Lab Manual

ComputerNetworksLab

IIIYearB.TechISemester(R22)

DepartmentofComputerScienceandEngineering

DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING

VISION

To foster collaborative and diverse community of Artificial Intelligence and Machine Learning experts who work together to advance the state of the art and address major societal challenges.

MISSION

To evolve as centre for academic excellence in learning through creative and modern teaching practices.

PROGRAMEDUCATIONALOBJECTIVES(PEOs)

PEO1:Have Knowledge and analytical skills including Mathematics, Science & basic Engineering.

PEO2:Graduates will be able to work effectively in cross-functional teams to develop Artificial Intelligence and Machine Learning solutions that meet business objectives & societal needs.

PEO3:Have extensive knowledge in state of art frame works in Artificial Intelligence and design industry acceptedAI solutions using modern tools.

PROGRAMSPECIFICOUTCOMES(PSOs)

PSO1: Understandingofstatisticalconceptsandtheirapplicationsin Machinelearning..

PSO2:Familiaritywithnaturallanguageprocessingandits applicationsinareassuchas sentiment analysis and language translation.

PSO3:AdoptnewandfastemergingtechnologiesinArtificialIntelligenceandMachine Learning.

ListofExperiments:

1. Implementthedatalinklayerframingmethodssuchascharactercount,character-stuffing and bit stuffing.
2. WriteaprogramtocomputeCRCcodeforthepolynomialsCRC-12,CRC-16andCRC CCIP
3. Developasimpledatalinklayerthatperformstheflowcontrolusingtheslidingwindow protocol, and loss recovery using the Go-Back-N mechanism.
4. ImplementDijkstra’salgorithmtocomputetheshortestpaththrougha network
5. Takeanexamplesubnet ofhostsand obtainabroadcast treeforthe subnet.
6. Implementdistancevectorroutingalgorithmforobtainingroutingtablesateachnode.
7. Implementdataencryption anddata decryption
8. Writeaprogramfor congestion controlusing Leakybucket algorithm.
9. Writeaprogramforframesortingtechniquesusedinbuffers.
10. Wireshark
    - i. PacketCaptureUsing Wireshark
    - ii. StartingWireshark
    - iii. ViewingCapturedTraffic
    - iv. AnalysisandStatistics&Filters.
11. HowtorunNmap scan
12. OperatingSystemDetectionusingNmap
13. Do thefollowing using NS2 Simulator
    - i. NS2Simulator-Introduction
    - ii. SimulatetoFind theNumber ofPacketsDropped
    - iii. Simulateto FindtheNumberofPacketsDropped by TCP/UDP
    - iv. Simulateto Findthe Numberof PacketsDropped dueto Congestion
    - v. SimulatetoCompareDataRate&Throughput
    - vi. SimulatetoPlotCongestionforDifferentSource/Destination
    - vii. SimulatetoDeterminethePerformancewithrespecttoTransmissionof Packets

Experiment-1

Aim: Implementthedatalinklayerframingmethodssuchascharactercount,character- stuffing and bit stuffing.

Program:Charactercount

```
int get_input();
voidmake_frames(int);
int count_chars(int s);
void main()
{
        intno_of_words=get_input();
        make_frames(no_of_words);
}
int get_input()
{
        intanswer;
        int i=0;do{
                 printf("\nEntertheWord:");
                 scanf("%s",input[i]);
                 fflush(stdin);
                printf("\nDoyouwanttocontinue:(y:1/n:0)?:");
                 scanf("%d",&answer);
                i++;
           }while(answer!=0);
           return i;
 }
voidmake_frames(int num_words)
{
         int i=0;
        for(i=0;i<num_words;i++)
                printf("%d%s",(count_chars(i)+1),input[i]);
        printf("\n\n");
}
```

```
 intcount_chars(int index)
{
            int i=0;
            while(input[index][i]!='\0')
                        i++;
            returni;
 }
```

Inputand Output:

```
EntertheWord:cat
Doyouwanttocontinue:(y:1/n:0)?:1 Enter
the Word: dog
Doyouwanttocontinue:(y:1/n:0)?:1 Enter
the Word: apple
Do you want to continue: (y: 1/n: 0)?:0
TheTransmittedDatais:4cat4dog6apple
```

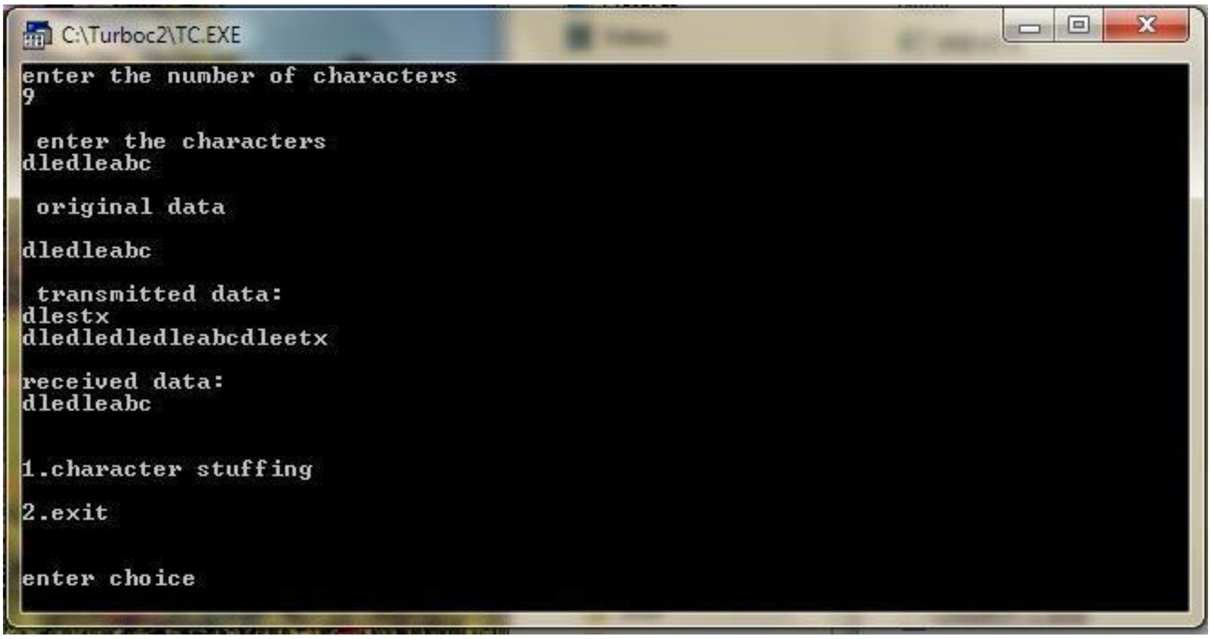
Program:Characterstuffing

```
voidcharc(void);
void main()
{
            intchoice;
              while(1)
            {
                        printf("\n\n\n1.characterstuffing");
                        printf("\n\n2.exit");
                        printf("\n\n\nenter choice");
                        scanf("%d",&choice);
                        printf("%d",choice);
                        if(choice>2)
                                    printf("\n\ninvalidoption..... pleaserenter");
                        switch(choice)
                        {
                                    case 1:
```

```
                charc();
                break;
            case 2:
                exit(0);
        }
    }
}

void charc(void)
{
    charc[50],d[50],t[50];
    int i,m,j;
    printf("enterthenumberofcharacters\n"); scanf("%d",&m);
    printf("\nenterthecharacters\n");
    for(i=0;i<m+1;i++)
    {
        scanf("%c",&c[i]);
    }
printf("\noriginaldata\n");
for(i=0;i<m+1;i++)
printf("%c",c[i]);
d[0]='d';
d[1]='l';
d[2]='e';
d[3]='s';
d[4]='t';
d[5]='x';
for(i=0,j=6;i<m+1;i++,j++)
 {
if((c[i]=='d'&&c[i+1]=='l'&&c[i+2]=='e'))
{
d[j]='d';    j++;
d[j]='l';    j++;
d[j]='e';    j++;
```

```
m=m+3;
}
d[j]=c[i];
}
m=m+6;
m++;
d[m]='d';
m++;
d[m]='l';
m++;
d[m]='e';
m++;
d[m]='e';
m++;
d[m]='t';
m++;
d[m]='x';
m++;
printf("\n\ntransmitteddata:\n");
for(i=0;i<m;i++)
{
printf("%c",d[i]);
}
for(i=6,j=0;i<m-6;i++,j++)
 {
if(d[i]=='d'&&d[i+1]=='l'&&d[i+2]=='e'&&d[i+3]=='d'&&d[i+4]=='l'&&d[i+5]=='e')
i=i+3;
t[j]=d[i];
}
printf("\n\nreceiveddata:");
for(i=0;i<j;i++)
 {
          printf("%c",t[i]);
   }
}
```

Inputand Output:



```
enter the number of characters
9

 enter the characters
dledleabc

 original data

dledleabc

 transmitted data:
dlestx
dledledledleabcdleetx

received data:
dledleabc


1.character stuffing

2.exit


enter choice
```

Program:Bitstuffing.

```
int main()
{
   int
   a[20],b[30],i,j,k,count,n;printf("Enterfra
   mesize(Example:8):"); scanf("%d",&n);
   printf("Entertheframeintheformof0and1:");
   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("AfterBitStuffing:");
    for(i=0; i<j; i++)
        printf("%d",b[i]);
    return 0;
}
```

Inputand Output:

```
Enterframesize (Example:8):12
Entertheframeintheformof0and1:010111111001 After Bit
Stuffing :0101111101001
```

Experiment-2

Aim: WriteaprogramtocomputeCRCcodeforthepolynomialsCRC-12,CRC-16andCRC CCIP

Program:

```
#define<stdio.h>
#define<string.h>#defin
eNstrlen(g)

chart[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()
{
   intflag=0;
   do{
       printf("\n1.crc12\n2.crc16\ncrcccit\n4.exit\n\nEnteryouroption.");
       scanf("%d",&b);
       switch(b)
       {
              case 1:strcpy(g,"1100000001111");
              break;
              case2:strcpy(g,"11000000000000101");
              break;
              case3:strcpy(g,"10001000000100001");
              break;
              case4: return 0;
       }
       printf("\nenterdata:"); scanf("%s",t);
       printf("\n----------------\n");
       printf("\ngeneratingpolynomial:%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("checksumis:%s",cs);
       for(e=a;e<a+N-1;e++)
              t[e]=cs[e-a];
       printf("\n----------------\n");
       printf("\nfinalcodewordis:%s",t);
       printf("\n-----------------\n");
       printf("\ntesterrordetection0(yes)1(no)?:");
       scanf("%d",&e);
```

```
        if(e==0)
        {
                do
                {
                printf("\n\tenterthepositionwhereerroristobeinserted:"); scanf("%d",&e);
                }while(e==0||e>a+N-1);

                t[e-1]=(t[e-1]=='0')?'1':'0';
                printf("\n---------------\n");
                printf("\n\terroneousdata:%s\n",t);
        }

        crc();
        for(e=0;(e<N-1)&&(cs[e]!='1');e++);
        if(e<N-1)
                printf("errordetected\n\n");
        else
                printf("\nnoerrordetected\n\n");
        printf("\n---------------");
        }while(flag!=1);
}
```

Inputand Output:

```
1.crc12
2.crc16
3.crcccit
4.exit
Enteryouroption.1
enter data:1100110011100011
generating polynomial:1100000001111
mod-ified data
is:110011001110001100000000000001100000001111
```

```
checksumis:1101110110001
final Codeword is :
11001100111000111101110110001100000001111
Testerrordetection0(yes)1(no)?:1
No error detected

1.crc12
2.crc16
3.crcccit
4.exit

Enteryouroption.2
enter data:11001100111000
generating
polynomial:11000000000000101
modifieddatais:11001100111000
```

Experiment-3

Aim: Developasimpledatalinklayerthatperformstheflowcontrolusingtheslidingwindow protocol, and loss recovery using the Go-Back-N mechanism.

Program:

```
 #include<stdio.h>vo
 idmain()
 {
         int w, i, f, frames[50]; printf("\nEnter
         thewindowsize:");scanf("%d",&w);
         printf("\nEnterthenumberofframestotransmit:");
         scanf("%d",&f);
         printf("\nEnter%dframes:",f);
         for(i=1;i<=f;i++)
                 scanf("%d",&frames[i]);

         printf("\nWithslidingwindowprotocol,theframeswillbesentasshownbelow");
         printf("\nAfter sending the %d frames, at each stage sender waits forAck by the
         receiver",f);
         for(i=1;i<=f;i++)
         {
                 if((i%w)==0)
                 {
                         printf("\n%d",frames[i]);
                         printf("\nAckofaboveframessentisreceivedbysender");

                 }
                  else
                         printf("\n%d",frames[i]);

         }
```

```
        if(f%w!=0)
                printf("\nAckofaboveframessentisreceivedby sender");
        return;
}
```

Inputand Output:

```
Enterthewindow size:3
Enterthenumberofframestotransmit:5 Enter 5
frames: 6
23
6
5
11
Withslidingwindowprotocol,theframeswillbesentasshown below
 Aftersendingthe5frames,ateachstagesenderwaitsforAckbythereceiver 6
23
6
 Ackofaboveframessentisreceivedbysender 5
11
Ackofaboveframessentisreceivedbysender
```

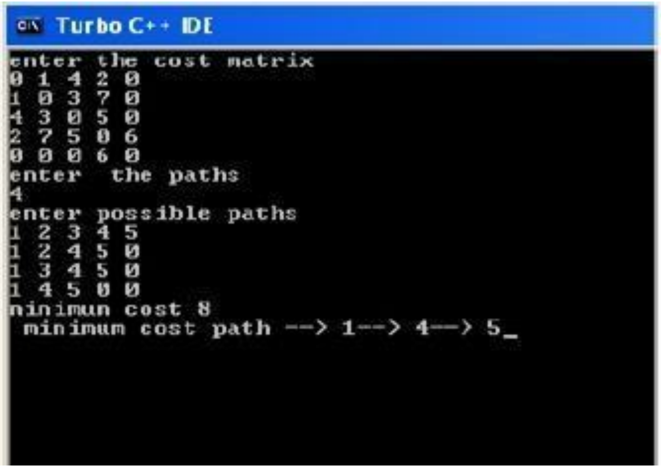
Experiment-4

Aim: ImplementDijkstra'salgorithm tocomputetheshortest paththroughanetwork

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int path[5][5],i,j,min,a[5][5],p,st=1,ed=5,stp,edp,t[5],index;
clrscr();
printf("enterthecostmatrix\n");
for(i=1;i<=5;i++)
for(j=1;j<=5;j++)
scanf("%d",&a[i][j]);
printf("enter the paths\n");
scanf("%d",&p);
printf("enterpossiblepaths\n");
for(i=1;i<=p;i++)
for(j=1;j<=5;j++)
scanf("%d",&path[i][j]);
for(i=1;i<=p;i++)
{
t[i]=0; stp=st;
for(j=1;j<=5;j++)
{
edp=path[i][j+1];
t[i]=t[i]+a[stp][edp];
if(edp==ed)
break;
else
stp=edp;
}
}min=t[st];
index=st;
for(i=1;i<=p;i++)
{
if(min>t[i])
{
min=t[i];
index=i;
}
}
printf("minimumcost%d",min);
printf("\nminimumcostpath");
for(i=1;i<=5;i++)
{
printf("-->%d",path[index][i]);
if(path[index][i]==ed)
break;
}
getch();
}
```

Inputand Output:



```
enter the cost matrix
0 1 4 2 0
1 0 3 7 0
4 3 0 5 0
2 7 5 0 6
0 0 0 6 0
enter  the paths
4
enter possible paths
1 2 3 4 5
1 2 4 5 0
1 3 4 5 0
1 4 5 0 0
minimun cost 8
 minimum cost path --> 1--> 4--> 5_
```

Experiment-5

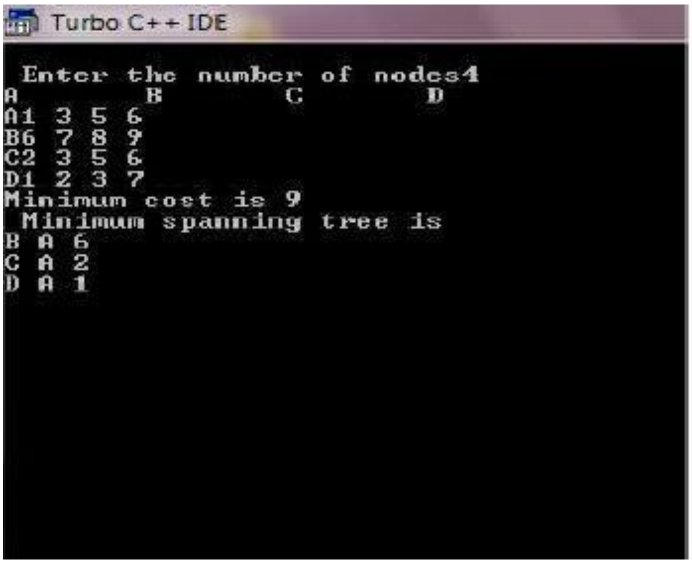
Aim: Take anexamplesubnet ofhostsandobtainabroadcasttree forthesubnet.

Program:

```
#include<stdio.h>
#include<conio.h>in
t 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("Minimumcost 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(intl,intm)
{
parent[l]=m;
}
find(intl)
{
if(parent[l]>0)
i=parent[i];
return i;
}
```

Inputand Output:



```
 Enter the number of nodes4
A            B            C            D
A1 3 5 6
B6 7 8 9
C2 3 5 6
D1 2 3 7
Minimum cost is 9
 Minimum spanning tree is
B A 6
C A 2
D A 1
```

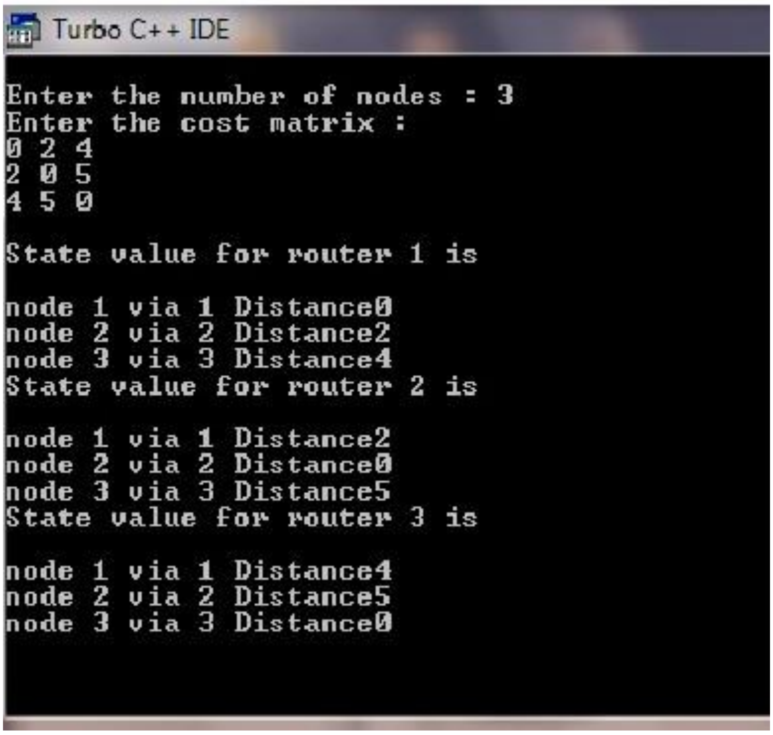
Experiment-6

Aim: Implementdistancevector routing algorithmforobtainingroutingtablesateach node.

Program:

```
#include<stdio.h>
#include<conio.h>st
ruct node
{
unsigned dist[20];
unsignedfrom[20];
}rt[10];
intmain()
{
int dmat[20][20]; int n,i,j,k,count=0; clrscr();
printf("\nEnterthe number of nodes : ");
scanf("%d",&n);printf("Enter the cost matrix
:\n"); for(i=0;i<n;i++)
for(j=0;j<n;j++)
{
scanf("%d",&dmat[i][j]);dmat[i][i]=0;rt[i].dist[j]=dmat[i][j];rt[i].from[j]=j;
}
do
{ count=0; for(i=0;i<n;i++) for(j=0;j<n;j++) for(k=0;k<n;k++)
if(rt[i].dist[j]>dmat[i][k]+rt[k].dist[j])
{
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<n;i++)
{
printf("\nStatevalueforrouter%dis\n",i+1);
for(j=0;j<n;j++)
{
printf("\nnode%dvia%dDistance%d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
}
}
printf("\n");
}
```

Inputand Output:



```
Enter the number of nodes : 3
Enter the cost matrix :
0 2 4
2 0 5
4 5 0

State value for router 1 is

node 1 via 1 Distance0
node 2 via 2 Distance2
node 3 via 3 Distance4
State value for router 2 is

node 1 via 1 Distance2
node 2 via 2 Distance0
node 3 via 3 Distance5
State value for router 3 is

node 1 via 1 Distance4
node 2 via 2 Distance5
node 3 via 3 Distance0
```

Experiment-7

Aim: Implementdata encryptionanddata decryption

Program:

```
 #include<stdio.h>
 int main()
 {
   int i, x;
   charstr[100];
   printf("\nPleaseenterastring:\t");
   gets(str);
   printf("\nPleasechoosefollowingoptions:\n");
   printf("1=Encryptthestring.\n");printf("2=Dec
   ryptthestring.\n");
   scanf("%d",&x);
   switch(x)
   {
   case1:
     for(i =0; (i <100 &&str[i]!='\0'); i++)
      str[i]=str[i]+3;//thekeyforencryptionis3that isadded toASCIIvalue printf("\nEncrypted
     string: %s\n", str);
     break;
   case 2:
     for(i =0; (i <100 &&str[i]!='\0'); i++)
      str[i]=str[i]-3;//thekeyforencryptionis3that issubtracted toASCIIvalue printf("\nDecrypted
     string: %s\n", str);
     break;default:
     printf("\nError\n");
      }
    return 0;
   }
```

Inputand Output:

```
Pleaseenterastring:cmrtcPlease
choose following options: Please
enter a string: vitsaiml Please
choose following options:
1=Encryptthestring.
2=Decryptthestring.
1
Encryptedstring:ylwvdlpoPlease
enter a string: ylwvdlpo Please
choose following options:
1=Encryptthestring.
2=Decryptthestring.
2
Decryptedstring: vitsaiml
```

Experiment-8

Aim: Write aprogramfor congestioncontrolusing Leakybucket algorithm.

Program:

```
#include<stdio.h>
#include<stdlib.h>st
ruct packet
{
    int time;
      intsize;
}p[50];

int main()
{
    int i,n,m,k=0;
 int bsize,bfilled,outrate;
 printf("Enterthenumberofpackets:");
 scanf("%d",&n);
 printf("Enterpacketsintheorderoftheyarearrivaltime\n");
 for(i=0;i<n;i++)
 {
     printf("Enterthetimeandsize: ");

     scanf("%d%d",&p[i].time,&p[i].size);
 }
 printf("Enter the bucket size: ");
 scanf("%d",&bsize);printf("Enterthe
 outputrate:");scanf("%d",&outrate);
 m=p[n-1].time;
 i=1;k=0;
 bfilled=0;
```

```
while(i<=m||bfilled!=0)
{
   printf("\n\nAttime%d",i);
   if(p[k].time==i )
   {
      if(bsize>=bfilled+p[k].size)
      {
         bfilled=bfilled+p[k].size;
         printf("\n%dbytepacketisinserted",p[k].size);
         k=k+1;
      }
      else
      {
         printf("\n%dbytepacketisdiscarded",p[k].size); k=k+1;
      }
   }
   if(bfilled==0)
   {
      printf("\nNopacketstotransmitte");
   }
   else if(bfilled>=outrate)
   {
      bfilled=bfilled-outrate;
      printf("\n%dbytestransfered",outrate);
      }
      else
      {
         printf("\n%dbytestransfered",bfilled);
         bfilled=0;
      }
```

```
    printf("\nPacketsinthebucket%dbyte",bfilled); i++;
  }
  return 0;
  }
```

Inputand Output:

```
Enterthenumberofpackets: 2
Enterpacketsintheorderoftheyarearrivaltime Enter
the time and size: 2 3
Enterthetimeandsize:54
Enter the bucket size: 3
Enter the output rate: 2

Attime1
No packets to transmitted
Packetsinthebucket0byte At
time 2
3bytepacketisinserted 2
bytes transferred
Packetsinthebucket1byte At
time 3
1bytes transferred
Packetsinthebucket0byte At
time 4
No packets to transmitted
Packetsinthebucket0byte At
time 5
4 byte packet is discarded No
packetstotransmittedPackets in
the bucket 0 byte
```

Experiment-9

Aim: Writeaprogram forframe sortingtechniquesusedin buffers.

Program:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>st
ruct frame{
int fslno;
charfinfo[20];
};
structframearr[10];
int n;
void sort()
{
 int i,j,ex;
 structframetemp;
 for(i=0;i<n;i++)
 {
 ex=0;
 for(j=0;j<n-i-1;j++)
 if(arr[j].fslno>arr[j+1].fslno)
 { temp=arr[j];
 arr[j]=arr[j+1];
 arr[j+1]=temp;
 ex++;
 }
 if(ex==0)break;
 }
}
```

```
void main()
{
 int i;
 clrscr();
 printf("\nEnterthenumberofframes\n");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 { arr[i].fslno=random(50);
 printf("\nEntertheframecontentsforsequence number
%d\n",arr[i].fslno);
 scanf("%s",arr[i].finfo);
 }
 sort();
 printf("\nTheframesinsequence\n");
 for(i=0;i<n;i++)
 printf("\n%d\t%s\n",arr[i].fslno,arr[i].finfo);
 getch();
}
```

Inputand Output:

```
Enterthenumberofframes:3
Entertheframecontentsforsequencenumber23 Lab
Entertheframecontentsforsequencenumber45 Program
Entertheframecontentsforsequencenumber9
Networks
Entertheframecontentsforsequencenumber2
Computer
```

Theframesinsequence 2

Computer

9 Networks

23 Lab

45 Program

Experiment-10

Aim: Understand the working ofWireshark

i. PacketCaptureUsing Wireshark

ii. StartingWireshark

iii. ViewingCapturedTraffic

iv. AnalysisandStatistics&Filters.

Implementation:

Wiresharkis anetworkprotocol analyser, oran application that captures packets from anetworkconnection,suchasfromyourcomputertoyourhomeofficeorthe internet.Packet is the name given to a discrete unit of data in a typical Ethernet network.

Wireshark is the most often-used packet sniffer in the world. Like any other packet sniffer, Wireshark does three things:

1. **PacketCapture:**Wiresharklistenstoanetworkconnectioninrealtimeandthengrabs entire streams of traffic – quite possibly tens of thousands of packets at a time.
2. **Filtering:**Wiresharkiscapableofslicinganddicingallofthisrandomlivedatausing filters. By applying a filter, you can obtain just the information you need to see.
3. **Visualization:**Wireshark, like any good packet sniffer, allows you to dive right into theverymiddleofanetworkpacket.Italsoallowsyoutovisualizeentireconversations and network streams.

Packetsniffingcanbecomparedtospelunking–goinginsideacaveandhikingaround. Folks who use Wireshark on a network are kind of like those who use flashlights to see what cool things they can find. After all, when using Wireshark on a network connection (or a flashlight in a cave), you’re effectively using a tool to hunt around tunnels and tubes to see what you can see.

WhatIsWiresharkUsedFor?

Wireshark has many uses, includingtroubleshooting networksthat have performance issues. Cybersecurity professionals often use Wireshark to trace connections, view the contents of suspect network transactions and identify bursts of network traffic. It’s a major part of any IT pro’s toolkit – and hopefully, the IT pro has the knowledge to use it.

WhenShouldWiresharkBeUsed?

Wireshark is a safe tool used by government agencies, educational institutions, corporations, small businesses and nonprofits alike to troubleshoot network issues. Additionally,Wiresharkcanbeusedasalearningtool.Thosenewtoinformationsecuritycan useWiresharkasatooltounderstandnetworktrafficanalysis,howcommunicationtakesplace whenparticularprotocolsareinvolvedandwhereitgoeswrongwhencertainissuesoccur.Of course, Wireshark can’t do everything.

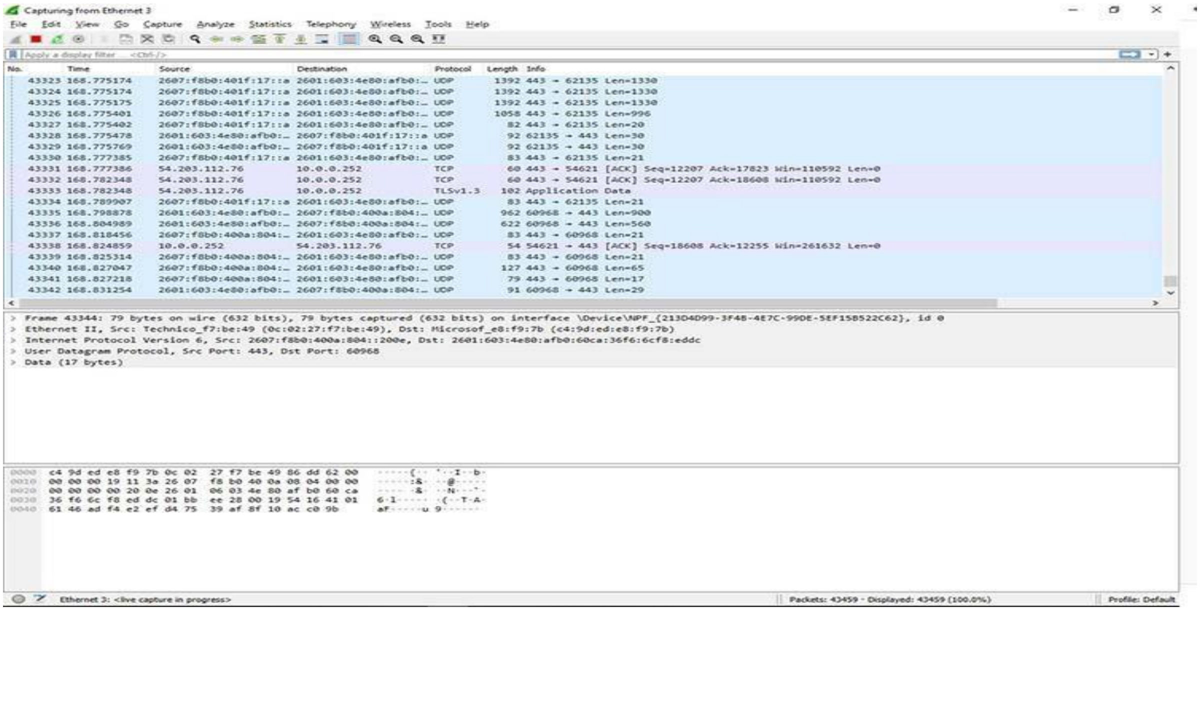
First of all, it can’t help a user who has little understanding ofnetwork protocols.No tool, no matter how cool, replaces knowledge very well. In other words, to properly use Wireshark, you need to learn exactly how a network operates.

Second,Wiresharkcan’tgrabtrafficfromalloftheothersystemsonthenetworkunder normal circumstances. On modern networks that use devices called switches, Wireshark (or any other standard packet-capturing tool) can only sniff traffic between your local computer and the remote system it is talking to.

Third,whileWiresharkcanshowmalformedpacketsandapplycolorcoding,itdoesn’t have actual alerts; Wireshark isn’t an intrusion detection system (IDS).

Fourth, Wireshark can’t help with decryption with regards to encrypted traffic. And finally, it is quite easy to spoof IPv4 packets.Wireshark can’t really tell you if a particular IP addressitfindsinacapturedpacketisarealoneornot.Thatrequiresabitmoreknow-howon the part of an IT pro, as well as additional software.

Result: ViewingapacketcaptureinWireshark



```
No.    Time        Source                  Destination             Protocol Length Info
 43323 168.775174  2607:f8b0:401f:17::a    2601:603:4e80:afb0:…    UDP       1392 443 → 62135 Len=1330
 43324 168.775174  2607:f8b0:401f:17::a    2601:603:4e80:afb0:…    UDP       1392 443 → 62135 Len=1330
 43325 168.775175  2607:f8b0:401f:17::a    2601:603:4e80:afb0:…    UDP       1392 443 → 62135 Len=1330
 43326 168.775401  2607:f8b0:401f:17::a    2601:603:4e80:afb0:…    UDP       1058 443 → 62135 Len=996
 43327 168.775402  2607:f8b0:401f:17::a    2601:603:4e80:afb0:…    UDP         82 443 → 62135 Len=20
 43328 168.775478  2601:603:4e80:afb0:…    2607:f8b0:401f:17::a    UDP         92 62135 → 443 Len=30
 43329 168.775769  2601:603:4e80:afb0:…    2607:f8b0:401f:17::a    UDP         92 62135 → 443 Len=30
 43330 168.777385  2607:f8b0:401f:17::a    2601:603:4e80:afb0:…    UDP         83 443 → 62135 Len=21
 43331 168.777386  54.203.112.76           10.0.0.252              TCP         60 443 → 54621 [ACK] Seq=12207 Ack=17823 Win=110592 Len=0
 43332 168.782348  54.203.112.76           10.0.0.252              TCP         60 443 → 54621 [ACK] Seq=12207 Ack=18608 Win=110592 Len=0
 43333 168.782348  54.203.112.76           10.0.0.252              TLSv1.3    102 Application Data
 43334 168.789907  2607:f8b0:401f:17::a    2601:603:4e80:afb0:…    UDP         83 443 → 62135 Len=21
 43335 168.798878  2601:603:4e80:afb0:…    2607:f8b0:400a:804:…    UDP        962 60968 → 443 Len=900
 43336 168.804989  2601:603:4e80:afb0:…    2607:f8b0:400a:804:…    UDP        622 60968 → 443 Len=560
 43337 168.818456  2607:f8b0:400a:804:…    2601:603:4e80:afb0:…    UDP         83 443 → 60968 Len=21
 43338 168.824859  10.0.0.252              54.203.112.76           TCP         54 54621 → 443 [ACK] Seq=18608 Ack=12255 Win=261632 Len=0
 43339 168.825314  2607:f8b0:400a:804:…    2601:603:4e80:afb0:…    UDP         83 443 → 60968 Len=21
 43340 168.827047  2607:f8b0:400a:804:…    2601:603:4e80:afb0:…    UDP        127 443 → 60968 Len=65
 43341 168.827218  2607:f8b0:400a:804:…    2601:603:4e80:afb0:…    UDP         79 443 → 60968 Len=17
 43342 168.831254  2601:603:4e80:afb0:…    2607:f8b0:400a:804:…    UDP         91 60968 → 443 Len=29
```

```
> Frame 43344: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface \Device\NPF_{213D4D99-3F48-4E7C-99DE-5EF158522C62}, id 0
> Ethernet II, Src: Technico_f7:be:49 (0c:02:27:f7:be:49), Dst: Microsof_e8:f9:7b (c4:9d:ed:e8:f9:7b)
> Internet Protocol Version 6, Src: 2607:f8b0:400a:804::200e, Dst: 2601:603:4e80:afb0:60ca:36f6:6cf8:eddc
> User Datagram Protocol, Src Port: 443, Dst Port: 60968
> Data (17 bytes)
```

```
0000  c4 9d ed e8 f9 7b 0c 02  27 f7 be 49 86 dd 62 00   ·····{·· '··I··b·
0010  00 00 00 19 11 3a 26 07  f8 b0 40 0a 08 04 00 00   ·····:&· ··@·····
0020  00 00 00 00 20 0e 26 01  06 03 4e 80 af b0 60 ca   ······&· ··N···`·
0030  36 f6 6c f8 ed dc 01 bb  ee 28 00 19 54 16 41 01   6·l····· ·(··T·A·
0040  61 46 ad f4 e2 ef d4 75  39 af 8f 10 ac c0 9b      aF·····u 9······
```

Ethernet 3: <live capture in progress>
Packets: 43459 · Displayed: 43459 (100.0%)
Profile: Default

Experiment-11

Aim: Howto runNmapscan

Implementation:

NMAPstandsforNetworkMapperwhichisanopen-sourcetoolusedfornetworkexploration and security auditing, in comparison to this, a tool named**Nessus**is used by industry professionals. These tools are mainly used by cybersecurity experts and hackers.
Itsmainpurposeis:

- Providethe list ofthelivehost.
- Findtheopen Ports.
- Thereal-timeinformation ofa network.
- OSand Port scanning.

Thehackers and the cybersecurity expert need to know the Operating System of themachine. Itbecomesveryeasyto accessasystemifwecan knowthespecificopenportsorthesecurity holes of the system. **Network Mapper(NMAP)**NMAP has a database that helps in**Operatingsystems(OS)**butitisnotautomaticallyupdated.ThedatabasetodetectanOS is located at '/usr/share/nmap/nmap-os-db'.

Operating System(OS)detection is a very long and hectic process. So, before we get our handsdirtyweshouldknowaboutthefiveseparateprobesbeingperformedtodeterminethe OS. This probe may consist of one or more packets. The response to each packet (which is sent by the probe) by the target system helps to determine the OS type.
Thefivedifferentprobesare:

- SequenceGeneration.
- ICMPEcho.
- TCPExplicitCongestionNotification.
- TCP.
- UDP.

1. SequenceGeneration: TheSequenceGenerationProbeconsistsofsixpacketsthatare sent 100 ms apart and are all TCP SYN packets. The result of all these packets will help in **Operating System(OS)**detection.

2. ICMPEcho: TwoICMPrequestpacketsaresenttothetargetsystemwithdifferent settings in the packet. The result of all these will help verify the OS type by NMAP.

3. TCP Explicit Congestion Notification: Congestion is a slowdown that occurs when a lot ofpacketsaregeneratedandpassedbyasinglerouter.Thepacketswhicharesentaremainly used to get back the responses from the target system. This helps to detect the OS because a specific OS returns a specific value and each OS handles a packet differently.

4. TCP: Sixpacketsaresentduringthisprobe,andsomepacketsaresenttoopenorclosed ports with specific packet settings by using the corresponding result we can determine the type of **Operating System(OS)**. The TCP Packets which are sent with varying flags are as follows:

- no flags.
- SYN,FIN,URG,andPSH.
- ACK.
- SYN.
- ACK.
- FIN,PSH,andURG.

5. UDP:UDPprobeconsistsofasinglepacketthatissenttoaclosedport. Iftheportusedon the target system is closed and an ICMP Port Unreachable message is returned it specifiesthat there is no Firewall.

Result: Theexperimentcompletedsuccessfully

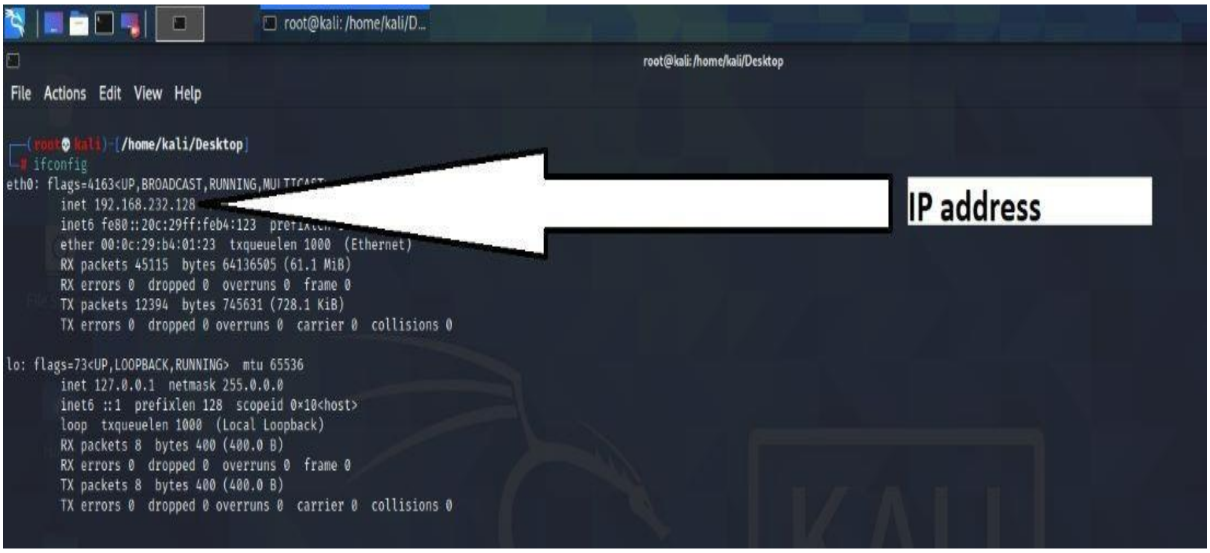
Experiment-12

Aim: UnderstandtheOperatingSystemDetectionusingNmap

Implementation:

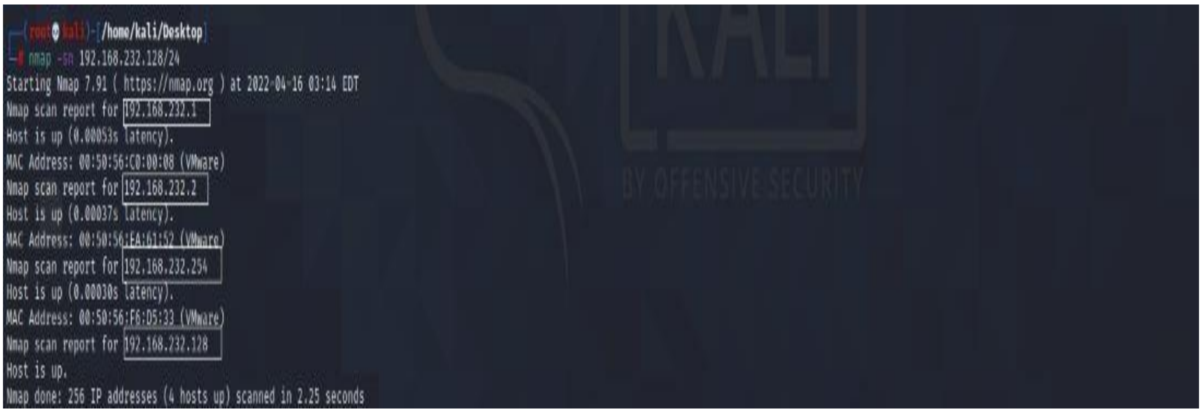
NowweneedtoruntheactualcommandstoperformOSdetectionusingNMAP,and at first, we will get the IP address of the host system, and then will perform a scan to get all active devices on the network.

Step 1: Gettingthe IP of theSystem-ifconfig


root@kali: /home/kali/D...
root@kali:/home/kali/Desktop
File Actions Edit View Help
IP address

```
┌──(root💀kali)-[/home/kali/Desktop]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST
        inet 192.168.232.128
        inet6 fe80::20c:29ff:feb4:123  prefixlen
        ether 00:0c:29:b4:01:23  txqueuelen 1000  (Ethernet)
        RX packets 45115  bytes 64136505 (61.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12394  bytes 745631 (728.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step2: ListofactivedevicesintheNetwork nmap
-sn 192.168.232.128/24

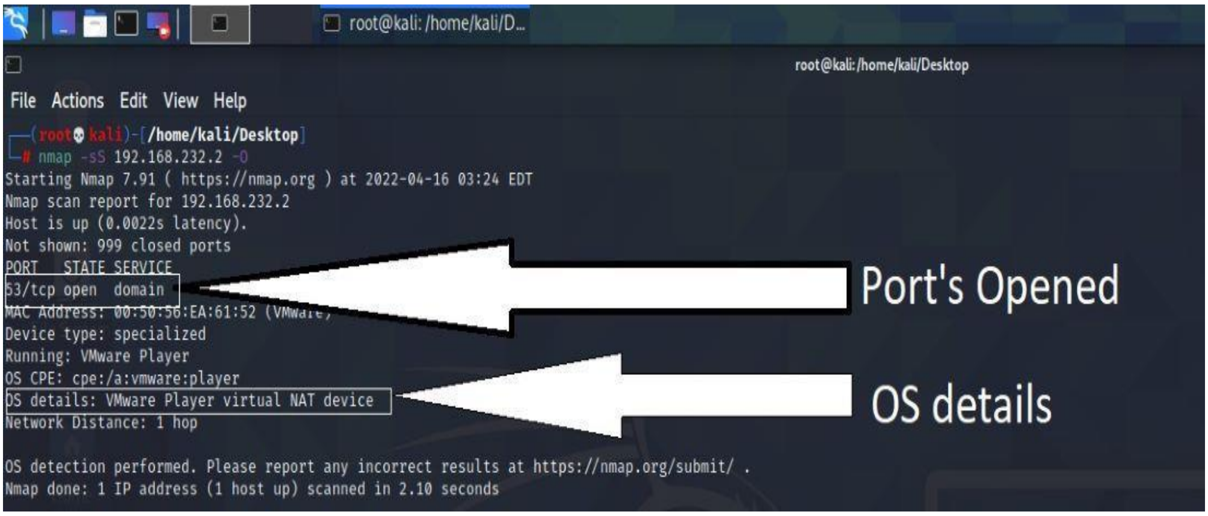


```
┌──(root💀kali)-[/home/kali/Desktop]
└─# nmap -sn 192.168.232.128/24
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-16 03:14 EDT
Nmap scan report for 192.168.232.1
Host is up (0.00053s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.232.2
Host is up (0.00037s latency).
MAC Address: 00:50:56:FA:61:52 (VMware)
Nmap scan report for 192.168.232.254
Host is up (0.00030s latency).
MAC Address: 00:50:56:F6:D5:33 (VMware)
Nmap scan report for 192.168.232.128
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.25 seconds
```

Let’sdoanSYNscanwith OS detectioninoneoftheactiveIPs

Let’sselectIP:192.168.232.2

nmap -sS 192.168.232.2 -O



```
┌──(root💀kali)-[/home/kali/Desktop]
└─# nmap -sS 192.168.232.2 -O
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-16 03:24 EDT
Nmap scan report for 192.168.232.2
Host is up (0.0022s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
53/tcp open  domain
MAC Address: 00:50:56:EA:61:52 (VMware)
Device type: specialized
Running: VMware Player
OS CPE: cpe:/a:vmware:player
OS details: VMware Player virtual NAT device
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.10 seconds
```

root@kali: /home/kali/D...
root@kali: /home/kali/Desktop
File Actions Edit View Help
Port's Opened
OS details

Running:VMwarePlayer.

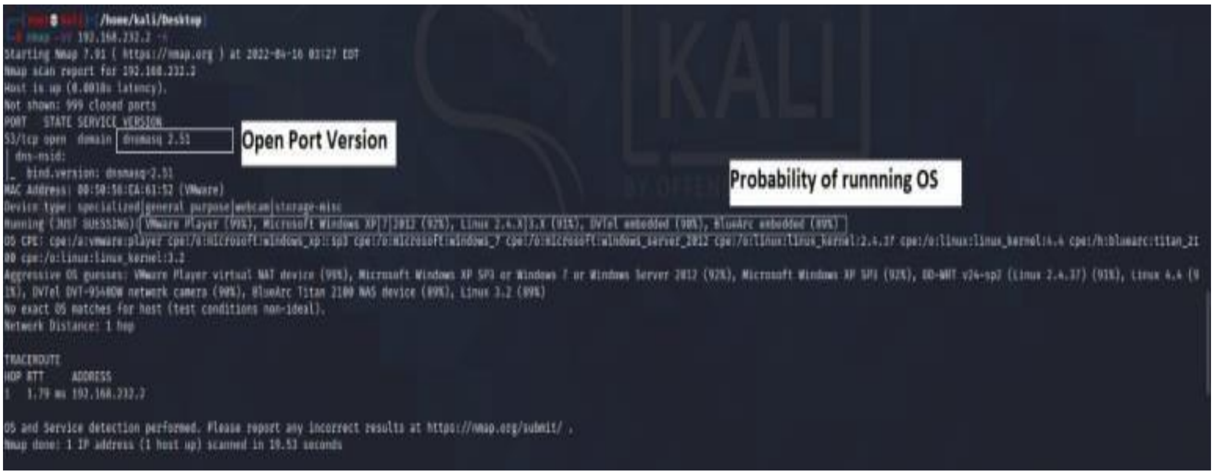
OSdetails:VMwarePlayervirtualNAT device.

Let’snowperformanAggressivescanToguesstheOS

- **-sV**standsforServiceversion.
- **-A**standsforAggressive.

ItwillonlydisplaythechanceofOperationSystem(OS)onthehostcomputerwiththehelp of Probability and Percentage.

nmap-sV192.168.232.2-A



```
┌──(root💀kali)-[/home/kali/Desktop]
└─# nmap -sV 192.168.232.2 -A
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-16 03:27 EDT
Nmap scan report for 192.168.232.2
Host is up (0.0018s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
53/tcp open  domain  dnsmasq 2.51
| dns-nsid:
|_  bind.version: dnsmasq-2.51
MAC Address: 00:50:56:EA:61:52 (VMware)
Device type: specialized|general purpose|webcam|storage-misc
Running (JUST GUESSING): VMware Player (99%), Microsoft Windows XP|7|2012 (92%), Linux 2.4.X|3.X (91%), DVTel embedded (90%), BlueArc embedded (89%)
OS CPE: cpe:/a:vmware:player cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2012 cpe:/o:linux:linux_kernel:2.4.37 cpe:/o:linux:linux_kernel:4.4 cpe:/h:bluearc:titan_2100 cpe:/o:linux:linux_kernel:3.2
Aggressive OS guesses: VMware Player virtual NAT device (99%), Microsoft Windows XP SP3 or Windows 7 or Windows Server 2012 (92%), Microsoft Windows XP SP3 (92%), DD-WRT v24-sp2 (Linux 2.4.37) (91%), Linux 4.4 (91%), DVTel DVT-9540DW network camera (90%), BlueArc Titan 2100 NAS device (89%), Linux 3.2 (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   1.79 ms 192.168.232.2

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 19.53 seconds
```

Result: Theexperimentcompletedsuccessfully

Experiment-13

Aim: Do thefollowing using NS2 Simulator

i. NS2Simulator-Introduction

ii. SimulatetoFind theNumber ofPacketsDropped

iii. Simulateto FindtheNumberofPacketsDropped by TCP/UDP

iv. Simulateto Findthe Numberof PacketsDropped dueto Congestion

v. SimulatetoCompareDataRate&Throughput

vi. SimulatetoPlotCongestionforDifferentSource/Destination

vii. SimulatetoDeterminethePerformancewithrespecttoTransmissionof Packets

NS2Simulator-Introduction

WhatisNS2

NS2standsforNetworkSimulatorVersion2.Itisanopen-sourceevent-drivensimulator designed specifically for research in computer communication networks.

FeaturesofNS2

1. Itisadiscrete eventsimulatorfornetworkingresearch.

2. ItprovidessubstantialsupporttosimulatebunchofprotocolslikeTCP,FTP,UDP,httpsand DSR.

3. Itsimulateswiredand wirelessnetwork.

4. ItisprimarilyUnixbased.

5. UsesTCLasitsscriptinglanguage.

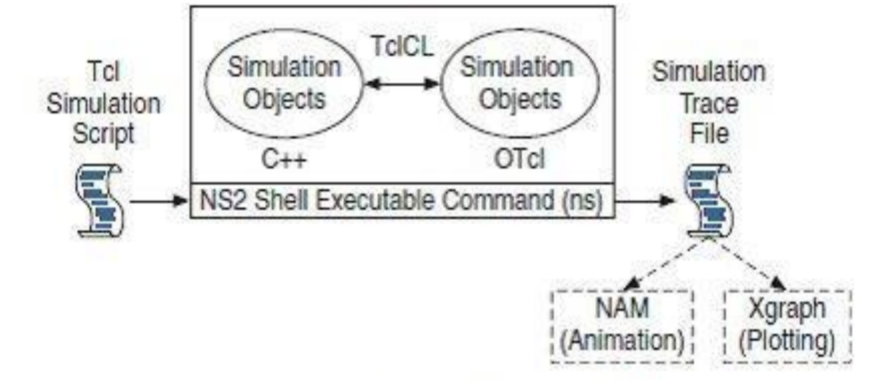
6. Otcl:Objectorientedsupport

7. Tclcl:C++andotcllinkage

8. Discreteevent scheduler

BasicArchitecture

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.

TCLand C++

NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks. NS2 uses OTcl to create and configure a network, and uses C++ to run simulation. All C++ codes need to be compiled and linked to create an executable file.

UsesofOTcl

For configuration, setup, or one time simulation, or To run simulation with existing NS2 modules. This option is preferable for most beginners, since it does not involve complicated internal mechanism of NS2. Unfortunately, existing NS2 modules are fairly limited. This option is perhaps not sufficient for most researchers.

UsesofC++

When you are dealing with a packet, or - when you need to modify existing NS2 modules. ThisoptionperhapsdiscouragesmostofthebeginnersfromusingNS2.Thisbookparticularly aimsathelpingthereadersunderstandthestructureofNS2andfeelmorecomfortablein modifying NS2 modules.

InstallationofNS2onUbuntu

Thefollowingstepsaretheguidetoinstallns2inwindowsaftertheubuntu(linux)installation.

Step1:Install thefollwing softwarebeforeinstallingNS2

```
sudoapt-getinstalltcl8.5-devtk8.5-dev
sudoapt-getinstallbuild-essentialautoconf automake
sudoapt-getinstall perlxgraph libxt-devlibx11-dev libxmu-dev
```

Step2:Download ns2 fromthefollowing link

https://www.isi.edu/nsnam/ns/ns-build.html

Step3:Extractns-allinone-2.35.tar.gzintothehomedirctory(/home/adminadminis username given in system) using the follwing command.

```
tar-zxvfns-allinone-2.35.tar.gz-C/home/admin
```

Step4:InstallNS2usingthefollwing command

```
cd/home/anupamj/ns-allinone-2.35
sudo ./install>
```

Step5:SetPATHenvironment as follows

1. YouMUSTput/home/admin/ns-allinone-2.35/otcl-1.14,/home/admin/ns-allinone-2.35/lib, into your LD_LIBRARY_PATH environment variable.

IfitcomplainsaboutXlibraries,addpathtoyourXlibrariesintoLD_LIBRARY_PATH. Ifyouareusingcsh,youcansetitlike:setenvLD_LIBRARY_PATHIfyouareusingsh,you can set it like: export LD_LIBRARY_PATH

2. You MUST put /home/admin/ns-allinone-2.35/tcl8.5.10/library into your TCL_LIBRARY environmental variable. Otherwise ns/nam will complain during startup.

Step6:Modify .bahrc

```
vi /home/admin/.bashrc
```

Goto the last lineand add thescripts below:

```
exportPATH=$PATH:/home/stan/ns-allinone-2.35/bin:/home/admin/ns-allinone-
```

```
2.35/tcl8.5.10/unix:/home/admin/ns-allinone-2.35/tk8.5.10/unix
exportLD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/admin/ns-allinone-2.35/otcl-1.14:/home/admin/ns-allinone-2.35/lib
exportTCL_LIBRARY=$TCL_LIBRARY:/home/admin/ns-allinone-2.35/tcl8.5.10/library
```

Enablethepathsetting:

Step7:SuccessfulInstallationofns2can beverified using thefollowing command

```
cd ns-2.35; ./validate
```

Implementation:

SimulatetoFindtheNumberofPackets Dropped:

```
#Threenodesnetwork&measurepackets dropped

set ns [new Simulator]
set tf [open out.tr
w]setnf[openout.namw
]
$nstrace-all $tf
$nsnamtrace-all$nf

#Createnodes set
num 3
for{seti0}{$i<$num}{incri}{ set
      node($i) [$ns node]
}

#Createlinks
$nsduplex-link$node(0)$node(1)1Mb10msDropTail
$nsduplex-link$node(1)$node(2)800Kb10msDropTail;#800,600,400, 200

#Createqueues
$nsduplex-link-op$node(1)$node(2)queuePos0.5
$nsqueue-limit$node(1)$node(2)10

#Labelnodes
$node(0)label"UDP"
$node(2)label"Null"

#Labelflows
$nscolor0Red
```

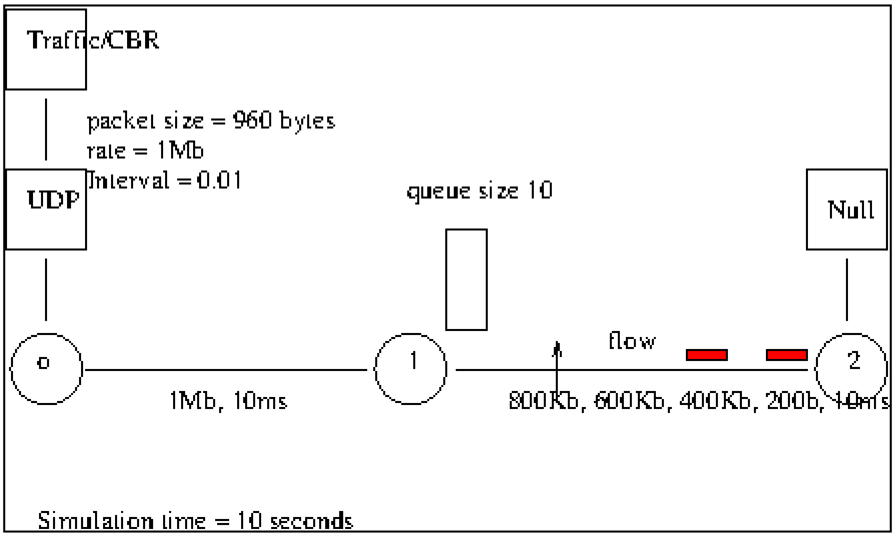
```
#Createconnections
setudp[$nscreate-connectionUDP$node(0)Null$node(2)0] set
cbr [$udp attach-app Traffic/CBR]

#Traffic
$cbrsetpacketSize_ 960
$cbrsetrate_1Mb
$cbrsetinterval_0.001;#choose0.01only;0.001,0.01,0.1

$nsat0.0"$cbr start"
$nsat 10 "finish"

procfinish{}{
        globalnstfnf
        $nsflush-trace
        close $tf
      close$nf
        exit 0
}

#Startsimulation
$ns run

#File 1.awk
#Countdroppedpackets


BEGIN{
      count=0;
}
{
      if($1=="d")count++;
}
```

```
END{
        printf("Numberofpacketsdroppedis%d\n",count);
}
```

RUN:

```
ns 1.tcl
namout.nam
awk -f 1.awk out.tr
BW(Kb/s)800600400200
Dropped0210470 730
```

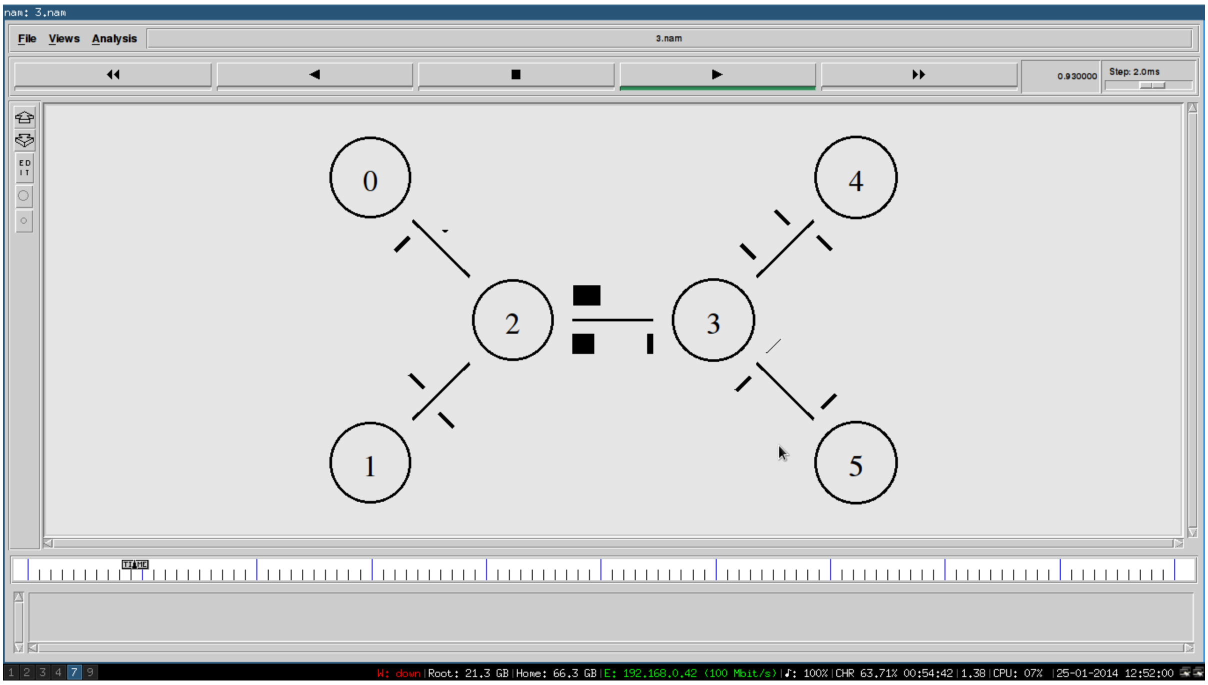
Result:



SimulatetoFindtheNumberofPacketsDroppedbyTCP/UDP

```
//creatinganagentobject
    setping0[newAgent/Ping]

 //agentobject node0
     $nsattach-agent$n0$ping0 set
    ping1 [new Agent/Ping]

 //agentobject node1
    $nsattach-agent$n1$ping1 set
    ping4 [new Agent/Ping]

    //agentobject node4
    $nsattach-agent$n4$ping4 set
    ping5 [new Agent/Ping]

    //agentobject node5
    $nsattach-agent$n5$ping5
    //node2andnode3actsasanintermediate nodes
    //$nsconnect$source$destination
    $nsconnect$ping0$ping4
    $nsconnect$ping1$ping5

    //functiontoconstantlypingthdestinationatanintervalof0.01s proc
    sendPingPacket {} {
        //global
        objectsglobalnsping0
        ping1
        //timeinterval
        set time 0.01
        //setsnowwiththecurrenttimeofsimulation set
        now [$ns now]
        //whenthecurrentsimulationtime($now)+time($time=0.01)occursapingissentto the
destination

        $nsat[expr$now + $time]"$ping0send"
        $nsat[expr$now + $time]"$ping1send"
        //pingPacketissent
        $nsat[expr$now+ $time]"sendPingPacket"
    }

    //IntheTclcode,aprocedure'Agent/Pingrecv{fromrtt}'hastobedefinedwhichallows the user to
react to the ping result.
 Agent/Pinginstprocrecv{fromrtt}{
        global seq
        $selfinstvarnode_
    }

    $nsat0.01"sendPingPacket"
    $nsat10.0"finish"
```

Result:


nam: 3.nam
File
Views
Analysis
3.nam
0.930000
Step: 2.0ms
EDIT
0
4
2
3
1
5
W: down|Root: 21.3 GB|Home: 66.3 GB|E: 192.168.0.42 (100 Mbit/s)|♪: 100%|CHR 63.71% 00:54:42|1.38|CPU: 07% |25-01-2014 12:52:00

SimulatetoFindtheNumberofPacketsDroppedduetoCongestion:

```
#File 2.tcl
#SimulatePing&countdroppedpacketsduetocongestion set ns
[new Simulator]
set tf [open out.tr
w]setnf[openout.namw
]

$nstrace-all $tf
$nsnamtrace-all$nf

# Create
nodesset
num 6
for{seti0}{$i<$num}{incri}{ set
      node($i) [$ns node]
}

#Createlinks
$nsduplex-link$node(0)$node(4)1Mb10msDropTail
$nsduplex-link$node(1)$node(4)1Mb10msDropTail
$nsduplex-link$node(2)$node(4)1Mb10msDropTail
$nsduplex-link$node(3)$node(4)1Mb10msDropTail
$nsduplex-link$node(4)$node(5)1Mb10msDropTail

#Createqueue
$nsduplex-link-op$node(4)$node(5)queuePos0.5
$nsqueue-limit$node(4)$node(5)2;#differentfromnormal3,2

#Labelflows
$nscolor1"red"
$nscolor2"blue"
$nscolor3"green"
```

```
$nscolor4"yellow"
$nscolor5"orange"

#Definea'recv'functionfortheclass'Agent/Ping'Agent/Ping
instproc recv {from rtt} {
        $selfinstvarnode_
        puts"node[$node_id]receivedpinganswerfrom$fromwithround-trip-time$rtt
ms."
}

#Createconnections
set p0 [$ns create-connection Ping $node(0) Ping $node(5) 1] set p1 [$ns
create-connection Ping $node(1) Ping $node(5) 2] set p2 [$ns create-
connectionPing$node(2)Ping$node(5)3]setp3[$nscreate-connectionPing
$node(3) Ping$node(5)4]set p5[$ns create-connectionPing$node(5) Ping
$node(4)5]

#Scheduleevents
for{seti 0}{$i< 10}{incri}{
        for{setj0}{$j<10}{incrj}{
                $nsat[expr$i+.1+$j/10]"$p0send"
                $nsat[expr$i+.1+$j/10]"$p5send"
                $nsat[expr$i+.2+$j/10]"$p1send"
                $nsat[expr$i+.3+$j/10]"$p2send"
                $nsat[expr$i+.4+$j/10]"$p3send"
                $nsat[expr$i+.5+$j/10]"$p5send"
        }
}
$nsat 10 "finish"

procfinish{}{
           globalnstfnf
           $nsflush-trace
```

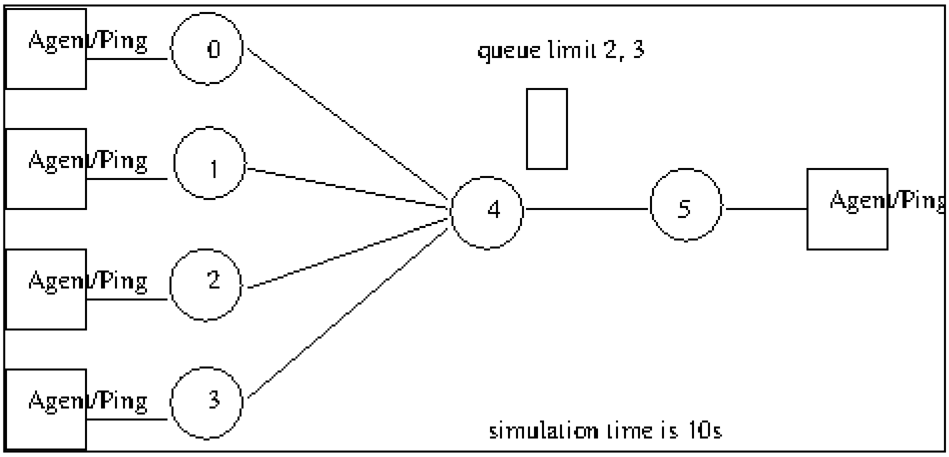
```
            close$tf
        close $nf
            exit0
}

#Startsimulation
$ns run

#File 2.awk
#Countdroppedpacketsduetocongestion

BEGIN{
        count=0;
}


{
        if($1=="d")count++;
}

END{
        printf("totalnoofpacketsdroppedduetocngestion:%d\n",count);
}
```

RUN:

```
ns2.tcl
namout.nam
awk-f2.awkout.tr
```

1. qsize(n4,n5)=2,30packetsdroppedduetocongestion
2. qsize(n4,n5)=3,20packetsdropped

Result:


Agent/Ping
0
Agent/Ping
1
Agent/Ping
2
Agent/Ping
3
queue limit 2, 3
4
5
Agent/Ping
simulation time is 10s

SimulatetoCompareDataRate& Throughput:

```
set val(chan) Channel/WirelessChannel
setval(prop)Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
setval(mac) Mac/802_11
setval(ifq)Queue/DropTail/PriQueue
set val(ll) LL
setval(ant)Antenna/OmniAntenna
set val(ifqlen) 50
setval(nn)3
setval(rp)DSDV
setns[new Simulator]

settf[openwireless.trw]
$nstrace-all$tf

settf1[openwireless1.nam w]
$nsnamtrace-all-wireless$tf1500500

settopo[newTopography]
$topoload_flatgrid500500

create-god $val(nn)

$nsnode-config-adhocRouting$val(rp)\
-llType$val(ll)\
-macType$val(mac) \
-ifqType$val(ifq) \
-ifqLen$val(ifqlen)\
-antType$val(ant)\
-propType$val(prop) \
-phyType$val(netif)\
-channelType$val(chan)\
-topoInstance$topo\
-agentTraceON\
-routerTraceOFF\
-macTraceOFF\
-movementTraceOFF

setnode0[$nsnode]
setnode1[$nsnode]
setnode2[$nsnode]

$nsinitial_node_pos$node010
$nsinitial_node_pos$node110
$nsinitial_node_pos$node210

$node0setX_ 5.0
$node0setY_ 5.0
$node0setZ_ 0.0

$node1setX_ 50.0
$node1setY_ 50.0
$node1setZ_ 0.0

$node2setX_ 100.0
$node2setY_ 100.0
$node2setZ_ 0.0

setudp1[newAgent/UDP]
$nsattach-agent$node0$udp1

setcbr1[newApplication/Traffic/CBR]
$cbr1 attach-agent $udp1
setnull1[newAgent/Null]
$nsattach-agent$node2$null1
$nsconnect$udp1$null1
```

```
$nsat0.0"$node0setdest5.010.00.0"
$nsat0.0"$node2setdest300.0300.00.0"
$nsat30.0"$node1 setdest30.0300.0 0.0"
$nsat50.0"$node1 setdest50.050.0 0.0"

$nsat0.5"$cbr1start"
$nsat159"$cbr1stop"

$nsat160"finish"

procfinish{}{
global ns tf tf1
$nsflush-trace
close $tf
close$tf1
execnamwireless1.nam& exit
0
}

$ns run
```

out.awk:

```
BEGIN{
Print“ThroughputCalculation”
}

{
if(( $1 ==“r”&& $7 ==“cbr”&& $3 ==“_2_“))
{
pkts=pkts+$8;
}
}

END{
Throughput=pkts*8/$2/1000000 print
“Throughput = “ Throughput print “
Datarate = “ Datarate
}
```

out1.awk:

```
{
if(( $1 ==“r”&& $7==“cbr”&& $3 ==“_2_“))
{
pkts=pkts+8;
print $2, pkts* 8/ $2 / 1000000
}
}
```

SimulatetoPlotCongestionforDifferent Source/Destination:

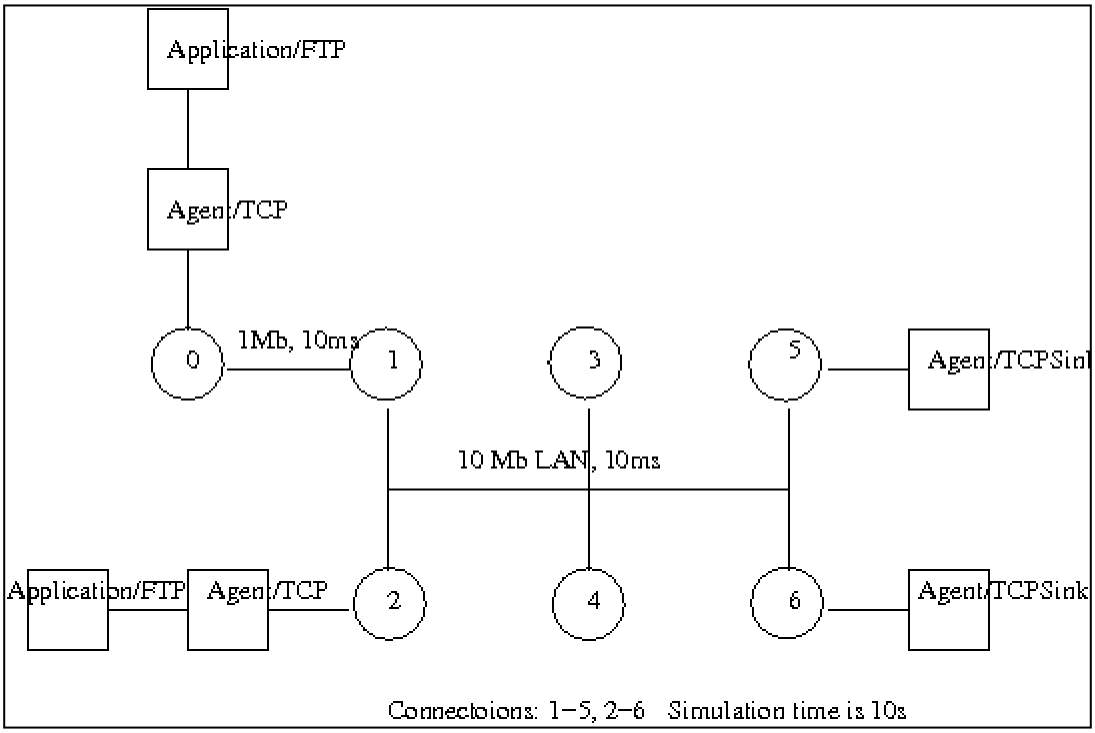
```
File3.tcl
#LANsimulation(congestionwindowsizewithtime) set
ns [new Simulator]
set tf [open out.tr
w]setnf[openout.namw
]

$nstrace-all $tf
$nsnamtrace-all$nf

#Createnodes
setnode(0)[$nsnode]

setnum6
for{seti1}{$i<=$num}{incri}{ set
        node($i) [$ns node]lappend
        nodelist $node($i)
}



#createLANand links
$nsmake-lan$nodelist10Mb10msLLQueue/DropTail

$nsduplex-link$node(0)$node(1)1Mb10msDropTail
$nsduplex-link-op$node(0)$node(1)queuePos0.5
$nsduplex-link-op$node(0)$node(1)orientright

#Createconnections
settcp0[$nscreate-connectionTCP$node(0)TCPSink$node(5)0]
settcp1[$nscreate-connectionTCP$node(2)TCPSink$node(6)0]   set
ftp0 [$tcp0 attach-app FTP]
setftp1[$tcp1attach-appFTP]
```

```
$tcp0attach$tf
$tcp0tracecwnd_

$tcp1attach$tf
$tcp1tracecwnd_

$nsat0.1"$ftp0 start"
$nsat0.2"$ftp1 start"
$nsat10"finish"
proc finish {} {
        globalnstfnf
        $nsflush-trace
        close $tf
      close$nf
        exit 0
}


#Startsimulator
$ns run

#File 3.awk
#PlotcongestionwindowX time

BEGIN{
}
{
if($6=="cwnd_")
{
if($2==0&&$4==5)printf("%4.2f\t%4.2f\t\n",$1,$7); #
$1=time, $7=cwnd size
#       if($2==2&& $4==6) printf("%4.2f\t%4.2f\t\n",$1,$7);
}
} END
{
   puts("DONE")
}
```

RUN:

```
ns3.tcl
namout.nam
awk-f3.awkout.tr>out.txtxgraph
out.txt
modifyawkscripttouseanothertcpconnection
```

Result:



SimulatetoDeterminethePerformancewithrespecttoTransmissionof Packets:

```
#File 4.tcl
#WirelessLANsimulation

set ns [new Simulator]
set tf [open out.tr
w]setnf[openout.namw
]

$nstrace-all$tf
$nsnamtrace-all-wireless
 $nf500500

settopo[newTopography]
$topoload_flatgrid500500



$nsnode-config\
        -adhocRoutingDSDV\
        -llType        LL\
        -macType       Mac/802_11\
        -ifqType       Queue/DropTail\
        -ifqLen        10\
        -phyType       Phy/WirelessPhy\
        -propType      Propagation/TwoRayGround\
        -antType       Antenna/OmniAntenna\
        -topoInstance$topo \
        -agentTrace    ON\
        -routerTraceON\
        -macTrace      ON\
        -channel       [new Channel/WirelessChannel]
create-god3;#GeneralOperationsDirectorsetnum3 for
{set i 0} {$i < $num} {incr i} {
```

```
         setnode($i)[$nsnode]
}

$node(0)label"TCP"
$node(1)label"TCPSink,TCP"
$node(2)label"TCPSink"

$node(0)setX_ 50
$node(0)setY_ 50
$node(0)setZ_0

$node(1)setX_ 100
$node(1)setY_ 100
$node(1)setZ_0

$node(2)setX_ 400
$node(2)setY_ 400
$node(2)setZ_0

#Createconnections
settcp0[$nscreate-connectionTCP$node(0)TCPSink$node(1)1]
settcp1[$nscreate-connectionTCP$node(1)TCPSink$node(2)2]

$nscolor1"red"
$nscolor2"blue"

setftp0[$tcp0attach-appFTP]
setftp1[$tcp1attach-appFTP]

$nsat0"$node(0)setdest5050100"
$nsat0"$node(1)setdest100100100"
$nsat0"$node(2)setdest400400100"
```

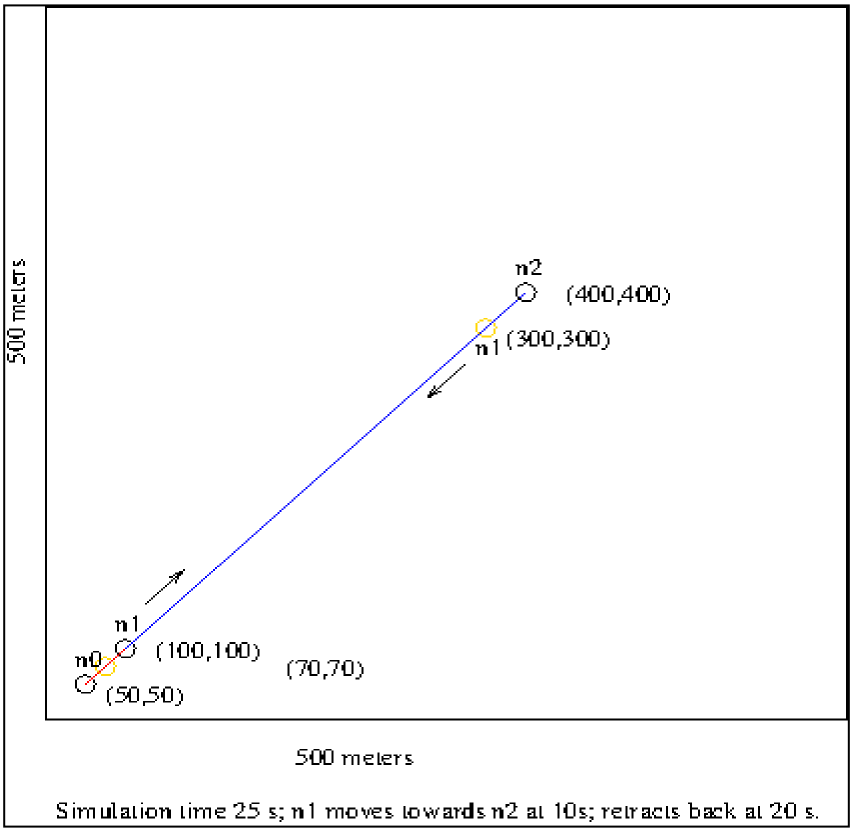
```
$nsat1"$ftp0start"
$nsat1"$ftp1start"

$nsat10"$node(1)setdest300300100"
$nsat15"$node(1)setdest100100100"
$nsat20"finish"
proc finish {} {
        globalnstfnf
        $nsflush-trace
        close $tf
      close$nf
        exit 0
}

#Startsimulation
$ns run

#File 4.awk
#WirelessLANlinkperformance

BEGIN{
        count1=0;
        count2=0;
        pack1=0;
        pack2=0;
        time1=0;
        time2=0;
}
{
        if($1=="r"&&$3=="_1_"&&$4=="AGT")
        {
                count1++;
                pack1=pack1+$8
```

```
                    time1=$2;
            }

printf("node(0)tonode(1)linkperformance:%6.2f
  Mbps\n",((count1*pack1*8)/(time1*1000000)));
            printf("node(0)tonode(1)linkperformance:%6.2f
  Mbps\n",((count2*pack2*8)/(time2*1000000)));
  }
```

RUN:

```
ns4.tcl
namout.nam
awk-f4.awkout.tr
Thethroughputfromnode(0)tonode(1):415.40Mb/s
Thethroughputfromnode(1)tonode(2):184.56 Mb/s
```

Result:



Simulation time 25 s; n1 moves towards n2 at 10s; retracts back at 20 s.