**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

# COMPUTER NETWORKS

## LAB MANUAL

**SubjectCode :**

**Regulation :R18/JNTUH**

**AcademicYear :2022-2023**

## IIIB. TECH I SEMESTER

## COMPUTERSCIENCEANDENGINEERING–DATASCIENCE

## KGREDDYCOLLEGEOFENGINEERINGANDTECHNOLOGY

**Autonomous, Chilkur Village, Moinabad Mandal, Hyderabad, Telangana501504**

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

# VISIONANDMISSIONOFTHEINSTITUTION

## VISION:

| |
|---|
| To become an institution which is internationally recognized for its holistic approach to engineering, innovative teaching and learning culture, research and entrepreneurial ecosystem, and sustainable social impact in the community |

## MISSION:

| |
|---|
| • To offer undergraduate and post-graduate programs which are supported through industry relevant Curriculum and innovative teaching and learning processes that would help students succeed in their professional careers. |
| • To provide faculty and students with an ecosystem that fosters innovation, research, entrepreneurship, andinternationalexposurethroughstrategicpartnershipswithgovernmentorganizationsand Collaboration within dustiest. |
| • To provide holistic learning environment to students, which will contribute to their personal and Professional growth and enable them to become leaders in their respective fields. |
| • To contribute to the development of the region by using our technological expertise to work with Nearby communities and support them in their social and economic development. |

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

# VISION AND MISSION OF THE DEPARTMENT

## VISION:

| |
|---|
| To be recognized as a department of excellence by stimulating a learning environment in which students and faculty will thrive and grow to achieve their professional, institutional and societal goals. |

## MISSION:

| |
|---|
| • To provide high quality technical education to students that will enable life-long learning and build Expertise in advanced technologies in Computer Science and Engineering. |
| • To promote research and development by providing opportunities to solve complex engineering Problems in collaboration with industry and government agencies. |
| • To encourage professional development of students that will inculcate ethical values and leadership Skills through entrepreneurship while working with the community to address societal issues. |

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

## PROGRAM EDUCATIONAL OBJECTIVES

| |
|---|
| **PEO1:** Graduates will provide solutions to difficult and challenging issues in their profession by Applying computer science and engineering theory and principles. |
| **PEO2:** Graduates have successful careers in computer science and engineering fields or will be Able to successfully pursue advanced degrees. |
| **PEO3:** Graduates will communicate effectively, work collaboratively and exhibit high levels of Professionalism, moral and ethical responsibility. |
| **PEO4:** Graduates will develop the ability to understand and analyze engineering issues in a Broader perspective with ethical responsibility towards sustainable development. |

## PROGRAM OUTCOMES

| |
|---|
| • **POI: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| • **POII: Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| • **POIII: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with for the public health and safety, and the cultural, societal, and environmental considerations. |
| • **POIV: 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. |
| • **POV:** Create, select, and apply appropriate techniques, resources, engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. |
| • **POVI: 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. |

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

| |
|---|
| • **POVII: 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. |
| • **POVIII: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and Norms of the engineering practice. |
| • **POIX: Individual and teamwork:** Function effectively as an individual, and as a member or leader in Diverse teams, and in multidisciplinary settings. |
| • **POX: 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. |
| • **PO XI: 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. |
| • **PO XII: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

## PROGRAM SPECIFIC OUTCOMES

| |
|---|
| **PSO1:** Graduates will be able to apply the knowledge of human cognition and modern tools to Solve complex real-world problems to meet the challenges of the future. |
| **PSO2:** Graduates will be able to utilize innovative Artificial Intelligence tools and techniques to develop the robotic systems, of inter-disciplinary domains for pursuing higher studies, research And entrepreneurship. |

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

Name of the Laboratory: **Computer**

**Networks Lab**

**Things to DO:**

1. Be on time. At the start of the lab period, there will be a short introduction to the experiment you will perform that day. Itis unfair to your partner and to others in the lab if you are not up to speed when the work begins.
2. Inform the instructor and/or TA if there is a problem. You will have their immediate attention if you have cut yourself (even if you consider it minor), if something broke and needs cleaning up, or if you are on fire.
3. Be aware of all the safety devices. Even though the instructor and TA will take care of emergencies, you should know where to find the first aid kit, the chemical spill kit, the eye wash and the safety shower.
4. Keep clutter to a minimum. There is a coat rack to hang your jackets and there are empty cabinets to store your backpacks. Anything left in the aisles is likely to be stepped on and is a hazard to everyone.
5. Wash your hands before you leave the lab for the day.
6. Be aware of others in the lab. Areas of the room may be crowded at times and you should take care not to disturb the experiments of others in the lab.
7. Bring your lab notebook and an open mind to every lab meeting.

**Things NOT TODO:**

1. Do not eat, drink, chew gum, smoke or apply cosmetics in the lab. Just being in lab makes your hands dirtier than you can imagine and you don't want to accidental lye at any re agent (see item 5on'things to do' list).
2. Do not put pieces of lab equipment in your mouth. It sounds obvious but you'd be surprised!
3. Do not work with chemicals until you are sure of their safe handling. This includes some awareness of their flammability, reactivity, toxicity, and disposal.
4. Do not use the phone or computer with gloves on your hands.

**HoD, CSE-CSD** **Principal**

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

## COURSE OBJECTIVES:

1. To understand the working principle of various communication protocols.
2. To understand the network simulator environment and visualize a network topology and observe its performance.
3. To Understanding the Routing Algorithms.
4. To Understand the Connection Management.
5. To analyze the traffic flow and the contents of protocol frames

## COURSEOUTCOMES:

1. **Implement** data link layer farming methods.
2. **Analyze** error detection and error correction codes.
3. **Implement** and analyze routing and congestion issues in network design
4. **Implemen**t Encoding and Decoding techniques used in presentation layer.
5. To be **able** to work with different network tools

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

**Name of the Lab: Computer Networks Lab**

## INDEX

| S. No | Name of the Experiment | Page No | Date | Signature |
|---|---|---|---|---|
| | Implement the data link layer farming methods such as character, character-stuffing and bit stuffing | | | |
| 2 | Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP | | | |
| 3 | Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism | | | |
| 4 | Implement Dijkstra's algorithm to compute the shortest path through a network | | | |
| 5 | Take an example subnet of hosts and obtain a broadcast tree for the subnet | | | |
| 6 | Implement distance vector routing algorithm for obtaining routing tables at each node | | | |
| 7 | Implement data encryption and data decryption | | | |
| 8 | Write a program for congestion control using Leaky bucket algorithm | | | |
| 9 | Write a program for frame sorting technique used in buffers. | | | |
| 10 | Wire shark<br>i. Packet Capture Using Wire shark<br>ii. Starting Wire shark<br>iii. Viewing Captured Traffic<br>iv. Analysis and Statistics & Filters. | | | |
| 11 | How to run Nmap scan | | | |
| 12 | Operating System Detection using Nmap | | | |
| 13 | Do the following using NS2 Simulator<br>i.NS2 Simulator-Introduction<br>ii. Simulate to Find the Number of Packets Dropped<br>iii. Simulate to Find the Number of Packets Dropped by TCP/UDP<br>iv. Simulate to Find the Number of Packets Dropped due to Congestion<br>v. Simulate to Compare Data Rate& Throughput.<br>vi. Simulate to Plot Congestion for Different Source/Destination<br>vii. Simulate to Determine the Performance with respect to Transmission of Packets | | | |

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

**Experiment-1**: Implement the data link layer farming methods such as character, character-stuffing and bit stuffing

## Source Code: character-stuffing

```
//program for character stuffing
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<process.h>
void main ()
{
int i =0,j=0, n, pos;
char a[20], b[50], ch;
clrscr();
printf("enter string:\n");
scanf("%s",&a);
n=strlen(a);
printf("enter position\n");
scanf("%d",&pos);
if(pos>n)
{
printf("invalid position,Enter again:");
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')
{
```


**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

```
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]='\0';
printf("\n frame after stuffing: \n");
printf("%s",b);
getch();
 }
```

**OUTPUT:**

```
Enter String:
haiarchana
Enter position:
4
Enter the Character
K
Frame after stuffing:
Dlestxhaidlekdlearchanadleetx
```

**Source Code:bit stuffing**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
 int a[20],b[30],i,j,k,count,n;
clrscr();
printf("enter frame length:");
scanf("%d",&n);
printf("enter input frame(0's&1's
only):");
 for(i=0;i<n;i++)
```




```
scanf("%d",&a[i]);
i=0;count=1;j=0;
 while(i<n)
 {
 if(a[i]==1)
 {
 b[j]=a[i];
for(k=i+1;a[k]==1&&k<n&&count<5;k
++)
 {
j++;
 b[j]=a[k];
 count++;
 if(count==5)
 {
j++;
 b[j]=0;
 }
i=k;
 }
 }
 else
 {
 b[j]=a[i];
 }
i++;
j++;
 }
printf("After stuffing the frame is:");
 for(i=0;i<j;i++)
printf("%d",b[i]);
getch();
}
```

**Output:**

```
Enter the number of bits:
1
0
1
1
1
1
1
1Data after stuffing: 10101111101
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

## Experiment 2:Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP

**Source Code:**

```
// program for Cyclic Redundancy Check
#include<stdio.h>
#include<conio.h>
int main(void)
{
int data[50],div[16],rem[16];
int datalen, divlen, i,j,k;
int ch;
clrscr();
printf("Enter the data: ");
i = 0;
while((ch = fgetc(stdin)) != '\n')
{
if(ch == '1')
data[i] = 1;
else
data[i] = 0;
i++;
}
datalen = i;
printf("\nEnter the divisor: ");
i = 0;


while((ch = fgetc(stdin)) != '\n')
{
if(ch == '1')
div[i] = 1;
else
div[i] = 0;
i++;
}
divlen = i;
for(i = datalen ; i<datalen + divlen - 1 ; i++)
data[i] = 0;
datalen = datalen + divlen - 1;
for(i = 0 ; i<divlen ; i++)
rem[i] = data[i];
k = divlen-1;
while(k <datalen)
if(rem[0] == 1)
{
```


**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

```
for(i = 0 ; i<divlen ; i++)
rem[i] = rem[i] ^ div[i];
}
else
{
if(k == datalen-1)
break;
for(i = 0 ; i< divlen-1 ; i++)
{
rem[i] = rem[i+1];
printf("%d",rem[i]);
}
rem[i] = data[++k];
printf("%d\n",rem[i]);
}
j=1;
for(i = datalen - divlen + 1 ; i<datalen ; i++)
{
data[i] = rem[j++];
}
printf("\nThe data to be sent is\n");
for(i = 0 ; i<datalen ; i++)
printf("%d",data[i]);
getch();
return 0;
}
```

**OUTPUT:**

```
Enter the data: 10101111
Enter the divisor: 1011
0011
0111
1111
1001
0100
1000
0110
The data to be sent is 10101111110
```

.

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

## Experiment-3: Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism

**Source code:**

```
#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("\acknowledgement 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
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

## Experiment-4: Implement Dijkstra's algorithm to compute the shortest path through a network

**Source code:**

```
#include<stdio.h>
#include<conio.h>
#define INFINITY 9999
#define MAX 10
void dijkstra(int G[MAX][MAX],int n,intstartnode);
int main()
{
 int G[MAX][MAX],i,j,n,u;
printf("Enter no. of vertices:");
scanf("%d",&n);
printf("\nEnter the adjacency matrix:\n");
 for(i=0;i<n;i++)
 for(j=0;j<n;j++)
scanf("%d",&G[i][j]);
printf("\nEnter the starting node:");
scanf("%d",&u);
dijkstra(G,n,u);
 return 0;
}
void dijkstra(int G[MAX][MAX],int n,intstartnode)
{
 int cost[MAX][MAX],distance[MAX],pred[MAX];
 int visited[MAX],count,mindistance,nextnode,i,j;
 //pred[] stores the predecessor of each node
 //count gives the number of nodes seen so far
 //create the cost matrix
for(i=0;i<n;i++)
 for(j=0;j<n;j++)
 if(G[i][j]==0)
 cost[i][j]=INFINITY;
 else
 cost[i][j]=G[i][j];
 //initialize pred[],distance[] and visited[]
 for(i=0;i<n;i++)
 {
 distance[i]=cost[startnode][i];
pred[i]=startnode;
 visited[i]=0;
 }
 distance[startnode]=0;
 visited[startnode]=1;
 count=1;
 while(count<n-1)
 {
mindistance=INFINITY;
```


**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

```
//nextnode gives the node at minimum distance
 for(i=0;i<n;i++)
 if(distance[i]<mindistance&&!visited[i])
 {
mindistance=distance[i];
nextnode=i;
 }
 //check if a better path exists through nextnode
 visited[nextnode]=1;
 for(i=0;i<n;i++)
 if(!visited[i])
 if(mindistance+cost[nextnode][i]<distance[i])
 {
 distance[i]=mindistance+cost[nextnode][i];
pred[i]=nextnode;
 }
 count++;
 }
 //print the path and distance of each node
 for(i=0;i<n;i++)
 if(i!=startnode)
 {
printf("\nDistance of node%d=%d",i,distance[i]);
printf("\nPath=%d",i);
 j=i;
 do
 {
 j=pred[j];
printf("<-%d",j);
}while(j!=startnode);
 }

 }
```

**Output:**

```
Enter no of vertices: 5
Enter the adjacency matrix:
0 10 0 30 100
10 0 50 0 0
0 50 0 20 10
30 0 20 0 60
100 0 10 60 0
Enter the starting node
Distance of node 1=10
Path =1<0
Distance of node2 =50
Path =2<3<0
Distance of node 3 =30
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

## Experiment-5:Take an example subnet of hosts and obtain a broadcast tree for the subnet.

**Source code:**

```
#include<stdio.h>
int a[10][10],n;
main()
{
int i,j,root;
clrscr();
printf("Enter no.of nodes:");
scanf("%d",&n);
printf("Enter adjacent matrix\n");
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{
printf("Enter connecting of %d>%d::",i,j);
scanf("%d",&a[i][j]);
}
printf("Enter root node:");
scanf("%d",&root);
adj(root);
}
adj(int k)
{
int i,j;
printf("Adjacent node of root node::\n");
printf("%d\n",k);
for(j=1;j<=n;j++)
{
if(a[k][j]==1 || a[j][k]==1)
printf("%d\t",j);
}
printf("\n");
for(i=1;i<=n;i++)
{if((a[k][j]==0) && (a[i][k]==0) && (i!=k))
Printf ("%d",i);
}}
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

**OUTPUT:**

```
Enter no.of nodes:5
Enter adjacent matrix
Enter connecting of 1–>1::0
Enter connecting of 1–>2::1
Enter connecting of 1–>3::1
Enter connecting of 1–>4::0
Enter connecting of 1–>5::0
Enter connecting of 2–>1::1
Enter connecting of 2–>2::0
Enter connecting of 2–>3::1
Enter connecting of 2–>4::1
Enter connecting of 2–>5::0
Enter connecting of 3–>1::1
Enter connecting of 3–>2::1
Enter connecting of 3–>3::0
Enter connecting of 3–>4::0
Enter connecting of 3–>5::0
Enter connecting of 4–>1::0
Enter connecting of 4–>2::1
Enter connecting of 4–>3::0
Enter connecting of 4–>4::0
Enter connecting of 4–>5::1
Enter connecting of 5–>1::0
Enter connecting of 5–>2::0
Enter connecting of 5–>3::0
Enter connecting of 5–>4::1
Enter connecting of 5–>5::0
Enter root node:2
Adjacent node of root node::
2
1 3 4
5
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

## Experiment-6:Implement distance vector routing algorithm for obtaining routing tables at each node

**Source code:**

```
#include<stdio.h>
#include<math.h>
 #include<conio.h>
main()
{
int i,j,k,nv,sn,noadj,edel[20],tdel[20][20],min;
char sv,adver[20],ch;
clrscr();
printf("\n ENTER THE NO.OF VERTECES:");
scanf("%d",&nv);
printf("\n ENTER THE SOURCE VERTEX
NUM,BER AND NAME:");
scanf("%d",&sn);
flushall(); sv=getchar();
printf("\n NETER NO.OF ADJ VERTECES TO
VERTEX %c",sv);
scanf("%d",&noadj);
for(i=0;i<noadj;i++)
{
printf("\n ENTER TIME DELAY and NODE NAME:");
scanf("%d %c",&edel[i],&adver[i]);
}
for(i=0;i<noadj;i++)
{
printf("\n ENTER THE TIME DELAY FROM %c to ALL OTHER
NODES: ",adver[i]);
for(j=0;j<nv;j++)
scanf("%d",&tdel[i][j]);
}
printf("\n DELAY VIA--VERTEX \n ");
for(i=0;i<nv;i++)
{
min=1000; ch=0;
for(j=0;j<noadj;j++)
if(min>(tdel[j][i]+edel[j]))
{
min=tdel[j][i]+edel[j];
ch=adver[j];
}
if(i!=sn-1)
printf("\n%d %c",min,ch);
else
printf("\n0 -");
}
getch();
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

**OUTPUT:**

```
ENTER THE NO.OF VERTECES:12
ENTER THE SOURCE VERTEX NUMBER AND NAME:10 J
ENTER NO.OF ADJ VERTECES TO
VERTEX 4 ENTER TIME DELAY and NODE
NAME:8 A ENTER TIME DELAY and NODE
NAME:10 I ENTER TIME DELAY and NODE
NAME:12 H ENTER TIME DELAY and
NODE NAME:6 K
ENTER THE TIME DELAY FROM A to ALL OTHER
NODES: 0 12 25 40 14 23 18 17 21 9 24 29
ENTER THE TIME DELAY FROM I to ALL OTHER
NODES: 24 36 18 27 7 20 31 20 0 11 22 33
ENTER THE TIME DELAY FROM H to ALL OTHER
NODES: 20 31 19 8 30 19 6 0 14 7 22 9
ENTER THE TIME DELAY FROM K to ALL OTHER
NODES: 21 28 36 24 22 40 31 19 22 10 0 9
DELAY VIA--VERTEX
8 a
20 a
28 i
20 h
17 i
30 i
18 h
12 h
10 i
0 -
6 k
15 k
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

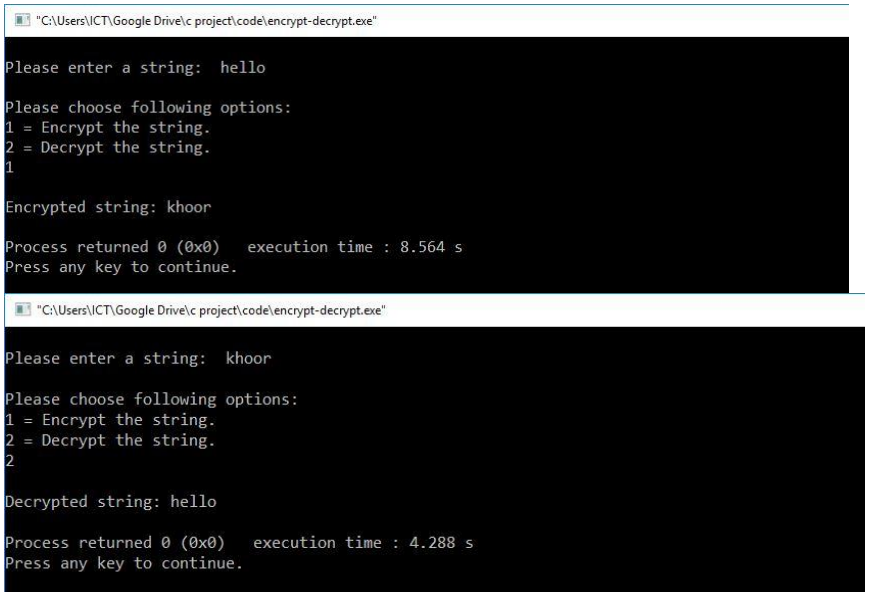
## Experiment-7: Implement data encryption and data decryption

**Source code:**

```
#include <stdio.h>
int main()
{
  int i, x;
  char str[100];

printf("\nPlease enter a string:\t");
  gets(str);

printf("\nPlease choose following options:\n");
printf("1 = Encrypt the string.\n");
printf("2 = Decrypt the string.\n");
scanf("%d", &x);

  //using switch case statements
  switch(x)
  {
  case 1:
for(i = 0; (i< 100 && str[i] != '\0'); i++)
      str[i] = str[i] + 3; //the key for encryption is 3 that is added to ASCII value

printf("\nEncrypted string: %s\n", str);
break;

  case 2:
for(i = 0; (i< 100 && str[i] != '\0'); i++)
      str[i] = str[i] - 3; //the key for encryption is 3 that is subtracted to ASCII value

printf("\nDecrypted string: %s\n", str);
break;

  default:
printf("\nError\n");
  }
  return 0;
}
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

**OUTPUT:**




**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

## Experiment- 8: Write a program for congestion control using Leaky bucket algorithm

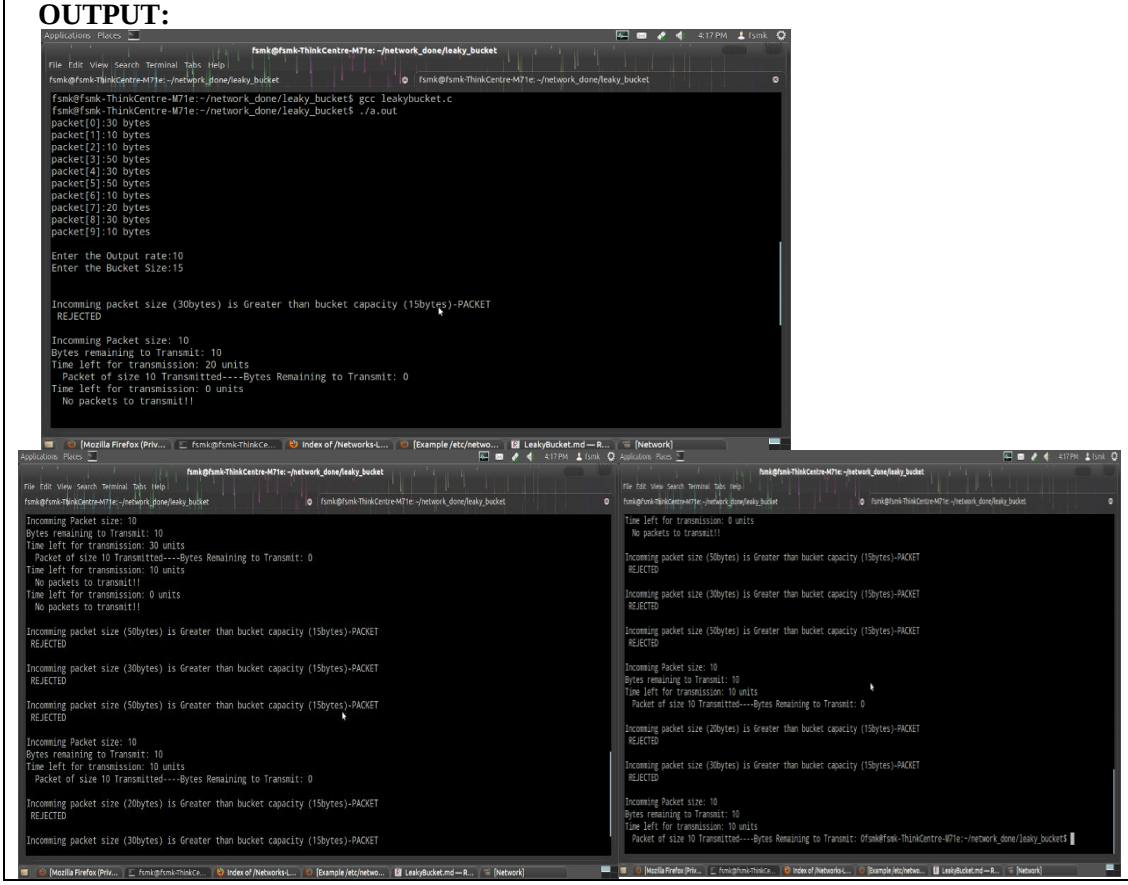
**Source code:**

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

#define NOF_PACKETS 10

int rand(int a)
{
   int rn = (random() % 10) % a;
return  rn == 0 ? 1 :rn;
}

int main()
{
   int packet_sz[NOF_PACKETS], i, clk, b_size, o_rate, p_sz_rm=0, p_sz, p_time, op;
for(i = 0; i<NOF_PACKETS; ++i)
packet_sz[i] = rand(6) * 10;
for(i = 0; i<NOF_PACKETS; ++i)
printf("\npacket[%d]:%d bytes\t", i, packet_sz[i]);
printf("\nEnter the Output rate:");
scanf("%d", &o_rate);
printf("Enter the Bucket Size:");
scanf("%d", &b_size);
for(i = 0; i<NOF_PACKETS; ++i)
   {
if( (packet_sz[i] + p_sz_rm) >b_size)
          if(packet_sz[i] >b_size)/*compare the packet siz with bucket size*/
printf("\n\nIncoming packet size (%dbytes) is Greater than bucket capacity (%dbytes)-
PACKET REJECTED", packet_sz[i], b_size);
          else
printf("\n\nBucket capacity exceeded-PACKETS REJECTED!!");
       else
       {
p_sz_rm += packet_sz[i];
printf("\n\nIncoming Packet size: %d", packet_sz[i]);
printf("\nBytes remaining to Transmit: %d", p_sz_rm);
p_time = rand(4) * 10;
printf("\nTime left for transmission: %d units", p_time);
for(clk = 10; clk<= p_time; clk += 10)
          {
sleep(1);
             if(p_sz_rm)
             {
if(p_sz_rm<= o_rate)/*packet size remaining comparing with output rate*/
                   op = p_sz_rm, p_sz_rm = 0;
                else
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

```
                    op = o_rate, p_sz_rm -= o_rate;
printf("\nPacket of size %d Transmitted", op);
printf("----Bytes Remaining to Transmit: %d", p_sz_rm);
            }
            else
            {
printf("\nTime left for transmission: %d units", p_time-clk);
printf("\nNo packets to transmit!!");
            }
          }
        }
    }
}
```

**OUTPUT:**




**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

## Experiment 9: Write a program for frame sorting technique used in buffers.

**Source code:**

```
include<stdio.h>
#include<string.h>
#define FRAM_TXT_SIZ 3
##define MAX_NOF_FRAM 127
char str[FRAM_TXT_SIZ*MAX_NOF_FRAM];
struct frame // structure maintained to hold frames
{ char text[FRAM_TXT_SIZ];
int seq_no;
}fr[MAX_NOF_FRAM], shuf_ary[MAX_NOF_FRAM];
int assign_seq_no() //function which splits message
{ int k=0,i,j; //into frames and assigns sequence no
for(i=0; i<strlen(str); k++)
{ fr[k].seq_no = k;
for(j=0; j < FRAM_TXT_SIZ && str[i]!='\0'; j++)
fr[k].text[j] = str[i++];
}
printf("\nAfter assigning sequence numbers:\n");
for(i=0; i< k; i++)
printf("%d:%s ",i,fr[i].text);
return k; //k gives no of frames
}
void generate(int *random_ary, const int limit) //generate array of random
nos
{ int r, i=0, j;
while(i< limit)
{ r = random() % limit;
for(j=0; j <i; j++)
if( random_ary[j] == r )
break;
if( i==j ) random_ary[i++] = r;
} }
void shuffle( const int no_frames ) // function shuffles the frames
{
int i, k=0, random_ary[no_frames];
generate(random_ary, no_frames);
for(i=0; i<no_frames; i++)
shuf_ary[i] = fr[random_ary[i]];
printf("\n\nAFTER SHUFFLING:\n");
for(i=0; i<no_frames; i++)
printf("%d:%s ",shuf_ary[i].seq_no,shuf_ary[i].text);
}
void sort(const int no_frames) // sorts the frames
{
int i,j,flag=1;
```


**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

```
struct frame hold;
for(i=0; i< no_frames-1 && flag==1; i++) // search for frames in sequence
{
flag=0;
for(j=0; j < no_frames-1-i; j++) //(based on seq no.) and display
if(shuf_ary[j].seq_no>shuf_ary[j+1].seq_no)
{
hold = shuf_ary[j];
shuf_ary[j] = shuf_ary[j+1];
shuf_ary[j+1] = hold;
flag=1;
}
}
}
int main()
{
int no_frames,i;
printf("Enter the message: ");
gets(str);
no_frames = assign_seq_no();
shuffle(no_frames);
sort(no_frames);
printf("\n\nAFTER SORTING\n");
for(i=0;i<no_frames;i++)
printf("%s",shuf_ary[i].text);
printf("\n\n");
}
```

**OUTPUT:**

```
Enter the message: Welcome To Acharya Institute of Technology
After assigning sequence numbers:
0:Wel 1:com 2:e T 3:o A 4:cha 5:rya 6: In 7:sti 8:tut 9:e o 10:f T 11:ech
12:nol 13:ogy
AFTER SHUFFLING:
1:com 4:cha 9:e o 5:rya 3:o A 10:f T 2:e T 6: In 11:ech 13:ogy 0:Wel 8:tut
12:nol 7:sti
AFTER SORTING
Welcome To Acharya Institute of Technology
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

# Experiment 10: Wireshark

## i. Packet Capture Using Wire shark
## ii. Starting Wire shark
## iii. Viewing Captured Traffic

Solution:

**What is Wireshark?**

Wireshark is an open-source network protocol analysis software program started by Gerald Combs in 1998. A global organization of network specialists and software developers support Wireshark and continue to make updates for new network technologies and encryption methods.

Wireshark is absolutely safe to use. Government agencies, corporations, non-profits, and educational institutions use Wireshark for troubleshooting and teaching purposes. There isn't a better way to learn networking than to look at the traffic under the Wireshark microscope.

There are questions about the legality of Wireshark since it is a powerful packet sniffer. The Light side of the Force says that you should only use Wireshark on networks where you have permission to inspect network packets. Using Wireshark to look at packets without permission is a path to the Dark Side

**Who uses Wireshark?**

Government, educational institutions, corporations, small businesses, non-profits

**How does Wireshark work?**

Wireshark is a packet sniffer and analysis tool. It captures network traffic on the local network and stores that data for offline analysis. Wireshark captures network traffic from Ethernet, Bluetooth, Wireless (IEEE.802.11), Token Ring, Frame Relay connections, and more.

Ed. Note: A "packet" is a single message from any network protocol (i.e., TCP, DNS, etc.)

Ed. Note 2: LAN traffic is in broadcast mode, meaning a single computer with Wireshark can see traffic between two other computers. If you want to see traffic to an external site, you need to capture the packets on the local computer.

Wireshark allows you to filter the log either before the capture starts or during analysis, so you can narrow down and zero into what you are looking for in the network trace. For example, you can set a filter to see TCP traffic between two IP addresses. You can set it only to show you the packets sent from one computer. The filters in Wireshark are one of the primary reasons it became the standard tool for packet analysis.

**How to Download Wireshark**

Downloading and installing Wireshark is easy. Step one is to check the official Wireshark Download page for the operating system you need. The basic version of Wireshark is free.

**Wireshark for Windows**

Wireshark comes in two flavors for Windows, 32 bit and 64 bit. Pick the correct version for your OS. The current release is 3.0.3 as of this writing. The installation is simple and shouldn't cause any issues.

**Wireshark for Mac**

Wireshark is available on Mac as a Homebrew install.

To install Homebrew, you need to run this command at your Terminal prompt:

/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"

Once you have the Homebrew system in place, you can access several open-source projects for your Mac. To install Wireshark run this command from the Terminal:

brew install wireshark

Homebrew will download and install Wireshark and any dependencies so it will run correctly.

**Wireshark for Linux**

Installing Wireshark on Linux can be a little different depending on the Linux distribution. If you aren't running one of the following distros, please double-check the commands.

**Ubuntu**

**From a terminal prompt, run these commands:**

sudo apt-get install wireshark
sudodpkg-reconfigure wireshark-common
sudoadduser $USER wireshark

Those commands download the package, update the package, and add user privileges to run Wireshark.

**Red Hat Fedora**

From a terminal prompt, run these commands:

sudodnf install wireshark-qt
sudousermod -a -G wireshark username
The first command installs the GUI and CLI version of Wireshark, and the second adds permissions to use Wireshark.

**Kali Linux**

Wireshark is probably already installed! It's part of the basic package. Check your menu to verify. It's under the menu option "Sniffing & Spoofing."

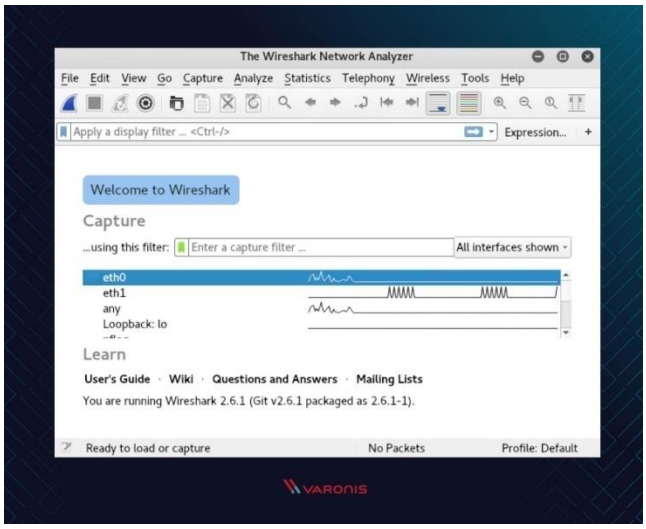
**Data Packets on Wireshark**

Now that we have Wireshark installed let's go over how to enable the Wireshark packet sniffer and then analyze the network traffic

**Capturing Data Packets on Wireshark**

When you open Wireshark, you see a screen that shows you a list of all of the network connections you can monitor. You also have a capture filter field, so you only capture the network traffic you want to see.

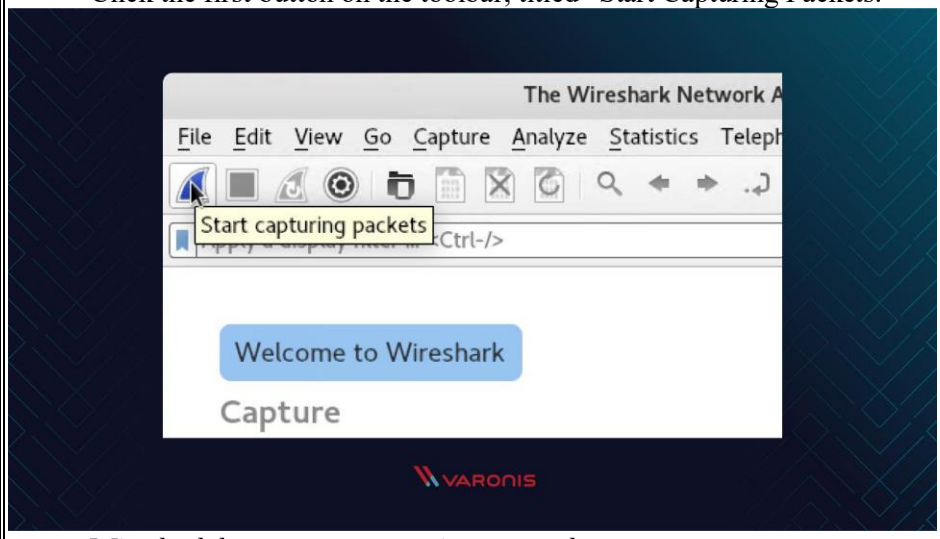
**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**




Wireshark capture filter screenshot

You can select one or more of the network interfaces using "shift left-click." Once you have the network interface selected, you can start the capture, and there are several ways to do that.

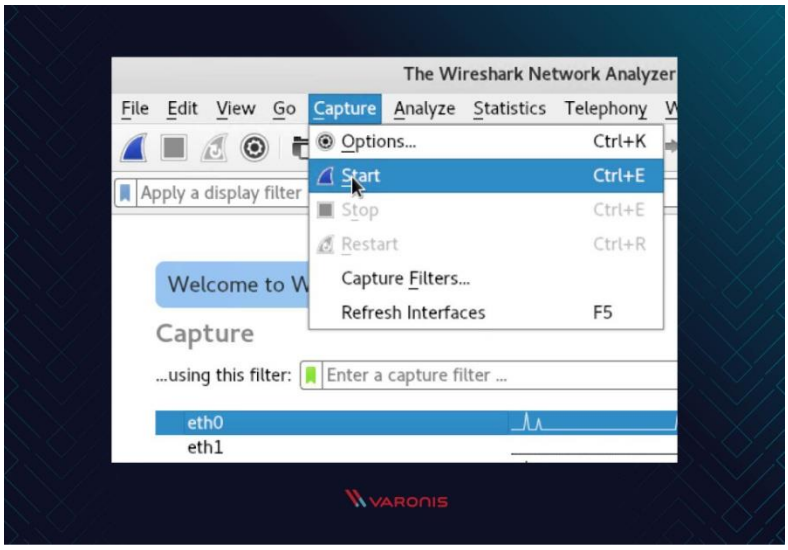
Click the first button on the toolbar, titled "Start Capturing Packets."




Wireshark how to start capturing screenshot

You can select the menu item Capture -> Start.

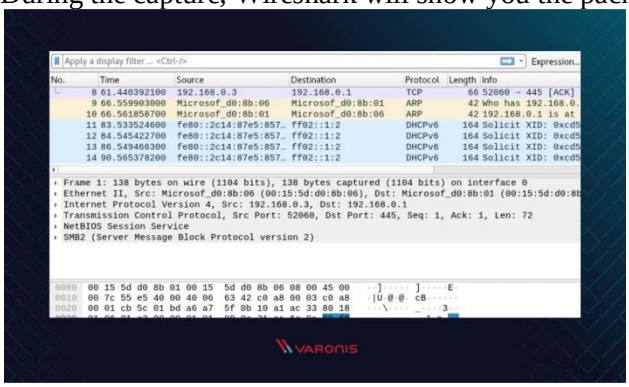
**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**




Wireshark capture packets screenshot

Or you could use the keystroke Control – E.

During the capture, Wireshark will show you the packets that it captures in real-time.




**Wireshark how to stop capture screenshot**

Once you have captured all the packets you need, you use the same buttons or menu options to stop the capture. Best practice says that you should stop Wireshark packet capture before you do analysis.

**Analyzing Data Packets on Wireshark**

Wireshark shows you three different panes for inspecting packet data. The Packet List, the top pane, is a list of all the packets in the capture. When you click on a packet, the other two panes change to show you the details about the selected packet. You can also tell if the packet is part of a conversation. Here are some details about each column in the top pane:

**No.**: This is the number order of the packet that got captured. The bracket indicates that this packet is part of a conversation.

**Time:** This column shows you how long after you started the capture that this packet got captured. You can change this value in the Settings menu if you need something different displayed.

**Source:** This is the address of the system that sent the packet.
**Destination**: This is the address of the destination of that packet.
**Protocol:** This is the type of packet, for example, TCP, DNS, DHCPv6, or ARP.
**Length:** This column shows you the length of the packet in bytes.
**Info:** This column shows you more information about the packet contents, and will vary depending on what kind of packet it is.

Packet Details, the middle pane, shows you as much readable information about the packet as possible, depending on what kind of packet it is. You can right-click and create filters based on the highlighted text in this field.
The bottom pane, Packet Bytes, displays the packet exactly as it got captured in hexadecimal.
When you are looking at a packet that is part of a conversation, you can right-click the packet and select Follow to see only the packets that are part of that conversation.

**Wireshark Filters**
One of the best features of Wireshark is the Wireshark Capture Filters and Wireshark Display Filters. Filters allow you to view the capture the way you need to see it so you can troubleshoot the issues at hand. Here are several filters to get you started.

**Wireshark Capture Filters**
Capture filters limit the captured packets by the filter. Meaning if the packets don't match the filter, Wireshark won't save them. Here are some examples of capture filters:

host IP-address: this filter limits the capture to traffic to and from the IP address

net 192.168.0.0/24: this filter captures all traffic on the subnet.

dst host IP-address: capture packets sent to the specified host.

port 53: capture traffic on port 53 only.

port not 53 and not arp: capture all traffic except DNS and ARP traffic

**Wireshark Display Filters**
Wireshark Display Filters change the view of the capture during analysis. After you have stopped the packet capture, you use display filters to narrow down the packets in the Packet List so you can troubleshoot your issue.

The most useful (in my experience) display filter is:

ip.src==IP-address and ip.dst==IP-address

This filter shows you packets from one computer (ip.src) to another (ip.dst). You can also use ip.addr to show you packets to and from that IP. Here are some others:

tcp.port eq 25: This filter will show you all traffic on port 25, which is usually SMTP traffic.

icmp: This filter will show you only ICMP traffic in the capture, most likely they are pings.

ip.addr != IP_address: This filter shows you all traffic except the traffic to or from the specified computer.


**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

Analysts even build filters to detect specific attacks, like this filter to detect the Sasser worm:

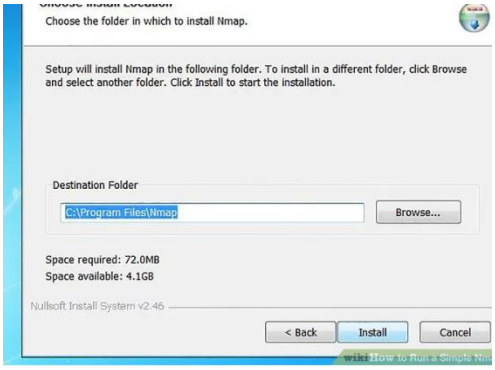
ls_ads.opnum==0x09

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

# Experiment-11: How to run Nmap scan

**Step 1: Download the Nmap installer.**

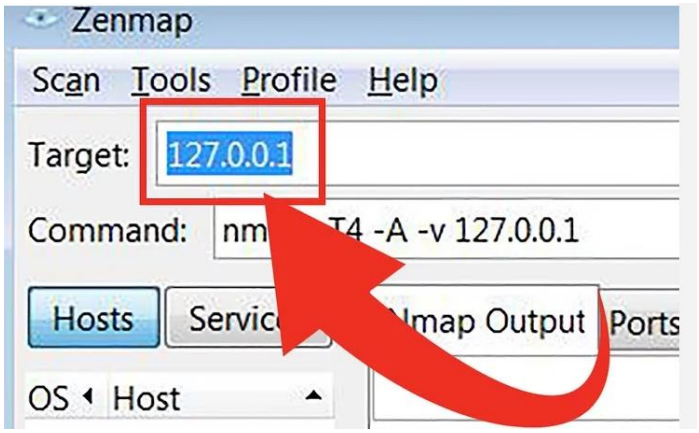
This can be found for free from the developer's website. It is highly recommended that you download directly from the developer to avoid any potential viruses or fake files. Downloading the Nmap installer includes Zenmap, the graphical interface for Nmap which makes it easy for newcomers to perform scans without having to learn command lines.TheZenmap program is available for Windows, Linux, and Mac OS X. You can find the installation files for all operating systems on the Nmap website.




**Step 2: Install Nmap.** 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 adware or spyware.

**Step3 :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.

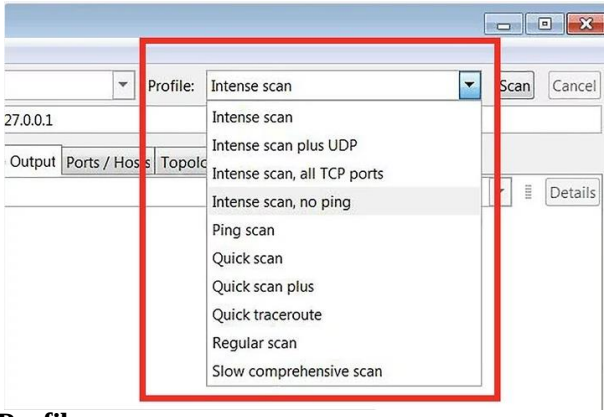



**Step4: Enter in the target for your scan.**

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.

Depending on the intensity and target of your scan, running an Nmap scan may be against the terms of your internet service provider, and may land you in hot water. Always check your local laws and your ISP contract before performing Nmap scans on targets other than your own network.

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**




**Step 5: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.

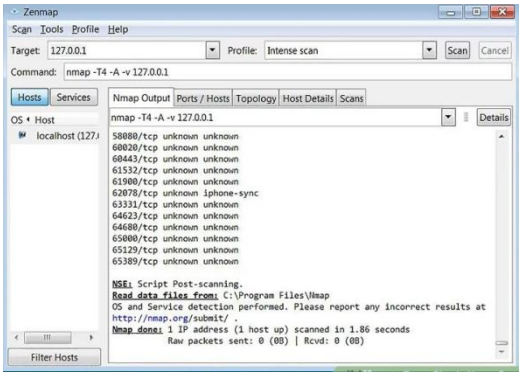
Quick scan - This is quicker than a regular scan due to aggressive timing and only scanning select ports.

Regular scan - This is the standard Nmap scan


without any modifiers. It will return ping and return open ports on the target.

**Step 6: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, network



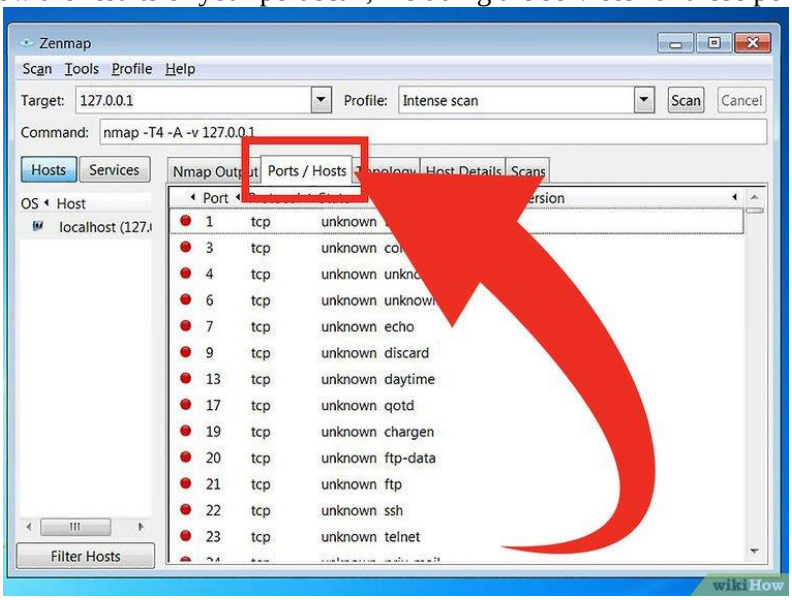

the physical distance to the target, and the target's configuration.

**Step 7: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.Ports/Hosts - This tab will show the results of your port scan, including the services for those

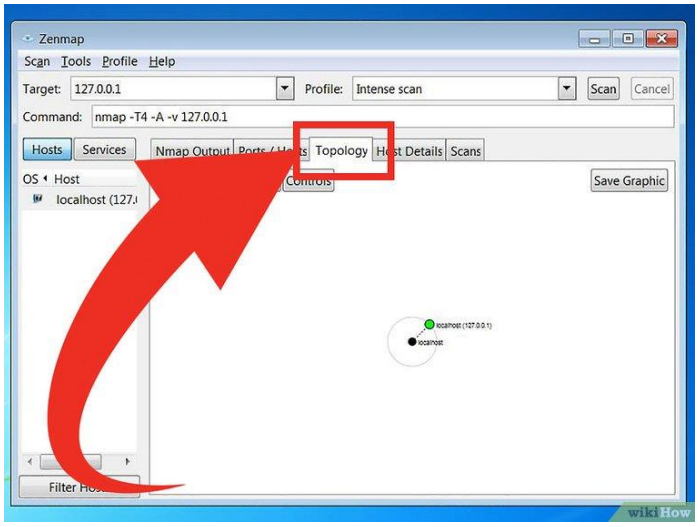
**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

ports.

Ports/Hosts - This tab will show the results of your port scan, including the services for those ports.

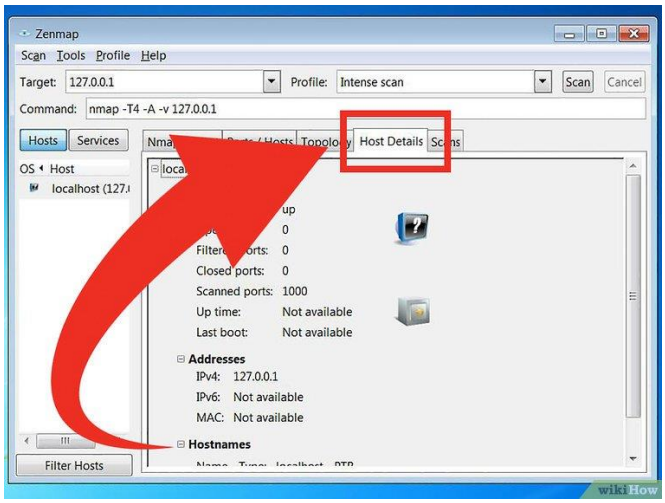



Topology - This shows the traceroute for the scan you performed. You can see how many hops your data goes through to reach the target.

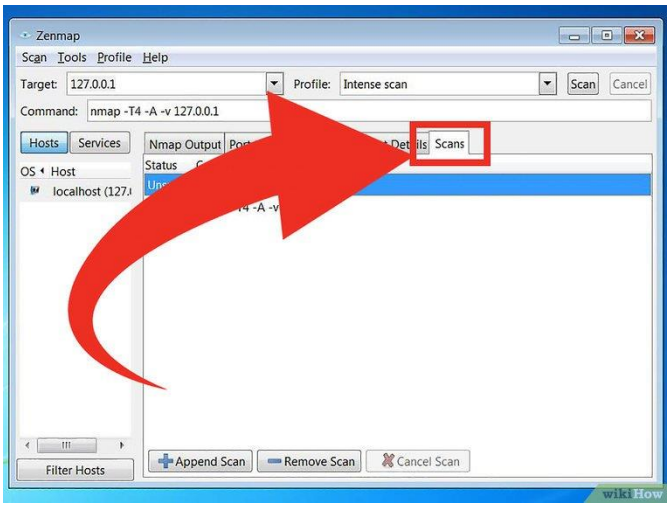



Host Details - This shows a summary of your target learned through scans, such as the number of ports, IP addresses, hostnames, operating systems, and more.

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**




Scans - This tab stores the commands of your previously-run scans. This allows you to quickly re-scan with a specific set of parameters.



**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

# Experiment-12: Operating System Detection using Nmap

Operating system (OS) detection can be enable with -O . On the other side -A parameter also provides operating system information. To use operating system detection nmap command should be run with root privileges because lower layer network manipulation will be done by nmap.

$ sudonmap -O localhost

```
ismail@ubu1:~$ nmap -O localhost
TCP/IP fingerprinting (for OS scan) requires root privileges.
QUITTING!
ismail@ubu1:~$ sudo nmap -O localhost
[sudo] password for ismail:

Starting Nmap 7.12 ( https://nmap.org ) at 2016-12-31 17:03 +03
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000070s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
631/tcp  open  ipp
3306/tcp open  mysql
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3.19 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.19, Linux 3.8 - 4.4
Network Distance: 0 hops
```

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

# Experiment-13: 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.**
**vi.Simulate to Plot Congestion for Different Source/Destination**
**vi.Simulate to Determine the Performance with respect to Transmission of Packets**

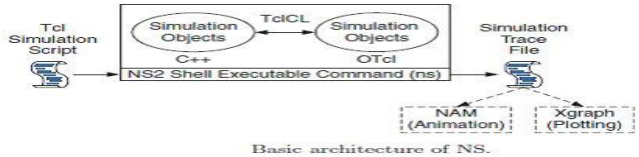
**i.NS2 Simulator-Introduction:**
NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

## Features of NS2

1. It is a discrete event simulator for networking research.
2. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR.
3. It simulates wired and wireless network.
4. It is primarily Unix based.
5. Uses TCL as its scripting language.
6. Otcl: Object oriented support
7. Tclcl: C++ and otcl linkage
8. Discrete event scheduler

## Basic Architecture

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL



Basic architecture of NS.

**ii. Simulate to Find the Number of Packets Dropped**
#Create a new Simulation Instance


**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

```
set ns [new Simulator]
set bandwidth 1.75Mb
#Turn on the Trace and the animation files
set f [open out.tr w]
set nf [open out.nam w]
$ns trace-all $f
$ns namtrace-all $nf
#Define the finish procedure to perform at the end of the simulation
proc finish {} {
 global f nf ns
 $ns flush-trace
 close $f
 close $nf
 exec namout.nam&
 #exec awk -f 1.awk out.tr &
 exit 0
}
#Create the nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
#Label the nodes
$n0 label "TCP Source"
$n1 label "UDP Source"
$n2 label "Sink"
#Set the color
$ns color 1 red
$ns color 2 yellow
#Create the Topology
$ns duplex-link $n0 $n1 $bandwidth 10ms DropTail
$ns duplex-link $n1 $n2 1.75Mb 20ms DropTail
#Attach a Queue of size N Packets between the nodes n1 n2
$ns queue-limit $n1 $n2 10
#Make the Link Orientation
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right
#Create a UDP Agent and attach to the node n1
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
#Create a CBR Traffic source and attach to the UDP Agent
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
#Specify the Packet Size and interval
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
#Create a Null Agent and attach to the node n2
set null0 [new Agent/Null]
$ns attach-agent $n2 $null0
#Connect the CBR Traffic source to the Null agent
```

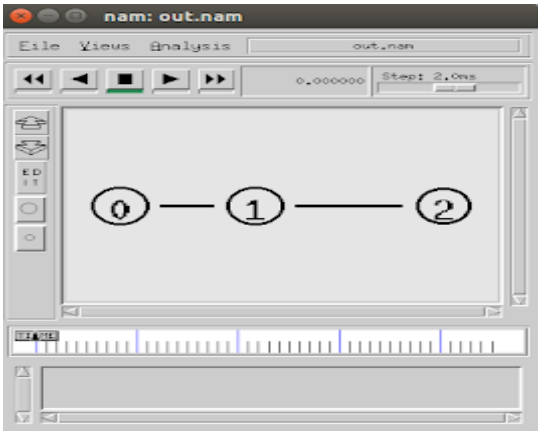
**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

```
$ns connect $udp0 $null0
#Create a TCP agent and attach to the node n0
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
#Create a FTP source and attach to the TCP agent
set ftp0 [new Application/FTP]
#Attach the FTP source to the TCP Agent
$ftp0 attach-agent $tcp0
#Create a TCPSink agent and attach to the node n2
set sink [new Agent/TCPSink]
$ns attach-agent $n2 $sink
#Specify the Max file Size in Bytes
$ftp0 set maxPkts_ 1000
#Connect the TCP Agent with the TCPSink
$ns connect $tcp0 $sink
$udp0 set class_ 1
$tcp0 set class_ 2
#Schedule the Events
$ns at 0.1 "$cbr0 start"
$ns at 1.0 "$ftp0 start"
$ns at 4.0 "$ftp0 stop"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"
$ns run
```

Now run the code using

**ns filename.tcl**

**OUTPUT:**



**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

# VIVA–VOCEQUESTIONS

1. **Explain What is Network?**

   A network is a set of devices connected by physical media links. A network is recursively is a connection of two or more nodes by a physical link or two or more networks connected by one or more nodes.

2. **What is a Link?**

   At the lowest level, a network can consist of two or more computers directly connected by some physical medium such as coaxial cable or optical fiber. Such a physical medium is called as Link.

3. **What is a node?**

   A network can consist of two or more computers directly connected by some physical medium such as coaxial cable or optical fiber. Such a physical medium is called as Links and the computer it connects is called as Nodes.

4. **What is a gateway or Router?**

   A node that is connected to two or more networks is commonly called as router or Gateway. It generally forwards message from one network to another.

5. **What is point-point link?**

   If the physical links are limited to a pair of nodes it is said to be point-point link.

6. **What is Multiple Access?**

   If the physical links are shared by more than two nodes, it is said to be Multiple Access.

7. **What are the advantages of Distributed Processing?**

   a. Security/Encapsulation

   b. Distributed database

   c. Faster Problem solving

   d. Security through redundancy

   e. Collaborative Processing

8. **What are the criteria necessary for an effective and efficient network?**

   a. Performance

   It can be measured in many ways, including transmit time and response time. b. Reliability

   It is measured by frequency of failure, the time it takes a link to recover from a failure, and the networks robustness.

c. Security

Security issues includes protecting data from unauthorized access and virus.

9. **Name the factors that affect the performance of the network?**

a. Number of Users

b. Type of transmission medium

c. Hardware

d. Software

10. **Name the factors that affect the reliability of the network?**

a. Frequency of failure

b. Recovery time of a network after a failure

11. **Name the factors that affect the security of the network?**

a. Unauthorized Access

b. Viruses

12. **What is Protocol?**

A protocol is a set of rules that govern all aspects of information communication.

13. **What are the key elements of protocols?**

The key elements of protocols are

a. Syntax

It refers to the structure or format of the data, that is the order in which they are presented.

b. Semantics

It refers to the meaning of each section of bits.

c. Timing

Timing refers to two characteristics: When data should be sent and how fast they can be sent

14. **What are the key design issues of a computer Network?**

a. Connectivity

b. Cost-effective Resource Sharing

c. Support for common Services

d. Performance

15. **Define Bandwidth and Latency?**

Network performance is measured in Bandwidth (throughput) and Latency (Delay). Bandwidth of a

network is given by the number of bits that can be transmitted over the network in a certain period of time. Latency corresponds to how long it t5akes a message to travel from one end of a network to the other. It is strictly measured in terms of time.

16. **Define Routing?**

The process of determining systematically hoe to forward messages toward the destination nodes based on its address is called routing.

17. **What is a peer-peer process?**

The processes on each machine that communicate at a given layer are called peer-peer process.

18. **When a switch is said to be congested?**

It is possible that a switch receives packets faster than the shared link can accommodate and stores in its memory, for an extended period of time, then the switch will eventually run out of buffer space, and some packets will have to be dropped and in this state is said to congested state.

19. **What is semantic gap?**

Defining a useful channel involves both understanding the applications requirements and recognizing the limitations of the underlying technology. The gap between what applications expects and what the underlying technology can provide is called semantic gap.

20. **What is Round Trip Time?**

The duration of time it takes to send a message from one end of a network to the other and back, is called RTT.

21. **Define the terms Unicasting, Multicasting and Broadcasting?**

If the message is sent from a source to a single destination node, it is called Unicasting.

If the message is sent to some subset of other nodes, it is called Multicasting.

If the message is sent to all the m nodes in the network it is called Broadcasting

22. **What is Multiplexing?**

Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link.

23. **Name the categories of Multiplexing?**

a. Frequency Division Multiplexing (FDM)

b. Time Division Multiplexing (TDM)

i. Synchronous TDM

ii. Asynchronous TDM Or Statistical TDM.

c. Wave Division Multiplexing (WDM)

**24. What is FDM?**

FDM is an analog technique that can be applied when the bandwidth of a link is greater than the combined bandwidths of the signals to be transmitted.

**25. What is WDM?**

WDM is conceptually the same as FDM, except that the multiplexing and demultiplexing involve light signals transmitted through fiber optics channel.

**26. What is TDM?**

TDM is a digital process that can be applied when the data rate capacity of the transmission medium is greater than the data rate required by the sending and receiving devices.

**27. What is Synchronous TDM?**

In STDM, the multiplexer allocates exactly the same time slot to each device at all times, whether or not a device has anything to transmit.

**28. List the layers of OSI**

a. Physical Layer

b. Data Link Layer

c. Network Layer

d. Transport Layer

e. Session Layer

f. Presentation Layer

g. Application Layer

**29. Which layers are network support layers?**

a. Physical Layer

b. Data link Layer and

c. Network Layers

**30. Which layers are user support layers?**

a. Session Layer

b. Presentation Layer and

c. Application Layer

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING–DATASCIENCE**

**31. Which layer links the network support layers and user support layers?**

The Transport layer links the network support layers and user support layers.

**32. What are the concerns of the Physical Layer?**

Physical layer coordinates the functions required to transmit a bit stream over a physical medium.

a. Physical characteristics of interfaces and media

b. Representation of bits

c. Data rate

d. Synchronization of bits

e. Line configuration

f. Physical topology

g. Transmission mode

**33. What are the responsibilities of Data Link Layer?**

The Data Link Layer transforms the physical layer, a raw transmission facility, to a reliable link and is responsible for node-node delivery.

a. Framing

b. Physical Addressing

c. Flow Control

d. Error Control

e. Access Control

**34. What are the responsibilities of Network Layer?**

The Network Layer is responsible for the source-to-destination delivery of packet possibly across multiple networks (links).

a. Logical Addressing

b. Routing

**35. What are the responsibilities of Transport Layer?**

The Transport Layer is responsible for source-to-destination delivery of the entire message.

a. Service-point Addressing

b. Segmentation and reassembly

c. Connection Control

d. Flow Control

e. Error Control

**36. What are the responsibilities of Session Layer?**

The Session layer is the network dialog Controller. It establishes, maintains and synchronizes the interaction between the communicating systems.

a. Dialog control

b. Synchronization

**37. What are the responsibilities of Presentation Layer?**

The Presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.

a. Translation

b. Encryption

c. Compression

**38. What are the responsibilities of Application Layer?**

The Application Layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as e-mail, shared database management and other types of distributed information services.

a. Network virtual Terminal

b. File transfer, access and Management (FTAM)

c. Mail services

d. Directory Services

**39. What are the two classes of hardware building blocks**?

Nodes and Links.

**40. What are the different link types used to build a computer network?**

a. Cables

b. Leased Lines

c. Last-Mile Links

d. Wireless Links

**41. What are the categories of Transmission media?**

a. Guided Media

i. Twisted Pair cable

1. Shielded TP

2. Unshielded TP

ii. Coaxial Cable

iii. Fiber-optic cable

b. Unguided Media

i. Terrestrial microwave

ii. Satellite Communication

**42. What are the types of errors?**

a. Single-Bit error

In a single-bit error, only one bit in the data unit has changed

b. Burst Error

A Burst error means that two or more bits in the data have changed.

**43. What is Error Detection? What are its methods?**

Data can be corrupted during transmission. For reliable communication errors must be deducted and Corrected. Error Detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination. The common Error Detection methods are

a. Vertical Redundancy Check (VRC)

b. Longitudinal Redundancy Check (VRC)

c. Cyclic Redundancy Check (VRC)

d. Checksum

**44. What is Redundancy?**

The concept of including extra information in the transmission solely for the purpose of comparison. This technique is called redundancy.

**45. What is VRC?**

It is the most common and least expensive mechanism for Error Detection. In VRC, a parity bit is added to every data unit so that the total number of 1s becomes even for even parity. It can detect all single-bit errors. It can detect burst errors only if the total number of errors in each data unit is odd.

**46. What is LRC?**

In LRC, a block of bits is divided into rows and a redundant row of bits is added to the whole block. It

can detect burst errors. If two bits in one data unit are damaged and bits in exactly the same positions in another data unit are also damaged, the LRC checker will not detect an error. In LRC a redundant data unit follows n data units.

**47. What is CRC?**

CRC, is the most powerful of the redundancy checking techniques, is based on binary division.

**48. What is Checksum?**

Checksum is used by the higher layer protocols (TCP/IP) for error detection

**49. List the steps involved in creating the checksum.**

a. Divide the data into sections

b. Add the sections together using 1s complement arithmetic

c. Take the complement of the final sum, this is the checksum.

**50. What are the Data link protocols?**

Data link protocols are sets of specifications used to implement the data link layer. The categories of Data Link protocols are

Asynchronous Protocols

Synchronous Protocols

a. Character Oriented Protocols

b. Bit Oriented protocols

**51. Compare Error Detection and Error Correction:**

The correction of errors is more difficult than the detection. In error detection, checks only any error has occurred. In error correction, the exact number of bits that are corrupted and location in the message are known. The number of the errors and the size of the message are important factors.

**52. What is Forward Error Correction?**

Forward error correction is the process in which the receiver tries to guess the message by using redundant bits.

**53. Define Retransmission?**

Re transmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Re sending is repeated until a message arrives that the receiver believes is error-freed.

**54. What are Data Words?**

In block coding, we divide our message into blocks, each of k bits, called data words. The block coding process is one-to-one. The same data word is always encoded as the same code word.

**55. What are Code Words?**

r redundant bits are added to each block to make the length n = k + r. The resulting n-bit blocks are called code words. 2n 2k code words that are not used. These code words are invalid or illegal.

**56. What is a Linear Block Code?**

A linear block code is a code in which the exclusive OR (addition modulo-2) of two valid code words creates another valid code word.

**57. What are Cyclic Codes?**

Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a code word is cyclically shifted (rotated), the result is another code word.

**58. Define Encoder?**

A device or program that uses predefined algorithms to encode, or compress audio or video data for storage or transmission use. A circuit that is used to convert between digital video and analog video.

**59. Define Decoder?**

A device or program that translates encoded data into its original format (e.g. it decodes the data). The term is often used in reference to MPEG-2 video and sound data, which must be decoded before it is output.

**60. What is Framing?**

Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet has to go and the sender address helps the recipient acknowledge the receipt.

**61. What is Fixed Size Framing?**

In fixed-size framing, there is no need for defining the boundaries of the frames. The size itself can be used as a delimiter.

**62. Define Character Stuffing?**

In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This

byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

**63. What is Bit Stuffing?**

Bit stuffing is the process of adding one extra 0 whenever five consecutive Is follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

**64. What is Flow Control?**

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

**65. What is Error Control ?**

Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term error control refers primarily to methods of error detection and re transmission.

**66. What Automatic Repeat Request (ARQ)?**

Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term error control refers primarily to methods of error detection and retransmission. Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).

**67. What is Stop-and-Wait Protocol?**

In Stop and wait protocol, sender sends one frame, waits until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame.

**68. What is Stop-and-Wait Automatic Repeat Request?**

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

**69. What is usage of Sequence Number in Reliable Transmission?**

The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame. Since we want to minimize

the frame size, the smallest range that provides unambiguous communication. The sequence numbers can wrap around.

**70. What is Pipelining?**

In networking and in other areas, a task is often begun before the previous task has ended. This is known as pipelining.

**71. What is Sliding Window?**

The sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver. In other words, he sender and receiver need to deal with only part of the possible sequence numbers.

**72. What is Piggy Backing?**

A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

**73. What are the two types of transmission technology available?**

(i) Broadcast and (ii) point-to-point

**74. What is subnet?**

A generic term for section of a large networks usually separated by a bridge or router.

**75. Difference between the communication and transmission.**

Transmission is a physical movement of information and concern issues like bit polarity, synchronization, clock etc.

Communication means the meaning full exchange of information between two communication media.

**76. What are the possible ways of data exchange?**

(i) Simplex (ii) Half-duplex (iii) Full-duplex.

**77. What is SAP?**

Series of interface points that allow other computers to communicate with the other layers of network protocol stack.

**78. What do you meant by "triple X" in Networks?**

The function of PAD (Packet Assembler Disassembler) is described in a document known as X.3. The standard protocol has been defined between the terminal and the PAD, called X.28; another standard

protocol exists between hte PAD and the network, called X.29. Together, these three recommendations are often called “triple X”.

**79. What is frame relay, in which layer it comes?**

Frame relay is a packet switching technology. It will operate in the data link layer.

**80. What is terminal emulation, in which layer it comes?**

Telnet is also called as terminal emulation. It belongs to application layer.

**81. What is Beaconing?**

The process that allows a network to self-repair networks problems. The stations on the network notify the other stations on the ring when they are not receiving the transmissions. Beaconing is used in Token ring and FDDI networks.

**82. What is redirector?**

Redirector is software that intercepts file or prints I/O requests and translates them into network requests. This comes under presentation layer.

**83. What is NETBIOS and NETBEUI?**

- **NETBIOS** is a programming interface that allows I/O requests to be sent to and received from a remote computer and it hides the networking hardware from applications
- **NETBEUI** is NetBIOS extended user interface. A transport protocol designed by microsoft and IBM for the use on small subnets

**84. What is RAID?**

A method for providing fault tolerance by using multiple hard disk drives.

**85. What is passive topology?**

When the computers on the network simply listen and receive the signal, they are referred to as passive because they don’t amplify the signal in any way. Example for passive topology -linear bus.

**86. What is Brouter?**

Hybrid devices that combine the features of both bridges and routers.

**87. What is cladding?**

A layer of a glass surrounding the center fiber of glass inside a fiber-optic cable.

**88. What is point-to-point protocol?**

A communications protocol used to connect computers to remote networking services including Internet service providers.

**89. How Gateway is different from Routers?**

A gateway operates at the upper levels of the OSI model and translates information between two completely different network architectures or data formats.

**90. What is attenuation?**

The degeneration of a signal over distance on a network cable is called attenuation.

**91. What is MAC address?**

The address for a device as it is identified at the Media Access Control (MAC) layer in the network architecture. MAC address is usually stored in ROM on the network adapter card and is unique.

**92. Difference between bit rate and baud rate.**

Bit rate is the number of bits transmitted during one second whereas baud rate refers to the number of signal units per second that are required to represent those bits.

baud rate = (bit rate / N)

where N is no-of-bits represented by each signal shift.

**93. What is Bandwidth?**

Every line has an upper limit and a lower limit on the frequency of signals it can carry. This limited range is called the bandwidth.

**94. What are the types of Transmission media?**

Signals are usually transmitted over some transmission media that are broadly classified in to two categories.

a.) Guided Media: These are those that provide a conduit from one device to another that include twisted-pair, coaxial cable and fiber-optic cable. A signal traveling along any of these media is directed and is contained by the physical limits of the medium. Twisted-pair and coaxial cable use metallic that accept and transport signals in the form of electrical current. Optical fiber is a glass or plastic cable that accepts and transports signals in the form of light.

b.) Unguided Media: This is the wireless media that transport electromagnetic waves without using a physical conductor. Signals are broadcast either through air. This is done through radio communication, satellite communication and cellular telephony.

**95. What is Protocol Data Unit?**

The data unit in the LLC level is called the protocol data unit (PDU). The PDU contains of four fields a destination service access point (DSAP), a source service access point (SSAP), a control field and an

information field. DSAP, SSAP are addresses used by the LLC to identify the protocol stacks on the receiving and sending machines that are generating and using the data. The control field specifies whether the PDU frame is a information frame (I – frame) or a supervisory frame (S – frame) or a unnumbered frame (U – frame).

**96. What are the different type of networking / internetworking devices?**

**1. Repeater:** Also called a regenerator, it is an electronic device that operates only at physical layer. It receives the signal in the network before it becomes weak, regenerates the original bit pattern and puts the refreshed copy back in to the link.

**2. Bridges:** These operate both in the physical and data link layers of LANs of same type. They divide a larger network in to smaller segments. They contain logic that allow them to keep the traffic for each segment separate and thus are repeaters that relay a frame only the side of the segment containing the intended recipient and control congestion.

**3. Routers:** They relay packets among multiple interconnected networks (i.e. LANs of different type). They operate in the physical, data link and network layers. They contain software that enables them to determine which of the several possible paths is the best for a particular transmission.

**4. Gateways:** They relay packets among networks that have different protocols (e.g. between a LAN and a WAN). They accept a packet formatted for one protocol and convert it to a packet formatted for another protocol before forwarding it. They operate in all seven layers of the OSI model.

**97. What is ICMP?**

ICMP is Internet Control Message Protocol, a network layer protocol of the TCP/IP suite used by hosts and gateways to send notification of datagram problems back to the sender. It uses the echo test / reply to test whether a destination is reachable and responding. It also handles both control and error messages.

**98. What is the minimum and maximum length of the header in the TCP segment and IP datagram?**

The header should have a minimum length of 20 bytes and can have a maximum length of 60 bytes.

**99. What are major types of networks and explain?**

- Server-based network: provide centralized control of network resources and rely on server computers to provide security and network administration
- Peer-to-peer network: computers can act as both servers sharing resources and as clients using the resources