PRAGNANAM BRAHMA
ANITS

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY & SCIENCES
Autonomous status accorded by UGC and Andhra University
Approved by AICTE, Permanently Affiliated to Andhra University
Accredited by NBA (IT,CSE,EEE,ECE, and Mech) & accredited by NAAC with "A" Grade
Sangivalasa - 531162, Bheemunipatnam (Mandal), Visakhapatnam (Dist.)
Phone: 08933 - 225083, 225084, 226131, Fax: 08933-226395
Email: principal@anits.edu.in
COLLEGE CODE - ANIL

A Laboratory Manual
For
COMPUTER NETWORKS LAB

DEPARTMENT OF COMPUTER SCIENCES & ENGINEERING

VISION:

Our vision is to emerge as a world class Computer Science and Engineering department through excellent teaching and strong research environment that responds swiftly to the challenges of changing computer science technology and addresses technological needs of the stakeholders.

MISSION:

To enable our students to master the fundamental principles of computing and to develop in them the skills needed to solve practical problems using contemporary computer-based technologies and practices to cultivate a community of professionals who will serve the public as resources on state-of- the-art computing science and information technology.

Course outcomes:

1.	To establish communicate among computers in the network using various network hardware components.
2.	Identification and Implement the various protocols for establishing the network using simulation tools like Wireshark.
3.	Simulating various network topologies, designing and implementation of various routing techniques
4.	Programmatical implementation of socket based communication and evaluating the various architectures.

PROGRAM OUTCOMES (POs):

Graduate Attribute1:	Engineering Knowledge
PO-1	Apply the knowledge of basic engineering sciences, humanities, core engineering and computing concept in modelling and designing computer based systems.
Graduate Attribute2:	Problem Analysis
PO-2	Identify, analyze the problems in different domains and define the requirements appropriate to the solution.
Graduate Attribute3:	Design/Development of Solution
PO-3	Design, implement & test a computer based system, component or process that meet functional constraints such as public health and safety, cultural, societal and environmental considerations.
Graduate Attribute4:	Conduct Investigations of Complex Problems
PO-4	Apply computing knowledge to conduct experiments and solve complex problems, to analyze and interpret the results obtained within specified timeframe and financial constraints consistently.
Graduate Attribute5:	Modern Tool Usage
PO-5	Apply or create modern techniques and tools to solve engineering problems that demonstrate cognition of limitations involved in design choices.

Graduate Attribute6:	The Engineer and Society
PO-6	Apply contextual reason and assess the local and global impact of professional engineering practices on individuals, organizations and society.
Graduate Attribute7:	Environment and Sustainability
PO-7	Assess the impact of engineering practices on societal and environmental sustainability.
Graduate Attribute8:	Ethics
PO-8	Apply professional ethical practices and transform into good responsible citizens with social concern.
Graduate Attribute9:	Individual and Team Work
PO-9	Acquire capacity to understand and solve problems pertaining to various fields of engineering and be able to function effectively as an individual and as a member or leader in a team.
Graduate Attribute10:	Communication
PO-10	Communicate effectively with range of audiences in both oral and written forms through technical papers, seminars, presentations, assignments, project reports etc.
Graduate Attribute11:	Project Management and Finance

PO-11	Apply the knowledge of engineering, management and financial principles to develop and critically assess projects and their outcomes in multidisciplinary areas.
Graduate Attribute12:	Life-long Learning
PO-12	Recognize the need and prepare oneself for lifelong self learning to be abreast with rapidly changing technology.

PROGRAM SPECIFIC OUTCOMES (PSOs):

1	Programming and software Development skills: Ability to acquire programming efficiency to analyze, design and develop optimal solutions, apply standard practices in software project development to deliver quality software product.
2	Computer Science Specific Skills: Ability to formulate, simulate and use knowledge in various domains like data engineering, image processing and information and network security, artificial intelligence etc., and provide solutions to new ideas and innovations

A Laboratory Manual
For
COMPUTER NETWORK (CSE 317)

Semester – 1

PRAGNANAM BRAHMA

ANITS

Prepared by

1. Dr. Jagadish G, Asst. Professor
2. Mrs.T.Anitha, Asst. Professor
3.CH. Roopesh, Asst. Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

S. No	List of Experiments	CO
1.	Implementation of Error Detection and Error Correction Techniques.	1

2.	**Implementation of Stop and Wait Protocol and Sliding Window Protocol.**	**1**
3.	**Implementation of High level data link control protocol.**	**2**
4	**Study the performance of network with CSMA/CA protocol and compare with CSMA/CD protocols.**	**4**
5	**Simulate network topology STAR ,BUS, RING**	**3**
6	**Study of Socket Programming and Client – Server model.**	**3**
7	**Write a code simulating ARP /RARP protocols.**	**2**
8	**Write a code simulating PING and TRACEROUTE commands.**	**4**
9	**Implementation of Distance vector routing algorithm.**	**4**
10	**Implementation of Link state routing algorithm.**	**4**

LIST OF INDUSTRY RELEVANT SKILLS:

- Computer Network Architect
- Computer Systems Analyst
- Network Programmer
- Wireless Network Engineer
- Network Administrator
- System Engineer

GUIDELINES TO TEACHERS

- Faculty must verify the observations and records before assigning the system.
- Faculty must verify Students Id cards before entering into Laboratory
- Faculty must take the attendance at the starting and ending of the lab time period.

This lab course consists of two set of programs

1) Minimum set of sample programs
2) Additional set of programs

Minimum set of sample programs are designed unit wise covering all the topics in the theory .

Additional set of programs are designed basing on problem solving.

Sessional marks : 50 marks

1) Daily Evaluation (Includes Record, Observation & regular performance) – 30 marks
2) Attendance – 5 marks
3) Internal Exam – 10 marks
4) Viva Voce – 5 marks

Daily Evaluation (30 marks)

Every Student must execute a minimum set of sample programs to secure 60% of marks in Daily Evaluation i.e. 18 Marks and to appear in external examination.
In addition to that if a student finishes the minimum set and 5 programs from an additional set of programs would secure 80% of marks in Daily Evaluation i.e. 24 Marks.
If a student finishes all the programs in both the set s will secure 100% of marks in Daily Evaluation

Internal Exam (10 marks)

- Every student is given 4 questions in the internal exam out of which the difficulty level of 2 questions is easy / medium and 2 questions of difficulty level is high
- Each easy / medium level question carries 20% of marks and difficulty level question carries 30% of marks

External Exam (50 marks)

- Viva voce – 10 marks
- Write up + Execution – 40 marks

Write up + Execution (40 marks)

- Every student is given 4 questions in the external exam out of which the difficulty level of 2 questions is easy / medium and 2 questions of difficulty level is high
- Each easy / medium level question carries 30% of marks and difficulty level question carries 20% of marks.

INSTRUCTIONS TO STUDENTS:

- Students should use computer related components smoothly
- Students should not carry other items into the lab.
- Students must wear the dress code and ID cards.
- Every student is given 4 questions in the external exam out of which the difficulty level of 2 questions is easy / medium and 2 questions of difficulty level is high

- Each easy / medium level question carries 30% of marks and difficulty level question carries 20% of marks.

GUIDELINES TO LAB PROGRAMMERS:

- Lab Programmers must verify All the Systems whether they are working properly or not.
- Lab Programmers must verify All the other equipments (devices like ACs).
- Lab Programmers must verify NS2 Tool installation & C Language in all Systems.

LAB RUBRICS

Name of the Programme: B.Tech **Session Duration: 3 Hours**

Semester:II **Course Code and Title : Computer Networks Lab (CSE 317)**

Key Performance Criteria(KPC) (25 pts)	4-Very Good	3-Good	2-Fair	1-Need to imp
Problem tatement (2)	The thorough knowledge of the problem statement.(3)	The better knowledge of the problem statement(2)	The basic knowledge of the problem statement(2)	The partial knowledge of th problem statement.(1)
xperimental rocedure Algorithm/Flo vchart/Tools Description)(4)	The experimental procedure is explained with the relevant implementation of the Tool.(4)	The experimental procedure is explained clearly and the details are covered.(3)	The experimental procedure is explained and few details are covered.(2)	The experiment procedure is explained, some minor implementation details are missing.(1)
Vorking with ools and imulation(4)	Simulation of the tools w.r.t the given Problem Statement is executed using the respective commands/ Source code effectively.(4)	Simulation of the tools w.r.t the given Problem Statement is executed using the respective commands/ Source code.(3)	Simulation of the tools w.r.t the given Problem Statement is executed using the respective commands/ Source code With tool.(2)	Simulation of th tools w.r.t the g Problem Statem executed partia using the respe commands/ Sou code.(1)
est Case erification(3)	Produces correct output for all mentioned test cases correctly in implementation /simulation(3)	Produces correct output for majority of test cases correctly in implementation /simulation(3)	Produces correct output for few important possible test cases correctly in implementation /simulation(3)	Produces Wron output for most the test cases i implementation /simulation (1)
ral Presentation/V va 5)	In depth knowledge on the concept and answered all the questions.(4)	Good knowledge on the concept and answered all the questions(4)	Basic knowledge on the concept and answered some of the questions(2)	With basic knowledge on t concept and answered few questions(2)

Presentation / Documentation ased on Observation 4)	Presented accurately all the prescribed documentation on time (4)	Presented all the required documentation on time as per the prescribed format(3)	Presented documents in a readable manner but not so neatly. Submitted documents on time.(2)	Submitted documents in ambiguity and r time.(1)
Code of Conduct- Courtesy , afety and ethics based on physical observation)(3	While conducting the procedure, the student is in proper dress code, always respectful of others and leaves the area clean.(3)	While conducting the procedure, the student is in proper dress code, many times respectful of others and leaves the area clean only after being reminded.(2)	While conducting the procedure, the student is in partial dress code, sometimes respectful of others and leaves the area clean only after being reminded.(2)	While conducti procedure, the student is not ir proper dress co not respectful o others and leav area messy eve after being reminded.(1)

Practical 1 : Implementation of Error Detection and Error Correction Techniques.

1. Practical significance :

Error detection and correction code plays an important role in the transmission of data from one source to another. The noise also gets added into the data when it transmits from one system to another, which causes errors in the received binary data at other systems. The bits of the data may change(either 0 to 1 or 1 to 0) during transmission.

It is impossible to avoid the interference of noise, but it is possible to get back the original data. For this purpose, we first need to detect either an error z is present or not using error detection codes. If the error is present in the code, then we will correct it with the help of error correction codes. The error detection codes are the code used for detecting the error in the received data bitstream. In these codes, some bits are included appended to the original bitstream.

Error detecting codes encode the message before sending it over the noisy channels. The encoding scheme is performed in such a way that the decoder at the receiving can find the errors easily in the receiving data with a higher chance of success.

- These codes are used when we use message backward error correction techniques for reliable data transmission. A feedback message is sent by the receiver to inform the sender whether the message is received without any error or not at the receiver side. If the message contains errors, the sender retransmits the message.
- In error detection codes, in fixed-size blocks of bits, the message is contained. In this, the redundant bits are added for correcting and detecting errors.
- These codes involve checking of the error. No matter how many error bits are there and the type of error.
- Parity check, Checksum, and CRC are the error detection technique.

2. Relevant Program Outcomes : PO1, PO2,PO3,PO4,PO5

3. Competency and practical skills :

The practical is expected to develop the fallowing skills :

1. Extract the knowledge from different error detecting and correcting coding schemes.

2. Ability to work on C/Java compilers.

4.Prerequisites :

1. Student should have knowledge on Computer Networks.

5. Resources required :

S.No	Name of the Resource Broad Specification(Approx.)
1	Computer System 1. Processor – 2GHz 2. RAM – 4GB 3. Hard-Drive Space – 20GB 4. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)
2	JDK(JAVA DEVELOPMENT KIT)

6. Precautions:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7. Algorithm/circuit/Diagram/Description:

1. Parity bit Method :

```
#include<stdio.h>

#include<string.h>

int main()

{

int count=0;

int i,parity;

char data[20];

printf("enter data\n");

scanf("%s",&data);

int l=strlen(data);

printf("choose parity\n0:Even\n1:Odd");

scanf("%d",&parity);

for(i=0;i<l;i++)

{

if(data[i]=='1')

{

count++;

}

}
if(parity==0 && count%2==0)
{
printf("No error detected\n");
}
else if(parity==1&&count%2==1)
{
printf("No error detected\n");
}
else
{
printf("error detected\n");
}
return 0;
}
```

2. CRC Method:

```
#include <stdio.h>
 #include <conio.h>
 #include <string.h>
 void main() {
        int i,j,keylen,msglen;
        char input[100], key[30],temp[30],quot[100],rem[30],key1[30];
        clrscr();
        printf("Enter Data: ");
        gets(input);
```

```
printf("Enter Key: ");
gets(key);
keylen=strlen(key);
msglen=strlen(input);
strcpy(key1,key);
for (i=0;i<keylen-1;i++) {
          input[msglen+i]='0';
}
for (i=0;i<keylen;i++)
 temp[i]=input[i];
for (i=0;i<msglen;i++) {
          quot[i]=temp[0];
          if(quot[i]=='0')
           for (j=0;j<keylen;j++)
           key[j]='0'; else
           for (j=0;j<keylen;j++)
           key[j]=key1[j];
          for (j=keylen-1;j>0;j--) {
                              if(temp[j]==key[j])
                               rem[j-1]='0'; else
                               rem[j-1]='1';
          }
          rem[keylen-1]=input[i+keylen];
          strcpy(temp,rem);
}
```

```
    strcpy(rem,temp);
    printf("\nQuotient is ");
    for (i=0;i<msglen;i++)
     printf("%c",quot[i]);
    printf("\nRemainder is ");
    for (i=0;i<keylen-1;i++)
     printf("%c",rem[i]);
    printf("\nFinal data is: ");
    for (i=0;i<msglen;i++)
     printf("%c",input[i]);
    for (i=0;i<keylen-1;i++)
     printf("%c",rem[i]);
    getch();
}
```

8. Test cases:

1. Outputs with different inputs must be given and appropriate results must be recorded after execution.

2. Students must also record the errors while executing the commands, so that they get deeper insights on how a command should be executed.

9. Sample output:

Parity bit Method :

```
1.Enter data  : 01011011
Choose Parity
0 : Even          1 : Odd
0
```

```
        Error Detected

        2. Enter data  : 01011011

        Choose Parity

        0 : Even          1 : Odd

        1

No Error Detected
```

10.Practical Related Questions:

1. What is Even parity and Odd parity ?

2. What is the use of CRC?

3. Name CRC standards?

4. Define generator polynomial?

Practical 2 : Implementation of Stop and Wait Protocol and Sliding Window Protocol.

1. Practical significance :

Stop and Wait Protocol :

It is the simplest flow control method. In this, the sender will transmit one frame at a time to the receiver. The sender will stop and wait for the acknowledgement from the receiver. This time (i.e. the time joining message transmitting and acknowledgement receiving) is the sender's waiting time, and the sender is idle during this time.

When the sender gets the acknowledgement (ACK), it will send the next data packet to the receiver and wait for the disclosure again, and this process will continue as long as the sender has the data to send. While sending the data from the sender to the receiver, the data flow needs to be controlled. If the sender is transmitting the data at a rate higher than the receiver can receive and process it, the data will get lost. The Flow-control methods will help in ensuring that the data doesn't get lost. The flow control method will check that the senders send the data only at a rate that the receiver can receive and process.

The features of Stop and Wait Protocol are as follows −

- It is used in Connection-oriented communication.
- It offers error and flows control.
- It can be used in data Link and transport Layers.
- Stop and Wait ARQ executes Sliding Window Protocol with Window Size 1

Sliding Window Protocol :

Sliding window protocols are data link layer protocols for reliable and sequential delivery of data frames. The sliding window is also used in Transmission Control Protocol.

In this protocol, multiple frames can be sent by a sender at a time before receiving an acknowledgment from the receiver. The term sliding window refers to the imaginary boxes to hold frames. Sliding window method is also known as windowing.

Working Principle

In these protocols, the sender has a buffer called the sending window and the receiver has buffer called the receiving window.

The size of the sending window determines the sequence number of the outbound frames. If the sequence number of the frames is an n-bit field, then the range of sequence numbers that can be assigned is 0 to $2n-1$. Consequently, the size of the sending window is $2n-1$. Thus in order to accommodate a sending window size of $2n-1$, a n-bit sequence number is chosen.

The sequence numbers are numbered as modulo-n. For example, if the sending window size is 4, then the sequence numbers will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, and so on. The number of bits in the sequence number is 2 to generate the binary sequence 00, 01, 10, 11.

The size of the receiving window is the maximum number of frames that the receiver can accept at a time. It determines the maximum number of frames that the sender can send before receiving acknowledgment.

2. Relevant Program Outcomes : PO1,PO2,PO3, PO5

3. Competency and practical skills :

The practical is expected to develop the fallowing skills :

1. Extract the knowledge from different Networking protocols and their implementations.

2. Ability to work on C/Java compilers.

4.Prerequisites :

1. Student should have knowledge on Computer Networks.

5. Resources required :

S.No	Name of the Resource Broad Specification(Approx.)

1	Computer System 1. Processor – 2GHz 2. RAM – 4GB 3. Hard-Drive Space – 20GB 4. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)
2	JDK(JAVA DEVELOPMENT KIT)

6. Precautions:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7. Algorithm/circuit/Diagram/Description:

i) Stop-and-wait ARQ:

RECEIVER(SERVER):

```
import java.io.*;
import java.net.*;
import java.util.*;
public class receiver {
    public static void main(String args[])
{
   String h="Serverhost";
   int q=5000;
   int i;
      try
      {
      ServerSocket ss2;
         ss2 = new ServerSocket(8000);
         Socket s1 =ss2.accept();
      DataInputStream dd1= new DataInputStream(s1.getInputStream());
      Integer i1 =dd1.read();
      for(i=0;i<i1;i++)
      {
         ServerSocket ss1;
         ss1 = new ServerSocket(9000+i);
         Socket s =ss1.accept();
      DataInputStream dd= new DataInputStream(s.getInputStream());
      String sss1 = dd.readUTF();
         System.out.println(sss1);
         System.out.println("Frame "+ i+" received");
      DataOutputStream d1 = new DataOutputStream(s.getOutputStream());
      d1.write(i);
       System.out.println("ACK sent for "+ i);
      }
      }
      catch(Exception ex)
      {
       System.out.println("Error"+ex);
            }
}
}
```

SENDER(CLIENT) :

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

```
public class sender {
   public static void main(String args[])
{
   int p=9000,i,q=8000;
   String h="localhost";
   try
   {
      Scanner scanner = new Scanner(System.in);
   System.out.print("Enter number of frames : ");
   int number = scanner.nextInt();
   if(number==0)
   {
      System.out.println("No frame is sent");
   }
   else
         {
        Socket s2;
      s2= new Socket(h,q);
      DataOutputStream d1 = new DataOutputStream(s2.getOutputStream());
      d1.write(number);
        }
   String str1;
      for (i=0;i<number;i++)
      {
   System.out.print("Enter message : ");
   String name = scanner.next();
   System.out.println("Frame " + i+" is sent");
   Socket s1;
      s1= new Socket(h,p+i);
      DataOutputStream d = new DataOutputStream(s1.getOutputStream());
      d.writeUTF(name);
      DataInputStream dd= new DataInputStream(s1.getInputStream());
      Integer sss1 = dd.read();
      System.out.println("Ack for :" + sss1 + " is  received");
      }
   }
```

```
    catch(Exception ex)
            {
               System.out.println("ERROR :"+ex);
            }
}
}
```

8. Test cases:

1. Outputs with different inputs must be given and appropriate results must be recorded after execution.

2. Students must also record the errors while executing the commands, so that they get deeper insights on how a command should be executed.

9. Sample output:

client class

server class

ii) SLIDING WINDOW PROTOCOL :

SENDER PROGRAM

```
import java.net.*;
import java.io.*;
import java.rmi.*;
public class slidsender
{
public static void main(String a[])throws Exception
{
ServerSocket ser=new ServerSocket(10);
Socket s=ser.accept();
DataInputStream in=new DataInputStream(System.in);
DataInputStream in1=new DataInputStream(s.getInputStream());
String sbuff[]=new String[8];
PrintStream p;
int sptr=0,sws=8,nf,ano,i;
String ch;
do
{
p=new PrintStream(s.getOutputStream());
System.out.print("Enter the no. of frames : ");
nf=Integer.parseInt(in.readLine());
p.println(nf);
if(nf<=sws-1)
{

System.out.println("Enter "+nf+" Messages to be send\n");
for(i=1;i<=nf;i++)
{
sbuff[sptr]=in.readLine();
p.println(sbuff[sptr]);
sptr=++sptr%8;
}
sws-=nf;
System.out.print("Acknowledgment received");
ano=Integer.parseInt(in1.readLine());
System.out.println(" for "+ano+" frames");
sws+=nf;
}
else
{
System.out.println("The no. of frames exceeds window size");
break;
}
System.out.print("\nDo you wants to send some more frames : ");
ch=in.readLine(); p.println(ch);
}
while(ch.equals("yes"));
s.close();
}
}
```

RECEIVER PROGRAM

```
import java.net.*;
import java.io.*;
class slidreceiver
{
public static void main(String a[])throws Exception
{
Socket s=new Socket(InetAddress.getLocalHost(),10);
 DataInputStream in=new DataInputStream(s.getInputStream());
PrintStream p=new PrintStream(s.getOutputStream());
int i=0,rptr=-1,nf,rws=8;
String rbuf[]=new String[8];
String ch; System.out.println();
do
{
nf=Integer.parseInt(in.readLine());
if(nf<=rws-1)
{
for(i=1;i<=nf;i++)
{
rptr=++rptr%8;
rbuf[rptr]=in.readLine();
System.out.println("The received Frame " +rptr+" is : "+rbuf[rptr]);
}
rws-=nf;
System.out.println("\nAcknowledgment sent\n");
p.println(rptr+1); rws+=nf; }
else
break;
ch=in.readLine();
}
while(ch.equals("yes"));
}
}
```

9. Sample output:

```
//SENDER OUTPUT
Enter the no. of frames : 4
Enter 4 Messages to be send

hiii
how r u
i am fine
how is everyone
Acknowledgment received for 4 frames

Do you wants to send some more frames : no

RECEIVER OUTPUT
The received Frame 0 is : hiii
The received Frame 1 is : how r u
The received Frame 2 is : i am fine
The received Frame 3 is : how is everyone
Acknowledgment sent
```

10.Practical Related Questions:

1. What is Go-back- N ARQ?
2. What are the differences between ‘Selective Repeat ARQ’ and ‘Go-back- N ARQ’?
3. What are the Primitives of Stop and Wait Protocol?
4. What are the disadvantages of Stop and Wait Protocol?

Practical 3 : Implementation of High level data link control protocol.

1. Practical significance :

HDLC is a group of data link (Layer 2) protocols used to transmit synchronous data packets between point-to-point nodes. Data is organized into addressable frames. This format has been used for other multipoint-to-multipoint protocols, and inspired the HDLC-like framing protocol described in RFC 1662.

2. Relevant Program Outcomes : PO1, PO2,PO3,PO4,PO5

3. Competency and practical skills :

The practical is expected to develop the fallowing skills :

1. Extract the knowledge from different error detecting and correcting coding schemes.

2. Ability to work on C/Java compilers.

4.Prerequisites :

1. Student should have knowledge on Computer Networks.

5. Resources required :

S.No	Name of the Resource Broad Specification(Approx.)

1	Computer System 1. Processor – 2GHz 2. RAM – 4GB 3. Hard-Drive Space – 20GB 4. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)
2	JDK(JAVA DEVELOPMENT KIT)

6. Precautions:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7. Algorithm/circuit/Diagram/Description:

```
import java.util.*;

public class Main

{

static int[] crc_fun(int[] crc)
```

```
{
int l_d = 4,l_dd = 8;
char divisor[] = {'1','1','0','0'};
char divident[] = new char[8];
for(int i = 0;i<8;i++){
divident[i] = (char)('0' + crc[i+8]);
}
int l = l_dd + l_d-1;
char n_str[] = new char[l+1];
for(int i = 0;i<l_dd;i++){
n_str[i] = divident[i];
}
for(int i = l_dd;i<l;i++){
n_str[i]='0';
}
char div_part[] = new char[l_d];
for(int i = 0;i<l_d;i++) div_part[i]=n_str[i];
int in = 0;
for(int i= 0;i<l-l_d+1;i++){
if(div_part[0]=='1'){
for(int com = 0;com<l_d;com++){
div_part[com] = (char)(((div_part[com]-'0') ^ (divisor[com]-'0')) + '0');
}
}
for(int sp = 0;sp<l_d-1;sp++){
div_part[sp] = div_part[sp+1];
}
div_part[l_d-1] = n_str[i+l_d];
in++;
}
int fs[] = new int[16];
for(int i = 5;i<13;i++){
fs[i] = crc[i + 3];
}
in = 0;
for(int i = 13;i<16;i++){
fs[i] = (div_part[in] - '0');
in++;
}
return fs;
}

public static void main(String[] args)
 {
Scanner s = new Scanner(System.in);
int[] fd = new int[] { 0, 1, 1, 1, 1, 1, 1, 0 };
```

```
int[] crc = {0,0,0,0,0,0,0,0,1,1,1,0,1,1,1,0};

int payload;

int[] address = new int[] { 0, 0, 1, 1, 0, 0, 1, 1 };

int[] control = new int[] { 0, 1, 1, 0, 1, 1, 0, 0 };

System.out.println("enter the length of payload");

payload = s.nextInt();

int payarr[] = new int[payload];

int hdlcframe[] = new int[48 + payload];

for (int i = 0; i < 8; i++) {

hdlcframe[i] = fd[i];

}

for (int i = 0; i < 8; i++) {

hdlcframe[i + 8] = address[i];

}

for (int i = 0; i < 8; i++) {

hdlcframe[i + 16] = control[i];

}

System.out.println("enter the payload data");

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

payarr[i] = s.nextInt();

}

crc = crc_fun(crc);

for (int i = 0; i < payload; i++) {
```

```
hdlcframe[i + 24] = payarr[i];

}

for (int i = 0; i < 16; i++) {

hdlcframe[i + 24 + payload] = crc[i];

}

for (int i = 0; i < 8; i++) {

hdlcframe[i + 40 + payload] = fd[i];

}

System.out.println("HDLC: ");

for(int i = 0;i<48 + payload;i++){

System.out.print(hdlcframe[i]);

}

}

}
```

8. Test cases:

1. Outputs with different inputs must be given and appropriate results must be recorded after execution.

2. Students must also record the errors while executing the commands, so that they get deeper insights on how a command should be executed.

9. Sample output:

10.Practical Related Questions:

1. In Asynchronous balanced mode (ABM), the link is ?

2. List the types of stations is HDLC.

3. Mention the types of frames in HDLC.

4. Write the types of frame fields contained in HDLC.

PRACTICAL 04: Study the performance of network with CSMA/CA protocol and compare with CSMA/CD protocols.

1. Practical significance:

To create scenario and study the performance of CSMA / CD protocol through simulation. Ethernet is a LAN (Local area Network) protocol operating at the MAC (Medium Access Control) layer. Ethernet has been standardized as per IEEE 802.3.

The underlying protocol in Ethernet is known as the CSMA / CD – Carrier Sense Multiple Access / Collision Detection. The working of the Ethernet protocol is as explained below, A node which has data to transmit senses the channel. If the channel is idle then, the data is transmitted. If the channel is busy then, the station defers transmission until the channel is sensed to be idle and then immediately transmitted. If more than one node starts data transmission at the same time, the data collides. This collision is heard by the transmitting nodes which enter into contention phase. The contending nodes resolve contention using an algorithm called Truncated binary exponential back off.

2. Relevant Program Outcomes: PO1, PO2, PO3, PO9

3. Competency and practical skills :

The practical is expected to develop the following skills

Ability to study and understand CSMA and Collision Detection.

4. Prerequisites:

1. Student should have knowledge on Data link layer

2. Student should have knowledge on Medium Access control mechanism

5. Resources required:

S.No	Name of the Resource Broad Specification(Approx.)
1	Computer System 1. Processor – 2.8 GHz 2. RAM – 4 GB 3. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)
2	Internet

6. Precautions:

1. Check whether the LAN simulator works.
2. Ensure the knowledge of CSMA fundamentals.
3. Ensure that there are no power fluctuations while executing the program.
4. Safe working conditions help prevent injury to people and damage to computer equipment.
5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

7. Algorithm/circuit/Diagram/Description:

1.Create a simulator object

2. Define different colors for different data flows

3. Open a nam trace file and define finish procedure then close the trace file, and execute nam on trace file.

4. Create six nodes that forms a network numbered from 0 to 5
5. Create duplex links between the nodes and add Orientation to the nodes for setting a LAN topology
6. Setup TCP Connection between n(0) and n(4)
7. Apply FTP Traffic over TCP
8. Setup UDP Connection between n(1) and n(5)
9. Apply CBR Traffic over UDP.
10. Apply CSMA/CA and CSMA/CD mechanisms and study their performance
11. Schedule events and run the program.

8. Program:

CSMA/CA:

```
set ns [new Simulator]

#Define different colors for data flows (for NAM)

$ns color 1 Blue

$ns color 2 Red

#Open the Trace files set file1 [open out.tr w]

set winfile [open WinFile w]

$ns trace-all $file1

#Open the NAM trace file set file2 [open out.nam w]

$ns namtrace-all $file2

#Define a 'finish' procedure proc finish {} {

global ns file1 file2
$ns flush-trace close $file1
close $file2
exec nam out.nam &
exit 0
}
#Create six nodes set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node] set n3 [$ns node] set n4 [$ns node] set n5 [$ns node]
$n1 color red
$n1 shape box
#Create links between the nodes
$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 $n2 0.3Mb 100ms DropTail
set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Ca
Channel] Setup a TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
```

```
set sink [new Agent/TCPSink/DelAck]

$ns attach-agent $n4 $sink

$ns connect $tcp $sink

$tcp set fid_ 1

$tcp set window_ 8000

$tcp set packetSize_ 552

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

$ftp attach-agent $tcp

$ftp set type_ FTP

#Setup a UDP connection set udp [new Agent/UDP]

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

$ns attach-agent $n5 $null

$ns connect $udp $null

$udp set fid_ 2

#Setup a CBR over UDP connection 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"
$ns at 124.0 "$ftp stop"
$ns at 124.5 "$cbr stop"
# next procedure gets two arguments: the name of the
# tcp source node, will be called here "tcp",
# and the name of output file.
proc plotWindow {tcpSource file} {
global ns
set time 0.1
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
set wnd [$tcpSource set window_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file" }
$ns at 0.1 "plotWindow $tcp $winfile"
```

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

# PPP

$ns at 125.0 "finish"

$ns run
```

CSMA/CD

```
 set ns [new Simulator]

#Define different colors for data flows (for NAM)

$ns color 1 Blue

$ns color 2 Red

#Open the Trace files set file1 [open out.tr w]

set winfile [open WinFile w]

$ns trace-all $file1

#Open the NAM trace file set file2 [open out.nam w]

$ns namtrace-all $file2

#Define a 'finish' procedure proc finish {} {

global ns file1 file2

$ns flush-trace close $file1 close $file2

exec nam out.nam &
exit 0

}

#Create six nodes set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

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

$n1 color red

$n1 shape box

#Create links between the nodes

$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

set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Cd
Channel] Setup a TCP connection

set tcp [new Agent/TCP/Newreno]

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink/DelAck]

$ns attach-agent $n4 $sink
```

```
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
#Setup a FTP over TCP connection set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#Setup a UDP connection set udp [new Agent/UDP]
$ns attach-agent $n1 $udp set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_ 2
#Setup a CBR over UDP connection 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"

$ns at 124.0 "$ftp stop"

$ns at 124.5 "$cbr stop"

# next procedure gets two arguments: the name of the

# tcp source node, will be called here "tcp",

# and the name of output file.

proc plotWindow {tcpSource file} {

global ns

set time 0.1

set now [$ns now]

set cwnd [$tcpSource set cwnd_]

set wnd [$tcpSource set window_]

puts $file "$now $cwnd"

$ns at [expr $now+$time] "plotWindow $tcpSource $file" }

$ns at 0.1 "plotWindow $tcp $winfile"

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

# PPP
```

```
$ns at 125.0 "finish"

$ns run
```

9. Test Cases:

Case1: CSMA –Normal process

Case 2: CSMA-CD

10.Practical Related Questions:

1. What happens if CSMA buffer size reaches to Maximum length i.e 65535 bytes?
2. Compare and contrast CSMA/CD with CSMA/CA with characerstics?
3. If N=10, How do you analyze how many Collisions are detected when CSMA-CD was deployed?

PRACTICAL 5: Simulate Network Topology STAR, BUS, RING

1. Practical significance:

Network Simulator (Version 2), widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors. Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community since its birth in 1989. Ever since, several revolutions and revisions have marked the growing maturity of the tool, thanks to substantial contributions from the players in the field. Among these are the University of California and Cornell University who developed the REAL network simulator,1 the foundation which NS is based on. Since 1995 the Defense Advanced Research Projects Agency (DARPA) supported development of NS through the Virtual Inter Network Testbed (VINT) project . Currently the National Science Foundation (NSF) has joined the ride in development. Last but not the least, the group of Researchers and developers in the community are constantly working to keep NS2 strong and versatile.

2. Relevant Program Outcomes: PO1, PO2, PO3, PO4

3. Competency and practical skills :

The practical is expected to develop the following skills

1. Ability to work on Network Simulators.
2. Familiarity with the Network Simulators and its usage

4. Prerequisites:

1. Student should have knowledge on Computer Networks

5. Resources required:

S.No	Name of the Resource Broad Specification(Approx.)
1	Computer System 1. Processor – 2GHz 2. RAM – 4GB 3. Hard-Drive Space – 20GB 4. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)
2	NS-2 (Network Simulator)

6. Precautions:

1. Check Whether the computer is getting proper power or not.
2. Ensure the keyboard, mouse and monitor are properly working.
3. Ensure that there are no power fluctuations while executing the program.
4. Safe working conditions help prevent injury to people and damage to computer equipment.
5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.
6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7. **Algorithm/circuit/Diagram/Description:**

BUS TOPOLOGY THEORY:

Token bus is a LAN protocol operating in the MAC layer. Token bus is standardized as per IEEE 802.4. Token bus can operate at speeds of 5Mbps, 10 Mbps and 20 Mbps. The operation of token bus is as follows: Unlike token ring in token bus the ring topology is virtually created and maintained by the protocol. A node can receive data even if it is not part of the virtual ring, a node joins the virtual ring only if it has data to transmit. In token bus data is transmitted to the destination node only where as other control frames is hop to hop. After each data transmission there is a solicit successor control frame transmitted which reduces the performance of the protocol.

ALGORITHM:

1. Create a simulator object
2. Define different colors for different data flows
3. Open a nam trace file and define finish procedure then close the trace file, and execute nam on trace file.
4. Create five nodes that forms a network numbered from 0 to 4
5. Create duplex links between the nodes and add Orientation to the nodes for setting a LAN topology
6. Setup TCP Connection between n(1) and n(3)
7. Apply CBR Traffic over TCP.
8. Schedule events and run the program.

PROGRAM:

```
#Create a simulator object
set ns [new Simulator]
```

```
#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
   global ns nf
   $ns flush-trace
   #Close the trace file
   close $nf
   #Executenam on the trace file
   exec nam out.nam &
   exit 0
}
#Create five nodes set
n0 [$ns node] set n1
[$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Create Lan between the nodes
set lan0 [$ns newLan "$n0 $n1 $n2 $n3 $n4" 0.5Mb 40ms LL Queue/DropTail
MAC/Csma/Cd Channel]
#Create a TCP agent and attach it to node n0
set tcp0 [new Agent/TCP]
$tcp0 set class_ 1
$ns attach-agent $n1 $tcp0
#Create a TCP Sink agent (a traffic sink) for TCP and attach it to
node n3 set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
#Connect the traffic sources with the traffic sink
$ns connect $tcp0 $sink0

# Create a CBR traffic source and attach it to tcp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.01
$cbr0 attach-agent $tcp0

#Schedule events for the CBR agents
$ns at 0.5 "$cbr0 start"
```

```
$ns at 4.5 "$cbr0 stop"


#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Run the simulation
$ns run
```

RING TOPOLOGY THEORY:

Token ring is a LAN protocol operating in the MAC layer. Token ring is standardized as per IEEE 802.5. Token ring can operate at speeds of 4mbps and 16 mbps. The operation of token ring is as follows: When there is no traffic on the network a simple 3-byte token circulates the ring. If the token is free (no reserved by a station of higher priority as explained later) then the station may seize the token and start sending the data frame. As the frame travels around the ring ach station examines the destination address and is either forwarded (if the recipient is another node) or copied. After copying4 bits of the last byte is changed. This packet then continues around the ring till it reaches the originating station. After the frame makes a round trip the sender receives the frame and releases a new token onto the ring.

ALGORITHM:

1. Create a simulator object
2. Define different colors for different data flows
3. Open a nam trace file and define finish procedure then close the trace file, and execute nam on trace file.
4. Create five nodes that forms a network numbered from 0 to 4
5. Create duplex links between the nodes to form a Ring Topology.
6. Setup TCP Connection between n(1) and n(3)
7. Apply CBR Traffic over TCP
8. Schedule events and run the program.

PROGRAM:

```
#Create a simulator object
set ns [new Simulator]

#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
```

```
#Define a 'finish' procedure
proc finish {} {
  global ns nf
  $ns flush-trace
  #Close the trace file
  close $nf
  #Executenam on the trace file
  exec nam out.nam &
  exit0
}

#Create five nodes set
n0 [$ns node] set n1
[$ns node] set n2 [$ns
node] set n3 [$ns
node] set n4 [$ns
node] set n5 [$ns
node]
#Create links between the nodes
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$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 $n0 1Mb 10ms DropTail

#Create a TCP agent and attach it to node n0
set tcp0 [new Agent/TCP]
$tcp0 set class_ 1
$ns attach-agent $n1 $tcp0
#Create a TCP Sink agent (a traffic sink) for TCP and attach it to
node n3 set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
#Connect the traffic sources with the traffic sink
$ns connect $tcp0 $sink0


# Create a CBR traffic source and attach it to tcp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.01
$cbr0 attach-agent $tcp0
```

```
#Schedule events for the CBR agents
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"


#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Run the simulation
$ns run
```

STAR TOPOLOGY THEORY:

Star networks are one of the most common computer network topologies. In its simplest form, a star network consists of one central switch, hub or computer, which acts as a conduit to transmit messages. This consists of a
central node, to which all other nodes are connected; this central node provides a common connection point for
all nodes through a hub. In star topology, every node (computer workstation or any other peripheral) is connected to a central node called a hub or switch. The switch is the server and the peripherals are the clients. Thus, the hub and leaf nodes, and the transmission lines between them, form a graph with the topology of a star. If the central node is passive, the originating node must be able to tolerate the reception of an echo of its own transmission, delayed by the two-way transmission time (i.e. to and from the central node) plus any delay generated in the central node. An active star network has an active central node that usually has the means to prevent echo-related problems.

The star topology reduces the damage caused by line failure by connecting all of the systems to a central node. When applied to a bus-based network, this central hub rebroadcasts all transmissions received from any peripheral node to all peripheral nodes on the network, sometimes including the originating node. All peripheral nodes may thus communicate with all others by transmitting to, and receiving from, the central node only. The failure of a transmission line linking any peripheral node to the central node will result in the isolation of that peripheral node from all others, but the rest of the systems will be unaffected.

ALGORITHM:

1. Create a simulator object
2. Define different colors for different data flows
3. Open a nam trace file and define finish procedure then close the trace file, and execute nam on trace file.

4. Create six nodes that forms a network numbered from 0 to 5
5. Create duplex links between the nodes to form a STAR Topology
6. Setup TCP Connection between n(1) and n(3)
7. Apply CBR Traffic over TCP
8. Schedule events and run the program.

PROGRAM:

```
    #Create a simulator object
    set ns [new Simulator]

    #Open the nam trace file
    set nf [open out.nam w]
    $ns namtrace-all $nf

    #Define a 'finish' procedure
    proc finish {} {
      global ns nf
      $ns flush-trace
      #Close the trace file
      close $nf
      #Executenam on the trace file
      exec nam out.nam &
      exit0
}
 #Create six nodes set n0 [$ns
 node] set n1 [$ns node] set n2
 [$ns node] set n3 [$ns node] set
 n4 [$ns node] set n5 [$ns node]

 #Change the shape of center node in a star topology
 $n0 shape square

 #Create links between the nodes
 $ns duplex-link $n0 $n1 1Mb 10ms DropTail
 $ns duplex-link $n0 $n2 1Mb 10ms DropTail
 $ns duplex-link $n0 $n3 1Mb 10ms DropTail
 $ns duplex-link $n0 $n4 1Mb 10ms DropTail
 $ns duplex-link $n0 $n5 1Mb 10ms DropTail

 #Create a TCP agent and attach it to node n0 set
 tcp0 [new Agent/TCP]
 $tcp0 set class_ 1
```

```
$ns attach-agent $n1 $tcp0
#Create a TCP Sink agent (a traffic sink) for TCP and attach it to
node n3 set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
#Connect the traffic sources with the traffic sink
$ns connect $tcp0 $sink0
# Create a CBR traffic source and attach it to tcp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.01
$cbr0 attach-agent $tcp0
#Schedule events for the CBR agents
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

8. Test cases:

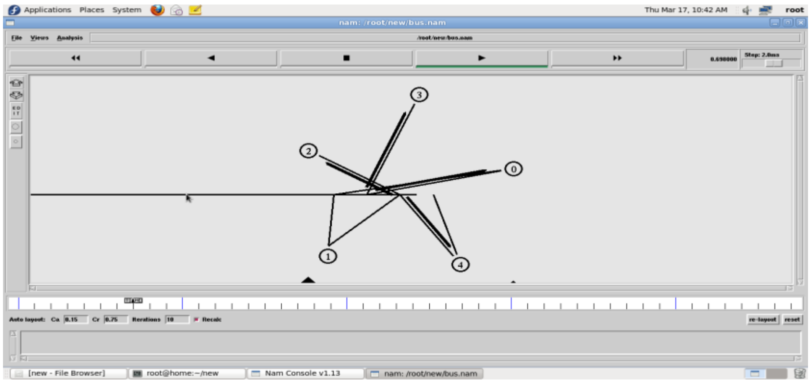
1. The Student should check the different variations in the nodes by applying different values.

2. Outputs with different inputs must be given and appropriate results must be recorded after execution.

3. Students must also record the errors while executing the commands, so that they get deeper insights on how a command should be executed.

9. Sample output:

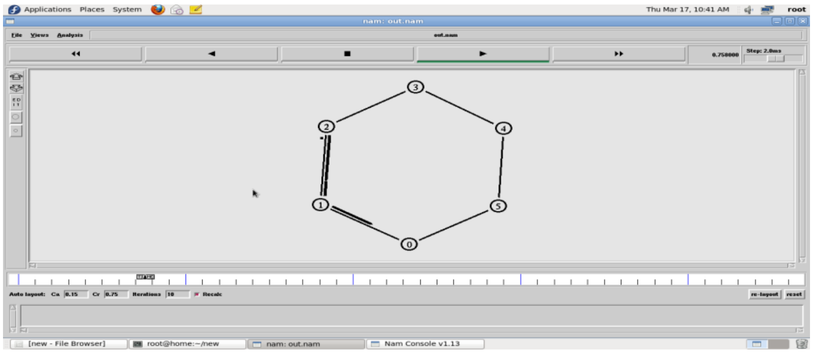
BUS TOPOLOGY OUTPUT:



Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc

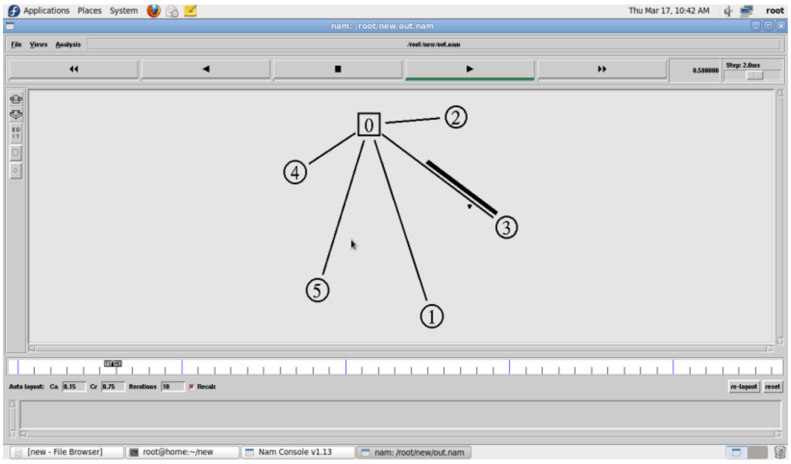
Thus the Bus Topology was Simulated and studied.

RING TOPOLOGY OUTPUT:



Thus the Ring Topology was simulated and studied.

STAR TOPOLOGY OUTPUT:



Thus the Star Topology was simulated and studied.

10.Practical Related Questions:

1.What are the Different topologies available in networks?
2.Which topology requires multipoint conection?
3.Data communication system with in a campus is called as?
4.What is meant by WAN?
5.Explain the working of Ring topology?

PRACTICAL 6: Study of Socket Programming and Client-Server Model.

1.Practical significance:

To implement socket programming date and time display from client to server using TCP Sockets.

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.Socket types define the communication properties visible to a user. The Internet family sockets provide access to the TCP/IP transport protocols. Datagram sockets allow processes to use UDP to communicate. ... A datagram socket supports bidirectional flow of messages.

2. Relevant Program Outcomes: PO1, PO2, PO3, PO4

3. Competency and practical skills :

The practical is expected to develop the following skills

1. Ability to work on Network Simulators.
2. Familiarity with the Network Simulators and its usage

4. Prerequisites:

1. Student should have knowledge on Computer Networks

5. Resources required:

S.No	Name of the Resource Broad Specification(Approx.)
1	Computer System 1. Processor – 2GHz 2. RAM – 4GB 3. Hard-Drive Space – 20GB 4. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)

2	NS-2 (Network Simulator)

6. Precautions:

1. Check Whether the computer is getting proper power or not.
2. Ensure the keyboard, mouse and monitor are properly working.
3. Ensure that there are no power fluctuations while executing the program.
4. Safe working conditions help prevent injury to people and damage to computer equipment.
5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.
6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7. Algorithm/circuit/Diagram/Description:

ALGORITHM:

Server

1. Create a server socket and bind it to port.
2. Listen for new connection and when a connection arrives, accept it.
3. Send server‟s date and time to the client.
4. Read client‟s IP address sent by the client.
5. Display the client details.
6. Repeat steps 2-5 until the server is terminated.
7. Close all streams.
8. Close the server socket.
9. Stop.

Client

1. Create a client socket and connect it to the server‟s port number.
2. Retrieve its own IP address using built-in function.
3. Send its address to the server.
4. Display the date & time sent by the server.
5. Close the input and output streams.
6. Close the client socket.
7. Stop.

8. Test cases:

1. The Student should check the different variations in the nodes by applying different values.
2. Outputs with different inputs must be given and appropriate results must be recorded after execution.
3. Students must also record the errors while executing the commands, so that they get deeper insights on how a command should be executed.

9. Sample output:

Server:

```
$ javac tcpdateserver.java $
java tcpdateserver
Press Ctrl+C to quit Client System/IP address is : localhost.localdomain/127.0.0.1 Client
System/IP address is : localhost.localdomain/127.0.0.1
```

Client:

```
$ javac tcpdateclient.java $ java tcpdateclient The date/time on server is: Wed Jul 06 07:12:03 GMT 2011
```

Every time when a client connects to the server, server‟s date/time will be returned to the client for synchronization.

Thus the program for implementing to display date and time from client to server using TCP Sockets was executed successfully and output verified using various samples

10.Practical Related Questions:

1.What are the Different topologies available in networks?
2.Which topology requires multipoint conection?
3.Data communication system with in a campus is called as?
4.What is meant by WAN?
5.Explain the working of Ring topology?

PRACTICAL 7: Write a code simulating ARP /RARP protocols.

Practical significance:

Address Resolution Protocol (ARP) is a low-level network protocol for translating network layer addresses into link layer addresses. ARP lies between layers 2 and 3 of the OSI model, although ARP was not included in the OSI framework and allows computers to introduce each other across a network prior to communication. Because protocols are basic network communication units, address resolution is dependent on protocols such as ARP, which is the only reliable method of handling required tasks.

The Address Resolution Protocol (ARP) is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address,

When configuring a new network computer, each system is assigned an Internet Protocol (IP) address for primary identification and communication. A computer also has a unique media access control (MAC) address identity. Manufacturers embed the MAC address in the local area network (LAN) card. The MAC address is also known as the computer's physical address.

Address Resolution Protocol (ARP) is used to resolve an IPv4 address (32 bit Logical Address) to the physical address (48 bit MAC Address). Network Applications at the Application Layer use IPv4 Address to communicate with another device.

Reverse Address Resolution Protocol (RARP) is a network protocol used to resolve a data link layer address to the corresponding network layer address. For example, RARP is used to resolve a Ethernet MAC address to an IP address.

The client broadcasts a RARP packet with an ethernet broadcast address, and it's own physical address in the data portion. The server responds by telling the client it's IP address. Note there is no name sent. Also note there is no security.

Media Access Control (MAC) addresses need to be individually configured on the servers by an administrator. RARP is limited to serving only IP addresses.Reverse ARP differs from the Inverse Address Resolution Protocol which is designed to obtain the IP address associated with a local Frame Relay data link connection identifier. InARP is not used in Ethernet.

2. Relevant Program Outcomes: PO 1, PO 2, PO 3, PO 4

3. Competency and practical skills :

The practical is expected to develop the following skills

1. Ability to work on Java Programming.

4. Prerequisites:

1. Student should have knowledge on Computer Networks

5. Resources required:

S.No	Name of the Resource Broad Specification(Approx.)
1	Computer System 1. Processor – 2GHz 2. RAM – 4GB 3. Hard-Drive Space – 20GB 4. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)
2	NS-2 (Network Simulator) JDK(JAVA DEVELOPMENT KIT)

6. Precautions:

1. Check Whether the computer is getting proper power or not.
2. Ensure the keyboard, mouse and monitor are properly working.
3. Ensure that there are no power fluctuations while executing the program.
4. Safe working conditions help prevent injury to people and damage to computer equipment.
5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.
6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7. Algorithm/circuit/Diagram/Description:
Algorithm for ARP:

Client

1. Start the program
2. Create socket and establish connection with the server.
3. Get the IP address to be converted into MAC address from the user.
4. Send this IP address to server.
5. Receive the MAC address for the IP address from the server.
6. Display the received MAC address
7. Terminate the connection

Server

1. Start the program
2. Create the socket, bind the socket created with IP address and port number and make it a listening socket.

3. Accept the connection request when it is requested by the client.
4. Server maintains the table in which IP and corresponding MAC addresses are stored.
5. Receive the IP address sent by the client.
6. Retrieve the corresponding MAC address for the IP address and send it to the client.
7. Close the connection with the client and now the server becomes a listening server waiting for the connection request from other clients
8. Stop

Algorithm for RARP:

Client

1. Start the program
2. Create datagram socket
3. Get the MAC address to be converted into IP address from the user.
4. Send this MAC address to server using UDP datagram.
5. Receive the datagram from the server and display the corresponding IP address.
6. Stop

Server

1. Start the program.
2. Server maintains the table in which IP and corresponding MAC addresses are stored.
3. Create the datagram socket
4. Receive the datagram sent by the client and read the MAC address sent.
5. Retrieve the IP address for the received MAC address from the table.
6. Display the corresponding IP address.
7. Stop

8. Test cases:

1. Outputs with different inputs must be given and appropriate results must be recorded after execution.
2. Students must also record the errors while executing the commands, so that they get deeper insights on how a command should be executed.

9. Sample output:

Sample Output for ARP:

```
E:\networks>java Serverarp
E:\networks>java Clientarp
Enter the Logical address(IP):
165.165.80.80
The Physical Address is: 6A:08:AA:C2
```

Sample Output for RARP:

```
I:\ex>java Serverrarp12
I:\ex>java Clientrarp12
 Enter the Physical address (MAC):
6A:08:AA:C2
The Logical Address is(IP): 165.165.80.80
```

10.Practical Related Questions:

1.What is the main purpose of ARP?
2. What do you mean by MTU (Maximum transfer Unit)?
3. What is the multiplexing
4. What does TTL shows?
5. What layer is ARP?
6. What is the function of RARP?
7. Difference between ARP and RARP.

PRACTICAL 08: Write a code simulating PING and TRACEROUTE commands

1. Practical significance:

The main difference between Ping and Traceroute is that Ping is a quick and easy utility to tell if the specified server is reachable and how long will it take to send and receive data from the server whereas Traceroute finds the exact route taken to reach the server and time taken by each step (hop).

2. Relevant Program Outcomes: PO1, PO2, PO3

3. Competency and practical skills :

The practical is expected to develop the following skills

- Ability to study and understand ping and traceroute command.

4. Prerequisites:

1. Student should have knowledge on network layer
2. Student should have knowledge on Application layer and its linkage with Network layer.

5. Resources required:

S.No	Name of the Resource Broad Specification(Approx.)
1	Computer System 1. Processor – 2GHz 2. RAM – 4GB 3. Hard-Drive Space – 20GB 4. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)
2	NS-2 (Network Simulator) JDK(JAVA DEVELOPMENT KIT)

6. Precautions:

1. Check whether the LAN works.

2. Ensure the knowledge of CN – traffic analysis fundamentals.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

7. Algorithm/circuit/Diagram/Description:

```
Step 1: start the program.
Step 2: Include necessary package in java.
Step 3: To create a process object p to implement the ping command.
Step 4: declare one BufferedReader stream class object.
Step 5: Get thedetails of the server
    5.1: length of the IP address.
    5.2: time required to get the details.
    5.3: send packets , receive packets and lost packets.
    5.4: minimum ,maximum and average times.
Step 6: print the results.
Step 7:Stop the program.
```

8. Program:

```
import java.io.*;
import java.net.*;
class pingserver
{
public static void main(String args[])
{
try
{
String str;
System.out.print(" Enter the IP Address to be Ping : ");
BufferedReader buf1=new BufferedReader(new
InputStreamReader(System.in));
String ip=buf1.readLine();
Runtime H=Runtime.getRuntime();
Process p=H.exec("ping " + ip);
InputStream in=p.getInputStream();
BufferedReader buf2=new BufferedReader(new
InputStreamReader(in));
while((str=buf2.readLine())!=null)
{
System.out.println(" " + str);
}
}
catch(Exception e)
{
System.out.println(e.getMessage());
}
}
}
```

9. Test Cases:

```
Enter the IP address to the ping:192.168.0.1

Pinging 192.168.0.1: with bytes of data =32

Reply from 192.168.0.11:bytes=32 time<1ms TTL =128
Reply from 192.168.0.11:bytes=32 time<1ms TTL =128
Reply from 192.168.0.11:bytes=32 time<1ms TTL =128
Reply from 192.168.0.11:bytes=32 time<1ms TTL =128
Ping statistics for 192.168.0.1
Packets: sent=4,received=4,lost=0(0% loss),approximate round trip time in milli seconds:
Minimum=1
ms,maximum=4ms,average=2ms
```

10.Practical Related Questions:

1. What happens if ping command is not working?
2. What is the objective of PING and TRACEROOT?
3. When do you choose ping command and traceroot command?

PRACTICAL 09: Implementation of Distance vector routing algorithm

1.Practical significance:

The Distance vector algorithm is iterative, asynchronous and distributed. Distributed: It is distributed in that each node receives information from one or more of its directly attached neighbours, performs calculation and then distributes the result back to its neighbours.

2. Relevant Program Outcomes: PO1, PO2, PO3,PO5,PO9

3. Competency and practical skills :

The practical is expected to develop the following skills

- Ability to study and understand Performance analysis of DVR Routing algorithm for LAN network among nodes.

4. Prerequisites:

1. Student should have practical exposure on network layer
2. Student should have knowledge on routing among nodes for data delivery.

5. Resources required:

S.No	Name of the Resource Broad Specification(Approx.)
1	Computer System 1. Processor – 2GHz 2. RAM – 4GB 3. Hard-Drive Space – 20GB 4. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)
2	NS-2 (Network Simulator) JDK(JAVA DEVELOPMENT KIT)

6. Precautions:

1. Check whether the LAN works.

2. Ensure the knowledge of routing algorithm – traffic analysis fundamentals.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

7. Algorithm/circuit/Diagram/Description:

NS 2 :
Simulating DVMRP with NS2

Now that you have studied the algorithm upon which DVMRP is based, we will simulate it using NS2 so you can visually see how it works. Startup the NS workbench again (as in part 1) and complete the following:

1. Use the NS-2 Workbench to create a script of 6 nodes.
2. The following links should be added:

a. Node 0 Æ Node 1
b. Node 0 Æ Node 4
c. Node 1 Æ Node 2
d. Node 1 Æ Node 3
e. Node 1 Æ Node 4
f. Node 2 Æ Node 3
g. Node 3 Æ Node 4
h. Node 4 Æ Node 5
i. Node 3 Æ Node 5
j. Node 1 Æ Node 2
k. Node 1 Æ Node 3
l. Node 1 Æ Node 4

You should have something that looks like the following:
Figure 6

3. You must now activate multicast routing in your simulation. This is done by doing the following steps: Click on the "Configuration" menu and select "Routing" at the top. Click on the Multicast on/off checkbox so the window looks like the following:

2004/2005 CS 4CC3/6CC3 -- Laboratory 2 page 2-10

4. You are now ready to generate the script that the NS2 simulator will be able to run. Click on the "Script" menu and select "Generate". This will create a file called test.tcl in your current working directory.

5. Unfortunately the NS-2 Workbench is not yet equipped to do most of the lab work for multicast so you are forced to finish off the script by hand. Open up your script file (test.tcl) and modify it so it looks like the following:

```
 # Creating New Simulator
set ns [new Simulator –multicast on]
set group [Node allocaddr]
$ns color 1 red
$ns color 30 purple
$ns color 31 green
# Setting up the traces
set f [open out.tr w]
$ns trace-all $f
#*******************
```

```
#Open the NAM trace file
set file2 [open out.nam w]
$ns namtrace-all $file2
proc finish {} {
 global ns nf f file2
 $ns flush-trace
 puts "Simulation completed."
 close $f
 close $file2
 exec nam out.nam &
 exit 0
}
Laboratory 2 page 2-11
set n0 [$ns node]
 put “n0: [$n0 id]”
set n1 [$ns node]
 put “n1: [$n1 id]”
set n2 [$ns node]
 put “n2: [$n2 id]”
set n3 [$ns node]
 put “n3: [$n3 id]”
set n4 [$ns node]
 put “n4: [$n4 id]”
set n5 [$ns node]
 put “n5: [$n5 id]”
$ns duplex-link $n2 $n1 2Mb 10ms DropTail
$ns duplex-link $n1 $n0 2Mb 10ms DropTail
$ns duplex-link $n0 $n4 2Mb 10ms DropTail
$ns duplex-link $n4 $n3 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 10ms DropTail
$ns duplex-link $n3 $n5 2Mb 10ms DropTail
$ns duplex-link $n4 $n5 2Mb 10ms DropTail
$ns duplex-link $n1 $n4 2Mb 10ms DropTail
$ns duplex-link $n1 $n3 2Mb 10ms DropTail
# Configure multicast protocol
DM set CacheMissMode dvmrp
set mproto DM
# all nodes will contain multicast protocol agents;
set mrthandle [$ns mrtproto $mproto]
set udp1 [new Agent/UDP]
set udp2 [new Agent/UDP]
$ns attach-agent $n0 $udp1
$ns attach-agent $n1 $udp2
set src1 [new Application/Traffic/CBR]
$src1 attach-agent $udp1
$udp1 set dst_addr_ $group
$udp1 set dst_port_ 0
$src1 set random_ false
set src2 [new Application/Traffic/CBR]
$src2 attach-agent $udp2
$udp2 set dst_addr_ $group
$udp2 set dst_port_ 1
$src2 set random_ false
# create receiver agents
```

```
set rcvr [new Agent/LossMonitor]
# joining and leaving the group;
$ns at 0.6 "$n2 join-group $rcvr $group"
$ns at 1.3 "$n3 join-group $rcvr $group"
$ns at 1.6 "$n4 join-group $rcvr $group"
$ns at 1.9 "$n3 leave-group $rcvr $group"
2004/2005 CS 4CC3/6CC3 -- Laboratory 2 page 2-12
$ns at 2.3 "$n5 join-group $rcvr $group"
$ns at 3.5 "$n2 leave-group $rcvr $group"
$ns at 0.4 "$src1 start"
$ns at 2.0 "$src2 start"
$ns at 4.0 "finish"
$ns run
```

6. Save The file and execute the script using the following command:

8. Program:

```
#include<stdio.h>
struct node
{
    unsigned dist[20];
    unsigned from[20];
}rt[10];
int main()
{
    int costmat[20][20];
    int nodes,i,j,k,count=0;
    printf("\nEnter the number of nodes : ");
    scanf("%d",&nodes);//Enter the nodes
    printf("\nEnter the cost matrix :\n");
    for(i=0;i<nodes;i++)
    {
        for(j=0;j<nodes;j++)
        {
            scanf("%d",&costmat[i][j]);
            costmat[i][i]=0;
            rt[i].dist[j]=costmat[i][j];//initialise the distance equal to cost matrix
            rt[i].from[j]=j;
        }
    }
        do
        {
            count=0;
            for(i=0;i<nodes;i++)//We choose arbitary vertex k and we calculate the direct distance from the node i to k using the cost matrix
            //and add the distance from k to node j
            for(j=0;j<nodes;j++)
            for(k=0;k<nodes;k++)
                if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
                {//We calculate the minimum distance
                    rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
                    rt[i].from[j]=k;
                    count++;
                }
        }while(count!=0);
        for(i=0;i<nodes;i++)
        {
            printf("\n\n For router %d\n",i+1);
            for(j=0;j<nodes;j++)
            {
                printf("\t\nnode %d via %d Distance %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);
            }
        }
    printf("\n\n");
    getch();
}
/*
```

9. Test Cases:

```
Enter the number of nodes :
3
Enter the cost matrix :
0 2 7
2 0 1
7 1 0
For router 1
node 1 via 1 Distance 0
node 2 via 2 Distance 2
node 3 via 3 Distance 3
For router 2
node 1 via 1 Distance 2
node 2 via 2 Distance 0
node 3 via 3 Distance 1
For router 3
node 1 via 1 Distance 3
node 2 via 2 Distance 1
node 3 via 3 Distance 0
*/
```

10.Practical Related Questions:

1. What happens Distance Vector routing working properly?
2. What is the objective of Distance vector routing strategy?
3. When do you choose Distance vector routing ?

PRACTICAL 10: Implementation of Link state routing algorithm

1. Practical significance:

In link state routing, each router shares its knowledge of its neighbourhood with every other router in the internet work. (i) Knowledge about Neighbourhood: Instead of sending its entire routing table a router sends info about its neighbourhood only. (ii) To all Routers: each router sends this information to every other router on the internet work not just to its neighbor .It does so by a process called flooding. (iii)Information sharing when there is a change: Each router sends out information about the neighbors when there is change.

The Dijkstra algorithm follows four steps to discover what is called the shortest path tree(routing table) for each router:The algorithm begins to build the tree by identifying its roots. The root router's trees the router itself. The algorithm then attaches all nodes that can be reached from the root. The algorithm compares the tree's temporary arcs and identifies the arc with the lowest cumulative cost. This arc and the node to which it connects are now a permanent part of the shortest path tree. The algorithm examines the database and identifies every node that can be reached from its chosen node. These nodes and their arcs are added temporarily to the tree.
The last two steps are repeated until every node in the network has become a permanent part of the tree.

2. Relevant Program Outcomes: PO1, PO2, PO3

3. Competency and practical skills :

The practical is expected to develop the following skills

- Ability to study and understand Link state routing process and its traffic analysis.

4. Prerequisites:

1. Student should have knowledge on NS2 programming i.e TCL
2. Student should have knowledge on Routing on Different topologies through which link state process operates.

5. Resources required:

S.No	Name of the Resource Broad Specification(Approx.)
1	Computer System 1. Processor – 2GHz 2. RAM – 4GB 3. Hard-Drive Space – 20GB 4. VGA with 1024×768 screen resolution (exact hardware requirement will depend upon the distribution that we choose to work with)
2	NS-2 (Network Simulator) JDK(JAVA DEVELOPMENT KIT)

6. Precautions:

1. Check whether the LAN works.
2. Ensure the knowledge of CN – traffic analysis fundamentals.
3. Ensure that there are no power fluctuations while executing the program.
4. Safe working conditions help prevent injury to people and damage to computer equipment.
5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

7. Algorithm/circuit/Diagram/Description:

1. Create a simulator object
2. Define different colors for different data flows
3. Open a nam trace file and define finish procedure then close the trace file, and execute nam on trace file.
4. Create n number of nodes using for loop
5. Create duplex links between the nodes
6. Setup UDP Connection between n(0) and n(5)
7. Setup another UDP connection between n(1) and n(5)
8. Apply CBR Traffic over both UDP connections
9. Choose Link state routing protocol to transmit data from sender to receiver.
10. Schedule events and run the program.

8. Program:

```
set ns [new Simulator]
set nr [open thro.tr w]
$ns trace-all $nr
set nf [open thro.nam w]
```

```
$ns namtrace-all $nf proc finish { } { global ns nr nf
$ns flush-trace close $nf
close $nr
exec nam thro.nam &
exit 0
}
for { set i 0 } { $i < 12} { incr i 1 } {
set n($i) [$ns node]}

for {set i 0} {$i < 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail  }

$ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms DropTail



set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set  interval_ 0.005
$cbr0 attach-agent $udp0 set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0

set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set  interval_ 0.005
$cbr1 attach-agent $udp1 set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp1 $null0

$ns rtproto LS
$ns rtmodel-at 10.0 down $n(11) $n(5)
$ns rtmodel-at 15.0 down $n(7) $n(6)
$ns rtmodel-at 30.0 up $n(11) $n(5)
$ns rtmodel-at 20.0 up $n(7) $n(6)

$udp0 set fid_ 1
$udp1 set fid_ 2
$ns color 1 Red
$ns color 2 Green

$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"

$ns at 45 "finish"
```

```
$ns run
```

9. Test Cases:

10.Practical Related Questions:

1. What is Routing?

2. What is Dynamic routing?

3. What are the two steps in link state routing?

4. Compare link state and Distance Vector routing

5. What are all the route metric used in Link state routing?