# COMPUTER NETWORK LABORATORY

# Computer Network Laboratory Manual (17CSL57)

:Instructors:

**Prof. E Santosh, Prof. Suhas G**

**Prof. Ganesh K, Prof. Ranjith K C, Prof. Bhavya M R**

Dept. of CS&E, MIT, Mysore

**5th Sem CS&E 2019-20**

# Part-A

# LAB PROGRAM – 1

## | Problem Statement |

Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.

## | TCL Source Code |

```
set ns [new Simulator]
set na [open Lab1.nam w]
$ns namtrace-all $na
set nt [open Lab1.tr w]
$ns trace-all $nt
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns queue-limit $n0 $n1 1
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns queue-limit $n1 $n2 1
set TCP [new Agent/TCP]
$ns attach-agent $n0 $TCP
set CBR [new Application/Traffic/CBR]
$CBR attach-agent $TCP
set SINK [new Agent/TCPSink]
$ns attach-agent $n2 $SINK
$ns connect $TCP $SINK
proc End {} {
        global ns na nt
        $ns flush-trace
        close $na
        close $nt
        exec nam Lab1.nam &
        exit 0
}

$ns at 0.0 "$CBR start"

$ns at 50.0 "End"

$ns run
```

## AWK Code

```
BEGIN{Count=0;}
{
            if($1=="d")
                Count++;
}
END{printf("\n\n\tNumber of Packets Droppped is %d\n\n\n",Count);}
```

# LAB PROGRAM – 2

## | Problem Statement |

Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

## | TCL Source Code |

```
set ns [new Simulator]

$ns color 1 Red

$ns color 2 Green

set na [open Lab2.nam w]

$ns namtrace-all $na

set nt [open Lab2.tr w]

$ns trace-all $nt

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 1000Mb 1ms DropTail

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

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

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

$ns duplex-link $n3 $n5 2Mb 1ms DropTail

$ns queue-limit $n2 $n3 3

$ns queue-limit $n3 $n2 3

set Ping1 [new Agent/Ping]

$ns attach-agent $n0 $Ping1

set Ping2 [new Agent/Ping]

$ns attach-agent $n1 $Ping2

set Ping3 [new Agent/Ping]

$ns attach-agent $n4 $Ping3
```

```
set Ping4 [new Agent/Ping]
$ns attach-agent $n5 $Ping4
Agent/Ping instproc recv {from rtt} {
        $self instvar node_
        puts "Node[$node_ id] --> Node$from : RTT = $rtt ms"
}
$ns connect $Ping1 $Ping4
$ns connect $Ping2 $Ping3
$Ping1 set class_ 1
$Ping2 set class_ 2
proc End {} {
        global ns na nt
        $ns flush-trace
        close $na
        close $nt
        exec nam Lab2.nam &
        exit 0
}
for {set t 0} {$t < 5} {set t [expr $t+0.001]} {
        $ns at $t "$Ping1 send"
        $ns at $t "$Ping2 send"
}
$ns at 5.0 "End"
$ns run
```

## AWK Code

```
BEGIN{Count=0;}
{
        if($1=="d")
            Count++;
}
END{printf("\n\n\tNumber of Packets Droppped is %d\n\n\n",Count);}
```

# LAB PROGRAM – 3

## Problem Statement

Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.

## TCL Source Code

```
set ns [new Simulator]
$ns color 1 Red
$ns color 2 Blue
set na [open Lab3.nam w]
$ns namtrace-all $na
set nt [open Lab3.tr w]
$ns trace-all $nt
set ng1 [open tcp1.xg w]
set ng2 [open tcp2.xg w]
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$ns make-lan "$n0 $n1 $n2" 1Mb 10ms LL Queue/DropTail Mac/802_3
$ns make-lan "$n3 $n4 $n5" 2Mb 10ms LL Queue/DropTail Mac/802_3
$ns duplex-link $n0 $n3 1Mb 10ms DropTail
set tcp1 [new Agent/TCP]
set tcp2 [new Agent/TCP]
set cbr1 [new Application/Traffic/CBR]
set cbr2 [new Application/Traffic/CBR]
$ns attach-agent $n4 $tcp1
$cbr1 attach-agent $tcp1
$ns attach-agent $n1 $tcp2
$cbr2 attach-agent $tcp2
```

```
set sink1 [new Agent/TCPSink]
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink1
$ns attach-agent $n5 $sink2
$ns connect $tcp1 $sink1
$ns connect $tcp2 $sink2
$tcp1 set class_ 1
$tcp2 set class_ 2
proc End {} {
        global ns na nt
        $ns flush-trace
        close $na
        close $nt
        exec nam Lab3.nam &
        exec xgraph tcp1.xg tcp2.xg &
        exit 0
}
proc Draw {Agent File} {
        global ns
        set Cong [$Agent set cwnd_]
        set Time [$ns now]
        puts $File "$Time $Cong"
        $ns at [expr $Time+0.01] "Draw $Agent $File"
}
$ns at 0.0 "$cbr1 start"
$ns at 0.7 "$cbr2 start"
$ns at 0.0 "Draw $tcp1 $ng1"
$ns at 0.0 "Draw $tcp2 $ng2"
$ns at 10.0 "End"
$ns run
```

# LAB PROGRAM – 4

## | Problem Statement |

Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.

## | TCL Source Code |

```
set ns [new Simulator]
set na [open Lab4.nam w]
$ns namtrace-all-wireless $na 500 500
set nt [open Lab4.tr w]
$ns trace-all $nt
set topo [new Topography]
$topo load_flatgrid 500 500
$ns node-config -adhocRouting DSDV
$ns node-config -llType LL
$ns node-config -macType Mac/802_11
$ns node-config -ifqType Queue/DropTail
$ns node-config -ifqLen 50
$ns node-config -phyType Phy/WirelessPhy
$ns node-config -channelType Channel/WirelessChannel
$ns node-config -propType Propagation/TwoRayGround
$ns node-config -antType Antenna/OmniAntenna
$ns node-config -topoInstance $topo
$ns node-config -agentTrace ON
$ns node-config -routerTrace ON
create-god 4
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$n0 set X_ 250.0
$n0 set Y_ 250.0
```

```
$n0 set Z_ 0.0
$n1 set X_ 200.0
$n1 set Y_ 250.0
$n1 set Z_ 0.0
$n2 set X_ 250.0
$n2 set Y_ 250.0
$n2 set Z_ 0.0
$n3 set X_ 250.0
$n3 set Y_ 250.0
$n3 set Z_ 0.0
$ns at 0.0 "$n0 setdest 400.0 300.0 50.0"
$ns at 0.0 "$n1 setdest 50.0 100.0 20.0"
$ns at 0.0 "$n2 setdest 75.0 180.0 5.0"
$ns at 0.0 "$n3 setdest 100.0 100.0 25.0"
set tcp1 [new Agent/TCP]
$ns attach-agent $n0 $tcp1
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
set sink2 [new Agent/TCPSink]
$ns attach-agent $n3 $sink2
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $tcp1
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $tcp2
$ns connect $tcp1 $sink1
$ns connect $tcp2 $sink2
proc End {} {
   global ns nt na
   $ns flush-trace
   close $na
```

```
    close $nt
    exec nam Lab4.nam &
}
$ns at 0.0 "$cbr1 start"
$ns at 0.0 "$cbr2 start"
$ns at 10.0 "End"
$ns run
```

## AWK Code

```
BEGIN{Num_of_pkts=0;}
{
            if ($1 == "r" && $3 == "_1_" && $4 == "AGT" && $7 == "tcp")
            {
                    Num_of_pkts = Num_of_pkts + $8;
            }
}
END{
Throughput  = Num_of_pkts * 8 / $2 /1000000;
printf("\n\n\tThroughput = %fbpms\n\n\n",Throughput);
}
```

# LAB PROGRAM – 5

## Problem Statement

Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.

## TCL Source Code

```
set gsmDB 96000
set gsmUB 96000
set gsmPD 0.5
set gsmQB 10
set ns [new Simulator]
set nt [open Lab5.tr w]
$ns trace-all $nt
set CM1 [$ns node]
set BS1 [$ns node]
set MS  [$ns node]
set BS2 [$ns node]
set CM2 [$ns node]
proc cell_topo {} {
        global ns CM1 BS1 MS BS2 CM2
        $ns duplex-link $CM1 $BS1 3Mbps 10ms DropTail
        $ns duplex-link $BS1 $MS 1 1 RED
        $ns duplex-link $MS $BS2 1 1 RED
        $ns duplex-link $BS2 $CM2 3Mbps 50ms DropTail
}
switch gsm {
        gsm -
        gprs -
        umts {cell_topo}
}

$ns bandwidth $BS1 $MS $gsmDB simplex
$ns bandwidth $MS $BS1 $gsmUB simplex
$ns bandwidth $BS2 $MS $gsmDB simplex
$ns bandwidth $MS $BS2 $gsmUB simplex

$ns delay $BS1 $MS $gsmPD simplex
$ns delay $MS $BS1 $gsmPD simplex
$ns delay $BS2 $MS $gsmPD simplex
$ns delay $MS $BS2 $gsmPD simplex
```

```
$ns queue-limit $BS1 $MS $gsmQB
$ns queue-limit $MS $BS1 $gsmQB
$ns queue-limit $BS2 $MS $gsmQB
$ns queue-limit $MS $BS2 $gsmQB

$ns insert-delayer $MS $BS1 [new Delayer]
$ns insert-delayer $BS1 $MS [new Delayer]
$ns insert-delayer $MS $BS2 [new Delayer]
$ns insert-delayer $BS2 $MS [new Delayer]

set tcp [new Agent/TCP]
$ns attach-agent $CM1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $CM2 $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns connect $tcp $sink

proc End {} {
        global ns nt
        $ns flush-trace
        close $nt
        exec awk -f Lab5.awk Lab5.tr &
        exec xgraph -P -bar -x TIME -y DATA gsm.xg &
        exit 0
}

$ns at 0.0 "$ftp start"
$ns at 10.0 "End"
$ns run
```

## | AWK Code |

```
BEGIN {n=0;}
{
if($1 == "r")
{
n = n + $6;
printf("%f %d\n",$2,n) > "gsm.xg"
}
}
END{}
```

# LAB PROGRAM – 6

## | Problem Statement |

Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or equivalent environment.

## | TCL Source Code |

```
set cdmaDB 384000
set cdmaUB 64000
set cdmaPD 0.15
set cdmaQB 20
set ns [new Simulator]
set nt [open Lab6.tr w]
$ns trace-all $nt
set CM1 [$ns node]
set BS1 [$ns node]
set MS  [$ns node]
set BS2 [$ns node]
set CM2 [$ns node]

proc cell_topo {} {
        global ns CM1 BS1 MS BS2 CM2
        $ns duplex-link $CM1 $BS1 3Mbps 10ms DropTail
        $ns duplex-link $BS1 $MS 1 1 RED
        $ns duplex-link $MS $BS2 1 1 RED
        $ns duplex-link $BS2 $CM2 3Mbps 50ms DropTail
}

switch umts {
        umts {cell_topo}
}

$ns bandwidth $BS1 $MS $cdmaDB simplex
$ns bandwidth $MS $BS1 $cdmaUB simplex
$ns bandwidth $BS2 $MS $cdmaDB simplex
$ns bandwidth $MS $BS2 $cdmaUB simplex

$ns delay $BS1 $MS $cdmaPD simplex
$ns delay $MS $BS1 $cdmaPD simplex
$ns delay $BS2 $MS $cdmaPD simplex
$ns delay $MS $BS2 $cdmaPD simplex
```

```
$ns queue-limit $BS1 $MS $cdmaQB
$ns queue-limit $MS $BS1 $cdmaQB
$ns queue-limit $BS2 $MS $cdmaQB
$ns queue-limit $MS $BS2 $cdmaQB

$ns insert-delayer $MS $BS1 [new Delayer]
$ns insert-delayer $BS1 $MS [new Delayer]
$ns insert-delayer $MS $BS2 [new Delayer]
$ns insert-delayer $BS2 $MS [new Delayer]

set tcp [new Agent/TCP]
$ns attach-agent $CM1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $CM2 $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns connect $tcp $sink

proc End {} {
        global ns nt
        $ns flush-trace
        close $nt
        exec awk -f Lab6.awk Lab6.tr &
        exec xgraph -P -bar -x TIME -y DATA cdma.xg &
        exit 0
}

$ns at 0.0 "$ftp start"
$ns at 10.0 "End"
$ns run
```

## AWK Code

```
BEGIN {n=0;}
{
if($1 == "r")
{
n = n + $6;
printf("%f %d\n",$2,n) > "cdma.xg"
}
}
END{}
```

# Part-B

# LAB PROGRAM – 7

## | Problem Statement |

7. Write a program for error detecting code using CRC-CCITT (16- bits).

## | Java Source Code |

```
import java.util.*;
public class crc{
        public static void main(String[] args) {
                Scanner in=new Scanner(System.in);
                String data_s=new String();
                String divisor_s=new String();
                String data_r=new String();
                int n;
                System.out.println("Enter the Data Bits :");
                data_s=in.nextLine();
                System.out.println("Enter the divisor Bits :");
                divisor_s=in.nextLine();
                int data[]= new int[data_s.length()+divisor_s.length()-1];
                for (int i=0;i<data_s.length();i++ ) {
                        data[i]=data_s.charAt(i)-48;
                }
                int divisor[] =new int[divisor_s.length()];
                for (int i=0;i<divisor_s.length();i++) {
                        divisor[i]=divisor_s.charAt(i)-48;
                }
                int rem[]=divide(data,divisor);
                for (int i=0;i<rem.length-1;i++ ) {
                        data[data_s.length()+i]=rem[i];
                }
                for (int i=0;i<data.length;i++) {
                        System.out.print(data[i]);
                }
                System.out.println();
                System.out.println("Enter the message recieved);
                data_r=in.nextLine();
                int data1[]= new int[data_r.length()+divisor_s.length()-1];
                for (int i=0;i<data_r.length();i++ ) {
                        data1[i]=data_r.charAt(i)-48;
                }
                int rem1[]=divide(data1,divisor);
                int flag=0;
                for (int i=0;i<rem1.length;i++ ) {
                        if(rem1[i]!=0)
                                flag=1;
                }
                if(flag==0) {
                        System.out.println("No error in the message");
                }
                else {
                        System.out.println("Error in the message");
                }
        }
        static int[] divide(int data[],int divisor[]) {
                int reminder[]=new int[divisor.length];
                System.arraycopy(data,0,reminder,0,divisor.length);
                for (int i=0;i<data.length-(divisor.length-1); i++) {
                        if (reminder[0]==1) {
                                for (int j=1;j<divisor.length;j++ ) {
                                        reminder[j-1]=Exor(reminder[j],divisor[j]);
                                }
                        }
                        else {
                                for (int j=1;j<divisor.length;j++ ) {
                                        reminder[j-1]=Exor(reminder[j],0);
```

```
                    }
                }
                if (i<data.length-divisor.length) {
                    reminder[divisor.length-1]=data[i+divisor.length];
                }
                }
                return reminder;
            }
            static int Exor(int a,int b) {
                if(a==b){
                    return 0;
                }
                return 1;
            }
        }
```

# LAB PROGRAM – 8

## | Problem Statement |

8. Write a program to find the shortest path between vertices using bellman-ford algorithm.

## | Java Source Code |

```
import java.util.Scanner;

public class Lab8

{

 private static int[] D=new int[10];

 private static int n;

 private static final int MAX=999;

 private static int A[][]=new int[10][10];

 private static int s;


 public static void shortest()

 {

  for(int i=1;i<=n;i++){

      D[i]=MAX;

  }

   D[s]=0;

   for(int k=1;k<=n;k++) {

     for(int i=1;i<=n;i++) {

       for(int j=1;j<=n;j++) {

          if(A[i][j]!=MAX) {

            if(D[j]>D[i]+A[i][j])

               D[j]=D[i]+A[i][j];

          }

       }

     }

   }

  for(int i=1;i<=n;i++){

   for(int j=1;j<=n;j++){
```

```
   if(A[i][j]!=MAX) {
    if(D[j]>D[i]+A[i][j]) {
     System.out.println("The graph contains negative cycle");
     s=-1;
    return;
    }
   }
  }
 }
}
public static void main(String args[])
{

 Scanner sc=new Scanner(System.in);
 System.out.println("Enter the number of vertices");
 n=sc.nextInt();

 System.out.println("Enter the weight matrix");
 for(int i=1;i<=n;i++) {
    for(int j=1;j<=n;j++) {
       A[i][j]=sc.nextInt();
       if(i==j){
        A[i][j]=0;
        continue;
       }
       if(A[i][j]==0){
         A[i][j]=MAX;
       }
    }
 }
 System.out.println("Enter the source vertex");
 s=sc.nextInt();
 shortest();
 if(s!=-1){
 for(int i=1;i<=n;i++){
   System.out.println("Distance from Source " +s+"to"+i+"is"+D[i]);
  }
 }
 sc.close();
}
}
```

# LAB PROGRAM – 9

## | Problem Statement |

9. Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.

## | Java Source Code – Client |

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

class Client
{
	public static void main(String a[]) throws Exception
	{
		Socket S = new Socket("localhost",5555);

		InputStream iStream = S.getInputStream();
		BufferedReader Read = new BufferedReader(new InputStreamReader(iStream));

		OutputStream oStream = S.getOutputStream();
		PrintWriter Write = new PrintWriter(oStream,true);

		System.out.print("\n\nData.txt file has been sent to the Server\n\n");
		Write.println("Data.txt");

		System.out.print("\n\nContent Sent by the Server is\n\n");
		System.out.println(Read.readLine());

		Read.close();
		Write.close();
		S.close();
	}
}
```

## | Java Source Code – Server |

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

class Server
{
	public static void main(String a[]) throws Exception
	{
		ServerSocket SS = new ServerSocket(5555);
		System.out.println("Server is Waiting for Connection…");
		Socket S = SS.accept();
```

```
                System.out.println("Connection is Successful and Waiting to Serve…");



                InputStream iStream = S.getInputStream();
                BufferedReader Read = new BufferedReader(new InputStreamReader(iStream));

                OutputStream oStream = S.getOutputStream();
                PrintWriter Write = new PrintWriter(oStream,true);

                String File = Read.readLine();

                BufferedReader Content = new BufferedReader(new FileReader(File));

                Write.println(Content.readLine());

                Content.close();
                Read.close();
                Write.close();
                S.close();
                SS.close();
        }
}
```

# LAB PROGRAM – 10

## Problem Statement

10. Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.

## Java Source Code – Client

```
import java.io.*;
import java.net.*;
class DC {
        public static void main(String args[]) throws Exception {
                DatagramSocket S = new DatagramSocket();

                byte sData[] = new byte[1024];
                sData = "Client".getBytes();
                InetAddress Ip = InetAddress.getByName("localhost");
                int Port = 4444;
                DatagramPacket sPack = new DatagramPacket(sData,sData.length,Ip,4444);

                S.send(sPack);

                byte rData[] = new byte[1024];
                DatagramPacket rPack = new DatagramPacket(rData,rData.length);
                S.receive(rPack);

                String Message = new String(rPack.getData());

                System.out.println("Message from Server : " + Message);

                S.close();
        }

}
```

## Java Source Code – Server

```
import java.io.*;
import java.net.*;
class DS {
        public static void main(String args[]) throws Exception {
                DatagramSocket S = new DatagramSocket(4444);

                byte rData[] = new byte[1024];
                DatagramPacket rPack = new DatagramPacket(rData,rData.length);
```

```
                    S.receive(rPack);
                    System.out.println("Client Waiting for Service...");

                    BufferedReader Input = new BufferedReader(new InputStreamReader(System.in));

                    System.out.println("Enter the Message : ");
                    String Data = Input.readLine();

                    byte sData[] = new byte[1024];

                    sData = Data.getBytes();
                    InetAddress Ip = rPack.getAddress();
                    int Port = rPack.getPort();

                    DatagramPacket sPack = new DatagramPacket(sData,sData.length,Ip,Port);

                    S.send(sPack);
                    S.close();
          }
}
```

# LAB PROGRAM – 11

## | Problem Statement |

11. Write a program for simple RSA algorithm to encrypt and decrypt the data.

## | Java Source Code |

```
import java.util.Scanner;
class RSA{
        public static void main(String[] argc){
                int PT[]=new int[100];
                int CT[]=new int[100];
                int n,d,p,q,i,z,e=0;
                Scanner scan=new Scanner(System.in);
                System.out.println("Enter 2 large prime numbers:");
                p=scan.nextInt();
                q=scan.nextInt();
                n=p*q;
                z=(p-1)*(q-1);
                 for(i=2;i<z;i++){
                        if(gcd(i,z)==1){
                                e=i;
                                break;
                        }
                }
                for(d=1;d<=z;d++){
                        if((e*d-1)%z==0)
                                break;
                }
                System.out.println("Enter the message:");
                String msg=scan.next();
                System.out.println("Public Key=("+ e +","+ n +")");
                System.out.println("Private Key=("+ d +","+ n +")");
```

```
		for(i=0;i<msg.length();i++)
			PT[i]=(int) msg.charAt(i);
		for(i=0;i<msg.length();i++){
		System.out.print("The ASCII Value of "+(char)PT[i]+" is"+PT[i]+" & is converted into :");
			CT[i]=endecoder(PT[i],e,n);
			System.out.println(CT[i]);
		}
		System.out.print("Cipher Text=");
		for(i=0;i<msg.length();i++)
			System.out.print(CT[i]);
		System.out.println();
		System.out.print("Plain Text=");
		for(i=0;i<msg.length();i++)
			PT[i]=endecoder(CT[i],d,n);
		for(i=0;i<msg.length();i++)
			System.out.print((char) PT[i]);
	}
	public static int endecoder(int x,int y,int n){
		int k=1;
		for(int j=1;j<=y;j++)
			k=(k*x)%n;
		return k;
	}
	public static int gcd(int a,int b){
		if(b==0)
			return a;
		else
			return gcd(b,(a%b));
	}
}
```

# LAB PROGRAM – 12

## Problem Statement

12. Write a program for congestion control using leaky bucket algorithm.

## Java Source Code

```
import java.util.Scanner;
import java.util.Random;
class Leaky {
    public static void main(String[] args) {
        int[] a=new int[20];
        int buckRem = 0, sent, recv, k=0;
        Scanner scan=new Scanner(System.in);
        System.out.println("Enter the Bucket Size");
        int buckCap=scan.nextInt();
        System.out.println("Enter the Transmission Rate ");
        int rate=scan.nextInt();
        System.out.println("Enter the number of clock ticks");
        int n=scan.nextInt();
        Random random=new Random();
        for(int i=0;i<=n;i++)
            a[i]=random.nextInt(100)%(buckCap+30);
        System.out.println("clock pacsize accptd sent remaining ");
        try{
            for(int i = 0; i < n; i++) {
                k=i+1;
                if(a[i] != 0) {
                    if(buckRem + a[i] > buckCap) {
                        recv = -1;
                    }
                    else {
                        recv = a[i];
                        buckRem += a[i];
                    }
                }
                else {
                recv = 0;
                }
            if(buckRem != 0) {
                Thread.sleep(1000);
                if(buckRem < rate) {
                sent = buckRem;
                buckRem = 0;
                }
                else {
                    sent = rate;
                    buckRem = buckRem - rate;
                }
            }
            else {
            sent = 0;
            }
            if(recv == -1) {
            System.out.println(k + "\t" + a[i] + "\tDropd " + sent + "\t" + buckRem);
        }
        else {
            System.out.println(k + "\t" + a[i] + "\t" +recv +"\t"+ sent +"\t"+ buckRem);
        }
        }
        }
        catch (Exception ex){
            System.out.println("Interrupted Exception");
        }
    }
}
```

## Or

```
 import java.util.Scanner;
public class Buck{
        public static void main(String[] s){
                Scanner In = new Scanner(System.in);
                int Bkt[] = new int[1000];
                int T=0,i=0,k=0;
                int j=0;
                System.out.println("\nEnter Maximum OutFlow Capacity : ");
                int maxCap = In.nextInt();
                while(true)
                {
                     System.out.println("\nEnter Number of Pkts at Time-"+T+": ");
                     int n = In.nextInt();
                     if(n>0)
                     {
                        System.out.println("\nEnter Size of each Pkts : ");
                        for(;i<n;i++)
                           Bkt[i] = In.nextInt();
                     }
                     System.out.println("\n\n\t\tAt T = "+T+1+" Outflow Pkts are :-");
                     int Cap = maxCap;
                     while(Bkt[i] <= Cap)
                     {
                               Cap = Cap - Bkt[i];
                               System.out.println("\n\t\tPacket["+Bkt[j]+"]");
                     }

                     T++;
          }
        }
}
```

# Viva Questions

1. What are functions of different layers?

2. Differentiate between TCP/IP Layers and OSI Layers

3. Why header is required?

4. What is the use of adding header and trailer to frames?

5. What is encapsulation?

6. Why fragmentation requires?

7. What is MTU?

8. Which layer imposes MTU?

9. Differentiate between flow control and congestion control.

10. Differentiate between Point-to-Point Connection and End-to-End connections.

11. What are protocols running in different layers?

12. What is Protocol Stack?

13. Differentiate between TCP and UDP.

14. Differentiate between Connectionless and connection oriented connection.

15. Why frame sorting is required?

16. What is meant by subnet?

17. What is meant by Gateway?

18. What is an IP address?

19. What is MAC address?

20. Why IP address is required when we have MAC address?

21. What is meant by port?

22. What are ephemerical port number and well known port numbers?

23. What is a socket?

24. What are the parameters of socket()?

25. Describe bind(), listen(), accept(),connect(), send() and recv().

26. What are system calls? Mention few of them.

27. What is IPC? Name three techniques.

28. Explain mkfifo(), open(), close() with parameters.

29. What is meant by file descriptor?

30. What is meant by traffic shaping?
31. How do you classify congestion control algorithms?
32. Differentiate between Leaky bucket and Token bucket.
33. How do you implement Leaky bucket?
34. How do you generate busty traffic?
35. What is the polynomial used in CRC-CCITT?
36. What are the other error detection algorithms?
37. What is difference between CRC and Hamming code?
38. Why Hamming code is called 7,4 code?
39. What is odd parity and even parity?
40. What is meant by syndrome?
41. What is generator matrix?
42. What is spanning tree?
43. Where Pirm's algorithm does finds its use in Networks?
44. Differentiate between Prim's and Kruskal's algorithm.
45. What are Routing algorithms?
46. How do you classify routing algorithms? Give examples for each.
47. What are drawbacks in distance vector algorithm?
48. How routers update distances to each of its neighbor?
49. How do you overcome count to infinity problem?
50. What is cryptography?
51. How do you classify cryptographic algorithms?
52. What is public key?
53. What is private key?
54. What are key, ciphertext and plaintext?
55. What is simulation?
56. What are advantages of simulation?
57. Differentiate between Simulation and Emulation.
58. What is meant by router?
59. What is meant by bridge?
60. What is meant by switch?
61. What is meant by hub?

62. Differentiate between route, bridge, switch and hub.

63. What is ping and telnet?

64. What is FTP?

65. What is BER?

66. What is meant by congestion window?

67. What is BSS?

68. What is incoming throughput and outgoing throughput?

69. What is collision?

70. How do you generate multiple traffics across different sender-receiver pairs?

71. How do you setup Ethernet LAN?

72. What is meant by mobile host?

73. Name few other Network simulators

74. Differentiate between logical and physical address.

75. Which address gets affected if a system moves from one place to another place?

76. What is ICMP? What are uses of ICMP? Name few.

77. Which layer implements security for data?