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: UnderstandingofstatisticalconceptsandtheirapplicationsinMachinelearning..

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
    1. PacketCaptureUsing Wireshark
    2. StartingWireshark
    3. ViewingCapturedTraffic
    4. AnalysisandStatistics&Filters.
11. HowtorunNmap scan
12. OperatingSystemDetectionusingNmap
13. Do thefollowing using NS2 Simulator
    1. NS2Simulator-Introduction
    2. SimulatetoFind theNumber ofPacketsDropped
    3. Simulateto FindtheNumberofPacketsDropped by TCP/UDP
    4. Simulateto Findthe Numberof PacketsDropped dueto Congestion
    5. SimulatetoCompareDataRate&Throughput
    6. SimulatetoPlotCongestionforDifferentSource/Destination
    7. 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:

case 2:

}

}

}

charc(); break;

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++)

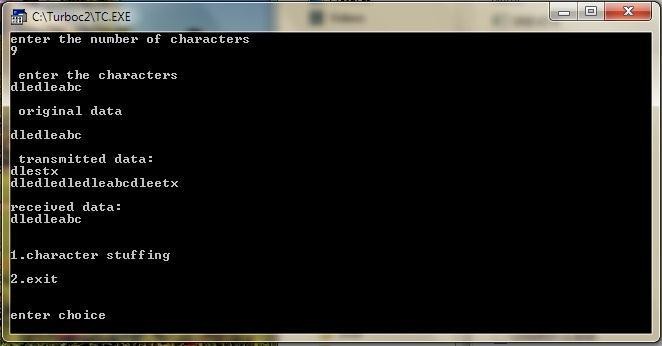
{

printf("%c",t[i]);

}

}

## Inputand Output:



**Program:Bitstuffing.**

int main()

{

int a[20],b[30],i,j,k,count,n;printf("Enterframesize(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>#defineNstrlen(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:11001100111000110000000000001100000001111

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>voidmain()

{

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)

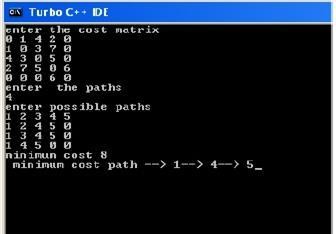
break;

}

getch();

}

## Inputand Output:



**Experiment-5**

**Aim:**Take anexamplesubnet ofhostsandobtainabroadcasttree forthesubnet.

## Program:

#include<stdio.h> #include<conio.h>int p,q,u,v,n;

int min=99,mincost=0; int t[50][2],i,j;

int parent[50],edge[50][50]; main()

{

clrscr();

printf("\n Enter the number of nodes"); scanf("%d",&n);

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

{

printf("%c\t", 65+i);

parent[i]=-1;

}

printf("\n"); for(i=0;i<n;i++)

{

printf("%c",65+i); for(j=0;j<n;j++) scanf("%d",&edge[ i][j]);

}

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

{for(j=0;j<n;j++) if(edge[i][j]!=99)if(min>edge[i][j])

{

min=edge[i][j]; u=i; **v** =j;

} p=find(u); q=find(v); if(p!=q)

{t[i][0]=u;

t[i][1]=v; mincost=mincost+edge[u][v]; sunion(p,q);

}

else

{

t[i][0]=-1;t[i][1]=-1;

}

min=99;

}

printf("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;

}

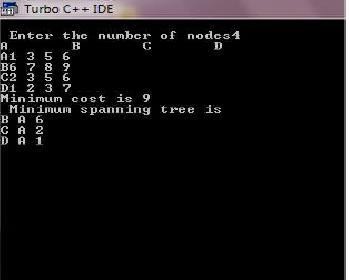
find(intl)

{

if(parent[l]>0) i=parent[i]; return i;

}

## Inputand Output:



**Experiment-6**

**Aim:**Implementdistancevector routing algorithmforobtainingroutingtablesateach node.

## Program:

#include<stdio.h> #include<conio.h>struct 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]);

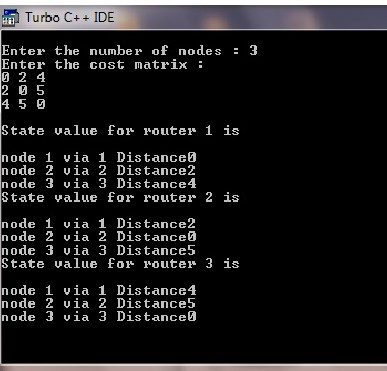
}

}

printf(“\n”);

}

## Inputand Output:



**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=Decryptthestring.\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>struct 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>struct 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

1. PacketCaptureUsing Wireshark
2. StartingWireshark
3. ViewingCapturedTraffic
4. 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, including[troubleshooting networks](https://www.comptia.org/content/guides/a-guide-to-network-troubleshooting)that 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.

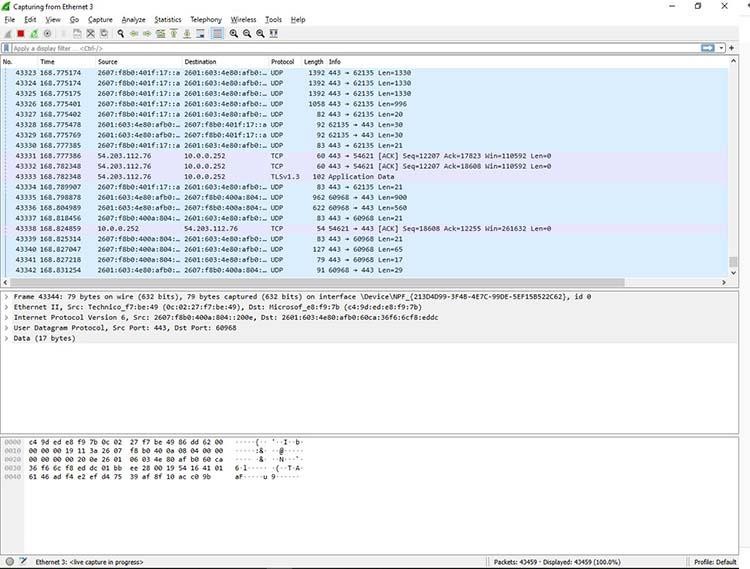
First of all, it can’t help a user who has little understanding of[network protocols.](https://www.comptia.org/content/guides/what-is-a-network-protocol)No tool, no matter how cool, replaces knowledge very well. In other words, to properly use Wireshark, you need to learn exactly how a network operates.

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



## Experiment-11

**Aim:**Howto runNmapscan

## Implementation:

[**NMAP**](https://www.geeksforgeeks.org/nmap-command-in-linux-with-examples/)standsforNetworkMapperwhichisanopen-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)**](https://www.geeksforgeeks.org/nmap-command-in-linux-with-examples/)NMAP has a database that helps

in**[Operatingsystems(OS)](https://www.geeksforgeeks.org/operating-systems/)**butitisnotautomaticallyupdated.ThedatabasetodetectanOS is located at ‘/usr/share/nmap/nmap-os-db’.

[**Operating System(OS)**](https://www.geeksforgeeks.org/operating-systems/)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)**](https://www.geeksforgeeks.org/operating-systems/)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**](https://www.geeksforgeeks.org/user-datagram-protocol-udp/)**:**UDPprobeconsistsofasinglepacketthatissenttoaclosedport. Iftheportusedon the target system is closed and an ICMP Port Unreachable message is returned it specifiesthat there is no Firewall.

**Result:**Theexperimentcompletedsuccessfully

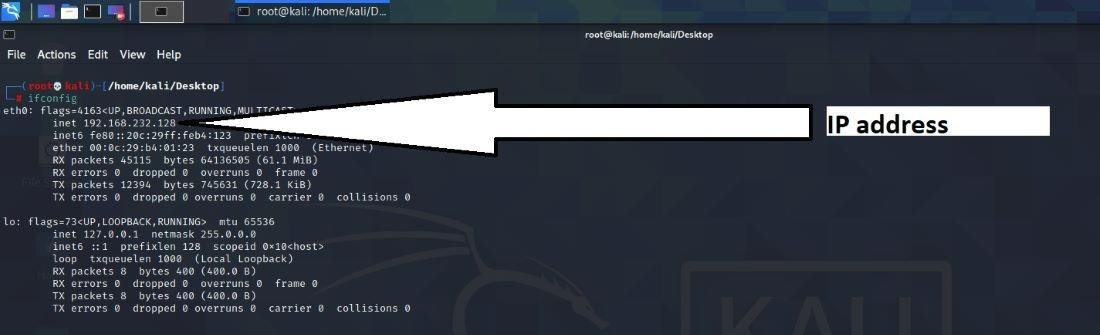
## Experiment-12

**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

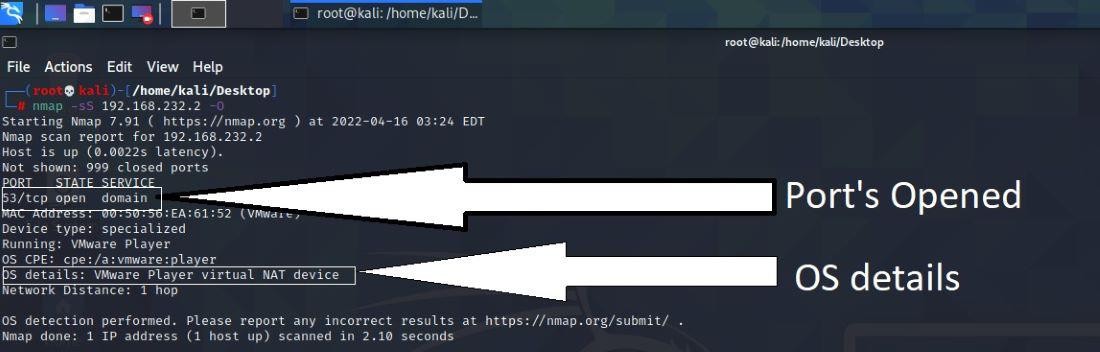


**Step2:**ListofactivedevicesintheNetwork nmap -sn 192.168.232.128/24



Let’sdoanSYNscanwith OS detectioninoneoftheactiveIPs

**Let’sselectIP**:192.168.232.2 nmap -sS 192.168.232.2 -O



**Running:**VMwarePlayer.

**OSdetails:**VMwarePlayervirtualNAT device.

Let’snowperformanAggressivescanToguesstheOS

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

ItwillonlydisplaythechanceofOperationSystem(OS)onthehostcomputerwiththehelp of Probability and Percentage.

nmap-sV192.168.232.2-A



**Result:**Theexperimentcompletedsuccessfully

## Experiment-13

**Aim:**Do thefollowing using NS2 Simulator

1. NS2Simulator-Introduction
2. SimulatetoFind theNumber ofPacketsDropped
3. Simulateto FindtheNumberofPacketsDropped by TCP/UDP
4. Simulateto Findthe Numberof PacketsDropped dueto Congestion
5. SimulatetoCompareDataRate&Throughput
6. SimulatetoPlotCongestionforDifferentSource/Destination
7. SimulatetoDeterminethePerformancewithrespecttoTransmissionof Packets

## NS2Simulator-Introduction

**WhatisNS2**

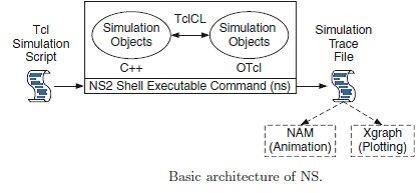
NS2standsforNetworkSimulatorVersion2.Itisanopen-sourceevent-drivensimulator designed specifically for research in computer communication networks.

## 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



## 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](http://www.isi.edu/nsnam/ns/ns-build.html)-[build.html](http://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

1. 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:

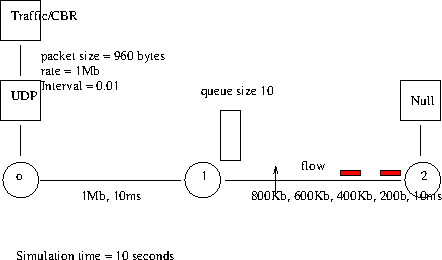
ns 1.tcl

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 objectsglobalnsping0ping1

//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

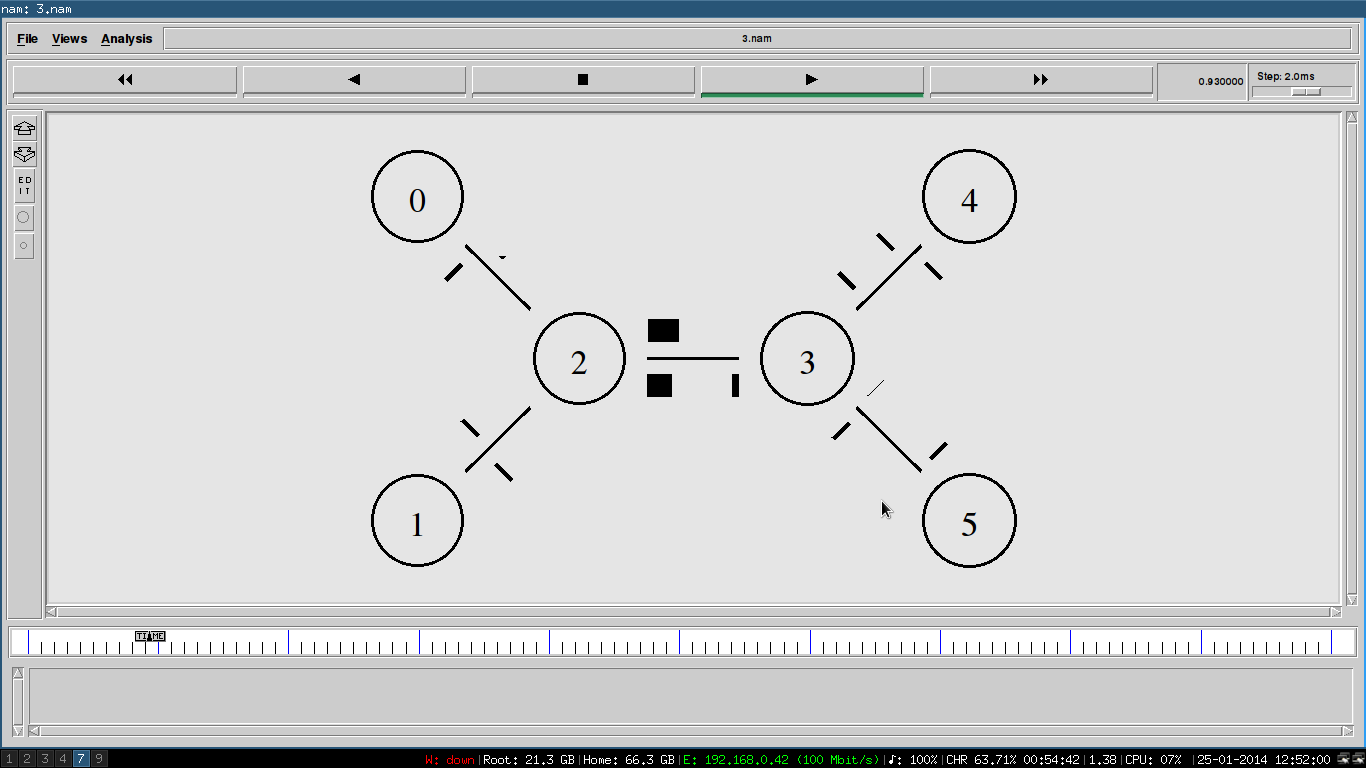
$selfinstvarnode\_

}

$nsat0.01"sendPingPacket"

$nsat10.0"finish"

## Result:



**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:

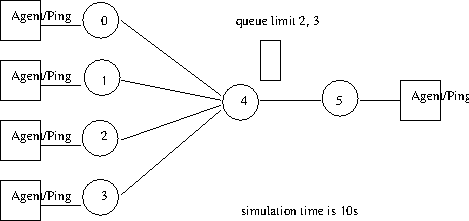
ns2.tcl

namout.nam

awk-f2.awkout.tr

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

## Result:



**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:

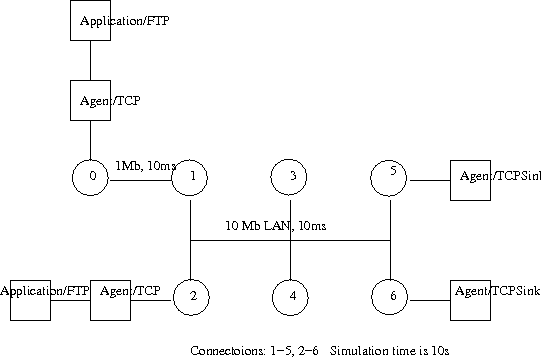
ns3.tcl

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:**

