Karnataka Law Society

Gogte Institute of Technology

Electronics & Communication Engineering Department

B.E

VII semester

LAB MANUAL

Sub: CCN Lab

Sub.Code: 16ECL77

2020-2021

List of Experiments

Expt. No.	Title of the Experiment
1	Implementation of bit stuffing.
2	Implementation of byte stuffing.
3	Implementation of Cyclic redundancy check algorithm.
4	Implementation of shortest path algorithm
5	Implementation of distance vector routing algorithm
6	Simulate a three-node point-to-point network with a duplex link between them. Set the queue size and vary the bandwidth and find the number of packets dropped
7	· Simulate a four node point to point network and connect the link as follows Apply a TCP agent between n0 to n3 and apply a UDP agent between n1 and n3.Apply relevant applications over TCP and UDP agents changing the parameters and determine the number of packets sent by two agents.
8	Simulate an ETHERNET LAN using n nodes (6-10), change error rate and data rate and compare throughput.
9	Simulate an Ethernet LAN using N nodes and set multiple traffic nodes and plot congestion window for different source/destination.
10	Simulate simple BSS and with transmitting nodes in wireless LAN by simulation and determine the performance with respect to transmission of packets.

Expt.No 1

Program Specifications you can write your program in any scripting or programming language you are comfortable with.

1. Create an array of the following characters ‘011111101111110’

2. Assume this is the data sent to the data link layer from the network layer. Notice that the combination of 0s and 1s conflicts with the synchronous flag.

3. Set up a loop to count the number of 1s and if there are 5 1s in sequence, insert a 0 into the data stream. Notice the data stream will require 2 ‘stuffed’ bits. 4. After bit stuffing append the flag to the front and end of the data stream. 5. Print you output (color is optional) but the your program output should be similar to the screen shot below.

a. Show the data stream before stuffing with the number of characters in the array
b. Show the data stream after stuffing with the number of characters in the array
c. Show the synchronous frame with the flag delimiters appended

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main() {
   int a[20],b[30],i,j,k,count,n;
printf("Enter frame size (Example: 8):");
scanf("%d",&n);
    printf("Enter the frame in the form of 0 and 1 :");
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 Bit Stuffing :");
for(i=0; i<j; i++)
printf("%d",b[i]);     getch();
}
```

Output:

```
Enter frame size (Example: 8):12
Enter the frame in the form of 0 and 1 :0 1 0 1 1 1 1 1 1 0 0 1
After Bit Stuffing :0101111101001
```

Expt.No 2

```
#include<stdio.h>
#include<string.h> main(){
        char a[20],b[20];
int i,n,j;         char f,s;
        printf("Enter the size of the frame : ");
scanf("%d",&n);         n=n*2;
        printf("\nEnter the characters in frame : \n");
for(i=0;i<n;i++)                 scanf("%c",&a[i]);
printf("\n FRAME \n ");         for(i=0;i<n;i++)
printf("%c",a[i]);         j=0;         for(i=0;i<n;i++)
        {
                if(a[i]=='f')
{
b[j]='s';
j++;
b[j]=a[i];

                }
                else if(a[i]=='s')
                {
b[j]='s';
j++;
b[j]=a[i];
                }
else
b[j]=a[i];

                j++;
        }
        printf("\n RESULT \n");
// printf("f");
        for(i=0;i<j;i++)
        {
printf("\n");
                printf("%c",b[i]);
        }
        //printf("\nf");
}
```

OR

```
#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')
{ 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("\nframe after stuffing:\n");
printf("%s",b); getch();}
```

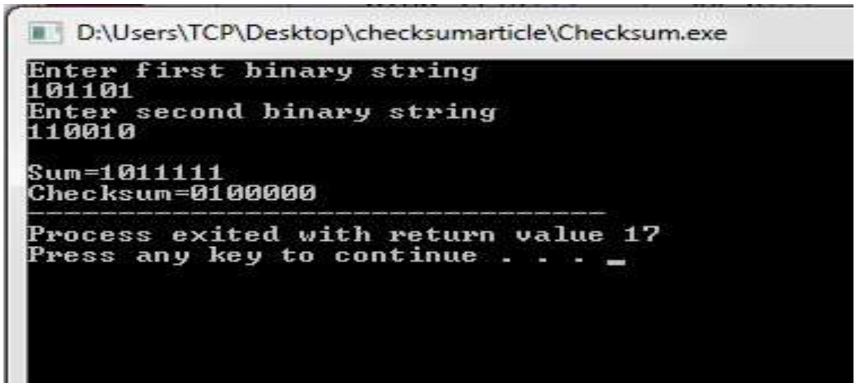
Expt.No 3

Checksum Algorithm

1. Take 2 binary input strings.
2. Do their binary sum to find out the checksum which will be sent to the destination or to the receiver.
3. In binary sum there are 6 cases:-
   a. If both bits are 0 and carry is 0, sum=0 and carry=0
   b. If both bits are 0 and carry is 1,sum=1 and carry=0
   c. If both bits are 1 and carry is 0,sum=0 and carry=1
   d. If both bits are 1 and carry is 1,sum=1 and carry=1
   e. If either bit is 1 and carry is 0,sum=1 and carry=0
   f. If either bit is 1 and carry is 1,sum=0 and carry=1
4. While doing the addition we have to add the binary strings from rightmost end i.e LSB to MSB.
5. When binary sum is done 1’s complement of it is taken by reversing 1’s to 0’s and vice versa.
6. The resulting 1’s complement is the Checksum.
7. Stop.

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

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

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

if(strlen(a)==strlen(b)){
length = strlen(a);
char carry='0';

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

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

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

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

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

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

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

            }
else
break;
        }

        printf("\nSum=%c%s",carry,sum);

        for(i=0;i<length;i++)
        {
            if(sum[i]=='0')
complement[i]='1';              else
                complement[i]='0';
        }
        if(carry=='1')
carry='0';
else
carry='1';

        printf("\nChecksum=%c%s",carry,complement);
    }
else {
        printf("\nWrong input strings");
    }
}
```

OUTPUT



D:\Users\TCP\Desktop\checksumarticle\Checksum.exe

```
Enter first binary string
101101
Enter second binary string
110010

Sum=1011111
Checksum=0100000
--------------------------------
Process exited with return value 17
Press any key to continue . . . _
```

Expt.No 4

Dijkstra's Algorithm

1. Create cost matrix C[ ][ ] from adjacency matrix adj[ ][ ]. C[i][j] is the cost of going from vertex i to vertex j. If there is no edge between vertices i and j then C[i][j] is infinity.

2. Array visited[ ] is initialized to zero.

```
            for(i=0;i<n;i++)
visited[i]=0;
```

3. If the vertex 0 is the source vertex then visited[0] is marked as 1.

4. Create the distance matrix, by storing the cost of vertices from vertex no. 0 to n-1 from the source vertex 0.

```
for(i=1;i<n;i++)
    distance[i]=cost[0][i];
```

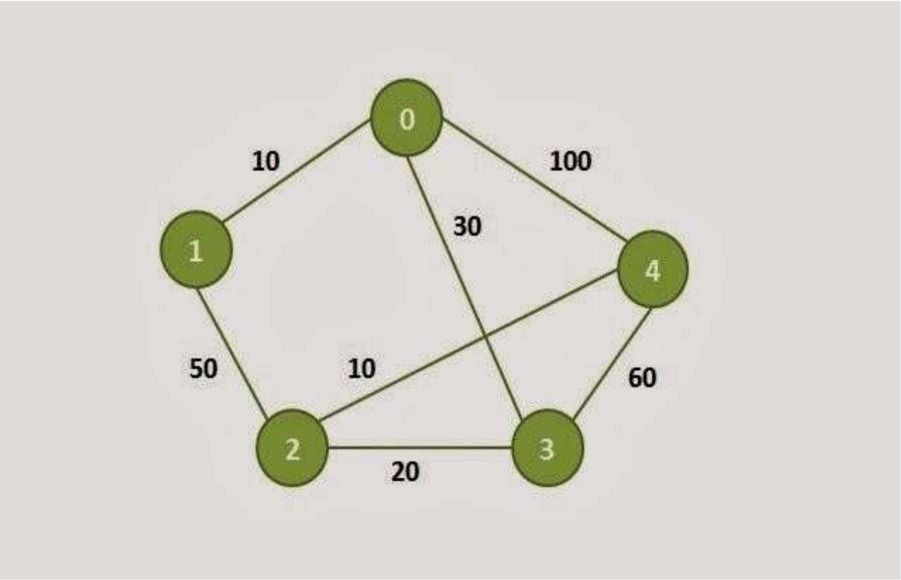
Initially, distance of source vertex is taken as 0. i.e. distance[0]=0;

5. for(i=1;i<n;i++)

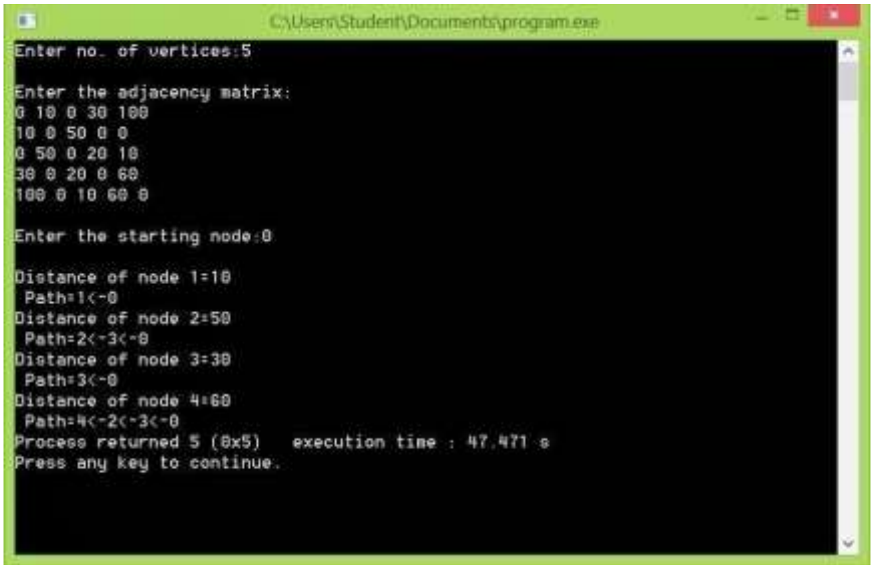
– Choose a vertex w, such that distance[w] is minimum and visited[w] is 0. Mark visited[w] as 1.

– Recalculate the shortest distance of remaining vertices from the source. – Only, the vertices not marked as 1 in array visited[ ] should be considered for recalculation of distance. i.e. for each vertex v

```
if(visited[v]==0)
distance[v]=min(distance[v],                    distance[w]+cost[w][v])
```



```
#include<stdio.h>
#include<conio.h>
#define INFINITY 9999
#define MAX 10

void dijkstra(int G[MAX][MAX],int n,int startnode);

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,int startnode)
{
   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;

       //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);
   }
}
```



```
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:0

Distance of node 1=10
 Path=1<-0
Distance of node 2=50
 Path=2<-3<-0
Distance of node 3=30
 Path=3<-0
Distance of node 4=60
 Path=4<-2<-3<-0
Process returned 5 (0x5)   execution time : 47.471 s
Press any key to continue.
```

Expt.No 5

In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in subnet. This entry contains two parts: the preferred outgoing line to use for that destination, and an estimate of the time or distance to that destination. The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar. The router is assumed to know the "distance" to each of its neighbour. If the metric is hops, the distance is just one hop. If the metric is queue length, the router simply examines each queue. If the metric is delay, the router can measure it directly with special ECHO packets that the receiver just time stamps and sends back as fast as possible.

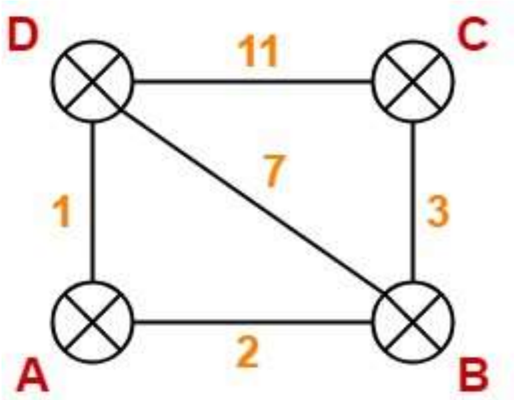
Algorithm:

1. Start
2. By convention, the distance of the node to itself is assigned to zero and when a node is

   unreachable the distance is accepted as 999.

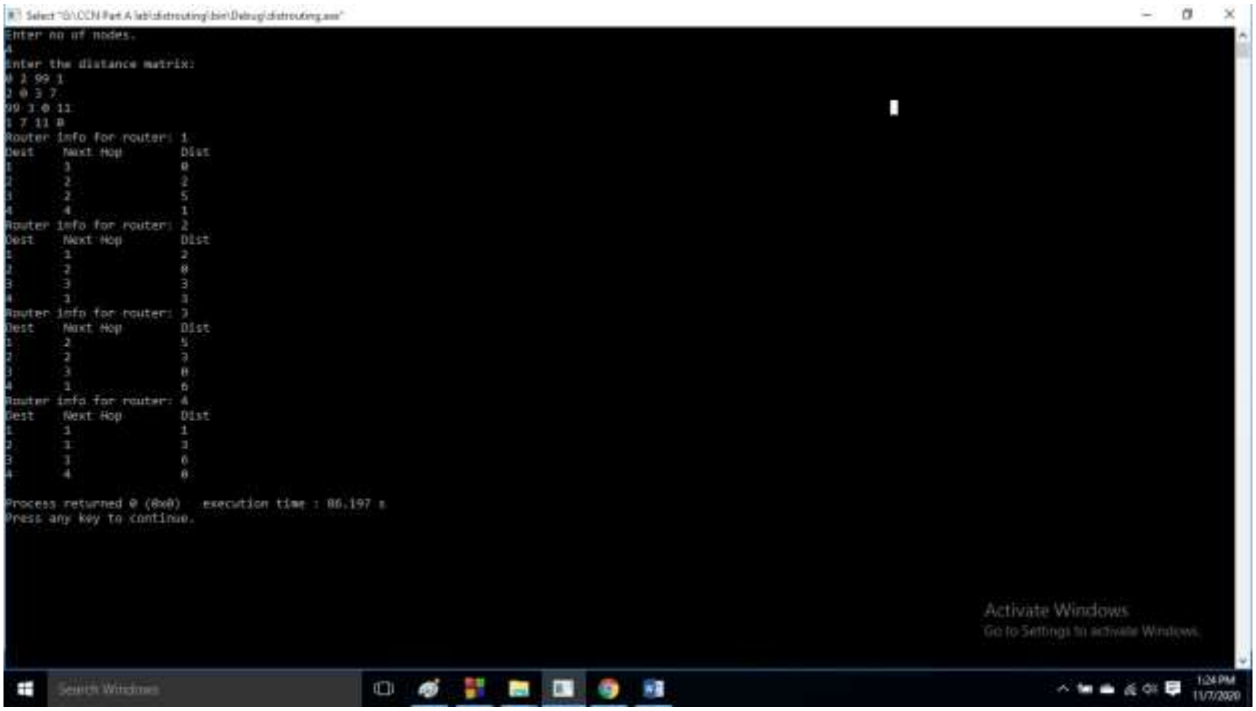
3. Accept the input distance matrix from the user (*dm[][]*) that represents the distance between each node in the network.

4. Store the distance between nodes in a suitable varible.
5. Calculate the minimum distance between two nodes by iterating.
   If the distance between two nodes is larger than the calculated alternate available path, replace the existing distance with the calculated distaance.
6. Print the shortest path calculated.
7. Stop.



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

using namespace std;
```

```
struct node {
int dist[20];
int from[20];
} route[10];

int main() {
   int dm[20][20], no;

   cout << "Enter no of nodes." << endl;
   cin >> no;
   cout << "Enter the distance matrix:" << endl;
   for (int i = 0; i < no; i++) {
for (int j = 0; j < no; j++) {
cin >> dm[i][j];
         /*  Set distance from i to i as 0 */
dm[i][i] = 0;             route[i].dist[j] =
dm[i][j];             route[i].from[j] = j;
      }
   }
    int flag;     do {          flag = 0;          for (int i = 0; i < no; i++) {
for (int j = 0; j < no; j++) {                  for (int k = 0; k < no; k++) {
if ((route[i].dist[j]) > (route[i].dist[k] + route[k].dist[j])) {
route[i].dist[j] = route[i].dist[k] + route[k].dist[j];
route[i].from[j] = k;
                  flag = 1;
               }
            }
         }
      }
   } while (flag);

   for (int i = 0; i < no; i++) {
      cout << "Router info for router: " << i + 1 << endl;
cout << "Dest\tNext Hop\tDist" << endl;
      for (int j = 0; j < no; j++)
         printf("%d\t%d\t\t%d\n", j+1, route[i].from[j]+1, route[i].dist[j]);
   }
return 0;}
```


Select "G:\CCN Part A lab\distrouting\bin\Debug\distrouting.exe"
Activate Windows
Go to Settings to activate Windows.
Search Windows
1:24 PM
11/7/2020

```
Enter no of nodes.
4
Enter the distance matrix:
0 2 99 1
2 0 3 7
99 3 0 11
1 7 11 0
Router info for router: 1
Dest    Next Hop        Dist
1       1               0
2       2               2
3       2               5
4       4               1
Router info for router: 2
Dest    Next Hop        Dist
1       1               2
2       2               0
3       3               3
4       1               3
Router info for router: 3
Dest    Next Hop        Dist
1       2               5
2       2               3
3       3               0
4       1               6
Router info for router: 4
Dest    Next Hop        Dist
1       1               1
2       1               3
3       1               6
4       4               0

Process returned 0 (0x0)   execution time : 86.197 s
Press any key to continue.
```

PART – B (SIMULATION)

INTRODUCTION

Network simulation is an important tool in developing, testing and evaluating network protocols. Simulation can be used without the target physical hardware, making it economical and practical for almost any scale of network topology and setup. It is possible to simulate a link of any bandwidth and delay, even if such a link is currently impossible in the real world. With simulation, it is possible to set each simulated node to use any desired software. This means that meaning deploying software is not an issue. Results are also easier to obtain and analyze, because extracting information from important points in the simulated network is as done by simply parsing the generated trace files.

Simulation is only of use if the results are accurate, an inaccurate simulator is not useful at all. Most network simulators use abstractions of network protocols, rather than the real thing, making their results less convincing. S.Y. Wang reports that the simulator OPNET uses a simplified finite state machine to model complex TCP protocol processing. [19] NS-2 uses a model based on BSD TCP, it is implemented as a set of classes using inheritance. Neither uses protocol code that is used in real world networking.

GETTING STARTED Setting up the environment

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

1. Set up environment variables:

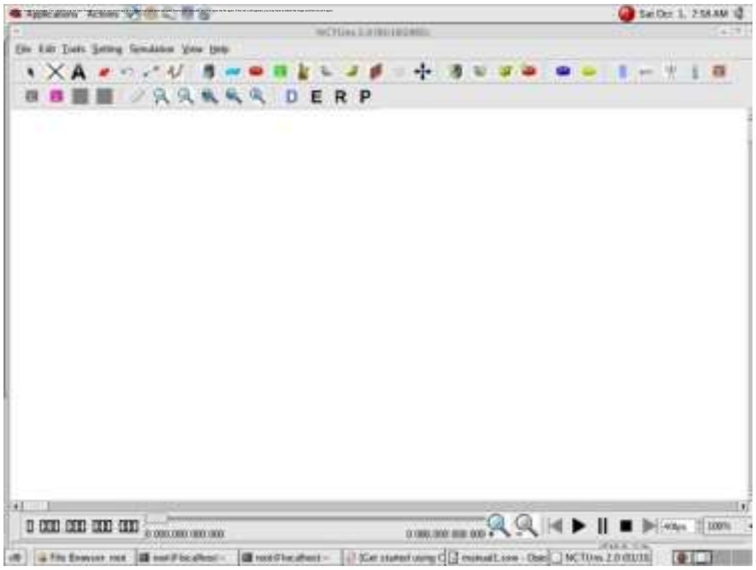
   Before the user can run up the dispatcher, coordinator, or NCTUns GUI program he/she must set up the NCTUNSHOME environment variable.

2. Start up the dispatcher on terminal 1.

3. Start up the coordinator on terminal 2.

4. Start up the nctunsclient on terminal 3.

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



Drawing A Network Topology

To draw a new network topology, a user can perform the following steps:

Choose Menu->File->Operating Mode-> and make sure that the “Draw Topology” mode is checked. This is the default mode of NCTUns when it is launched. It is only in this mode that a user can draw a new network topology or change an existing simulation topology. When a user switches the mode to the next mode “Edit Property”, the simulation network topology can no longer be changed.

1. Move the cursor to the toolbar.

2. Left-Click the router icon on the toolbar.

3. Left-Click anywhere in the blank working area to add a router to the current network topology. In the same way we can add switch, hub, WLAN access point, WLAN mobile node, wall (wireless signal obstacle) etc.

4. Left-Click the host icon on the toolbar. Like in step 4, add the required number of hosts to the current topology.

5. To add links between the hosts and the router, left-click the link icon on the toolbar to select it.

6. Left-Click a host and hold the mouse button. Drag this link to the router and then release the mouse left button on top of the router. Now a link between the selected host and the router has been created.

7. Add the other, required number of links in the same way. This completes the creation of a simple network topology.

8. Save this network topology by choosing Menu->File->Save. It is saved with a .tpl extension.

9. Take the snapshot of the above topology.

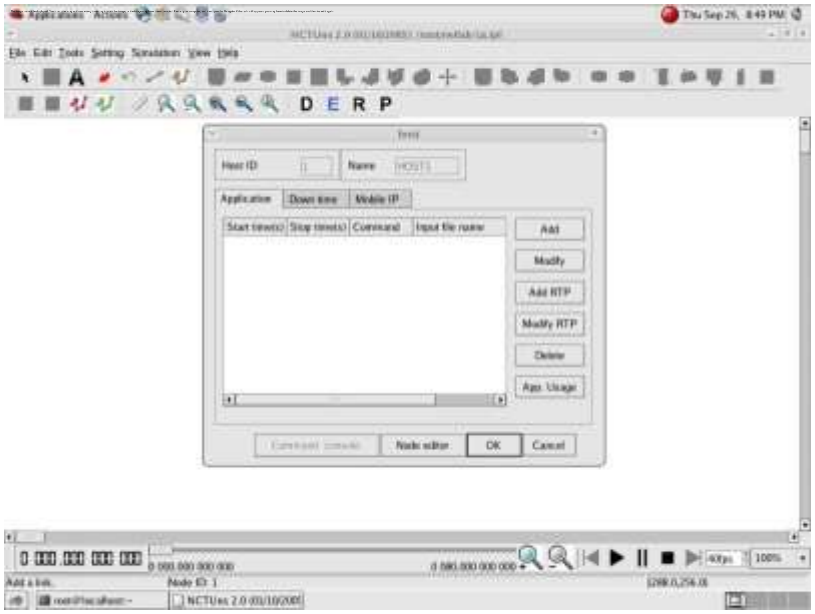
Editing Node's Properties

1. A network node (device) may have many parameters to set. For example, we may have to set the maximum bandwidth, maximum queue size etc to be used in a network interface. For another example, we may want to specify that some application programs (traffic generators) should be run on some hosts or routers to generate network traffic.

2. Before a user can start editing the properties of a node, he/she should switch the mode from the “Draw Topology” to “Edit Property” mode. In this mode, topology changes can no longer be made. That is, a user cannot add or delete nodes or links at this time.

3. The GUI automatically finds subnets in a network and generates and assigns IP and MAC addresses to layer 3 network interfaces.

4. A user should be aware that if he/she switches the mode back to the “Draw Topology” mode when he/she again switches the mode back to the “Edit Topology” mode, node's IP and MAC addresses will be regenerated and assigned to layer 3 interfaces.



Host ID: 1

Name: HOST1

Therefore the application programs now may use wrong IP addresses to communicate with their partners.

Running the Simulation

When a user finishes editing the properties of network nodes and specifying application programs to be executed during a simulation, he/she can start running the simulation.

2. In order to do so, the user must switch mode explicitly from “Edit Property” to “Run Simulation”. Entering this mode indicates that no more changes can (should) be made

to the simulation case, which is reasonable. This simulation is about to be started at this moment; of course, any of its settings should be fixed.

3. Whenever the mode is switched to the “ Run Simulation” mode, the many simulation files that collectively describe the simulation case will be exported. These simulation files will be transferred to the (either remote or local) simulation server for it to execute the simulation. These files are stored in the “ main File Name.sim” directory, where main Filename is the name of the simulation case chosen in the “Draw Topology” mode.

Playing Back the Packet Animation Trace

After the simulation is finished, the simulation server will send back the simulation result files to the GUI program after receiving these files, the GUI program will store these files in the “results directory” .It will then automatically switch to “play back mode”.

1. These files include a packet animation trace file and all performance log files that the user specifies to generate. Outputting these performance log files can be specified by checking some output options in some protocol modules in the node editor. In addition to this, application programs can generate their own data files.

3. The packet animation trace file can be replayed later by the packet animation player. The performance curve of these log files can be plotted by the performance monitor.

Post Analysis

1. When the user wants to review the simulation results of a simulation case that has been finished before, he /she can run up the GUI program again and then open the case's topology file

2. The user can switch the mode directly to the “Play Back” mode. The GUI program will then automatically reload the results (including the packet animation trace file and performance log file.

3. After the loading process is finished, the user can use the control buttons located at the bottom of the screen to view the animation.

Simulation Commands

The following explains the meaning of each job control command:

Run: Start to run the simulation.

Pause: Pause the currently -running simulation.

Continue: Continue the simulation that was just paused.

Stop: Stop the currently -running simulation

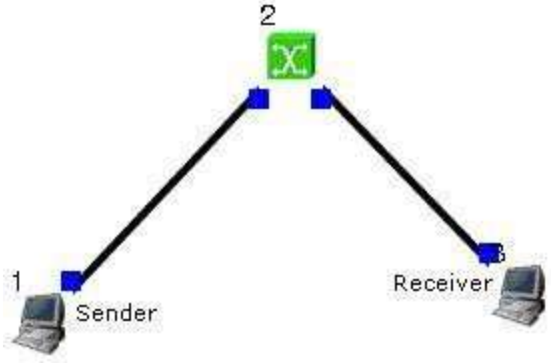
Abort: Abort the currently running simulation. The difference between “stop” and “abort” is that a stopped simulation job's partial results will be transferred back to GUI files.

Reconnect: The Reconnect command can be executed to reconnect to a simulation job that was previously disconnected. All disconnected jobs that have not finished their simulations or have finished their simulations but the results have not been retrieved back to be a GUI program by the user will appear in a session table next to the “Reconnect” command. When executing the reconnect command, a user can choose a disconnected job to reconnect from this session table.

Disconnect: Disconnect the GUI from the currently running simulation job. The GUI now can be used to service another simulation job. A disconnected simulation will be given a session name and stored in a session table.

Expt.No 6

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



Step1: Drawing topology

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

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

2. Select/click the HUB icon on the toolbar and click the left mouse button on the editor, to place HUB1 on the editor.

3. Click on the LINK icon on the toolbar and connect HOST1 to HUB1 and HUB1 to HOST2

4. Click on the "E" icon on the toolbar to save the current topology
   e.g: file1.tpl
   (Look for the ******.tpl extension.)

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

Step2: Configuration

1. Double click the left mouse button while cursor is on HOST1 to open the HOST window.

2. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

```
stg –u 1024 100 1.0.1.2
```

3. Click OK button on the command window to exit and once again click on the OK button on the HOST window to exit.

4. Double click the left mouse button while cursor is on HOST2 to open the HOST window.

5. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

```
rtg –u –w log1
```

6. Click OK button on the command window to exit.

7. Click NODE EDITOR Button on the HOST window and select the MAC tab from the modal window that pops up.

8. Select LOG STATISTICS and select checkboxes for Number of Drop Packet and Number of Collisions in the MAC window

9. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

Note: To set QUEUE size

1. Double click the left mouse button while cursor is on HOST2 to open the HOST window.

2. Click NODE EDITOR Button on the HOST window and select the FIFO tab from the modal window that pops up.

3. Change Queue size (Default 50).

4. Click OK button on the FIFO window to exit and once again click on the OK button on the HOST window to exit.

Step3: Simulate

i. Click “R” icon on the tool bar

ii. Select Simulation in the menu bar and click/ select RUN in the dropdown list to execute the simulation.

iii. venTo start playback select “►” icon located at the bottom right corner of the editor.

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

Caution: file1 is the hypothetical name given to this simulation. **(Refer Step 1.4)**

Changing configurations

Change 1

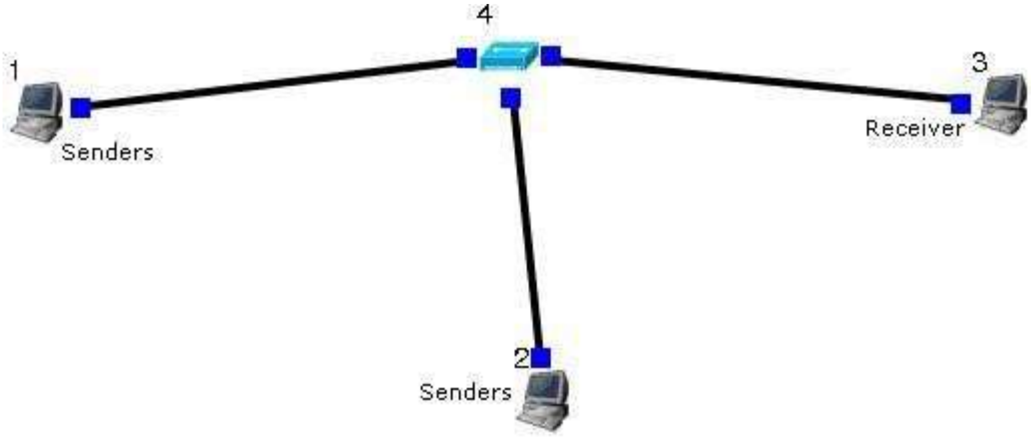
1. Open the above file,
2. Do not change the topology or any other configuration,
3. Select E icon on the toolbar
4. Reduce the bandwidth at link2 by double clicking the left mouse button while cursor is on link2 .(Change bandwidth on both tabs Uplink/Downlink)
5. Repeat Step3 (Simulate)

Change 2

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

Expt.No 7

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



Step1: Drawing topology

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

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

2. Select/click the HUB (or SWITCH) icon on the toolbar and click the left mouse button on the editor, to place a HUB (or SWITCH) on the editor.

3. Click on the LINK icon on the toolbar and connect HOST1 to HUB, HOST2 to HUB and HUB to HOST3

4. Click on the “E” icon on the toolbar to save the current topology
   e.g: file2.tpl
   (Look for the ******.tpl extension.)

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

Step2: Configuration

1. Double click the left mouse button while cursor is on HOST1 to open the HOST window.

2. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

```
stcp –p 21 –l 1024  1.0.1.3
```

3. Click OK button on the command window to exit

4. Click NODE EDITOR Button on the HOST window and select the MAC tab from the modal window that pops up.

5. Select LOG STATISTICS and select checkbox for output throughput in the MAC window

6. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

7. Double click the left mouse button while cursor is on HOST2 to open the HOST window.

8. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

```
stg –u 1024 100 1.0.1.3
```

9. Click OK button on the command window to exit

10. Click NODE EDITOR Button on the HOST window and select the MAC tab from the modal window that pops up.

11. Select LOG STATISTICS and select checkbox for output throughput in the MAC window

12. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

13. Double click the left mouse button while cursor is on HOST3 to open the HOST window.

14. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

    ```
    rtcp –p 21 –l 1024
    ```

15. Click OK button on the command window to exit.

16. Also add the following command on

    ```
    HOST3 rtg –u –w log1
    ```

17. Click NODE EDITOR Button on the HOST window and select the MAC tab from the modal window that pops up.

18. Select LOG STATISTICS and select checkbox for input and output throughput in the MAC window

19. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

Step3: Simulate

i. Click “R” icon on the tool bar

ii. Select Simulation in the menu bar and click/ select RUN in the dropdown list to execute the simulation.

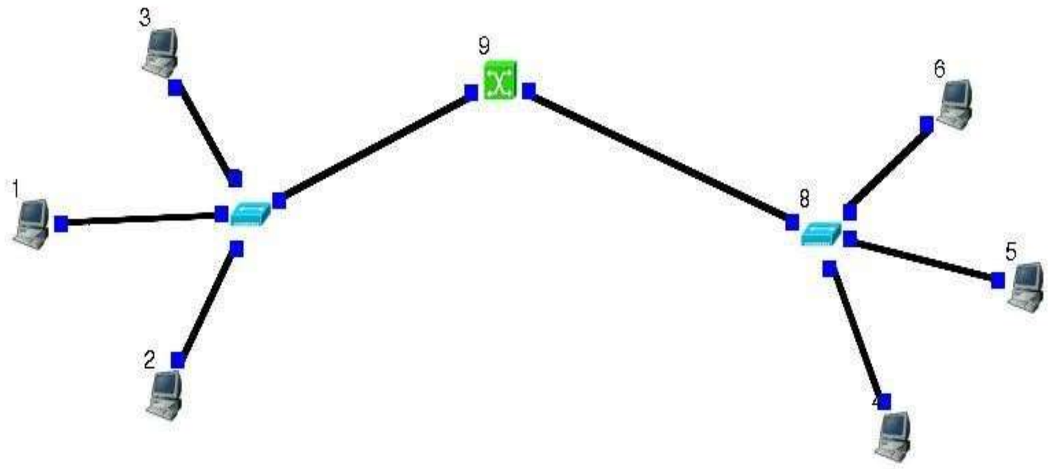
iii. To start playback select “►” icon located at the bottom right corner of the editor.

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

Caution: file2 is the hypothetical name given to this simulation. **(Refer Step 1.4)**

Expt. No 8

Simulate an ETHERNET LAN using n nodes (6-10), change error rate and data rate and compare throughput.



Step1: Drawing topology

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

   i. Repeat the above procedure and place 5 other hosts “HOST2”, “HOST3”, “HOST4”, “HOST5”, and “HOST6”on the editor.

2. Select/click the HUB icon on the toolbar and click the left mouse button on the editor, to place HUB1 on the editor.

Repeat the above procedure and place another host “HUB2” on the editor

3. Click on the LINK icon on the toolbar and connect HOST1, HOST2 and HOST3 to HUB1, HOST4, HOST5 and HOST6 to HUB2.

4. Select/click the SWITCH icon on the toolbar and click the left mouse button on the editor, to place SWITCH1 on the editor.

5. Click on the LINK icon on the toolbar and connect HUB1 to SWITCH1 and HUB2 to SWITCH1.

6. Click on the “E” icon on the toolbar to save the current topology
   e.g: file5.tpl
   (Look for the ******.tpl extension.)

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

Step2: Configuration

1. Double click the left mouse button while cursor is on HOST1 to open the HOST window.

2. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

```
stcp –p 21 –l 1024  1.0.1.4
```

3. Click OK button on the command window to exit and once again click on the OK button on the HOST window to exit.

4. Repeat this step at HOST 2 and HOST3, but use different commands

```
stcp –p 21 –l 1024  1.0.1.5 at HOST2
stcp –p 21 –l 1024  1.0.1.6 at  HOST3
```

5. Double click the left mouse button while cursor is on HOST4 to open the HOST window.

6. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

```
rtcp –p 21 –l 1024
```

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

9. Select LOG STATISTICS and select checkbox for output throughput in the MAC window

10. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

11. Repeat this step at HOST 5 and HOST6, but use different commands

```
rtcp –p 21 –l 1024  at HOST5 rtcp
–p 21 –l 1024  at  HOST6
```

12. Double click the left mouse button while cursor is on HOST5 to open the HOST window.

13. Click NODE EDITOR Button on the HOST5 window and select the PHYSICAL tab from the modal window that pops up.

14. Change Bit Error Rate

15. Click OK button on the PHYSICAL window to exit and once again click on the OK button to return to the HOST window

16. Click NODE EDITOR Button on the HOST window and select the MAC tab from the modal window that pops up.

17. Select LOG STATISTICS and select checkbox for output throughput in the MAC window

18. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

19. Repeat this step HOST6, Change Bandwidth this time while undoing the change in Bit Error Rate, also select the output throughput at HOST6.

Step3: Simulate

i. Click “R” icon on the tool bar

ii. Select Simulation in the menu bar and click/ select RUN in the dropdown list to execute the simulation.

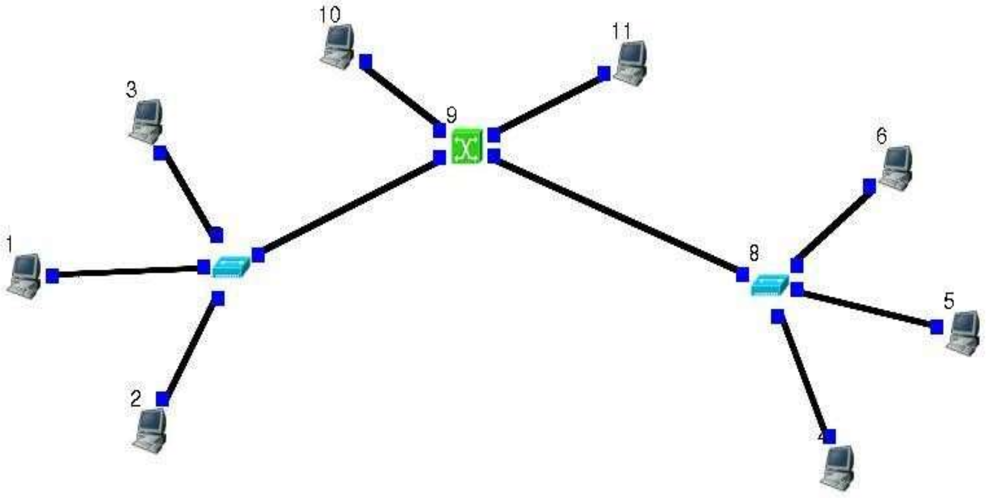
iii. To start playback select “►” icon located at the bottom right corner of the editor.

iv. To view results, Open up new TERMINAL window, move to file5.results folder and open output throughput log files in separate TERMINAL window.

Caution: file5 is the hypothetical name we gave to this simulation **(Refer Step 1.7)**

Expt. No 9

Simulate an ETHERNET LAN using n nodes and set multiple traffic nodes and plot congestion window for different source/destination.



Step1: Drawing topology

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

   i. Repeat the above procedure and place 3 other hosts “HOST2”, “HOST3”, “HOST4”, “HOST5”, and “HOST6”on the editor.

2. Select/click the HUB icon on the toolbar and click the left mouse button on the editor, to place HUB1 on the editor.

   Repeat the above procedure and place another host “HUB2” on the editor

3. Click on the LINK icon on the toolbar and connect HOST1, HOST2 and HOST3 to HUB1, HOST4, HOST5 and HOST6 to HUB2.

4. Select/click the SWITCH icon on the toolbar and click the left mouse button on the editor, to place SWITCH1 the editor.

5. Click on the LINK icon on the toolbar and connect HUB1 to SWITCH1 and HUB2 to SWITCH1.

6. Click on the “E” icon on the toolbar to save the current topology **e.g:** file7.tpl
   (Look for the ******.tpl extension.)

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

Step2: Configuration

1. Double click the left mouse button while cursor is on HOST1 to open the HOST window.

2. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

```
stcp –p 21 –l 1024  1.0.1.4
```

3. Click OK button on the command window to exit and once again click on the OK button on the HOST window to exit.

4. Repeat this step at HOST 2 and HOST3, but use different commands

```
stcp –p 23 –l 1024  1.0.1.5 at HOST2 stcp
–p 25 –l 1024  1.0.1.6 at  HOST3
```

5. Double click the left mouse button while cursor is on HOST4 to open the HOST window.

6. Select Add button on the HOST window to invoke the command window and provide the following command in the command textbox.

```
rtcp –p 21 –l 1024
```

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

9. Select LOG STATISTICS and select checkbox for Number of drop and collisions packets in the MAC window

10. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

11. Repeat this step at HOST 5 and HOST6, but use different commands

```
rtcp –p 23 –l 1024  at HOST5 rtcp
–p 25 –l 1024  at  HOST6
```

12. Double click the left mouse button while cursor is on HOST5 to open the HOST window.

13. Click NODE EDITOR Button on the HOST5 window and select the MAC tab from the modal window that pops up.

14. Select LOG STATISTICS and select checkbox for Number of drop and collisions packets in the MAC window

15. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

16. Also select the drop and collisions at HOST6.

Step3: Simulate

i. Click “R” icon on the tool bar

ii. Select Simulation in the menu bar and click/ select RUN in the dropdown list to execute the simulation.

iii. To start playback select “►” icon located at the bottom right corner of the editor.

iv. To plot congestion window select Tools in the menu bar and select PLOT GRAPH in the drop down list.

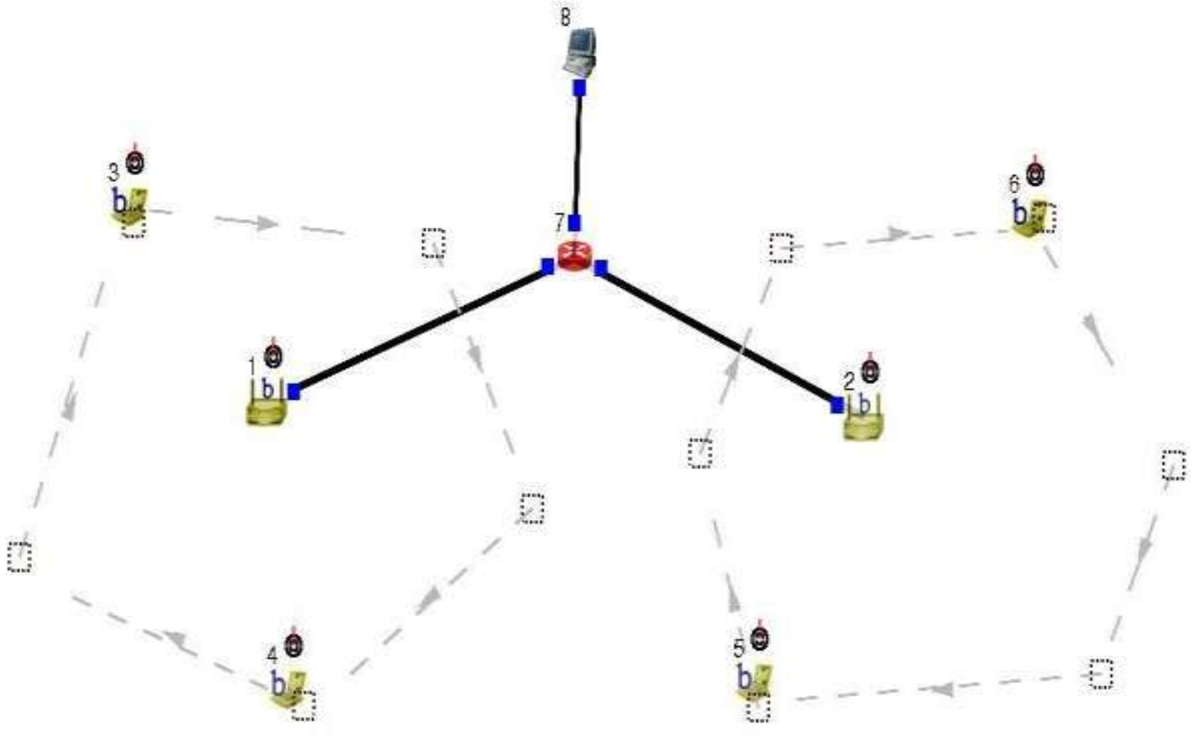
v. In the Graph window, select File->OPEN, move to file7.results folder and the drop and collision log file.

vi. To open another Graph window, Select File->New tab on the drop down list to open up to a maximum of 6 windows **vii.** To view results, Open up new TERMINAL window, move to file7.results folder and open input and output throughput log files in separate TERMINAL window.

Caution: file7 is the hypothetical name given to this simulation.

Expt. No 10

Simulate simple BSS and with transmitting nodes in wireless LAN by simulation and determine the performance with respect to transmission of packets.



Step1: Drawing topology

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

2. Select/click the ROUTER icon on the toolbar and click the left mouse button on the editor, to place ROUTER1 on the editor.

3. Select/click the WIRELESS ACCESS POINT(802.11b) icon on the toolbar and click the left mouse button on the editor, to place ACCESS POINT 1 on the editor.

   Repeat this procedure and place ACCESS POINT 2 on the editor.
4. Select/click the MOBILE NODE (infrastructure mode) icon on the toolbar and click the left mouse button on the editor, to place MOBILE NODE 1 on the editor.

Repeat this procedure and place MOBILE NODE 2, MOBILE NODE3 and MOBILE NODE 4 on the editor.

5. Click on the LINK icon on the toolbar and connect ACCESS POINT1 to ROUTER1 and ACCESS POINT2 to ROUTER1

6. Click on the "Create a moving path" icon on the toolbar and draw moving path across MOBILE NODE 1 and 2, Repeat for MOBILE NODE 3and 4 (Accept the default speed value 10 and close the window, Click the right mouse button to terminate the path).

To create Subnet

7. Select wireless subnet icon in the toolbar now select MOBILE NODE1, MOBILE NODE2 and ACCESS POINT1 by clicking on left mouse button, and clicking right mouse button will create a subnet.

8. Repeat the above step for MOBILE NODE3, MOBILE NODE4 and ACCESS POINT2.

9. Click on the "E" icon on the toolbar to save the current topology
   e.g: file8.tpl
   (Look for the ******.tpl extension.)

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

Step2: Configuration

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

```
ttcp –r –u –s –p 8001
```

3. Click OK button on the command window to exit

4. Repeat this step and add the following commands at

   HOST1 ttcp –r –u –s –p 8002

   ttcp –r –u –s –p 8003 ttcp
   –r –u –s –p 8004

5. Click NODE EDITOR Button on the HOST1 window and select the MAC tab from the modal window that pops up.

6. Select LOG STATISTICS and select checkbox for Input throughput in the MAC window

7. Click OK button on the MAC window to exit and once again click on the OK button on the HOST window to exit.

8. Double click the left mouse button while cursor is on MOBILE NODE 1 to open the MOBILE NODE window.

9. Select Application tab and select Add button to invoke the command window and provide the following command in the command textbox.

ttcp –t –u –s –p 80011.0.2.2 (host’s ip address)

10. Click NODE EDITOR Button on the MOBILE NODE1 window and select the MAC tab from the nodal window that pops up.

11. Select LOG STATISTICS and select checkbox for Output throughput in the MAC window

12. Click OK button on the MAC window to exit and once again click on the OK button on the MOBILE NODE1 window to exit.

13. Repeat the above steps (step 8 to step12) for the MOBILE NODE2,3 and 4 and add the following commands at

MOBILE NODE2:- ttcp –t –u –s –p 8002 1.0.2.2

MOBILE NODE 3:- ttcp –t–u –s –p 8003 1.0.2.2

MOBILE NODE4:- ttcp –t –u –s –p 8004 1.0.2.2

14. Double click the left mouse button while cursor is on ROTER1 to open the ROUTER window.

15. Click NODE EDITOR Button on the ROUTER1 window and you can see three stacks. two stacks for two ACCESS POINTS and another stack for HOST1 which is connected to the ROUTER1.

16. Select the MAC tab of ACCESS POINT1 and Select LOG STATISTICS and select checkbox for Input throughput in the MAC window. Click OK button on the MAC window to exit.

17. Select the MAC tab of ACCESS POINT2 and Select LOG STATISTICS and select checkbox for Input throughput in the MAC window. Click OK button on the MAC window to exit.

18. Select the MAC tab of HOST1 and Select LOG STATISTICS and select checkbox for Output throughput in the MAC window. Click OK button on the MAC window to exit.

Step3: Simulate

I. Click "R" icon on the tool bar

II. Select Simulation in the menu bar and click/ select RUN in the dropdown list to execute the simulation.

III. To start playback select "►" icon located at the bottom right corner of the editor. IV. MOBILE NODE's start moving across the paths already drawn.

Caution: file8 is the hypothetical name given to this simulation.