Set 1

Write a program to simulate Carrier Sense Multiple Access (CSMA/CD).

Server.java

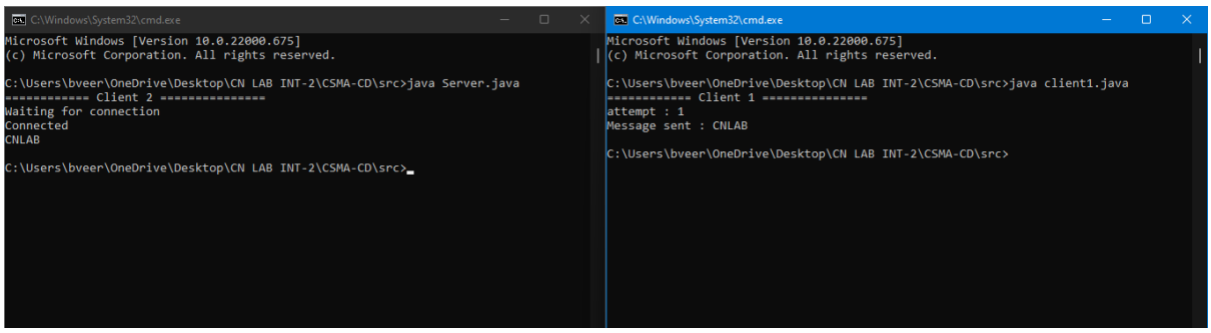
```
import java.io.*;
import java.net.*;
public class Server
{
	public static void main(String[] args)
	{
		try
		{
		System.out.println("============ Client 2 ===============");
		ServerSocket ss = new ServerSocket(3000);
		System.out.println("Waiting for connection");

			Socket con = ss.accept();
			System.out.println("Connected");
	ObjectInputStream in = new ObjectInputStream(con.getInputStream());
			System.out.println((String)in.readObject());

			in.close();
			ss.close();
		}
			catch(Exception e)
			{
				System.out.println(e);
			}
	}
}
```

Client1.java

```
import java.io.*;
import java.net.*;
public class client1
{
	public static void main(String[] args)
	{
		try
		{
		System.out.println("============ Client 1 ===============");
			client1 cli = new client1();
			int Tp = 2000;
			int R = 0;
			int Tb = 0;

			for(int i=1; i<=15;i++)
			{
				System.out.println("attempt : "+i);
				if(cli.send() == "sent")
				{
					break;
				}
				else
				{
					R = 2^i-1;
				System.out.println("Selected Random number :"+R);
```

```
                                    Tb = R*Tp;
System.out.println("waiting for next attempt in seconds): "+Tb);
                                    Thread.sleep(Tb);
                              }
                        }
                }
                catch (InterruptedException e)
                {
                        System.out.println(e);
                }
        }
        String send()
        {
                String str=null;
                try
                {
                        Socket soc = new Socket("localhost",3000);
ObjectOutputStream out = new ObjectOutputStream(soc.getOutputStream());
                        String msg = "CNLAB";
                        out.writeObject(msg);
                        System.out.println("Message sent : "+msg);
                        str = "sent";
                        soc.close();
                }
                catch(Exception e)
                {
                        str = "collision occured";
                        String msg = null;
                        System.out.println("Message sent : "+msg);
                }
                return str;
        }
}
```

OUTPUT



Set 2

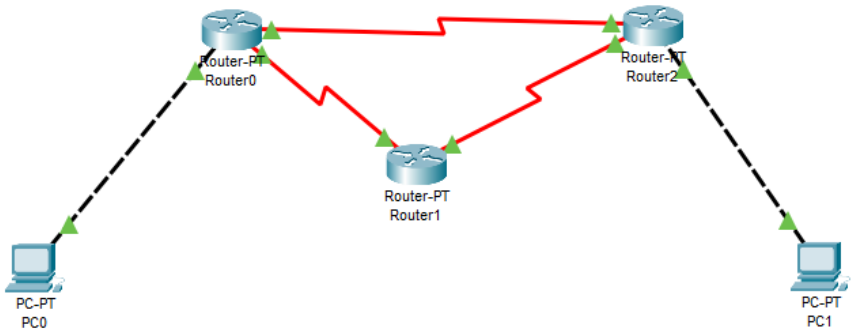
Configure a network using Distance Vector Routing protocol using Packet tracer tool.

Implementation:

Follow the below steps to configure a Network using Distance Vector Routing protocol.

1) Open cisco packet tracer and make a overview of the connection using Pc and Router.

2) consider the below model for DVR protocol using cisco packet.



3) Place all the PT-routers and Pcs as shown in the model without Establishing any connection among Each others.

4) Firstly, take Pc_0 connect it with PT-$router_0$ using FastEthernet 0 in both the cases. Once, its done Provide IP address to the Pc i.e; 192.168.1.2 and change the Default Gateway to 192.168.1.1. The DG Provided in Pc should be the IP address to the PT-$router_0$.

5) Now consider PT-$router_0$ and PT-$router_1$, connect them using Serial 2/0 in $router_1$ and Serial 3/0 is $router_2$. We need to create a separate class path to all the routers which are interconnected, class ip address can be 10.0.0.1, 20.0.0.1, 30.0.0.1

6) In $router_0$ and $router_1$ go into the configuration of Serial 2/0 or 3/0 into their respective routers and give class ip address as listed above. Repeat the same steps to the $router_2$ and Pc_2 from the above steps Provided.

7) Once, the model is successful created and connected with Each other. Now its the task to Provide routers with the ip addresses of the connected systems, switches, hubs, Etc.

8) Go to RIP configuration into the respective router and place all the others ip address of systems which are connected to it such as Pc_0 - 192.168.1.1, routers - 10.0.0.1, 20.0.0.1, 30.0.0.1.

9) Now transmit messages from Pc_0 to Pc_1.

10) In initial state messages get failed, try another time to activate the connections Established and finally message gets transmitted to the other End / destination.

Set 3

write a program to implement client server communication for chat using TCP.

Server.java

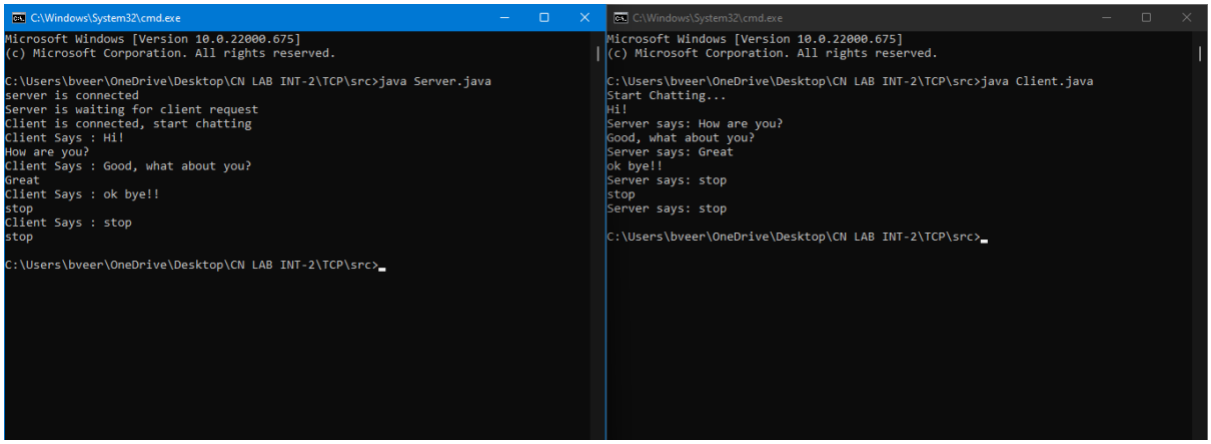
```
import java.net.*;
import java.io.*;
public class Server {
	public static void main(String[] args) throws Exception {
		System.out.println("server is connected");
		ServerSocket ss=new ServerSocket(3333);
		System.out.println("Server is waiting for client request");
		Socket s=ss.accept();
		System.out.println("Client is connected, start chatting");

		DataInputStream din=new DataInputStream(s.getInputStream());
	DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
		String str="",str2="";
		while(!str.equals("stop"))
		{
			str=din.readUTF();
			System.out.println("Client Says : "+str);
			str2=br.readLine();
			dout.writeUTF(str2);
			dout.flush();
		}
		din.close();
		ss.close();
	}
}
```

Client.java

```
import java.net.*;
import java.io.*;
public class Client {
	public static void main(String[] args) throws Exception {
		try {
			Socket s=new Socket("localhost",3333);
			System.out.println("Start Chatting...");
		DataInputStream din=new DataInputStream(s.getInputStream());
	DataOutputStream dout=new DataOutputStream(s.getOutputStream());
	BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

			String str="",str2="";
			while(!str.equals("stop"))
			{
				str=br.readLine();
				dout.writeUTF(str);
				dout.flush();
				str2=din.readUTF();
				System.out.println("Server says: "+str2);
			}
			dout.close();
			s.close();
		}
```

```
                catch(ConnectException e) {
                        System.out.println("Server is offline");
                }
        }
}
```

OUTPUT



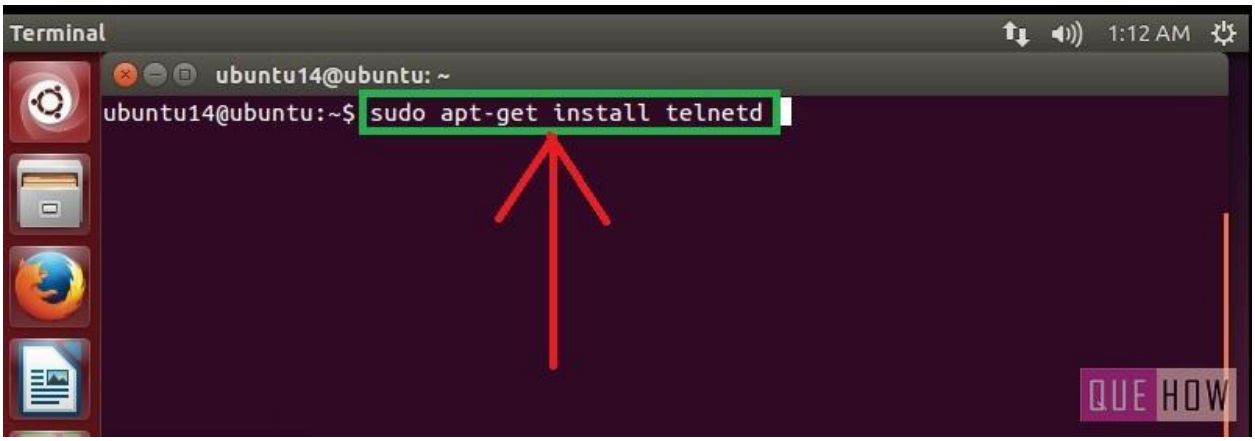
Set 4

Install Telnet on one of the systems connected by a switch and telnet it from other system.

Installation of Telnet:

Step 1: sudo apt-get install telnetd

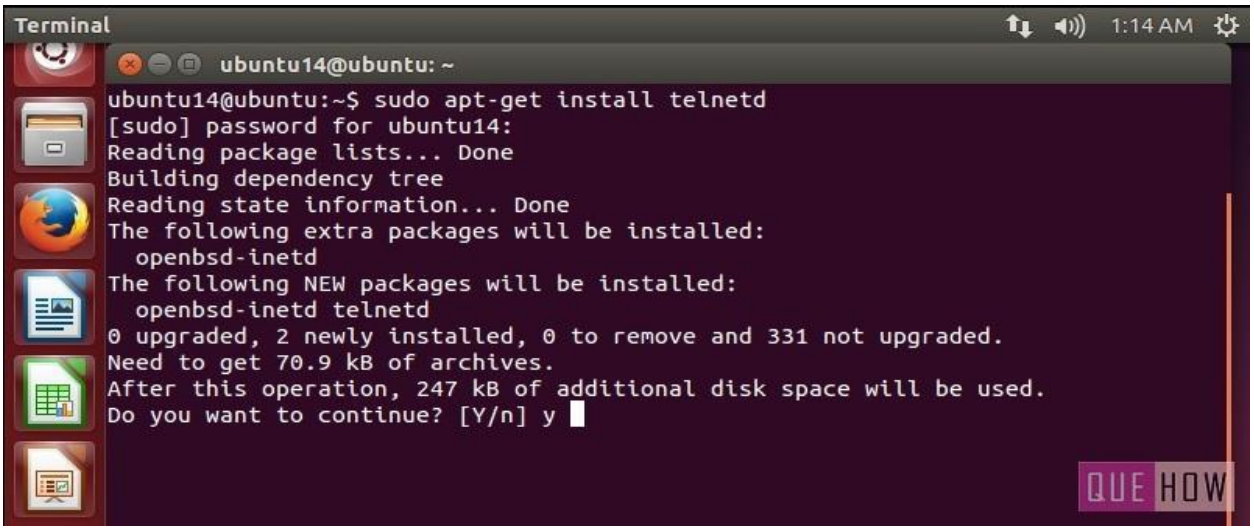
This will complete the installation of Telnet. **"telnetd"** is a daemon that gets invoked by *"inetd" or its extension "xinetd"*, both are the internet servers.



Step 2:

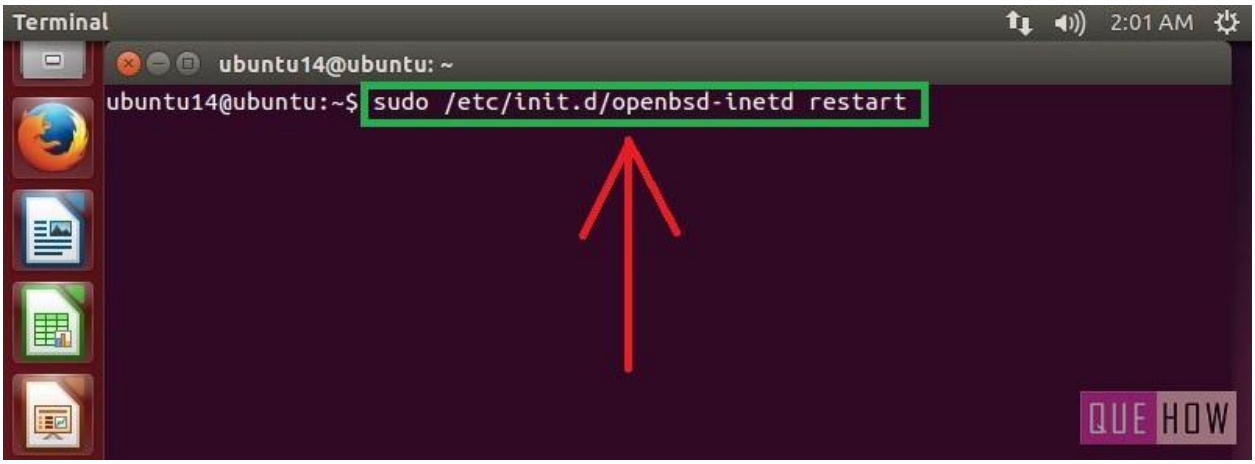
Then you are asked to enter the user password and then press enter. Processing will start as soon as you press enter. After this, I have noticed a line **"274 KB additional disk space will be used"** on the terminal screen.

You may also observe some sort of a message like this and then you"ll be asked to continue or not. Just write **"y"** and then press enter to continue.

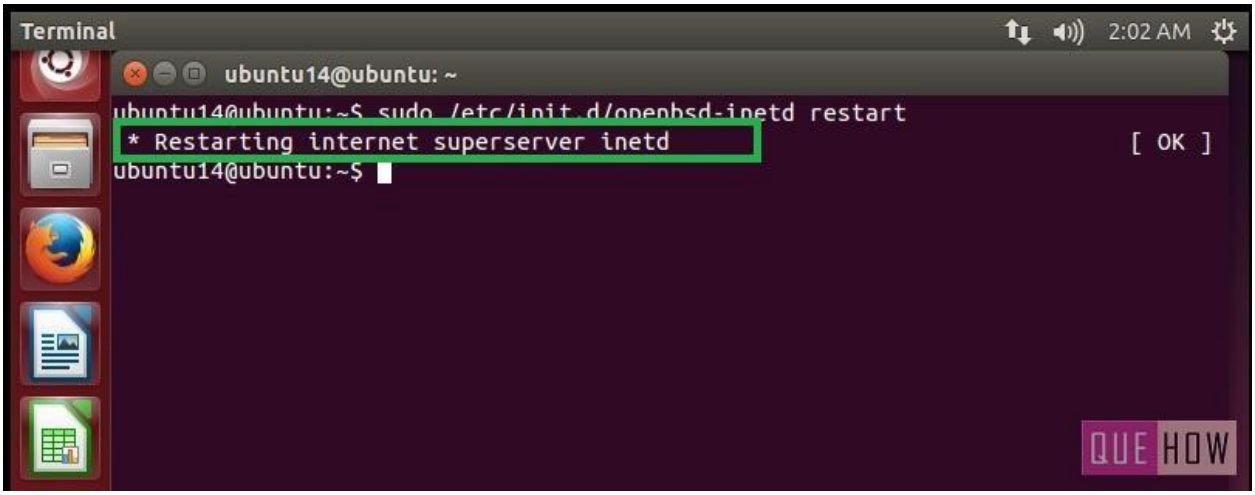


Step 3: Now when you are done with it, **restart "inetd".** Type **"sudo /etc/init.d.open-bsd-inetd restart"**.

"inetd" is daemon used for *dealing with incoming network* and it is responsible for deciding which program to run when a request comes.



Step 4: To ensure "inetd" is started, press enter after writing the above command.



To connect with any remote client:

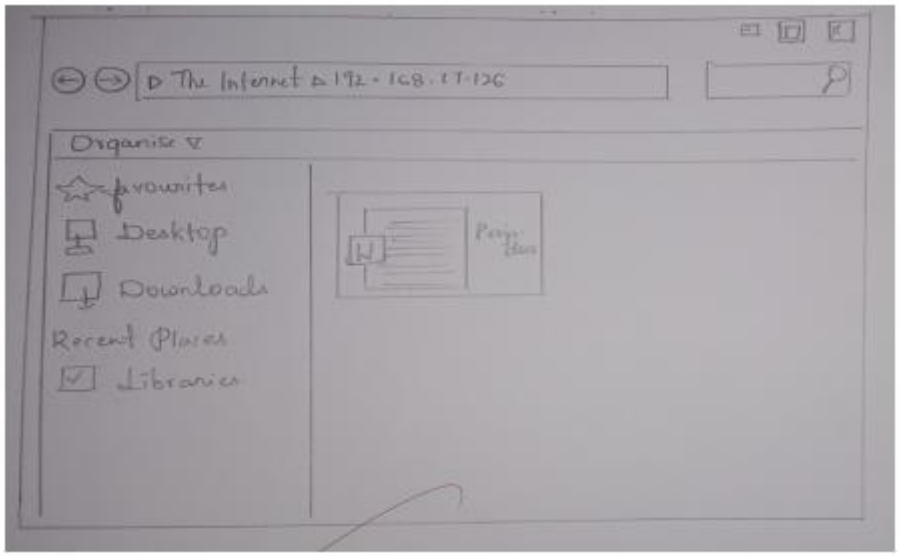
Step 5: Just type:**"telnet hostipaddress"**. For an example: "telnet 192.168.0.68" and press enter.

Step 6: Then you"ll see, it is connected to **"host ip address"**. For security reasons, you are required to provide "username" and "password" as well.

Set 5

Configure FTP Server on a Linux or Windows machine using FTP Client.

1) Type in cmd ip config to find the ip address of the system
2) Create a folder in desktop and store files that you would want to share
3) To configure the FTP server go to the control panel, click on programs -> programs and features.
4) Click on turn windows features on /off
5) Check mark IIS (internet and information services)
6) Expand IIS and check mark ftp server
7) Expand ftp server check mark ftp extensibility and ftp services
8) Make sure web management tools and World Wide Web services are checked.
9) Click ok .10) Click on control panel home
11) Click on administrative tools (if windows 10 or 11, search for administrative tools).
12) Click on IIS MANAGER
13) Under connections look for the root (folder name) (available on top right)
14) Expand root 15) Expand sites
16) Right click on sites and select add ftp site
17) Name the ftp site
18) Browse the physical path for the folder created in step 2.
19) Click on next
20) Select the ip address of your computer. And select no ssl under ssl and click on next.
21) Select basic authentication
22) Select authorization as specific users and type in Gcet (username)
23) Check in required permissions.
24) Click on finish 25) FTP site is configured.
26) Go to MyComputer, type in ftp://ip address and click enter2
7) Provide user credentials and access the files available in the ftp server. This access is internally with in a network
28) To access this ftp outside a network we need to enable port no 21 at the router



Set 6

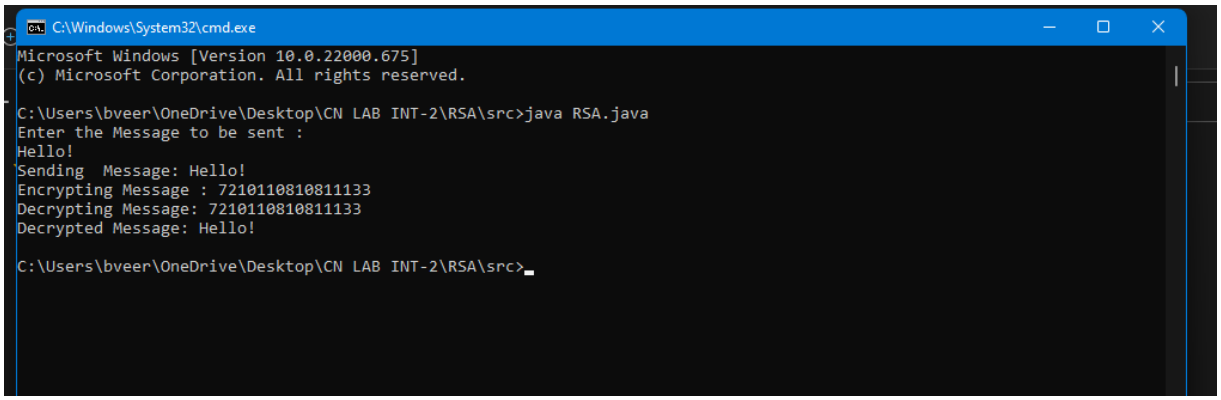
Using RSA Algorithm Encrypt a Text data and Decrypt the same.

RSA.java

```
import java.io.DataInputStream;
import java.io.IOException;
import java.math.BigInteger;
import java.util.Random;

public class RSA
{
 private BigInteger p;
 private BigInteger q;
 private BigInteger N;
 private BigInteger phi;
 private BigInteger e;
 private BigInteger d;
 private int bitlength = 1024;
 private Random r;

 public RSA()
 {
 r = new Random();
 p = BigInteger.probablePrime(bitlength, r);
 q = BigInteger.probablePrime(bitlength, r);
 N = p.multiply(q);
 phi = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
 e = BigInteger.probablePrime(bitlength / 2, r);

while (phi.gcd(e).compareTo(BigInteger.ONE) > 0 && e.compareTo(phi) < 0)
 {
 e.add(BigInteger.ONE);
 }
 d = e.modInverse(phi);
 }
 public RSA(BigInteger e, BigInteger d, BigInteger N)
 {
 this.e = e;
 this.d = d;
 this.N = N;
 }
 @SuppressWarnings("deprecation")
 public static void main(String[] args) throws IOException
 {
 RSA rsa = new RSA();
 DataInputStream in = new DataInputStream(System.in);
 String teststring;
System.out.println("Enter the Message to be sent :");
 teststring = in.readLine();
 System.out.println("Sending  Message: " + teststring);
 System.out.println("Encrypting Message : "+
bytesToString(teststring.getBytes()));
// encrypt
 byte[] encrypted = rsa.encrypt(teststring.getBytes());
 // decrypt
 byte[] decrypted = rsa.decrypt(encrypted);
 System.out.println("Decrypting Message: " + bytesToString(decrypted));
 System.out.println("Decrypted Message: " + new String(decrypted));
```

```
 }
 private static String bytesToString(byte[] encrypted)
 {
 String test = "";
 for (byte b : encrypted)
 {
 test += Byte.toString(b);
 }
 return test;
 }
 // Encrypt message
 public byte[] encrypt(byte[] message)
{
 return (new BigInteger(message)).modPow(e, N).toByteArray();
 }
 // Decrypt message
 public byte[] decrypt(byte[] message)
 {
 return (new BigInteger(message)).modPow(d, N).toByteArray();
 }
}
```

OUTPUT



Set 7

Write a program to simulate Carrier Sense Multiple Access (CSMA/CA).

Server.java

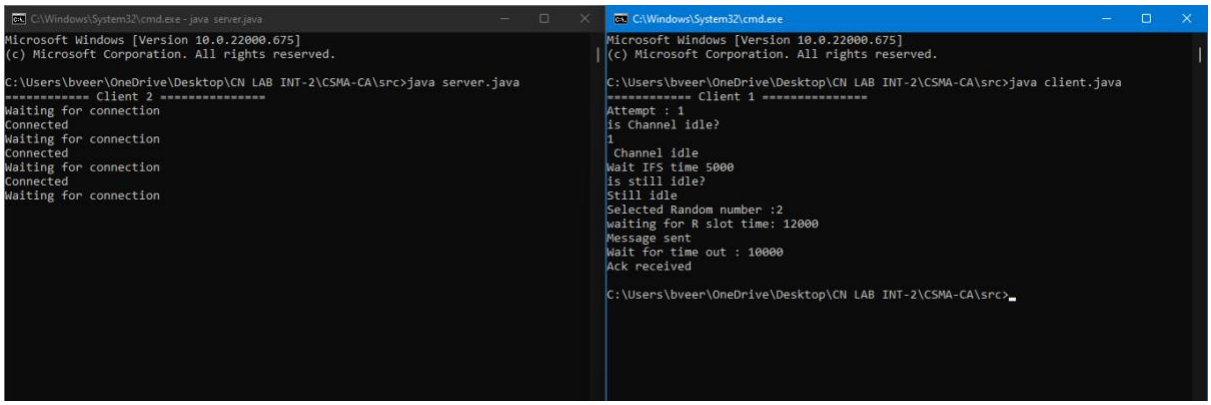
```
import java.net.*;
public class server
{
	public static void main(String[] args)
	{
		try
		{
		System.out.println("============ Client 2 ===============");
			while(true)
			{
				ServerSocket ss = new ServerSocket(3000);
				System.out.println("Waiting for connection");
				ss.accept();
				ss.close();
				System.out.println("Connected");
			}
		}
		catch(Exception e)
		{
			System.out.println(e);
		}
	}
}
```

Client.java

```
import java.net.*;

public class client
{
public static void main(String[] args)
{
try
	{
	System.out.println("============ Client 1 ===============");
	client cli = new client();
	int R = 0;
	Boolean bln = false;

	for(int k=1; k<=15;k++)
	{
		System.out.println("Attempt : "+k);
		System.out.println("is Channel idle? ");

		int i = 0;
		while(true)
		{
		System.out.print(i=i+1);
		if(cli.isidle())
		{
		System.out.println("\n Channel idle");
		System.out.println("Wait IFS time 5000");
		Thread.sleep(8000);
		System.out.println("is still idle?");
```

```
            if(cli.isidle())
            {
                  System.out.println("Still idle");
                  R = 2^k-1;
                  System.out.println("Selected Random number :"+R);
                  System.out.println("waiting for R slot time: "+R*6000);
                  Thread.sleep(R*6000);
                  System.out.println("Message sent");
                  System.out.println("Wait for time out : "+10000);
                  Thread.sleep(10000);

                  if(cli.isidle())
                  {
                        System.out.println("Ack received");
                        bln = true;
                        break;
                  }
                  else
                  {
                  System.out.println("Ack not received");
                  break;
                  }
            }
            else
            {
            System.out.println("Busy, goes to channel idle check");
                  }
            }
      }

      if(bln == true)
      {
            break;
                  }
            }
      }
            catch (InterruptedException e)
            {
                  System.out.println(e);
            }
      }

Boolean isidle()
 {
      try
      {
            Socket soc= new Socket("localhost",3000);
            soc.close();
            return true;
      }
      catch (Exception e)
      {
            return false;
      }
      }
}
```

OUTPUT


C:\Windows\System32\cmd.exe - java server.java
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.
C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CA\src>java server.java
========== Client 2 ==========
Waiting for connection
Connected
Waiting for connection
Connected
Waiting for connection
Connected
Waiting for connection
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.
C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CA\src>java client.java
========== Client 1 ==========
Attempt : 1
is Channel idle?
1
Channel idle
Wait IFS time 5000
is still idle?
Still idle
Selected Random number :2
waiting for R slot time: 12000
Message sent
Wait for time out : 10000
Ack received
C:\Users\bveer\OneDrive\Desktop\CN LAB INT-2\CSMA-CA\src>