NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Common to CSE (AI&ML), AI&ML, CSE (Data Science)

III YEAR I SEMESTER

(COMPUTER NETWORKS-R23CC21L2)

LAB MANUAL

Narasaraopeta Engineering College

Kotappakonda Road, Yellamanda (Post), Narasaraopet-522601, Guntur District, AP

Approved by AICTE, New Delhi & Permanent Affiliation to JNTUK, Kakinada, code: 47

Accredited by NBA & NAAC, RTA Approved Pollution test Centre,

ISO 9001:2015 Certified Institution

Phone: 08647-239905, Website: www.nrtec.in

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE'S VISION, MISSION & VALUES

Vision:

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

Mission:

M1: Provide the best class infra-structure to explore the field of engineering and research.

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills.

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems.

Values:

- **Student-centric education:** Meeting the community's and student's needs by developing a world-class educational environment with cultural values.
- **Excellence**: Giving special attention towards the standards of integrity and performance to help the institute in leading academic achievements and professional goals.
- **Collaboration**: Seeking the latest input and working closely with all the industrial sectors and Society for the continuous up gradation of the quality of education.
- **Diversity**: Creating a favourable on-campus environment in which the goals and learning styles of all students are recognised and nurtured.
- **Continuous** Development: Encouraging enthusiastic, innovative thinkers and learners to strive for personal growth in the world of inventions and start-ups.
- **Technological Advancement**: Keeping pace with evolving technology and professional trends to prepare all its students to achieve success in the workplace.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Department of CSE (Artificial Intelligence & Machine Learning)

Vision:

To empower students to become AI and ML professionals, driving industry innovation and positively impacting society through cutting-edge technologies.

Mission:

M1: To establish a solid foundation in Artificial Intelligence and Machine Learning and effectively address real-world challenges.

M2: To develop resilient professionals by encouraging them to create applications for industry innovation.

M3: To cultivate a research-oriented mindset in students, encouraging them to create applications that have practical value and make a positive impact on society.

PROGRAM SPECIFIC OBJECTIVES (PSOs)

PSO1: Apply a range of AI and ML techniques to analyze and solve real-world problems effectively.

PSO2: Design and develop AI systems by integrating appropriate algorithms, models, and technologies to address specific problem domains.

PSO3: Develop the ethical implications and societal impact of AI and ML technologies.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates possess a strong knowledge of Artificial Intelligence and Machine Learning principles, enabling them to effectively address real-world challenges.

PEO2: Graduates continuously enhance their skills and adapt to the evolving field of AI and ML, empowering them to confidently face future challenges.

PEO3: Graduates apply their AI and ML knowledge to develop practical applications that positively impact society and address societal challenges.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Department of Artificial Intelligence & Machine Learning

Vision:

To empower students to become AI and ML professionals, driving industry innovation and positively impacting society through cutting-edge technologies.

Mission:

M1: To establish a solid foundation in Artificial Intelligence and Machine Learning and effectively address real-world challenges.

M2: To develop resilient professionals by encouraging them to create applications for industry innovation.

M3: To cultivate a research-oriented mindset in students, encouraging them to create applications that have practical value and make a positive impact on society.

PROGRAM SPECIFIC OBJECTIVES (PSOs)

PSO1: Apply a range of AI and ML techniques to analyze and solve real-world problems effectively.

PSO2: Design and develop AI systems by integrating appropriate algorithms, models, and technologies to address specific problem domains.

PSO3: Develop the ethical implications and societal impact of AI and ML technologies.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates possess a strong knowledge of Artificial Intelligence and Machine Learning principles, enabling them to effectively address real-world challenges.

PEO2: Graduates continuously enhance their skills and adapt to the evolving field of AI and ML, empowering them to confidently face future challenges.

PEO3: Graduates apply their AI and ML knowledge to develop practical applications that positively impact society and address societal challenges.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Department of CSE (Data Science)

Vision:

To nurture skilled professionals in the Data Science for industry innovation and create meaningful societal impact through advanced analytics, machine learning, and impactful data-driven solutions.

Mission:

M1: To develop skilled data scientists who can effectively solve challenges in data analytics through comprehensive education and practical training in statistical analysis, machine learning, data visualization, and data manipulation.

M2: To develop students with strong research capabilities who can revolutionize multiple fields through the application of data science.

M3: To develop ethical data science professionals who utilize data for the welfare of society.

PROGRAM SPECIFIC OBJECTIVES (PSOs)

PSO1: Apply Data Science Techniques, statistical analysis, machine learning algorithms, data visualization, and data manipulation effectively to solve complex data problems.

PSO2: Demonstrate proficiency in conducting data collection, pre-processing, analysis, and interpretation, contributing to the advancement of the field.

PSO3: Able to independently carry out research and investigation to solve societal problems.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates be proficient in applying advanced analytics techniques to solve complex data challenges.

PEO2: Graduates contribute to the advancement of the field through the development of innovative methodologies, algorithms, and models.

PEO3: Graduates utilize data science principles to create meaningful societal impact. They will prioritize ethical considerations in data usage and develop solutions that address societal challenges, and benefit individuals and communities.

NARASARAOPETA
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

PROGRAM OUTCOMES (POs)

P01	**Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
P02	**Problem Analysis**: Identify, formulate, review research literature and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
P03	**Design/development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural, societal and environmental considerations.
P04	**Conduct investigations of complex problems**: Use research based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.
P05	**Modern tools usage**: Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
P06	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
P07	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work**: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broad test context of technological change.

NARASARAOPETA
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

GENERAL LABORATORY INSTRUCTIONS

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with:
    a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session.
    b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab.
    c. Proper Dress code and Identity card.
4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain the discipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions.Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out ; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

SYLLABUS

List of Experiments:

1. Study of Network devices in detail and connect the computers in Local Area Network.
2. Write a Program to implement the data link layer farming methods such as
   i) Character stuffing ii) bit stuffing.
3. Write a Program to implement data link layer farming method checksum.
4. Write a program for Hamming Code generation for error detection and correction.
5. Write a Program to implement on a data set of characters the three CRC polynomials –CRC 12, CRC 16and CRC CCIP.
6. Write a Program to implement Sliding window protocol for Goback N.
7. Write a Program to implement Sliding window protocol for Selective repeat.
8. Write a Program to implement Stop and Wait Protocol.
9. Write a program for congestion control using leaky bucket algorithm
10. Write a Program to implement Dijkstra‘s algorithm to compute the Shortest path througha graph.
11. Write a Program to implement Distance vector routing algorithm by obtaining routing table at eachnode (Take an example subnet graph with weights indicating delay between nodes).
12. Write a Program to implement Broadcast tree by taking subnet of hosts.
13. Wireshark
    i) Packet Capture Using Wire shark
    ii) Starting Wire shark using command line
    iii) Viewing Captured Traffic
    iv) Analysis and Statistics & Filters.
14. How to run Nmap scan
15. Operating System Detection using Nmap
16. Do the following using NS2 Simulator
    i) NS2 Simulator-Introduction
    ii) Simulate to Find the Number of Packets Dropped
    iii) Simulate to Find the Number of Packets Dropped by TCP/UDP
    iv) Simulate to Find the Number of Packets Dropped due to Congestion
    v) Simulate to Compare Data Rate& Throughput.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

COURSE OUTCOMES

CO	OUTCOMES
1	**Study** the Network devices, physical layer, data link layer and network layer algorithms
2	**Implemen**t the various computer networks algorithms in any programming language
3	**Implement** the packet capture, captured traffic and perform analysis using Wireshark
4	**Study** and detect the operating system using Nmap
5	**Find** the number of packets dropped in various scenarios using NS2/NS3 Simulator

CO-PO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	-	2	-	-	-	-	-	-	-	2	-	-
CO2	2	1	2	-	-	-	-	-	-	-	-	1	-	-
CO3	1	1	-	-	-	-	-	-	-	-	-	-	1	-
CO4	1	1	-	-	-	-	-	-	-	-	-	1	-	-
CO5	-	-	1	-	-	-	-	-	-	-	-	2	1	-
AVG	1.5	1	1.5	2	-	-	-	-	-	-	-	1.5	1	-

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

CO-PO MAPPING JUSTIFICATION

CO1	PO 1	2	They could just apply the knowledge acquired to understand the layers based on its function
CO1	PO 2	1	Understanding the layer functions helps the students to identify and formulate the problems based on the layer.
CO1	PO 4	2	Understanding the layer functions and understanding the network factors, helps in analyzing and interpreting the quality of networks.
CO1	PO 12	2	The student will become aware of the need for lifelong learning and the continued upgrading of technical knowledge of networks.
CO2	PO 1	2	They could just apply the knowledge acquired to implement various networkalgorithms based on its function
CO2	PO 2	1	Study of Different computer network algorithms helps the students to identify and formulate the problems based on the layer.
CO2	PO 3	2	Studies about the various network algorithms helps the students to fix up thethe network.
C02	PO 12	1	The student will become aware of the need for lifelong learning and thecontinued upgrading of technical knowledge of network algorithms.
CO3	PO 1	1	The knowledge of wireshark can be applied to solve complex engineering problems.
CO3	PO 2	1	The knowledge of packet capturing, captured traffic can be solve complex engineering problems.
CO3	PSO1	1	Student can apply wireshark and other concepts in networking for development of current technical concepts.
CO4	PO 1	1	They could just apply the knowledge acquired to detect Operating System using Nmap.
CO4	PO 2	1	The knowledge of Nmap can be solve complex engineering problems
CO4	PO 12	1	The student will become aware of the need for lifelong learning and the continued upgrading of technical knowledge Nmap.
CO5	PO 3	1	The knowledge of NS2/NS3 simulator help to analyze and design solutionsto complex problems.
CO5	PO12	2	The student will become aware of the need for lifelong learning and the continued upgrading of technical knowledge of Simulator NS2/NS3.
CO5	PS O1	1	Student can analyze the simulator in networking for development of current technical concepts.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INDEX

S.No	Name of Experiment	CO	Page No
1	Study of Network devices in detail and connect the computers in Local Area Network.	CO1	1
2	Write a Program to implement the data link layer farming methods such as i) Character stuffing ii) bit stuffing.	CO2	7
3	Write a Program to implement data link layer farming method checksum.	CO2	9
4	Write a program for Hamming Code generation for error detection and correction.	CO2	12
5	Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.	CO2	14
6	Write a Program to implement Sliding window protocol for Goback N.	CO2	17
7	Write a Program to implement Sliding window protocol for Selective repeat.	CO2	19
8	Write a Program to implement Stop and Wait Protocol.	CO2	22
9	Write a program for congestion control using leaky bucket algorithm.	CO2	27
10	Write a Program to implement Dijkstra's algorithm to compute the Shortest path through a graph.	CO2	30
11	Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).	CO2	32
12	Write a Program to implement Broadcast tree by taking subnet of hosts.	CO2	34
13	Study about Wireshark i. Packet Capture Using Wire shark ii. Starting Wire shark using command line iii. Viewing Captured Traffic iv. Analysis and Statistics & Filters.	CO3	36
14	How to run Nmap scan.	CO4	41
15	Operating System Detection using Nmap	CO4	43
16	Do the following using NS2 Simulator i. NS2 Simulator-Introduction ii. Simulate to find the number of packets Dropped iii. Simulate to find the number of packets	CO5	44

EXPERIMENT-1

AIM : Study of Network devices in detail and connect the computers in Local Area Network.

Study of Network Devices in Detail and connect the computers in Local Area Network

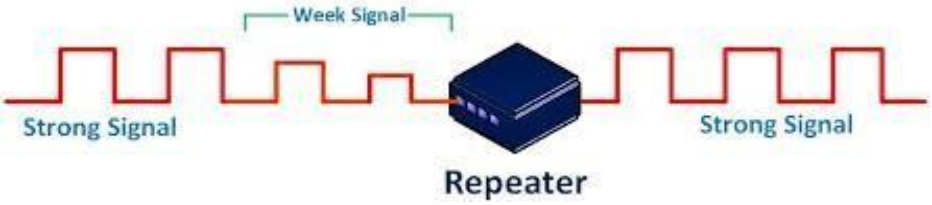
a) Study of Network Devices in Detail

- Repeater
- Hub
- Switch
- Bridge
- Router
- GateWay

Apparatus(Software): No software or hardware needed.

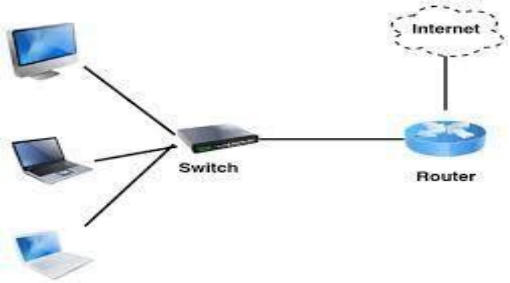
Procedure: following should be done to understand this practical.

1. **Repeater:** Functioning at Physical Layer. A **repeater** is an electronic device that receives a signal and retransmits it at a higher level and/or higher power, or onto the other side of an obstruction, so that the signal can cover longer distances. Repeater have two ports, so can not be use to connect for more than two devices.



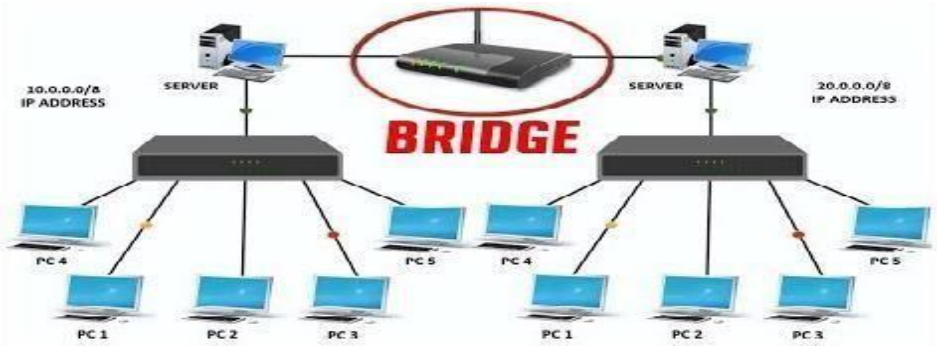
2. **Hub**: An Ethernet hub, active hub, network hub, repeater hub, hubor concentrator is a device for connecting multiple twisted pair or fiber optic Ethernet devices together and making them act as a single network segment. Hubs work at the physical layer (layer 1) of the OSI model. The device is a form of multiport repeater. Repeater hubs also participate in collision detection, forwarding a jam signal to all ports if it detects a collision.

3. **Switch:** A **network switch** or **switching hub** is a computer networking device that connects network segments. The term commonly refers to a network bridge that processes and routes data at the data link layer (layer 2) of the OSI model.Switches that additionally process data at the network layer(layer3andabove)are often referred to as Layer 3switches or multi layer switches.

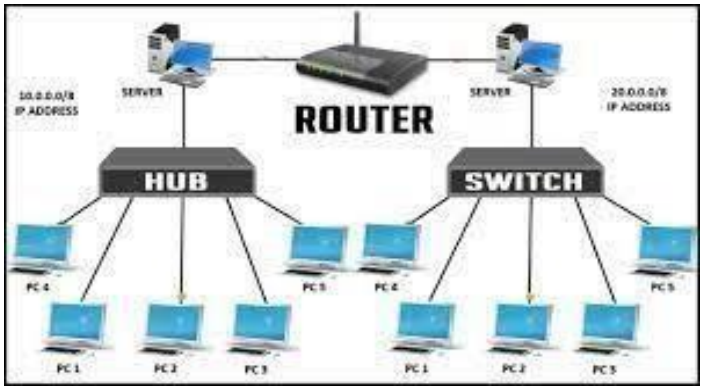


Switch

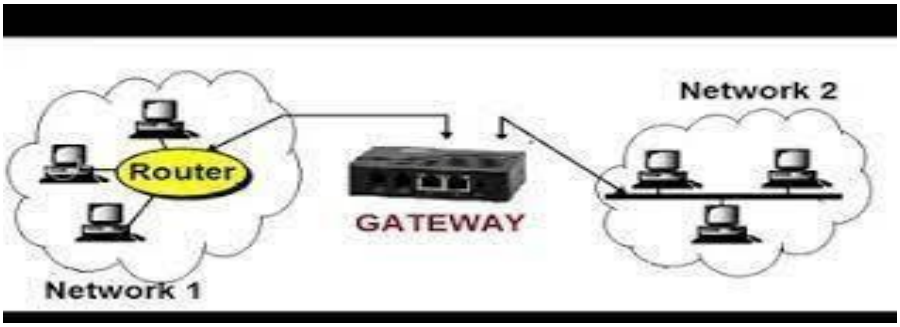
4. **Bridge:** A **network bridge** connects multiple network segments at the data link layer (Layer 2) of the OSI model. In Ethernet networks, the term *bridge* formally means a device that behaves according to the IEEE 802.1D standard. A bridge and switch are very much alike; a switch being a bridge with numerous ports. *Switch* or *Layer2 switch* is often used interchangeably with *bridge*. Bridges can analyze incoming data packets to determine if the bridge is able to send the given packet to another segment of the network.



5. **Router:** A **router** is an electronic device that interconnects two or more computer networks, and selectively interchanges packets of data between them. Each data packet contains address information that a router can use to determine if the source and destination are on the same network, or if the data packet must be transferred from one network to another. Where multiple routers are used in a large collection of interconnected networks, the routers exchange information about target system addresses, so that each router can build up a table showing the preferred paths between any two systems on the interconnected networks.



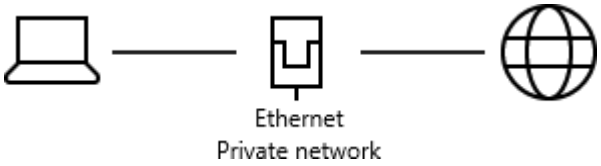
6. **Gate Way:** In a communications network, a network node equipped for interfacing with another network that uses different protocols.
    a. A gateway may contain devices such as protocol translators, impedance matc hing devices, rate converters, fault isolators, or signal translators as necessary to provide system interoperability. It also requires the establishment of mutua lly acceptable administrative procedures between both networks.



b. A protocol translation/ mapping gateway interconnects networks with different network protocol technologies by performing the required protocol conversions.

c. Aim: Connect thecomputersin LocalAreaNetwork.
Procedure: On the host computer

On the host computer, follow these steps to share the Internet connection:



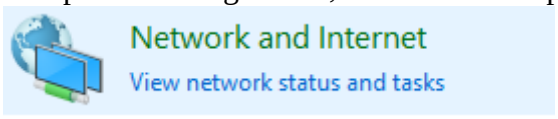
1. Log on to the host computer as Administrator or as Owner.
2. Click **Start**, and then click **Control Panel**.
3. Click Network and Internet Connections.
4. Click **Network Connections**.
5. Right-click the connection that you use to connect to the Internet. For example, if you connect to the Internet by using a modem, right-click the connection that you want under Dial-up / other network available.
6. Click **Properties**.
7. Click the **Advanced** tab.
8. Under **Internet Connection Sharing**, select the **Allow other network users to connect through this computer's Internet connection** checkbox.
9. If you are sharing a dial-up Internet connection, select the **Establish a dial- up connection whenever a computer on my network attempts to access the Internet** check box if you want to permit your computer to automatically connect to the Internet.
10. Click **OK**. You receive the following message:

When Internet Connection Sharing is enabled, your LAN adapter will be set to use IP address 192.168.0.1. Your computer may lose connectivity with other computers on your network. If these other computers have static IP addresses, it is a good idea to set them to obtain their IP addresses automatically. Are you sure you want to enable Internet Connection Sharing?

Click **Yes**.

The connection to the Internet is shared to other computers on the local area network(LAN). The network adapter that is connected to the LAN is configured with a static IP addressof192.168.0.1and a subnet maskof255.255.255.0 On the client computer to connect to the Internet by using the shared connection, you must confirm the LAN adapter IP configuration, and then configure the client computer.

To confirm the LAN adapter IP configuration, follow these steps:

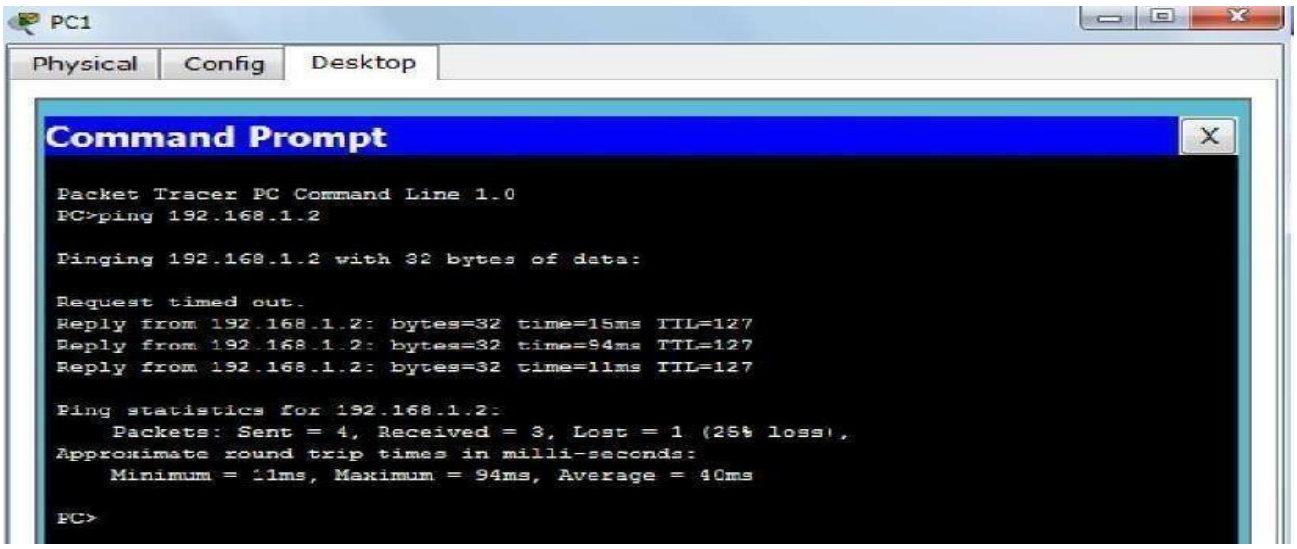


1. Log on to the client computer as Administrator or as Owner.
2. Click **Start**, and then click **Control Panel**.
3. Click Network and Internet Connections.
4. Click **Network Connections**.
5. Right-click **Local Area Connection** and then click **Properties**.
6. Click the **General** tab, click **Internet Protocol (TCP/IP)** in the **Connection uses the following items** list, and then click **Properties**.
7. In the **Internet Protocol (TCP/IP) Properties** dialog box, click **Obtain an IP address automatically**(if it is not already selected),and then click **OK**.

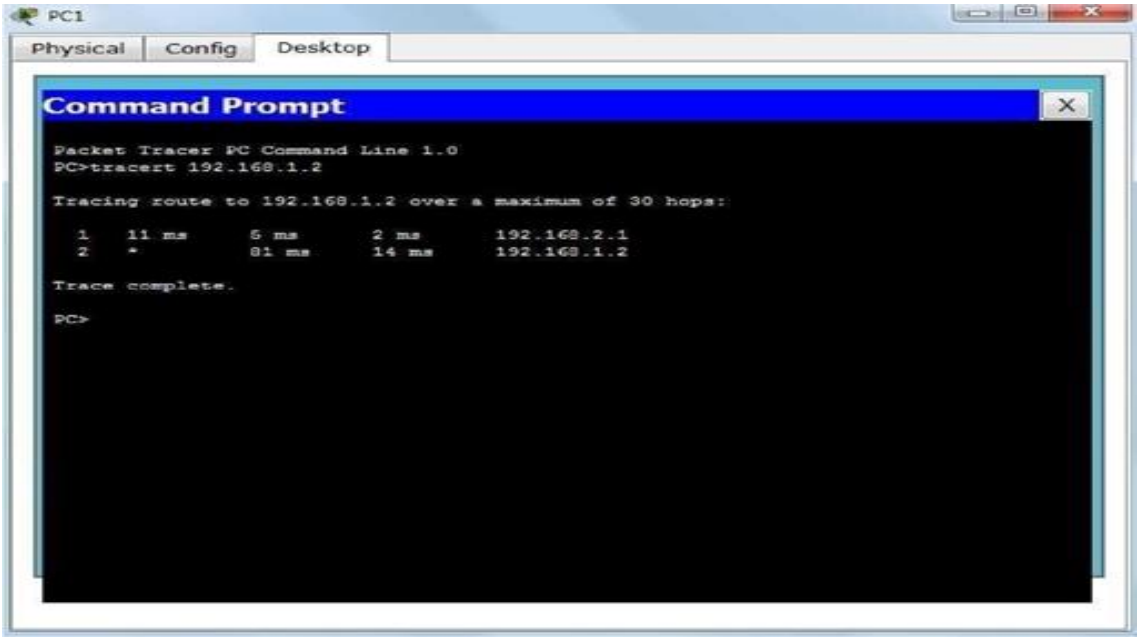
Note: You can also assign a unique static IP address in the range of 192.168.0.2 to192.168.0.254. For example, you can assign the following static IP address, subnet mask, and default gateway:

8. IPAddress192.168.31.202
9. Subnet mask255.255.255.0
10. Default gateway192.168.31.1
11. In the **Local Area Connection Properties** dialog box, click **OK**.
12. Quit Control Panel

Ping Output:



Trace out Output:



Result: the study achieved its objective of understanding and practically implementing a LAN using standard networking components**.**

Viva Questions:

1.What is the function of a router?

Ans:

2.How is a hub different from a switch?

Ans:

3.What is an IP address?

Ans:

4.Why is file sharing used in a LAN?

Ans:

EXPERIMENT-2

AIM: Write a Program to implement the data link layer farming methods such as
i) Character stuffing ii) bit stuffing.

i) Character Stuffing

Program:

```
#include<stdio.h>
#include<string.h>
#include<process..h>
void main()
{
int i=0,j=0,n,pos;
char a[20],b[50],ch;
printf("enter string\n");
scanf("%s",&a);
n=strlen(a);
printf("enter position:\n");
scanf("%d",&pos);
if(pos>n)
{
printf("invalid position");
scanf("%d",&pos);
}
//printf("enter the character\n");
//ch=getche();
b[0]='D';
b[1]='L';
b[2]='E';
b[3]='S';
b[4]='T';
b[5]='X';
j=6;
while(i<n)
{
if(i==pos-1)
{
b[j]='D';
b[j+1]='L';
b[j+2]='E';
b[j+3]='ch';
b[j+4]='D';
b[j+5]='L';
b[j+6]='E';
j=j+7;
}
if(a[i]=='d'&&a[i+1]=='l'&&a[i+2]=='e')
{
b[j]='D';
b[j+1]='L';
b[j+2]='E';
```

```
j=j+3;
}
b[j]=a[i];
i++;
j++;
b[j]='D';
b[j+1]='L';
b[j+2]='E';
b[j+3]='E';
b[j+4]='T';
b[j+5]='X';
b[j+6]='/o';
printf("\n frame after setting:\n");
printf("%s",b);
}
getch();
}
```

Output: Enter string: necnrt

Enter position 2
Enter the character frame after stuffing:
DLESTXnDLEETX
frame after stuffing:
DLESTXnDLEcDLEeDLEETX
frame after stuffing:
DLESTXnDLEcDLEecDLEETX
frame after stuffing:
DLESTXnDLEcDLEecnDLEETX
frame after stuffing:
DLESTXnDLEcDLEecnrDLEETX
frame after stuffing:
DLESTXnDLEcDLEecnrtDLEETX

ii) Bit stuffing

Program:

```
#include<stdio.h>
#include<string.h>
void main()
{
char data[50],stuff[50];
int i,j,k,count,length;
printf("Enter your data");
scanf("%s",data);
length=strlen(data);
count=0,j=0;
for(i=0;i<length;i++)
{
if(data[i]=='1')
{
count++;
}
else
{
count=0;
}
stuff[j]=data[i];
j++;
if(count==5 && data[i+1]=='1')
{
stuff[j]='0';
j++;
count=0;
}
}
stuff[j]='\0';
printf("your stuffing data is: %s",stuff);
getch();
}
```

Program Output:

```
Enter your data: 0101111101011111101011111111
Your stuffing data is: 01011111010111110101011111011
```

Viva Questions:

1.What is the main difference between bit stuffing and character stuffing?

Ans:

2.Give an example of bit stuffing.

Ans:

3.In which layer of the OSI model do stuffing techniques apply?

Ans:

EXPERIMENT-3

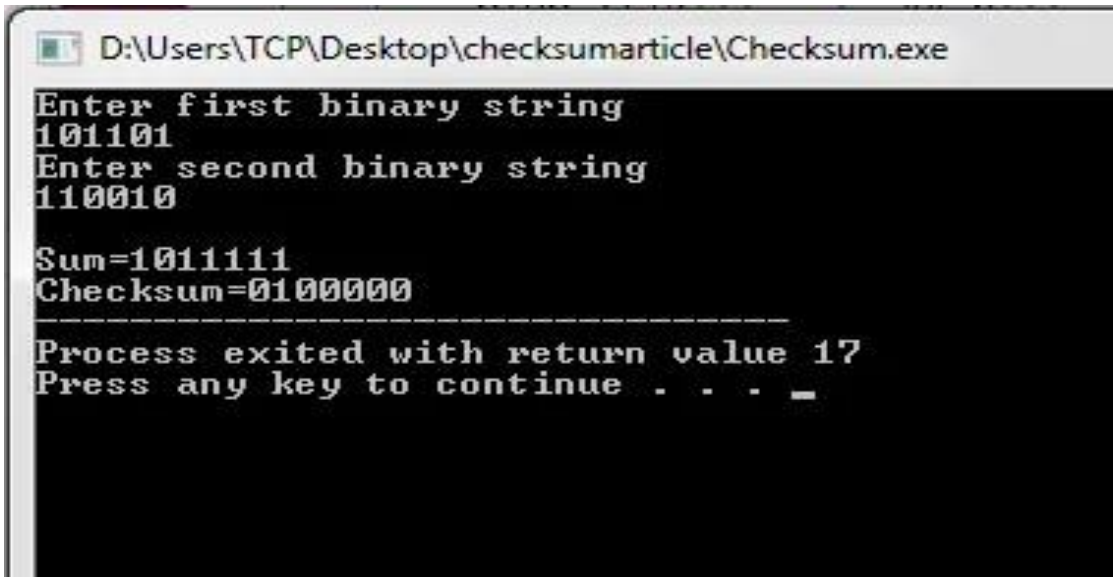
AIM: Write a Program to implement data link layer farming method checksum.

Program:

```
#include<stdio.h>
#include<string.h>

int main()
{
   char a[20],b[20];
   char sum[20],complement[20];
   int i,length;

printf("Enter first binary string\n");
scanf("%s",&a);
printf("Enter second binary string\n");
scanf("%s",&b);

   if(strlen(a)==strlen(b)){
length = strlen(a);
char carry='0';
for(i=length-1;i>=0;i--)
      {
if(a[i]=='0' && b[i]=='0' && carry=='0')
         {
            sum[i]='0';
            carry='0';
         }
         else if(a[i]=='0' && b[i]=='0' && carry=='1')
         {
            sum[i]='1';
            carry='0';
         }
         else if(a[i]=='0' && b[i]=='1' && carry=='0')
         {
            sum[i]='1';
            carry='0';
         }
         else if(a[i]=='0' && b[i]=='1' && carry=='1')
         {
            sum[i]='0';
            carry='1';

         }
         else if(a[i]=='1' && b[i]=='0' && carry=='0')
         {
            sum[i]='1';
            carry='0';
         }
```

```
        else if(a[i]=='1' && b[i]=='0' && carry=='1')
        {
           sum[i]='0';
           carry='1';
        }
        else if(a[i]=='1' && b[i]=='1' && carry=='0')
        {
           sum[i]='0';
           carry='1';
        }
        else if(a[i]=='1' && b[i]=='1' && carry=='1')
        {
           sum[i]='1';
           carry='1';
        }
        else
           break;
     }

printf("\nSum=%c%s",carry,sum);
for(i=0;i<length;i++)
     {
        if(sum[i]=='0')
           complement[i]='1';
        else
           complement[i]='0';
     }
     if(carry=='1')
        carry='0';
     else
        carry='1';

printf("\nChecksum=%c%s",carry,complement);


}
else {
printf("\nWrong input strings");
}
}
```

OUTPUT:



Viva Questions:

1. What is a checksum?

Ans:

2. How is a checksum calculated in the data link layer?

Ans:

3.What happens if the checksum verification fails at the receiver?

Ans:

EXPERIMENT-4

AIM: Write a program for Hamming Code generation for error detection and correction.

Program:

```
#include<stdio.h>
 void main() {
    int data[10];
    int dataatrec[10],c,c1,c2,c3,i;

printf("Enter 4 bits of data one by one\n");
scanf("%d",&data[0]);

scanf("%d",&data[1]);
scanf("%d",&data[2]);
scanf("%d",&data[4]);

    //Calculation of even parity
    data[6]=data[0]^data[2]^data[4];
data[5]=data[0]^data[1]^data[4];
data[3]=data[0]^data[1]^data[2];

printf("\nEncoded data is\n");
for(i=0;i<7;i++)
printf("%d",data[i]);

printf("\n\nEnter received data bits one by one\n");
    for(i=0;i<7;i++)
scanf("%d",&dataatrec[i]);

    c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
c=c3*4+c2*2+c1 ;

    if(c==0) {
printf("\nNo error while transmission of data\n");
    }
else {
printf("\nError on position %d",c);

printf("\nData sent : ");
       for(i=0;i<7;i++)
printf("%d",data[i]);

printf("\nData received : ");
```

```
        for(i=0;i<7;i++)
printf("%d",dataatrec[i]);
printf("\nCorrect message is\n");

//if errorneous bit is 0 we complement it else vice versa
if(dataatrec[7-c]==0)
dataatrec[7-c]=1;
        else
dataatrec[7-c]=0;
for (i=0;i<7;i++) {
printf("%d",dataatrec[i]);
}
}
}
```

OUTPUT:

```
Enter 4 bits of data one by one
1
0
1
0
Encoded data is
1010010
Enter received data bits one by one
1
0
1
0
0
1
0

No error while transmission of data
```

Viva Questions:

1. What is Hamming Code?

Ans:

2. What is the formula to calculate the number of redundant bits?

Ans:

3. Where are the parity bits placed in Hamming Code?

Ans:

EXPERIMENT-5

AIM: Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.

Program:

```
#include<stdio.h>
# include<string.h>
#define N strlen(g)
char t[28],cs[28],g[28];
int a,e,c,b;
void xor()
{
for(c=1;c<N;c++)
cs[c]=((cs[c]==g[c])?'0':'1');
}
void crc()
{
for(e=0;e<N;e++)
cs[e]=t[e];
do
{ if(cs[0]=='1')
xor();
for(c=0;c<N-1;c++)
cs[c]=cs[c+1];
cs[c]=t[e++];
}while(e<=a+N-1);
} int main()
{ int flag=0;
do{
printf("\n1.crc12\n2.crc16\ncrcccit\n4.exit\n\nEnter your option.");
scanf("%d",&b);
switch(b)
{ case 1:strcpy(g,"1100000001111");
break;
case 2:strcpy(g,"11000000000000101");
break;
case 3:strcpy(g,"10001000000100001");
break;
case 4:return 0;
}
printf("\n enter data:");
scanf("%s",t);
printf("\n----------------\n");
printf("\n generating polynomial:%s",g);
a=strlen(t);
```

```
for(e=a;e<a+N-1;e++)
t[e]='0';
printf("\n----------------\n");
printf("mod-ified data is:%s",t);
printf("\n----------------\n");
crc();
printf("checksum is:%s",cs);
for(e=a;e<a+N-1;e++)
t[e]=cs[e-a];
printf("\n----------------\n");
printf("\n final codeword is : %s",t);
printf("\n----------------\n");
printf("\ntest error detection 0(yes) 1(no)?:");
scanf("%d",&e);
if(e==0)
{
do{
printf("\n\tenter the position where error is to be inserted:");
scanf("%d",&e);
}
while(e==0||e>a+N-1);
t[e-1]=(t[e-1]=='0')?'1':'0';
printf("\n----------------\n");
printf("\n\terroneous data:%s\n",t);
} crc();
for(e=0;(e<N-1)&&(cs[e]!='1');e++);
if(e<N-1)
printf("error detected\n\n");
else
printf("\n no error detected \n\n");
printf("\n----------------");
}while(flag!=1);
} crc();
for(e=0;(e<N-1)&&(cs[e]!='1');e++);
if(e<N-1)
printf("error detected\n\n");
else
printf("\n no error detected \n\n");
printf("\n----------------");
}while(flag!=1);
```

OUTPUT:

```
1.crc12
2.crc16
3.crc ccit
4.exit
Enter your option.1
enter data:1100110011100011
----------
generating polynomial:1100000001111
----------
mod-ified data is:11001100111000110000000000001100000001111
----------
checksum is:1101110110001
----------
final codeword is : 11001100111000111101110110001100000001111
----------
test error detection 0(yes) 1(no)?:1
no error detected
----------
1. crc12
2.crc16
3.crc ccit
4.exit
enter data:11001100111000
----------
generating polynomial:11000000000000101
----------
mod-ified data is:11001100111000000000000000000000000000000101
```

Viva Questions:

1. What is a generator polynomial?
   Ans:

2.What type of errors can CRC detect?
   Ans:
3. What is CRC?

   Ans:

EXPERIMENT-6

AIM: Write a Program to implement Sliding window protocol for Goback N.

Program:

```
#include<stdio.h>
int main()
{
        int windowsize,sent=0,ack,i;
        printf("enter window size\n");
        scanf("%d",&windowsize);
        while(1)
        {
                for( i = 0; i<windowsize; i++)
                        {
                                printf("Frame %d has been transmitted.\n",sent);
                                sent++;
                                if(sent == windowsize)
                                        break;
                        }
                        printf("\nPlease enter the last Acknowledgement received.\n");
                        scanf("%d",&ack);

                        if(ack == windowsize)
                                break;
                        else
                                sent = ack;
        }
return 0;
}
```

OUTPUT:

```
enter window size
8
Frame 0 has been transmitted.
Frame 1 has been transmitted.
Frame 2 has been transmitted.
Frame 3 has been transmitted.
Frame 4 has been transmitted.
Frame 5 has been transmitted.
Frame 6 has been transmitted.
Frame 7 has been transmitted.

Please enter the last Acknowledgement received.
2
Frame 2 has been transmitted.
```

Frame 3 has been transmitted.
Frame 4 has been transmitted.
Frame 5 has been transmitted.
Frame 6 has been transmitted.
Frame 7 has been transmitted.

Please enter the last Acknowledgement received.
8

Viva Questions:

1. What is the sliding window concept?
Ans:
2.What determines the window size (N)?
Ans:
3.How does Go-Back-N differ from Selective Repeat?
Ans:

EXPERIMENT-7

AIM: Write a Program to implement Sliding window protocol for Selective repeat.

Program:

```
#include<stdio.h>
int main()
{
   int w,i,f,frames[50];

printf("Enter window size: ");
scanf("%d",&w);

printf("\nEnter number of frames to transmit: ");
scanf("%d",&f);

printf("\nEnter %d frames: ",f);

   for(i=1;i<=f;i++)
scanf("%d",&frames[i]);

printf("\nWith sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)\n\n");
printf("After sending %d frames at each stage sender waits for acknowledgement sent by the receiver\n\n",w);

   for(i=1;i<=f;i++)
   {
      if(i%w==0)
      {
printf("%d\n",frames[i]);
printf("Acknowledgement of above frames sent is received by sender\n\n");
      }
      else
printf("%d ",frames[i]);
   }

   if(f%w!=0)
printf("\nAcknowledgement of above frames sent is received by sender\n");

   return 0;
}
```

OUTPUT:

Enter window size: 3
Enter number of frames to transmit: 5
Enter 5 frames: 12 5 89 4 6
With sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)

After sending 3 frames at each stage sender waits for acknowledgement sent by the receiver

12 5 89
Acknowledgement of above frames sent is received by sender

4 6
Acknowledgement of above frames sent is received by sender

Viva Questions:
1. Compare Selective Repeat with Go-Back-N.

Ans:

2. What happens when an ACK is lost?

Ans:

3. What does the receiver do with out-of-order frames?
Ans:

EXPERIMENT-8

AIM: Write a Program to implement Stop and Wait Protocol.

Program:

```
#include <stdlib.h>
#define TIMEOUT 5
#define MAX_SEQ 1
#define TOT_PACKETS 8
#define inc(k) if(k<MAX_SEQ) k++; else k=0;
typedef struct
{
 int data;
}packet;
typedef struct
{
 int kind;
 int seq;
 int ack;
 packet info;
 int err;
}frame;
frame DATA;
typedef enum{frame_arrival,err,timeout,no_event} event_type;

void from_network_layer(packet *);
void to_network_layer(packet *);
void to_physical_layer(frame *);
void from_physical_layer(frame *);
void wait_for_event_sender(event_type *);
void wait_for_event_reciever(event_type *);
void reciever();
void sender();

int i=1;        //Data to be sent by sender
char turn;      //r , s
int DISCONNECT=0;
/*_____________________________________________________________________
_*/
void main()
{
 clrscr();
 randomize();
 while(!DISCONNECT)
```

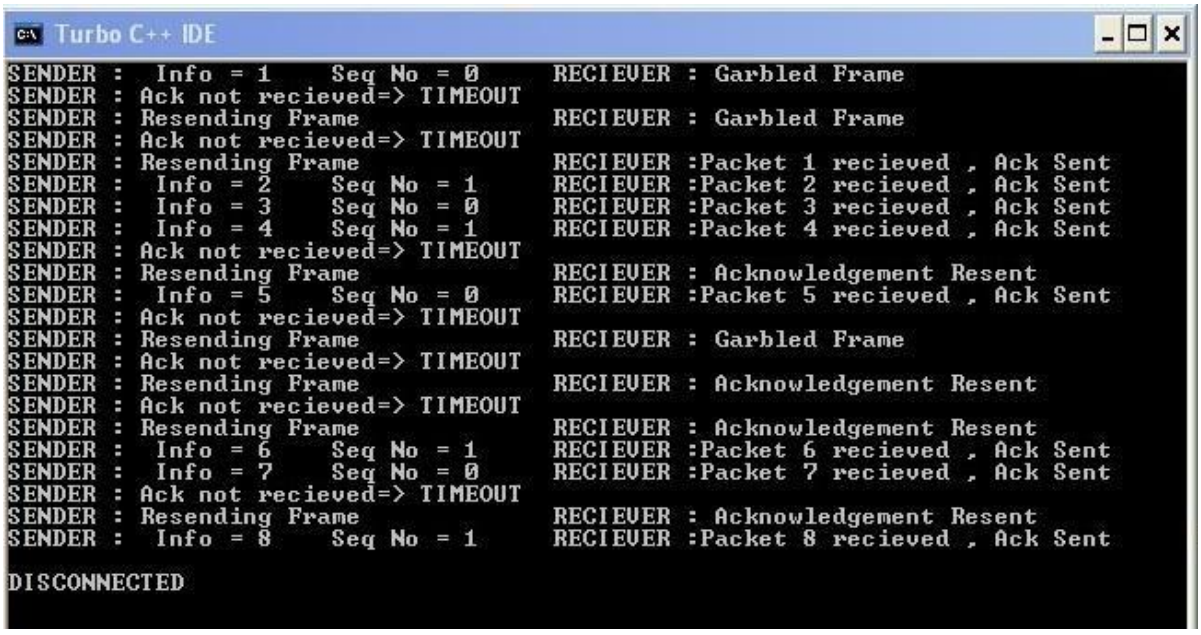
```
 {
  sender();
  delay(400);
  reciever();
 }

}
/*_____________________________________________________________________
_*/
void sender()
{
 static int frame_to_send=0;
 static frame s;
 packet buffer;
 event_type event;
 static int flag=0;

 if(flag==0)
 {
  from_network_layer(&buffer);
  s.info = buffer;
  s.seq = frame_to_send;
  printf("SENDER :  Info = %d    Seq No = %d     ",s.info,s.seq);
  turn = 'r';
  to_physical_layer(&s);
  flag = 1;
 }
 wait_for_event_sender(&event);
 if(turn=='s')
  {
   if(event==frame_arrival)
    {
      from_network_layer(&buffer);
      inc(frame_to_send);
      s.info = buffer;
      s.seq = frame_to_send;
      printf("SENDER :  Info = %d    Seq No = %d     ",s.info,s.seq);
      turn = 'r';
      to_physical_layer(&s);
    }
   if(event==timeout)
    {
      printf("SENDER : Resending Frame             ");
```

```
      turn = 'r';
      to_physical_layer(&s);
    }
  }
}
/*_____________________________________________________________________
_*/
void reciever()
{
 static int frame_expected=0;
 frame r,s;
 event_type event;

 wait_for_event_reciever(&event);
 if(turn=='r')
  {
   if(event==frame_arrival)
    {
     from_physical_layer(&r);
     if(r.seq==frame_expected)
      {
       to_network_layer(&r.info);
       inc(frame_expected);
      }
     else
       printf("RECIEVER : Acknowledgement Resent\n");

     turn = 's';
     to_physical_layer(&s);
    }
   if(event==err)
    {
      printf("RECIEVER : Garbled Frame\n");
      turn = 's';     //if frame not recieved
    }                 //sender shold send it again
  }
}
/*_____________________________________________________________________
_*/
void from_network_layer(packet *buffer)
{
   (*buffer).data = i;
   i++;
```

```
}
/*_____
__*/
void to_physical_layer(frame *s)
{                   // 0 means error
 s->err = random(4);  //non zero means no error
 DATA = *s;          //probability of error = 1/4
}
/*_____
__*/
void to_network_layer(packet *buffer)
{
 printf("RECIEVER :Packet %d recieved , Ack Sent\n",(*buffer).data);
 if(i>TOT_PACKETS)          //if all packets recieved then disconnect
{
   DISCONNECT = 1;
   printf("\nDISCONNECTED");
  }
}
/*_____
__*/
void from_physical_layer(frame *buffer)
{
 *buffer = DATA;
}
/*_____
__*/
void wait_for_event_sender(event_type * e)
{
 static int timer=0;

 if(turn=='s')
  {
   timer++;
   if(timer==TIMEOUT)
    {
     *e = timeout;
     printf("SENDER : Ack not recieved=> TIMEOUT\n");
     timer = 0;
     return;
    }
   if(DATA.err==0)
     *e = err;
```

```
   else
    {
     timer = 0;
     *e = frame_arrival;
    }
  }
}
/*_____
___*/
void wait_for_event_reciever(event_type * e)
{
 if(turn=='r')
  {
   if(DATA.err==0)
     *e = err;
   else
     *e = frame_arrival;
  }
```

OUTPUT:



Viva Questions:
1. What is Stop-and-Wait protocol?
Ans:
2. Is Stop-and-Wait used in real protocols?
Ans:
3. Difference between Go-Back-N and Stop-and-Wait?

Ans:

EXPERIMENT-9

AIM: Write a program for congestion control using leaky bucket algorithm.

Program:

```
#include <stdio.h>

#include<stdio.h>

#include<stdlib.h>

#define MIN(x,y) (x>y)?y:x

int main()

{

 int orate,drop=0,cap,x,count=0,inp[10]={0},i=0,nsec,ch;

 printf("\n enter bucket size : ");

 scanf("%d",&cap);

 printf("\n enter output rate :");

 scanf("%d",&orate);

 do{

printf("\n enter number of packets coming at second %d :",i+1);

 scanf("%d",&inp[i]);

 i++;

 printf("\n enter 1 to contiue or 0 to quit.....");

 scanf("%d",&ch);

 }while(ch);

 nsec=i;

printf("\n second \t recieved \t sent \t dropped \tremained \n");

 for(i=0;count || i<nsec;i++)

 {

 printf(" %d",i+1);

 printf(" \t%d\t ",inp[i]);
```

```
printf(" \t %d\t ",MIN((inp[i]+count),orate));
if((x=inp[i]+count-orate)>0)
{
if(x>cap)
{
count=cap;
drop=x-cap;
}
else
{
count=x;
drop=0;
}
}
else
{
drop=0;
count=0;
}
printf(" \t %d \t %d \n",drop,count);
}
return 0;
}
```

OUTPUT:

```
enter bucket size : 5
enter output rate :3
enter number of packets coming at second 1 :1
enter 1 to contiue or 0 to quit .....1
enter number of packets coming at second 2 :2
```

enter 1 to contiue or 0 to quit 0

enter 1 to contiue or 0 to quit 0

second	recieved		sent	dropped	remained
1	1	1	0	0	
2	2	2	0	0	

Viva questions:
1. What is the Leaky Bucket Algorithm?
Ans:
2. What is the role of the bucket size?
Ans:
3. What is the difference between Leaky Bucket and Token Bucket?
Ans

EXPERIMENT-10

AIM: Write a Program to implement Dijkstra's algorithm to compute the Shortest path through a graph.

Program:

```
#include <limits.h>
#include <stdio.h>
#define V 9
int minDistance(int dist[], bool sptSet[]) {
  int min = INT_MAX, min_index;
  for (int v = 0; v < V; v++)
  if (sptSet[v] == false &&dist[v] <= min)
    min = dist[v], min_index = v;
  return min_index;
}
int printSolution(int dist[], int n) {
printf("Vertex Distance from Source\n");
  for (int i = 0; i< V; i++)
printf("%d \t %d\n", i, dist[i]);
}
void dijkstra(int graph[V][V], int src) {
  int dist[V];
  bool sptSet[V];
  for (int i = 0; i< V; i++)
dist[i] = INT_MAX, sptSet[i] = false;
dist[src] = 0;
  for (int count = 0; count < V - 1; count++) {
    int u = minDistance(dist, sptSet);
sptSet[u] = true;
    for (int v = 0; v < V; v++)
      if (!sptSet[v] && graph[u][v] &&dist[u] != INT_MAX &&dist[u] + graph[u][v]
<dist[v]) dist[v] = dist[u] + graph[u][v];
  }
printSolution(dist, V);
}
int main() {
  int graph[V][V] = { { 0, 6, 0, 0, 0, 0, 0, 8, 0 },
    { 6, 0, 8, 0, 0, 0, 0, 13, 0 },
    { 0, 8, 0, 7, 0, 6, 0, 0, 2 },
    { 0, 0, 7, 0, 9, 14, 0, 0, 0 },
    { 0, 0, 0, 9, 0, 10, 0, 0, 0 },
    { 0, 0, 6, 14, 10, 0, 2, 0, 0 },
```

```
    { 0, 0, 0, 0, 0, 2, 0, 1, 6 },
    { 8, 13, 0, 0, 0, 0, 1, 0, 7 },
    { 0, 0, 2, 0, 0, 0, 6, 7, 0 }
  };
dijkstra(graph, 0);
  return 0;
}
```

OUTPUT:

Vertex Distance from Source

0 0
1 6
2 14
3 21
4 21
5 11
6 9
7 8
8 15

Viva Questions:

1. Compare Dijkstra with Bellman–Ford.

Ans:

2. How would you detect unreachable vertices?

Ans:

3. Explain the difference between single-source and all-pairs shortest paths.

Ans:

EXPERIMENT-11

AIM: Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).

Program:

```
/*
Distance Vector Routing in this program is implemented using Bellman Ford Algorithm:-
*/
#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();
}
/*
```

OUTPUT:

A sample run of the program works as:-

```
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
*/
```

Viva Questions:

1. Explain the Bellman–Ford equation used in DV.

Ans:

2. Why do DV protocols typically choose 16 as ‘infinity’ in RIP?

Ans:

3. **What tables does a router keep in DV?**

Ans:

EXPERIMENT-12

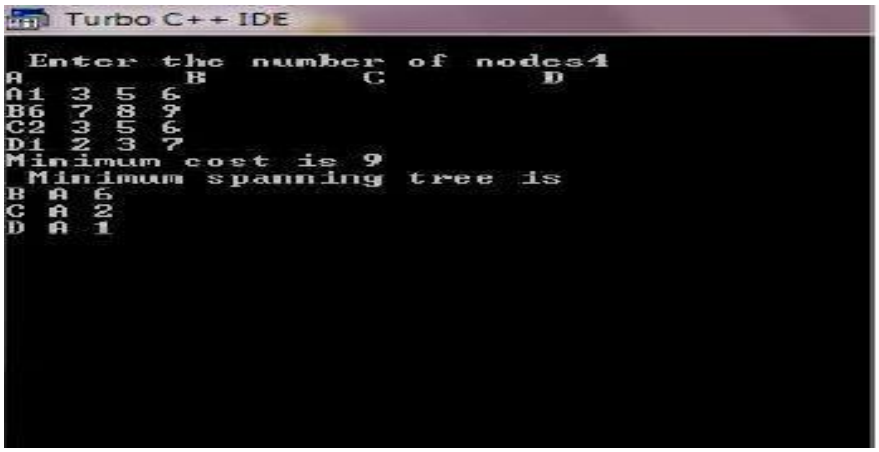
AIM: Write a Program to implement Broadcast tree by taking subnet of hosts.

Program:

```
#include<stdio.h>
#include<conio.h>
int p,q,u,v,n;
int min=99,mincost=0;
int t[50][2],i,j;
int parent[50],edge[50][50];
main()
{
clrscr();
printf("\n Enter the number of nodes");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("%c\t",65+i);
parent[i]=-1;
}
printf("\n");
for(i=0;i<n;i++)
{
printf("%c",65+i);
for(j=0;j<n;j++)
scanf("%d",&edge[i][j]);
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
if(edge[i][j]!=99)
if(min>edge[i][j])
{
min=edge[i][j];
u=i;
v=j;
}
p=find(u);
q=find(v);
if(p!=q)
{
```

```
t[i][0]=u;
t[i][1]=v;
mincost=mincost+edge[u][v];
sunion(p,q);
}
Else
{
t[i][0]=-1;t[i][1]=-1;
}
min=99;
}
printf("Minimum cost is %d\n Minimum spanning tree is\n" ,mincost);
for(i=0;i<n;i++)
if(t[i][0]!=-1 && t[i][1]!=-1)
{
printf("%c %c %d", 65+t[i][0],65+t[i][1],edge[t[i][0]][t[i][1]]);printf("\n");
}
getch();
}
sunion(int l,int m)
{
parent[l]=m;
}
find(int l)
{
if(parent[l]>0)
i=parent[i];
return i;
}
```

OUTPUT:



Viva Questions;

1.What is a broadcast tree, and why is it needed?

Ans:

2. How would you modify the program for weighted links (delays)?

Ans:

3.Compare broadcast tree with minimum spanning tree (MST).

Ans:

EXPERIMENT-13

AIM: Study about Wireshark

Wireshark is an open-source packet analyzer, which is used for **education, analysis, software development, communication protocol development, and network troubleshooting**.

It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a **sniffer, network protocol analyzer, and network analyzer**. It is also used by network security engineers to examine security problems.

Wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as a free packet sniffer computer application. It puts the network card into an unselective mode, i.e., to accept all the packets which it receives.

A packet is a unit of data which is transmitted over a network between the origin and the destination. Network packets are small, i.e., maximum **1.5 Kilobytes for Ethernet packets and 64 Kilobytes for IP packets**. The data packets in the Wireshark can be viewed online and can be analyzed offline

Installation of Wireshark Software

Below are the steps to install the Wireshark software on the computer:

- Open the web browser.
- Search for '**Download Wireshark**.'
- Select the Windows installer according to your system configuration, either 32-bt or 64-bit. Save the program and close the browser.
- Now, open the software, and follow the install instruction by accepting the license.
- The Wireshark is ready for use.

On the network and Internet settings option, we can check the interface connected to our computer.

If you are Linux users, then you will find Wireshark in its package repositories.

i. Packet Capture Using Wire shark

Capturing your traffic with Wireshark
After starting Wireshark, do the following:

1. Select **Capture** | **Interfaces**
2. Select the interface on which packets need to be captured. This will usually be the interface where the Packet/s column is constantly changing, which would indicate the presence of live traffic). If you have multiple network interface cards (i.e. LAN card and Wi-Fi adapter) you may need to check with your IT administrator to determine the right interface.

3. Click the **Start** button to start the capture.
4. Recreate the problem. The capture dialog should show the number of packets increasing. Try to avoid running any other internet applications while capturing, closing other browsers, Instant messengers etc.
5. Once the problem which is to be analyzed has been reproduced, click on **Stop**. It may take a few seconds for Wireshark to display the packets captured.
6. Save the packet trace in the default format. Click on the **File** menu option and select **Save As**. By default Wireshark will save the packet trace in libpcap format. This is a filename with a.pcap extension.

ii. Starting Wire shark using command line

You can start Wireshark from the command line, but it can also be started from most Window managers as well. In this section we will look at starting it from the command line.

Wireshark supports a large number of command line parameters. To see what they are, simply enter the command *wireshark –h*

Wireshark 3.7.3 (v3.7.3rc0-25-g72703582d587)
Interactively dump and analyze network traffic.
See https://www.wireshark.org for more information.

Usage: wireshark [options] ... [<infile>]

Capture interface:
 -i <interface>, --interface <interface>
 name or idx of interface (def: first non-loopback)
 -f <capture filter> packet filter in libpcap filter syntax

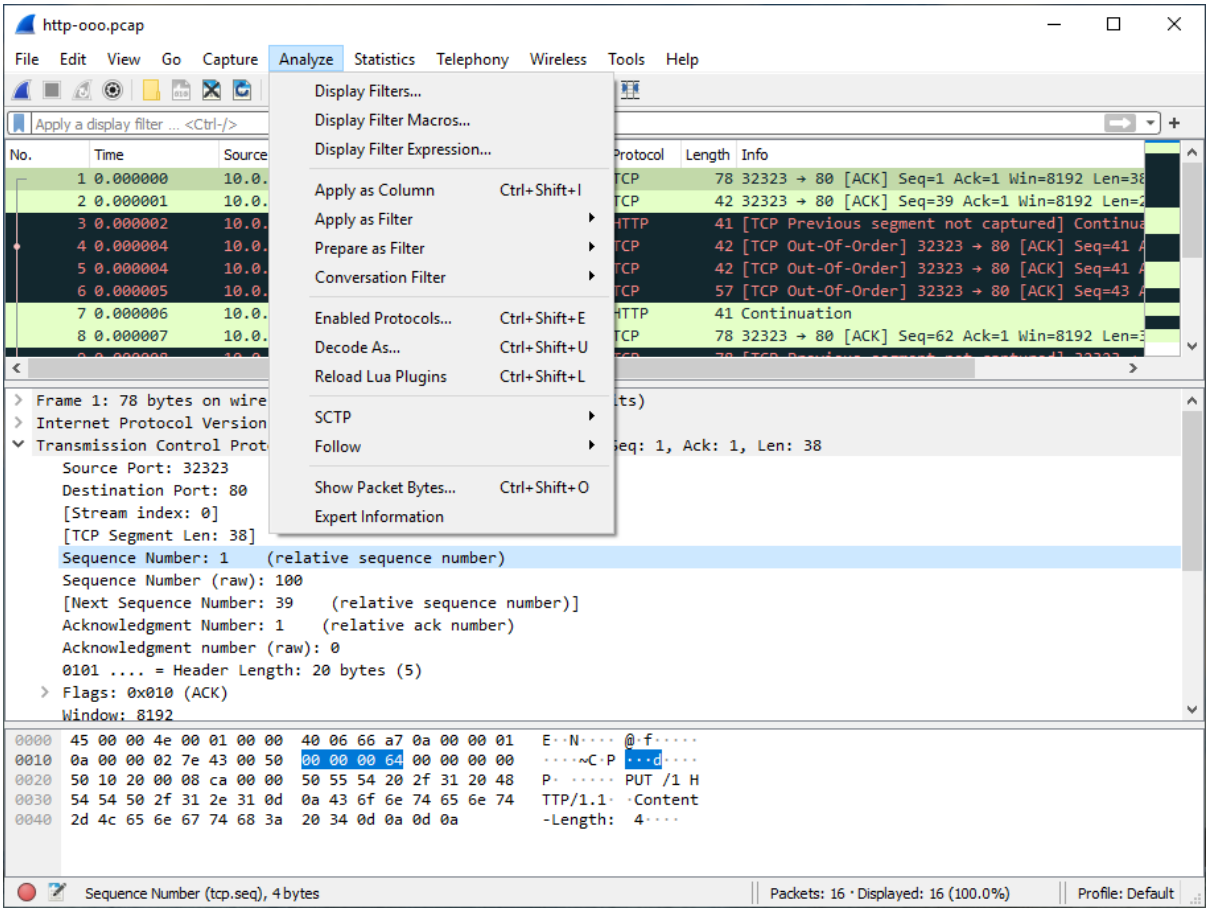
iii. Viewing Captured Traffic

Once you have captured some packets or you have opened a previously saved capture file, you can view the packets that are displayed in the packet list pane by simply **clicking on a packet in the packet list pane**, which will bring up the selected packet in the tree view and byte view panes. Once you have captured some packets or you have opened a previously saved capture file, you can view the packets that are displayed in the packet list pane by simply **clicking on a packet in the packet list pane**, which will bring up the selected packet in the tree view and byte view panes.

iv. Analysis and Statistics & Filters.

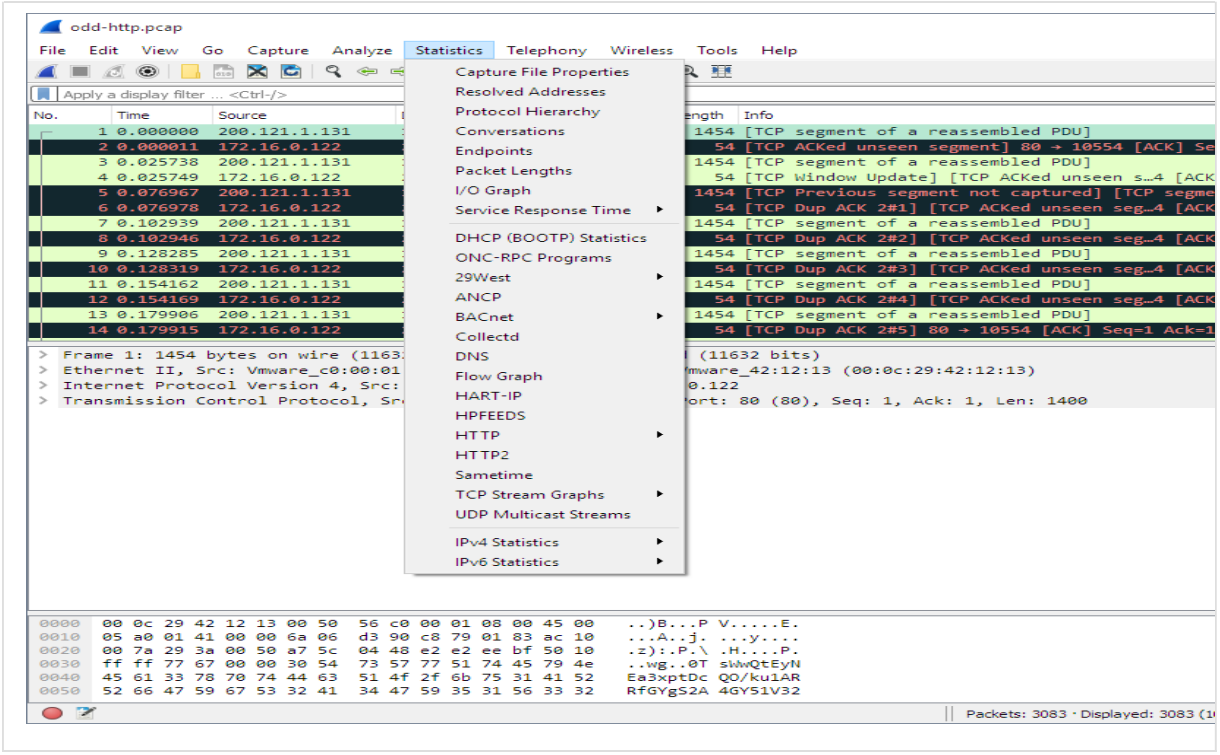
Analysis menu in wireshark

The “Analyze” Menu The Wireshark Analyze menu contains the fields shown in Analyze menu items.



The "Statistics" Menu in wireshark

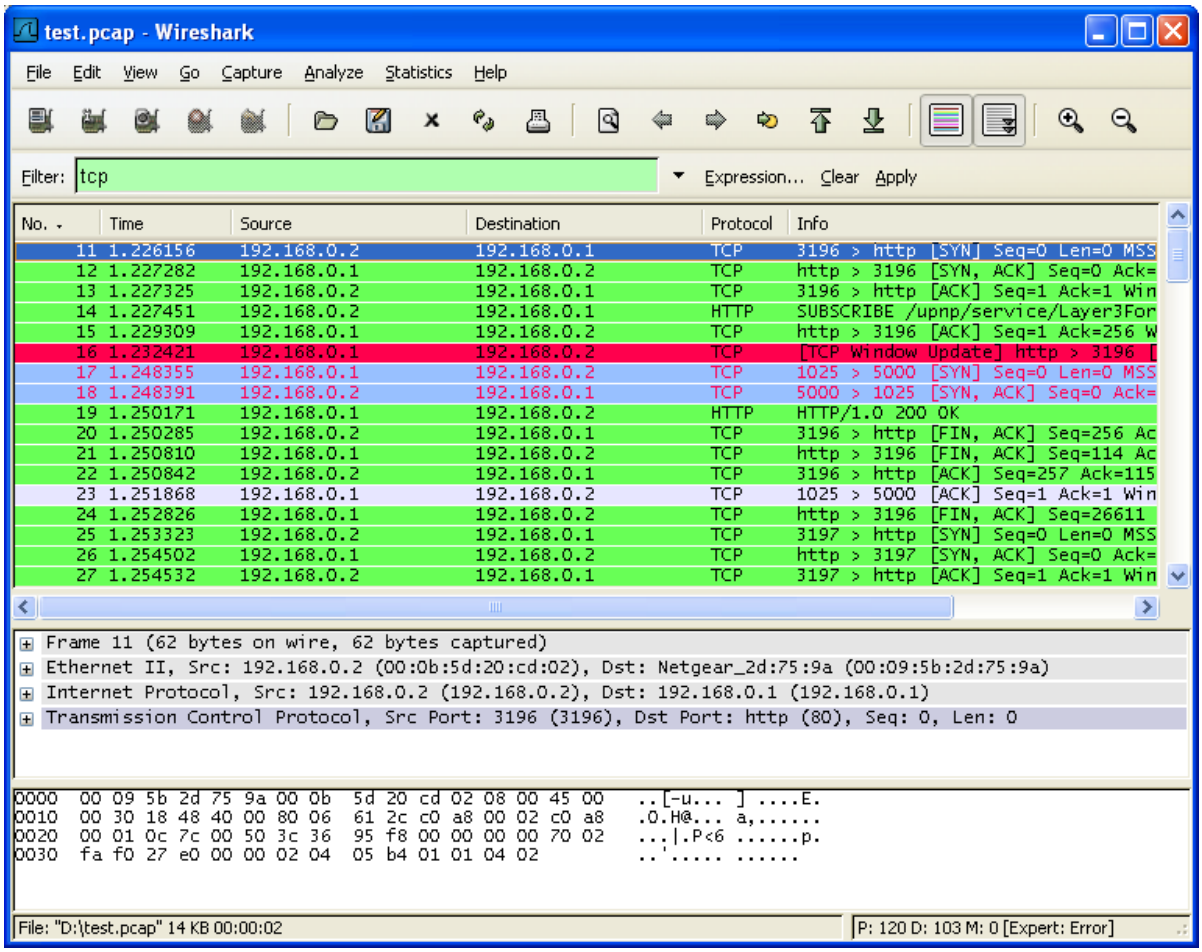
The Wireshark Statistics menu contains the fields



shown in Table 3.9, “Statistics menu items”.

Filters in wireshark

Wireshark has two filtering languages: **capture filters and display filters**. Capture filters are used for filtering when capturing packets, “Filtering while capturing”. Display filters are used for filtering which packets are displayed.



Viva Questions:
1. What is Wireshark?
Ans:

2. Which layer of the OSI model does Wireshark capture?
Ans:

3. Can Wireshark capture encrypted data like HTTPS?
Ans

EXPERIMENT-14

AIM: How to run Nmap scan.

1. Download the Nmap installer. This can be found for free from the developer's website. It is highly recommended that you download directly from the developer to avoid any potential viruses or fake files. Downloading the Nmap installer includes Zenmap, the graphical interface for Nmap which makes it easy for newcomers to perform scans without having to learn command lines.

- The Zenmap program is available for Windows, Linux, and Mac OS X. You can find the installation files for all operating systems on the Nmap website.

- **Install Nmap.** Run the installer once it is finished downloading. You will be asked which components you would like to install. In order to get the full benefit of Nmap, keep all of these checked. Nmap will not install any adware or spyware.

- **Run the "Nmap – Zenmap" GUI program.** If you left your settings at default during installation, you should be able to see an icon for it on your desktop. If not, look in your Start menu. Opening Zenmap will start the program.

- **Enter in the target for your scan.** The Zenmap program makes scanning a fairly simple process. The first step to running a scan is choosing your target. You can enter a domain (example.com), an IP address (127.0.0.1), a network (192.168.1.0/24), or a combination of those.

Choose your Profile. Profiles are preset groupings of modifiers that change what is scanned. The profiles allow you to quickly select different types of scans without having to type in the modifiers on the command line. Choose the profile that best fits your needs:[1]

- **Intense scan** - A comprehensive scan. Contains Operating System (OS) detection, version detection, script scanning, traceroute, and has aggressive scan timing. This is considered an intrusive scan.
- **Ping scan** - This scan simply detects if the targets are online, it does not scan any ports.
- **Quick scan** - This is quicker than a regular scan due to aggressive timing and only scanning select ports.
- **Regular scan** - This is the standard Nmap scan without any modifiers. It will return ping and return open ports on the target.

- **Click Scan to start scanning.** The active results of the scan will be displayed in the Nmap Output tab. The time the scan takes will depend on the scan profile you chose, the physical distance to the target, and the target's network configuration.

Read your results. Once the scan is finished, you'll see the message "Nmap done" at the bottom of the Nmap Output tab. You can now check your results, depending on the type of

scan you performed. All of the results will be listed in the main Nmap Output tab, but you can use the other tabs to get a better look at specific data.[2]

- **Ports/Hosts** - This tab will show the results of your port scan, including the services for those ports.
- **Topology** - This shows the traceroute for the scan you performed. You can see how many hops your data goes through to reach the target.
- **Host Details** - This shows a summary of your target learned through scans, such as the number of ports, IP addresses, hostnames, operating systems, and more.
- **Scans** - This tab stores the commands of your previously-run scans. This allows you to quickly re-scan with a specific set of parameters.

Viva Questions:

1. What is Nmap?

Ans:

2. What are the uses of Nmap?

Ans:

3. Can Nmap detect operating systems?

Ans:

EXPERIMENT-15

AIM: Operating System Detection using Nmap

Program:

```
nmap -O -sVx.x.x.x
Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-17 11:02 PDT
Nmap scan report for x.x.x.x
Host is up (0.22s latency).
Not shown: 994 closed tcp ports (reset)


PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2701/tcp open  cmrcservice   Microsoft Configuration Manager Remote Control service
(CmRcService.exe)
3306/tcp open  mysql         MySQL 5.1.60-community-log
3389/tcp open  ms-wbt-server Microsoft Terminal Services

Aggressive OS guesses: Microsoft Windows Server 2016 (93%),
Microsoft Windows Server 2008 R2 SP1 (92%), Microsoft Windows Server 2012 R2 (91%),
Microsoft Windows 10 1607 (90%), Microsoft Windows Server 2008 R2 (90%),
Microsoft Windows Server 2012 (89%), Microsoft Windows Server 2012 or
Windows Server 2012 R2 (89%), Microsoft Windows Server 2008 R2 or Windows 8 (87%),
Microsoft Windows Server 2008 R2 SP1 or Windows 8 (87%), Microsoft Windows 7 SP1
(86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 18 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 33.77 seconds
```

Viva questions

1. What is Nmap?
Ans:

2. What is OS detection in Nmap?
Ans:

3. Which Nmap option is used for OS detection?
Ans:

EXPERIMENT-16

AIM: Do the following using NS2 Simulator

i. NS2 Simulator-Introduction

ii. Simulate to Find the Number of Packets Dropped

iii. Simulate to Find the Number of Packets Dropped by TCP/UDP

iv. Simulate to Find the Number of Packets Dropped due to Congestion

v. Simulate to Compare Data Rate& Throughput.

Description:

i) NS2 Simulator-Introduction

1. NS2 stands for Network Simulator Version
2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

Features of NS2

1. It is a discrete event simulator for networking
2. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and
3. It simulates wired and wireless
4. Uses TCL as its scripting

Network simulation is an important tool in developing, testing and evaluating network protocols. Simulation can be used without the target physical hardware, making it economical and practical for almost any scale of network topology and setup.

GETTING STARTED

Setting up the environment

A user using the NCTUns in single machine mode, needs to do the following steps before he/she starts the GUI program:

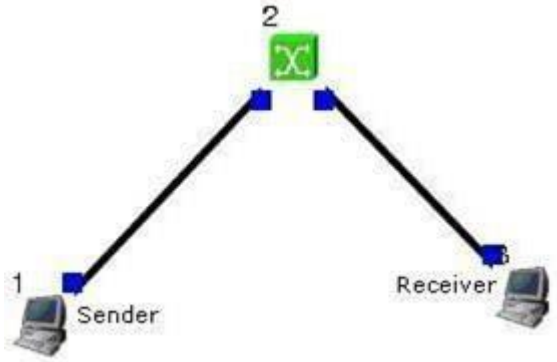
1. Set up environment variables:

Before the user can run up the dispatcher, coordinator, or NCTUns GUI program he/she must set up the NCTUNSHOME environment variable.

2. Start up the dispatcher on terminal
3. Start up the coordinator on terminal
4. Start up the nctunsclient on terminal

After the above steps are followed, the starting screen of NCTUns disappears and the user is presented with the working window as shown below:

ii) Simulate a three node point to point network with a duplex link between them. Set the queue size and vary the bandwidth and find the number of packets dropped.



Step1: Drawing topology

1. Select/click the HOST icon on the toolbar and click the left mouse button on the editor, to place a HOST1 on the editor.

Repeat the above procedure and place another host "HOST2" on the editor.

2. Select/click the HUB icon on the toolbar and click the left mouse button on the editor, to place HUB1 on the editor.
3. Click on the LINK icon on the toolbar and connect HOST1 to HUB1 and HUB1 toHOST2
4. Click on the "E" icon on the toolbar to save the current topology

e.g: file1.tpl

(Look for the ******.tpl extension.)

NOTE: Changes cannot / (should not) be done after selecting the "E" icon.

Step2: Configuration

1. Double click the left mouse button while cursor is on HOST1 to open the
2. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

stg –u 1024 100 1.0.1.2

3. Click OK button on the command window to exit and once again click on the OK button on the HOST window to exit.
4. Double click the left mouse button while cursor is on HOST2 to open the
5. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

rtg –u –w log1

6. Click OK button on the command window to
7. Click NODE EDITOR Button on the HOST window and select the MAC tab from the modal window that pops up.
8. Select LOG STATISTICS and select checkboxes for Number of Drop Packet and Number of Collisions in the MAC window
9. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

Note: To set QUEUE size

1. Double click the left mouse button while cursor is on HOST2 to open the
2. Click NODE EDITOR Button on the HOST window and select the FIFO tab from the modal window that pops up.
3. Change Queue size (Default 50).
4. Click OK button on the FIFO window to exit and once again click on the OK button on the HOST window to exit.

Step3: Simulate

1. Click "R" icon on the tool bar
2. Select Simulation in the menu bar and click/ select RUN in the drop down list to execute the simulation.

- venTo start playback select "►" icon located at the bottom right corner of the

1. To view results, Open up new TERMINAL window, move to file1.resultsfolder and open collision and drop log files in separate TERMINAL window.

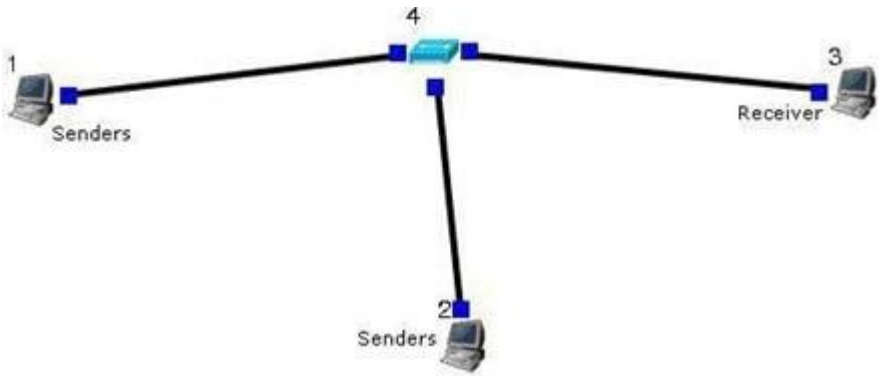
Changing configurations Change 1

1. Open the above file,
2. Do not change the topology or any other configuration,
3. Select E icon on the toolbar
4. Reduce the bandwidth at link2 by double clicking the left mouse button while cursor is on link2 .(Change bandwidth on both tabs Uplink/Downlink)
5. Repeat Step3 (Simulate)

Change 2

1. Open the above file,
2. Remove HUB and replace it with
3. Do not change anything in the configuration
4. Repeat Step3(Simulate)

iii) Simulate a four node point to point network and connect the link as follows Apply a TCP agent between n0 to n3 and apply a UDP agent between n1 and Apply relevant applications over TCP and UDP agents changing the parameters and determine the number of packets sent by two agents.



Step1: Drawing topology

1. Select/click the HOST icon on the toolbar and click the left mouse button on the editor, to place a host on the editor.

Repeat the above procedure and place two other hosts "HOST2" and"HOST3" on the editor.

2. Select/click the HUB (or SWITCH) icon on the toolbar and click the left mouse button on the editor, to place a HUB (or SWITCH) on the editor.
3. Click on the LINK icon on the toolbar and connect HOST1 to HUB, HOST2 to HUB and HUB to HOST3
4. Click on the "E" icon on the toolbar to save the current topology **g:** file2.tpl(Look for the ******.tpl extension.)

NOTE: Changes cannot / (should not) be done after selecting the "E" icon.

Step2: Configuration

1. Double click the left mouse button while cursor is on HOST1 to open the
2. Change simulation time from 0 to 20 for HOST1
3. Select Add button on the HOST1 window to invoke the command window and provide the following command in the command textbox.

ttcp –t –u –s –p 8000 1.0.1.3

4. Click OK button on the command window to exit
5. Click NODE EDITOR Button on the HOST window and select the MAC tab from the modal window that pops up.

6. Select LOG STATISTICS and select checkbox for output throughput in the MAC window
7. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.
8. Double click the left mouse button while cursor is on HOST2 to open the
9. Change simulation time from 21 to 40 for HOST2
10. Select Add button on the HOST2 window to invoke the command window and provide the following command in the command textbox.

stg –u 1024 100 1.0.1.3

11. Click OK button on the command window to exit
12. Click NODE EDITOR Button on the HOST window and select the MAC tab from the modal window that pops up.
13. Select LOG STATISTICS and select checkbox for output throughput in the MAC window
14. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.
15. Double click the left mouse button while cursor is on HOST3 to open the
16. Change simulation time from 0 to 20 for HOST3
17. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

ttcp –r –u –s –p 8000

18. Click OK button on the command window to
19. Change simulation time from 21 to 40 for HOST3
20. Also add the following command on HOST3 rtg –u –

w log1

21. Click NODE EDITOR Button on the HOST window and select the MAC tab fromthe modal window that pops up.
22. Select LOG STATISTICS and select checkbox for input and output throughput in the MAC window
23. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

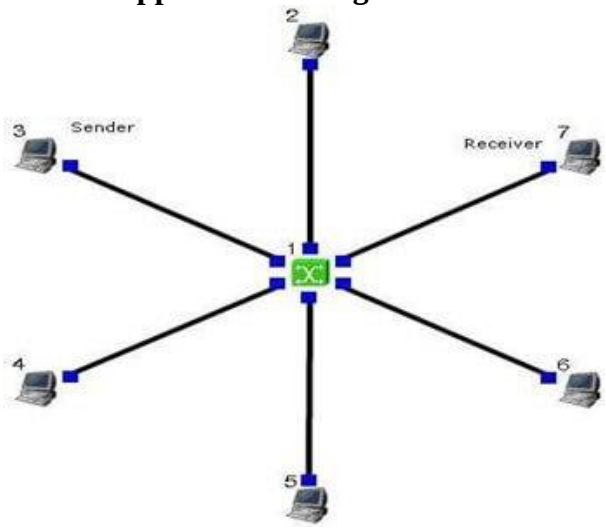
Step3: Simulate

1. Click “R” icon on the tool bar
2. Select Simulation in the menu bar and click/ select RUN in the drop down list to execute the simulation.

- To start playback select “►” icon located at the bottom right corner of the

2. To view results, Open up new TERMINAL window, move to file2.resultsfolder and open input and output throughput log files in separate TERMINAL

Iv) Simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.



Step1: Drawing topology

1. Select/click the SUBNET icon on the toolbar and click the left mouse button on the editor, to place a SUBNET on the editor.
2. A pop up window appears requesting the number of nodes and radius for the subnet

Set number of nodes=6; Set radius of subnet >150

3. Click on the "E" icon on the toolbar to save the current topology **g:** file4.tpl(Look for the ******.tpl extension.)

NOTE: Changes cannot / (should not) be done after selecting the "E" icon.

Step2: Configuration

1. Make 5 HOST as sender and make 1 HOST as receiver and give the commands with particular port number
2. **Sender Commands**

Sender 1 : stcp –p 21 –l 1024 1.0.1.1
Sender 2 : stcp –p 22 –l 1024 1.0.1.1
Sender 3 : stcp –p 23 –l 1024 1.0.1.1
Sender 4 : stcp –p 24 –l 1024 1.0.1.1
Sender 5 : stcp –p 25 –l 1024 1.0.1.1

Receiver Commands

rtcp –p 21 –l 1024
rtcp –p 22 –l 1024
rtcp –p 23 –l 1024
rtcp –p 24 –l 1024
rtcp –p 25 –l 1024

3. For each sender HOST select the Out through put, Collision and Drop
4. For receiver HOST select the In through tput, Collision and Drop

Step3: Simulate

1. Click "R" icon on the tool bar
2. Select Simulation in the menu bar and click/ select RUN in the dropdown list to execute the simulation.

- During simulation, double click the mouse button on any sender HOST, the HOST window pops up, select / click on command console button located at the bottom.

1. A terminal window appears, type **ping IP address** of a receiver HOST in the subnet atthe command prompt.
2. To view results, Open up new TERMINAL window, move to results folder and open drop and collision log files in separate TERMINAL window.

Viva Questions:
1. What is NS2?
Ans:
2. Which scripting language is used in NS2?
Ans:
3. What is NAM in NS2?
Ans: