accept();
        System.out.println("connection successfull");

        InputStream istream = s.getInputStream();
        OutputStream ostream = s.getOutputStream();

        BufferedReader fileRead = new BufferedReader(new
InputStreamReader(istream));
        String fname = fileRead.readLine();
        BufferedReader contentRead = new BufferedReader(new
FileReader(fname));
        PrintWriter pwrite = new PrintWriter(ostream, true);
        String str;
        while((str = contentRead.readLine()) != null)
            pwrite.println(str);
        s.close();
        ss.close();
    }
}

```

### **TCPClient.java**

```

import java.io.*;
import java.util.*;
import java.nio.file.*;
import java.nio.charset.*;
import java.net.*;

public class TCPClient
{
    public static void main(String[] args) throws IOException
    {
        Socket s = new Socket("localhost",9889);
        System.out.println("Enter the filename: ");
        BufferedReader keyRead = new BufferedReader(new
InputStreamReader(System.in));
        String fname = keyRead.readLine();

        OutputStream ostream = s.getOutputStream();
        PrintWriter pwrite = new PrintWriter(ostream, true);
        pwrite.println(fname);
    }
}

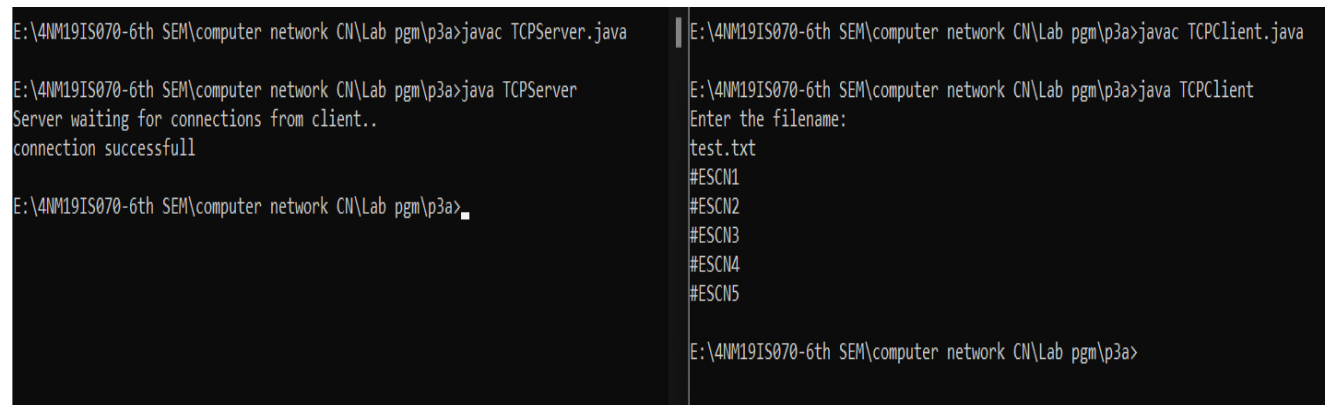
```



```

        InputStream istream = s.getInputStream();
        BufferedReader sRead = new BufferedReader(new
InputStreamReader(istream));
        String str;
        while ((str = sRead.readLine()) != null)
        System.out.println(str);
        s.close();
    }
}

```



```

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p3a>javac TCPServer.java
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p3a>java TCPServer
Server waiting for connections from client..
connection successfull
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p3a>

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p3a>javac TCPClient.java
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p3a>java TCPClient
Enter the filename:
test.txt
#ESC1
#ESC2
#ESC3
#ESC4
#ESC5
E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p3a>

```

### 3b. Design a program to implement client server model (UDP) using socket programming.

#### UdpServer.java

```

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

class UDPServer {
    public static void main(String args[]) throws Exception
    {
        DatagramSocket ss = new DatagramSocket(7999);
        System.out.println("Server Running");
        while(true)

        {
            byte[] receiveData = new byte[1024];
            byte[] sendData = new byte[1024];
            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);

```

```

        String message = "";
        String modifiedmsg = "";
        ss.receive(receivePacket);
        message = new String(receivePacket.getData());
        InetAddress IPAddress = receivePacket.getAddress();
        int port = receivePacket.getPort();
        System.out.println("From Client "+IPAddress+": "+message);
        modifiedmsg = message.toUpperCase();
        sendData = modifiedmsg.getBytes();

        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, IPAddress, port);

        ss.send(sendPacket);
        System.out.println("Data Sent Successfully");
    }
}

```

### **UdpClient.java**

```

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

class UDPClient {
    public static void main(String args[]) throws Exception
    {
        DatagramSocket cs = new DatagramSocket();
        BufferedReader inputstream = new BufferedReader(new
InputStreamReader(System.in));

        InetAddress IPAddress = InetAddress.getByName("localhost");
        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];

        System.out.println("Enter your message here (Type END to stop)");

        while(true)

        {

            String message = inputstream.readLine();

            if(message.equals("END"))
            {
                cs.close();
                System.exit(0);
            }
        }
    }
}

```

```

    }

    else
    {
        sendData = message.getBytes();

        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, IPAddress, 7999);

        cs.send(sendPacket);

        DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);

        cs.receive(receivePacket);

        String modifiedmsg = new String(receivePacket.getData());

        System.out.println("From Server: "+modifiedmsg);

    }
}
}
}
}

```

The screenshot shows two terminal windows side-by-side. The left window is the server, and the right window is the client.

**Left Window (Server):**

```

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p3b>java UDPServer
Server Running
From Client /127.0.0.1: abc def

Data Sent Successfully
From Client /127.0.0.1: ghi jkl

Data Sent Successfully

```

**Right Window (Client):**

```

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p3b>java UDPClient
Enter your message here (Type END to stop)
abc def
From Server: ABC DEF

ghi jkl
From Server: GHI JKL

END

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p3b>

```

#### **4. Design a program for congestion control using leakybucket algorithm.**

```
import java.util.*;

public class LeakyBucket {
    static void flow(int pktsize, int output)
    {
        int buketsize = 512;
        if (pktsize > buketsize)
        {
            System.out.println("Bucket overflow");
        }
        else
        {
            try{
                Thread.sleep(1000);
                while (pktsize > output)
                {
                    System.out.println (output + "bytes
outflow");

                    pktsize = pktsize-output;
                }

                if (pktsize > 0)
                {
                    System.out.println(pktsize + "bytes
outflow");
                }
            }
            catch (InterruptedException e) {
                System.out.println("Interrupted!");
            }
        }
    }

    public static void main(String[] args)
    {
        int output, pktsize, n;
        Scanner input = new Scanner(System.in);
        Random rand = new Random();
        System.out.println("Enter output rate: ");
        output = input.nextInt();
        System.out.println("Enter number of packets: ");
    }
}
```

```

        n = input.nextInt();
        for(int i=1; i<=n;i++)
        {
            pktsize = rand.nextInt(1000);
            System.out.println("Packet No: "+i+" packetsize = "+pktsize);
            flow(pktsize, output);
        }
    }
}

```

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p4>javac LeakyBucket.java

E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\p4>java LeakyBucket

Enter output rate:

250

Enter number of packets:

10

Packet No: 1 packetsize = 629

Bucket overflow

Packet No: 2 packetsize = 467

250bytes outflow

217bytes outflow

Packet No: 3 packetsize = 219

219bytes outflow

Packet No: 4 packetsize = 812

Bucket overflow

Packet No: 5 packetsize = 994

Bucket overflow

Packet No: 6 packetsize = 111

111bytes outflow

Packet No: 7 packetsize = 870

Bucket overflow

Packet No: 8 packetsize = 605

Bucket overflow

Packet No: 9 packetsize = 726

Bucket overflow

Packet No: 10 packetsize = 196

196bytes outflow

## 5.Design and implement a program to route the packet in a network using distance vector algorithm.

```
import java.io.*;
import java.util.*;
public class Dist
{
    public static void main(String args[]) throws IOException
    {
        int n,i,j,k,count=0;
        int rt[][] = new int[20][20];
        int via[][] = new int[20][20];
        int costmat[][] = new int[20][20];
        Scanner sc = new Scanner(System.in);
        System.out.println("\nEnter the number of nodes : ");
        n = sc.nextInt();
        System.out.println("\nEnter the cost matrix :\n");
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            {
                costmat[i][j]=sc.nextInt();
                costmat[i][i]=0;
                rt[i][j]=costmat[i][j];
                via[i][j]=j;
            }
        }
        do
        {
            count=0;
            for(i=0;i<n;i++)
            for(j=0;j<n;j++)
            for(k=0;k<n;k++)
            if(rt[i][j]>costmat[i][k]+rt[k][j])
            {
                rt[i][j]=rt[i][k]+rt[k][j];
                via[i][j]=k;
                count++;
            }
        }while(count!=0);
        for(i=0;i<n;i++)
        {
            System.out.println("\nFor router: " + (i+1));
            for(j=0;j<n;j++)
            {
                System.out.println("\t\tnode: " + (j+1) + " via: " + (via[i][j]+1) + " Distance: " + (rt[i][j]));
            }
        }
    }
}
```

```
PS E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\Part-A> java Dist
```

```
Enter the number of nodes :
```

```
5
```

```
Enter the cost matrix :
```

```
0   3   999   999   5
3   0   4     2   999
999 4   0     7   999
999 2   7     0   8
5   999 999   8   0
```

```
For router: 1
```

```
node: 1 via: 1 Distance: 0
```

```
node: 2 via: 2 Distance: 3
```

```
node: 3 via: 2 Distance: 7
```

```
node: 4 via: 2 Distance: 5
```

```
node: 5 via: 5 Distance: 5
```

```
For router: 2
```

```
node: 1 via: 1 Distance: 3
```

```
node: 2 via: 2 Distance: 0
```

```
node: 3 via: 3 Distance: 4
```

```
node: 4 via: 4 Distance: 2
```

```
node: 5 via: 1 Distance: 8
```

For router: 3

node: 1 via: 2 Distance: 7

node: 2 via: 2 Distance: 4

node: 3 via: 3 Distance: 0

node: 4 via: 2 Distance: 6

node: 5 via: 2 Distance: 12

For router: 4

node: 1 via: 2 Distance: 5

node: 2 via: 2 Distance: 2

node: 3 via: 2 Distance: 6

node: 4 via: 4 Distance: 0

node: 5 via: 5 Distance: 8

For router: 5

node: 1 via: 1 Distance: 5

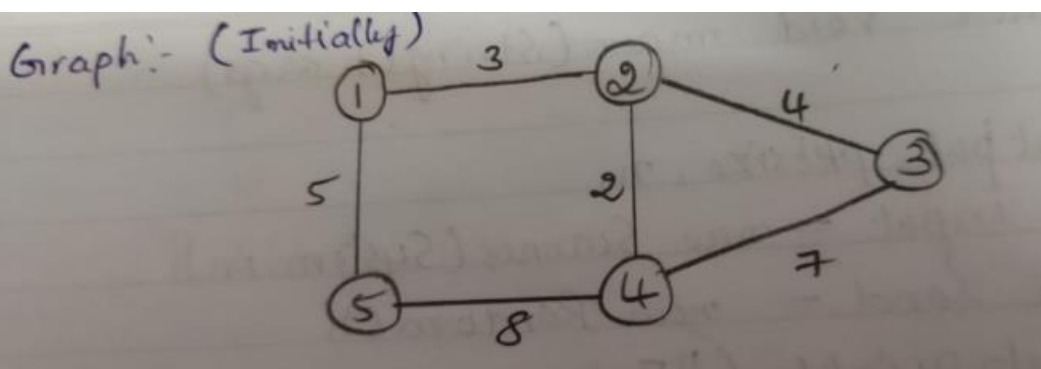
node: 2 via: 1 Distance: 8

node: 3 via: 1 Distance: 12

node: 4 via: 4 Distance: 8

node: 5 via: 5 Distance: 0

PS E:\4NM19IS070-6th SEM\computer network CN\Lab pgm\Part-A> |





## PART-B

1) Simulate peer-to-peer communication between a client and a server using Point-to-Point protocol. Apply NetAnim software to demonstrate the scenario graphically. Analyze packet parameters by creating trace file using Ascii trace metrics

```
// Default Network Topology
//
// 10.1.1.0
// n0 ----- n1
//   point-to-point
//
```

1 — first - C C  
2 — L N → C s m → second - C C  
3 — C s m a → third - C C  
5 — ping → fifth - C C

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/netanim-module.h"

using namespace ns3;

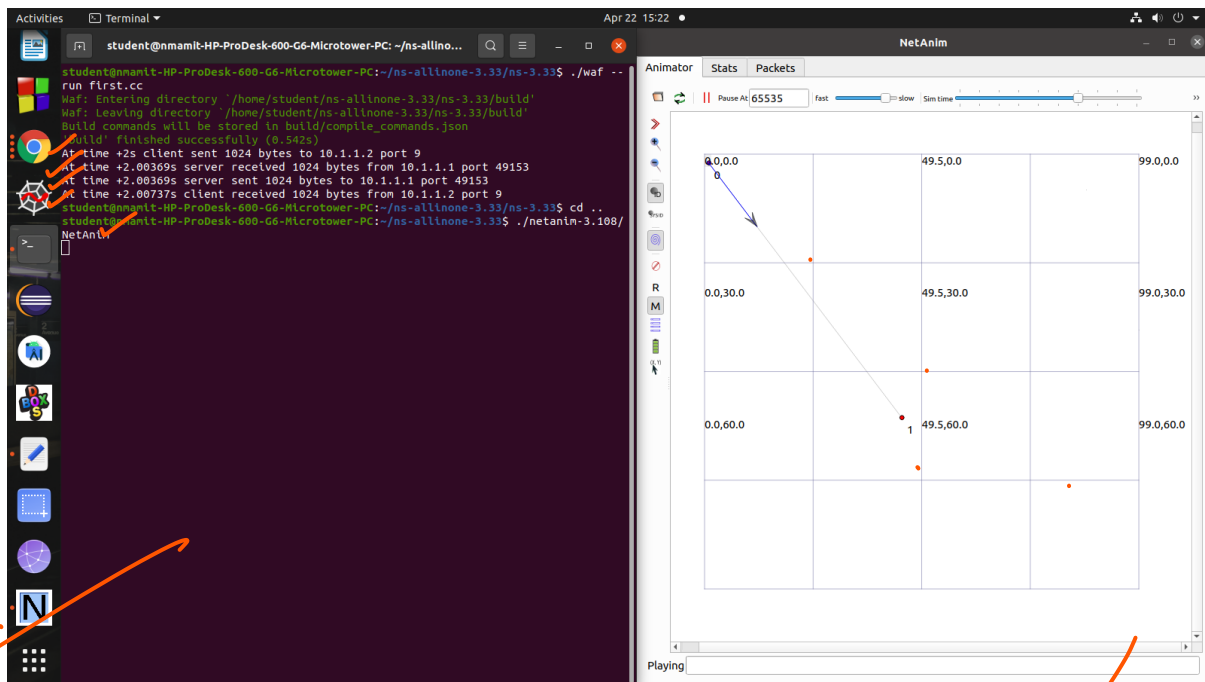
NS_LOG_COMPONENT_DEFINE("FirstScriptExample");

int main(int argc, char *argv[])
{
    CommandLine cmd(__FILE__);
    cmd.Parse(argc, argv);
    Time::SetResolution(Time::NS);
    LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
    LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
    std::string animFile = "first.xml";
    NodeContainer nodes;
    nodes.Create(2);
    PointToPointHelper pointToPoint;
    pointToPoint.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
    pointToPoint.SetChannelAttribute("Delay", StringValue("2ms"));
    NetDeviceContainer devices;
    devices = pointToPoint.Install(nodes);
    InternetStackHelper stack;
    stack.Install(nodes);
    Ipv4AddressHelper address;
    address.SetBase("10.1.1.0", "255.255.255.0");
    Ipv4InterfaceContainer interfaces = address.Assign(devices);
    UdpEchoServerHelper echoServer(9);
    ApplicationContainer serverApps = echoServer.Install(nodes.Get(1));
    serverApps.Start(Seconds(1.0));
    serverApps.Stop(Seconds(10.0));
}
```

```

UdpEchoClientHelper echoClient(interfaces.GetAddress(1), 9);
echoClient.SetAttribute("MaxPackets", UIntegerValue(10));
echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0)));
echoClient.SetAttribute("PacketSize", UIntegerValue(1024));
ApplicationContainer clientApps = echoClient.Install(nodes.Get(0));
clientApps.Start(Seconds(2.0));
clientApps.Stop(Seconds(10.0));
AnimationInterface anim(animFile);
anim.SetConstantPosition(nodes.Get(0), 1.0, 2.0);
anim.SetConstantPosition(nodes.Get(1), 45.0, 60.0);
AsciiTraceHelper ascii;
pointToPoint.EnableAsciiAll(ascii.CreateFileStream("first.tr"));
Simulator::Run();
Simulator::Destroy();
return 0;
}

```



— P2P + CSMA

2) Simulate to implement a bus topology using Point-to-Point protocol between a client and a LAN with 4 nodes. The LAN use CSMA during packet transmission. Apply NetAnim software to demonstrate the scenario graphically. Analyze packet parameters by creating trace file using Ascii trace metrics.

```

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/csma-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"

```

```
#include "ns3/netanim-module.h"
```

```
// Default Network Topology
//
//      10.1.1.0
// n0  ----- n1  n2  n3  n4
//      point-to-point | | | |
//                      =====
//                      LAN 10.1.2.0
```

```
using namespace ns3;
```

```
int main(){
```

```
    LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
```

```
    LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
```

```
    std::string animFile = "second.xml";
```

```
    NodeContainer p2pNodes;
```

```
    p2pNodes.Create(2);
```

```
    NodeContainer csmaNodes;
```

```
    csmaNodes.Add(p2pNodes.Get(1));
```

```
    csmaNodes.Create(3);
```

```
    PointToPointHelper pointToPoint;
```

```
    pointToPoint.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
```

```
    pointToPoint.SetChannelAttribute("Delay", StringValue("2ms"));
```

```
    NetDeviceContainer p2pDevices;
```

```
    p2pDevices = pointToPoint.Install(p2pNodes);
```

```
    CsmHelper csma;
```

```
    csma.SetChannelAttribute("DataRate", StringValue("100Mbps"));
```

```
    csma.SetChannelAttribute("Delay", StringValue("2ms"));
```

```
    NetDeviceContainer csmaDevices;
```

```
    csmaDevices = csma.Install(csmaNodes);
```

```
    InternetStackHelper stack;
```

```
    stack.Install(p2pNodes.Get(0));
```

```
    stack.Install(csmaNodes);
```

```
    Ipv4AddressHelper address;
```

```
    address.SetBase("10.1.1.0", "255.255.255.0");
```

```
    Ipv4InterfaceContainer p2pInterfaces = address.Assign(p2pDevices);
```

```
    address.SetBase("10.1.2.0", "255.255.255.0");
```

```
    Ipv4InterfaceContainer csmaInterfaces = address.Assign(csmaDevices);
```

```
UdpEchoServerHelper echoServer(9);

ApplicationContainer serverApps = echoServer.Install(csmaNodes.Get(3));
serverApps.Start(Seconds(1.0));
serverApps.Stop(Seconds(10.0));

UdpEchoClientHelper echoClient(csmaInterfaces.GetAddress(3),9);
echoClient.SetAttribute("MaxPackets",UIntegerValue(1));
echoClient.SetAttribute("Interval",TimeValue(Seconds(1.0)));
echoClient.SetAttribute("PacketSize",UIntegerValue(1024));

ApplicationContainer clientApps = echoClient.Install(p2pNodes.Get(0));
clientApps.Start(Seconds(2.0));
clientApps.Stop(Seconds(10.0));

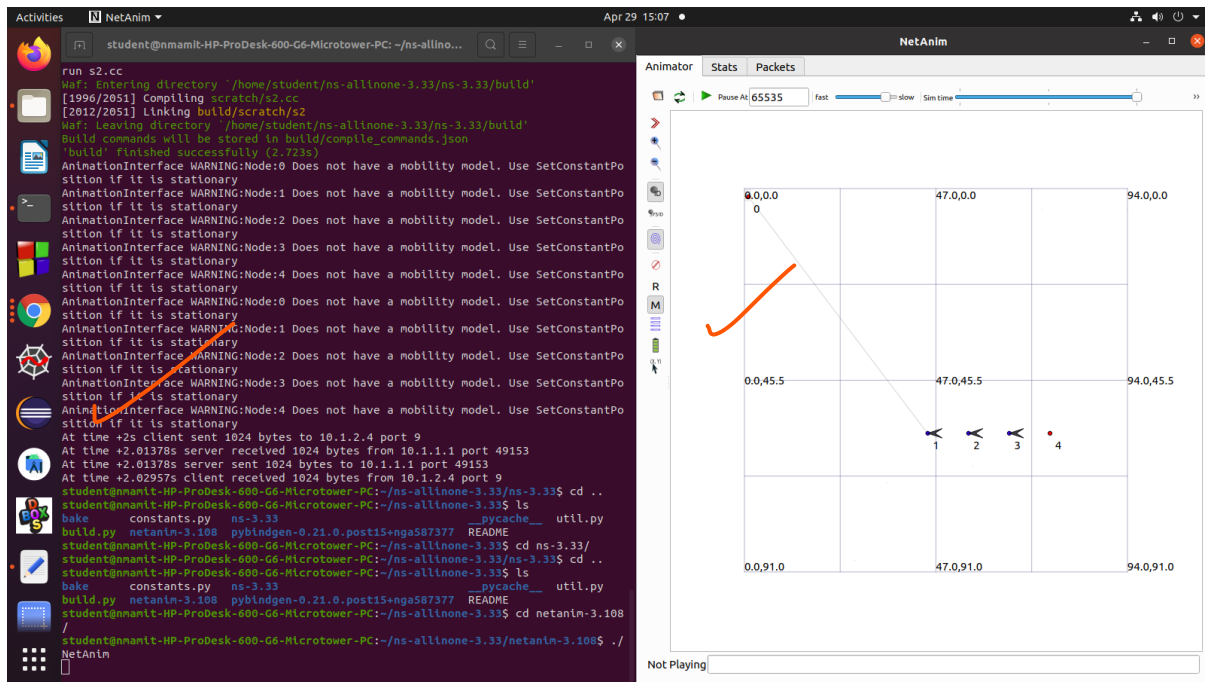
Ipv4GlobalRoutingHelper::PopulateRoutingTables();
AnimationInterface anim(animFile);

anim.SetConstantPosition(p2pNodes.Get(0),1.0,2.0);
anim.SetConstantPosition(csmaNodes.Get(0),45.0,60.0);
anim.SetConstantPosition(csmaNodes.Get(1),55.0,60.0);
anim.SetConstantPosition(csmaNodes.Get(2),65.0,60.0);
anim.SetConstantPosition(csmaNodes.Get(3),75.0,60.0);

AsciiTraceHelper ascii;
pointToPoint.EnableAsciiAll(ascii.CreateFileStream("second.tr"));
csma.EnableAsciiAll(ascii.CreateFileStream("second1.tr"));

Simulator::Run();
Simulator::Destroy();
return 0;

}
```



NO P2P  
↓

3) Simulate peer-to-peer communication between a client and a server using CSMA protocol. Apply NetAnim software to demonstrate the scenario graphically. Analyze packet parameters by creating trace file using Ascii trace metrics.

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/csma-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/netanim-module.h"

using namespace ns3;

int main(){
    LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
    LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
    std::string animFile = "third.xml";

    NodeContainer csmaNodes;
    csmaNodes.Create(4);
```

```
// Default Network Topology
//
//      10.1.1.0
// n0 ----- n1
//      point-to-point
//
```

```
CsmaHelper csma;
csma.SetChannelAttribute("DataRate", StringValue("5Mbps"));
csma.SetChannelAttribute("Delay", StringValue("2ms"));

NetDeviceContainer csmaDevices;
csmaDevices = csma.Install(csmaNodes);

InternetStackHelper stack;
stack.Install(csmaNodes);

Ipv4AddressHelper address;
address.SetBase("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfaces = address.Assign(csmaDevices);

UdpEchoServerHelper echoServer(9);

ApplicationContainer serverApps = echoServer.Install(csmaNodes.Get(3));
serverApps.Start(Seconds(1.0));
serverApps.Stop(Seconds(10.0));

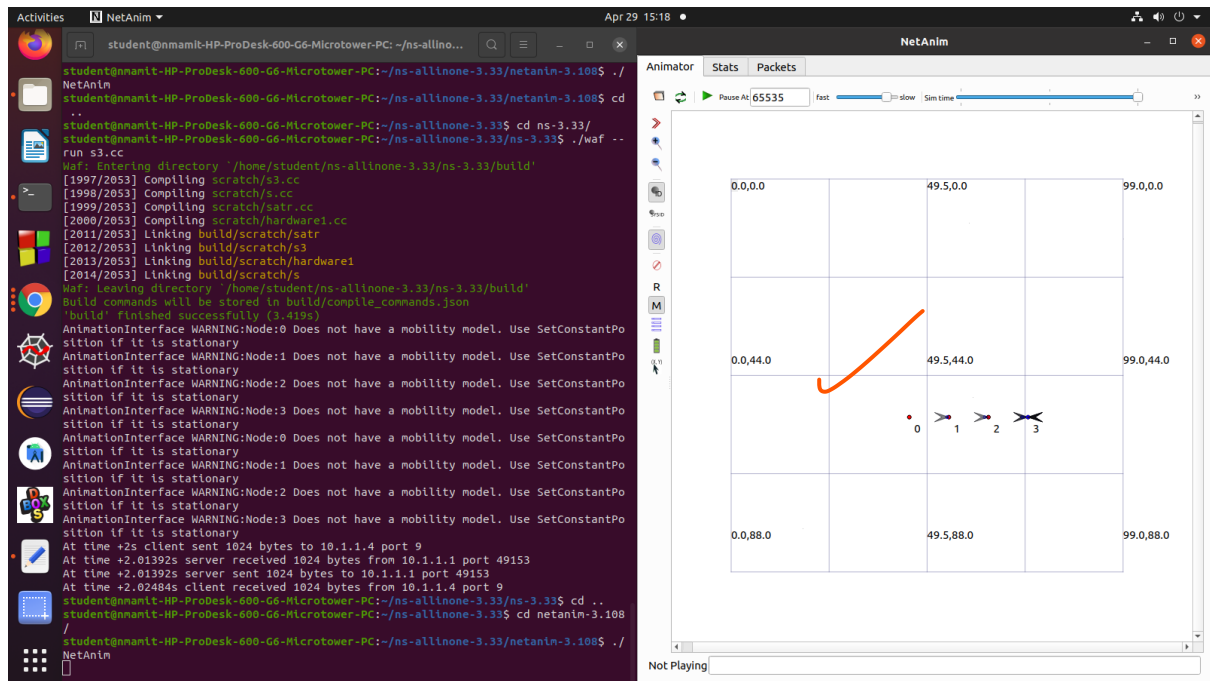
UdpEchoClientHelper echoClient(csmaInterfaces.GetAddress(3),9);
echoClient.SetAttribute("MaxPackets",UIntegerValue(1));
echoClient.SetAttribute("Interval",TimeValue(Seconds(1.0)));
echoClient.SetAttribute("PacketSize",UIntegerValue(1024));

ApplicationContainer clientApps = echoClient.Install(csmaNodes.Get(0));
clientApps.Start(Seconds(2.0));
clientApps.Stop(Seconds(10.0));

Ipv4GlobalRoutingHelper::PopulateRoutingTables();
AnimationInterface anim(animFile);
anim.SetConstantPosition(csmaNodes.Get(0),45.0,60.0);
anim.SetConstantPosition(csmaNodes.Get(1),55.0,60.0);
anim.SetConstantPosition(csmaNodes.Get(2),65.0,60.0);
anim.SetConstantPosition(csmaNodes.Get(3),75.0,60.0);

AsciiTraceHelper ascii;
csma.EnableAsciiAll(ascii.CreateFileStream("third.tr"));

Simulator::Run();
Simulator::Destroy();
return 0;
}
```



4) Simulate to implement the star topology using Point-to-Point protocol. Apply NetAnim software to demonstrate the scenario graphically. Analyze packet parameters by creating trace file using Ascii trace metrics

```

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/netanim-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/point-to-point-layout-module.h"
#include "ns3/netanim-module.h"

// Network topology (default)
//
//      n2 n3 n4
//      \ | /
//      n1---n0---n5
//      / | \
//      n8 n7 n6

```

```

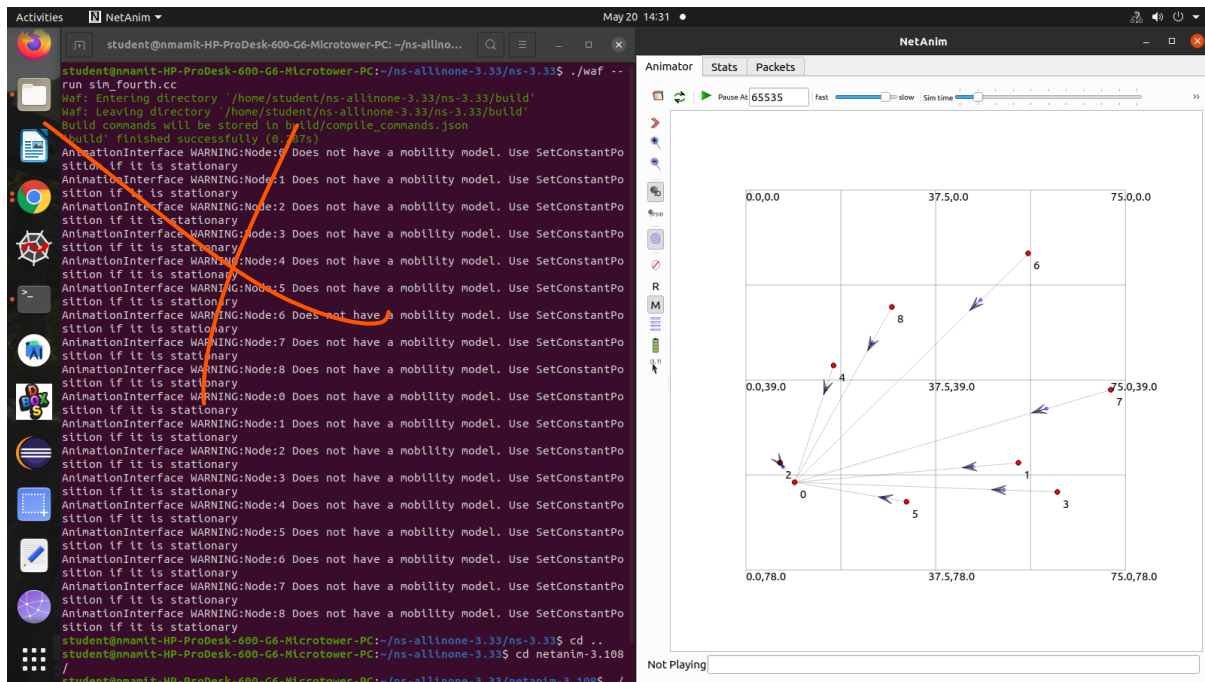
using namespace ns3;
int main ()
{
    std::string animFile="fourth.xml";

    PointToPointHelper pointToPoint;
    pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
    pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
    PointToPointStarHelper star (8, pointToPoint); //8 nodes
    InternetStackHelper internet;
    star.InstallStack (internet);
    star.AssignIpv4Addresses (Ipv4AddressHelper ("10.1.1.0", "255.255.255.0"));
    Address hubLocalAddress (InetSocketAddress (Ipv4Address::GetAny (), 50000));
    PacketSinkHelper packetSinkHelper("ns3::TcpSocketFactory", hubLocalAddress);
    ApplicationContainer hubApp = packetSinkHelper.Install (star.GetHub ());
    hubApp.Start (Seconds (1.0));
    hubApp.Stop (Seconds (10.0));
    //
    // Create OnOff applications to send TCP to the hub, one on each spoke node.
    //
    OnOffHelper onOffhelper ("ns3::TcpSocketFactory", Address ());
    onOffHelper.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable
[Constant=1]"));
    onOffHelper.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable
[Constant=0]"));
    ApplicationContainer spokeApps;
    for (uint32_t i = 0; i < star.SpokeCount (); ++i)
    {
        AddressValue remoteAddress (InetSocketAddress (star.GetHubIpv4Address (i),
50000)); // 50000 is the port number
        onOffHelper.SetAttribute ("Remote", remoteAddress);
        spokeApps.Add (onOffHelper.Install (star.GetSpokeNode (i)));
    }
    spokeApps.Start (Seconds (1.0));
    spokeApps.Stop (Seconds (10.0));
    Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
    pointToPoint.EnablePcapAll ("star");
    AnimationInterface anim(animFile);
    anim.SetConstantPosition(star.GetHub(),10.0,60.0);

    Simulator::Run ();
    Simulator::Destroy ();
    return 0;
}

```





5) Simulate the transmission of ping messages over a network topology consisting of 3 nodes n0, n1 and n2, where node n0 and n1 are the pingers. Analyze the working of ping using wireshark.

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/csma-module.h"
#include "ns3/applications-module.h"
#include "ns3/internet-apps-module.h"
#include "ns3/internet-module.h"
#include "ns3/netanim-module.h"
using namespace ns3;
int
main ()
{
    std::string animFile="fifth67.xml";
    NodeContainer nodes;
    nodes.Create (3);
    CsmaHelper csma;
    csma.SetChannelAttribute ("DataRate", DataRateValue (DataRate (5000000)));
    csma.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (2)));
    NetDeviceContainer devs = csma.Install (nodes);
    InternetStackHelper ipStack;
    ipStack.Install (nodes);
    Ipv4AddressHelper ip;
    ip.SetBase ("10.0.1.0", "255.255.255.0");
    Ipv4InterfaceContainer interface = ip.Assign (devs);

    V4PingHelper ping = V4PingHelper (interface.GetAddress (2));
    NodeContainer pingers;
```

```

pingers.Add (nodes.Get (0));
pingers.Add (nodes.Get (1));
ApplicationContainer apps = ping.Install (pingers);
apps.Start (Seconds (2.0));
apps.Stop (Seconds (5.0));
csma.EnablePcapAll ("csma-ping", true);

AnimationInterface anim(animFile);
anim.SetConstantPosition(nodes.Get(0),10.0,60.0);
anim.SetConstantPosition(nodes.Get(1),10.0,100.0);
anim.SetConstantPosition(nodes.Get(2),50.0,60.0);

Simulator::Run ();
Simulator::Destroy ();
}

```

