Certificate

Name: Shaurya Gupta Class: V^th Sem - A

Roll No: 1RV18CS150 Exam No: 18CS54

Institution R.V. College of Engineering

This is certified to be the bonafide work of the student in the Network Programming & Security *Laboratory during the academic year 2020/ 2021.*

No. of practicals certified _____ *out of* 20 7 *in the subject of* NPS (18CS54)

.....

Teacher In-charge

( )

.....

Examiner's Signature

.....

Principal

Date: 07/01/21.

Institution Rubber Stamp

(N.B: The candidate is expected to retain his/her journal till he/she passes in the subject.)

Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
1	Certificate	0			
2	Index	0			
3	Program - 1	1-3			
4	Program - 2	3 5			
5	Program - 3	5-10			
6	Program - 4	10-13			
7	Program - 5	13-17			
8	Program - 6	17-23			
9	Program - 8 7	23-26			
10	Program - 8	26-32			
11	Quabet) Question 1	35			
12	Question 2	36			
13	Question 3	37			
14	Question 4	38.			

Date

Expt. No. 1 Page No. 1

(Q) Implement a client - server Communicating through Socket programming.

* Client code

```
#include <arpa/inet.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/sockets.h>
#include <stdlib.h>
#include <sys/types.h>

int main( int argc, char* argv[])
{
 int sockfd, cont, fd, create_socket;
   int bufsize = 1024;
 Char* buffer = malloc(bufsize);
  Char *fname[256];
  Struct sockaddr_in address;
  if (( sockfd =socket (AF_inet, Sock stream, 0))>0)
       printf ("socket was created");
address.sin_family = AF_inet;
address.sin_port = htons (15000);
   inet_pton (AF_inet, argv[1], &address.sin_addr).
if (connect ( sockfd, (Struct (sockaddr*) & address),
       sizeof (address)) == 0).
```

Teacher's Signature

Output

Server

The Socket was created

Binding Socket.

The client 127.0.0.1 is Connected.

A request for a text is received.

request Completed.

Client.

The Socket was cred

Connection accepted

filename: a.txt

The Contents are

Make once great again.

EOF.

```
printf (" The connection is accepted the server %s--\n",
        argv[1] );
printf (" Enter filename "); scanf ("%s", frame);
send ( create_socket, frame, sizeof(frame), 0);
printf (" accepted request");
printf (" The Contents of the file are --");
while ((cont = recv (create_socket, buffer,
    bufsize, 0)) > 0) { write (1, buffer, cont);
}
printf ("\n EOF \n");
close ( create_socket)
return 0;
}
```

server code

```
// include all the headers from previous
int main ()
{
int cont, sockfd, newfd, addrlen, fd;
int bufsize = 1024;
char * buffer = malloc (bufsize);
char str [1024] = " file not found on server"
char frame [256];
struct sockaddr_in address;
if ((sockfd = socket (AF_inet, SockStream, 0)) > 0)
    printf (" socket was created Successfully \n");
address . sin_family = AF_INET
address . sin_addr . s_addr = INADDR_ANY
```

Teacher's Signature _____

```
address.sin_port = htons(15000)
if (bind (create_socket, (struct sockaddr*)& address,
    sizeof(address))==0)
printf ("Binding socket");
  listen (create_socket, 3);
addrlen = sizeof (struct sockaddr_in).
  new_socket = accept (create_socket, (struct sockaddr*)
      &address, &addrlen));
  recv (new_socket, frame 255,0).
  printf ("A request for file name is received \n", frame);
 if ((fd = open (frame, O_RDONLY))<0).
     { perror ("File error");
         write (new_socket, str, sizeof (str));
       exit(0);
     }
  while ((cont = read (fd, buffer, bufsize))>0) {
      send (new_socket, buffer, cont, 0);
  }
  printf ("Request Completed \n");
   close (new_socket);
    close (create_socket).
   return 0;
}
```

Teacher's Signature _____

Write a program to implement distance vector routing for a simple topology of routers.

```
#include <stdio.h>
struct node
{
   unsigned dist[20];
   unsigned from[20];
} rt[10];
int main()
{
    int costmat[20][20];
    int nodes, i, j, k, count=0;
    printf("Enter No of nodes:");
     scanf("%d", &nodes)
printf("Enter cost Matrix:");
    for(i=0; i<nodes; i++)
    {
       for(j=0; j<nodes; j++)
       {
          scanf("%d", &costmat[i][j]);
          costmat[i][j]=0;
          rt[i].dist[j] = costmat[i][j];
          rt[i].from[j]=j;
       }
    }
do
  {
```

Teacher's Signature _____

Output:

Enter the number of nodes

3

Enter the Cost Matrix:

0 2 7
2 0 1
7 1 0

for router 1

node 1 via 1 distance 0
node 2 via 2 distance 2
node 3 via 3 distance 3

for router 2.

node 1 via 1 distance 2
node 2 via 2 distance 0
node 3 via 3 distance 1

Expt. No. _____ Date _____

Page No. _____

```
count = 0;
for (i=0; i<nodes; i++)
  for (j=0; j<nodes; j++)
    for (k=0; k<nodes; k++)
      if (rt[i].dist[j] > costmat[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<nodes; i++)
  {
    printf ("\n for router %d \n", i+1)
    for (j=0; j<nodes; j++)
    {
      printf ("\n node %d via %d distance ",
              j+1, rt[i].from[j]+1, rt[i].dist[j]);
    }
printf ("\n\n");
getch();
}
```

Teacher's Signature _____

Enter IP header Information in 16 words

Field 1 abc
Field 2 aas
Field 3
Field 4
Field 5
Field 6
Field 7
Field 8
Field 9

Computed Checksum F499

Enter header

Field 1
Field 2
Field 3
Field 4
Field 5
Field 6
Field 7
Field 8
Field 9

Computed at receiver F499

Date

Expt. No. Page No.

3 Write a program to implement error detection and correction using Checksum and hamming code.

Checksum.c

```
#include <stdio.h>
unsigned fields[10];
unsigned short checksum(){
    int i;
    int sum=0;
  printf("IP header is in the format: 16 bit words");
    for(i=0;i<9;i++)
      {
      printf("\n Enter field %d:",i+1);
      scanf("%x",&fields[i]);

      sum=sum+(unsigned short)fields[i];
      while(sum>>16).
      sum=(sum&0xFFFF)
    }
    sum=~sum;
    return(unsigned short) sum;
  }
int main(){
  unsigned short r1,r2;
  r1=checksum();
  printf("Computed Checksum at sender %0x\n\n",r1);
  r2=checksum();
  printf("\n Computed checksum at rec %0x\n\n",r2);
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. _____

```
if (r1 == r2)
  printf ("\n No error");
else
  printf ("\n error in data recieved");
}
```

hamming code - c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void main()
{
  int n=6; t,i,j,k,nd,n,ch,sum=0,pos=0;
  int a[50], temp[70], temp2[70];
  printf ("Enter length of data string");
  Scanf ("%d", &nd);
  printf ("Enter data string")
  for (i=0; i<nd; i++)
    Scanf ("%d", &a[i]);
  for (i=0, j=0, i<nd, i++)
  {
    for (k=0; k<n; k++)
    {
      t = pow(2,k) - 1
      if (j==t).
    {
      temp[j]=0]
      j++;
    }
  }
}
```

Teacher's Signature _____

Enter length of data String 10

Enter data String 1011010101.

Length of hamming code = 14 bits

Number of parity bits 4

P1: 0
P2: 1
P3: 0
P4: 1

hamming code : receiver side
: 11011110101 01

P1 = 1
P2 = 0
P3 = 1
P4 = 0

Position of error 5.

hamming code corrected 01100111010101

Date

Expt. No. Page No.

```
temp[j] = a[i];
    j++;
}
nh = j;
printf("length of harming code = %d bits\n", nh);
n = nh - nd
printf("Number of parity bits: %d\n", n);
int b[n];
int m = n - 1;
for(k = 0; k < n; k++)
{
  t = pow(2, k) - 1
  for(i = t; i < nh;)
  {
   for(j = 0; j <= t; j++)
   {
     sum = sum + temp[i]
     i++;
     if(i >= nh) break;
   }
  if(i >= nh) break;
  for(j = 0; j < t; j++)
  {
     i++;
     if(i >= nh) break;
  }
  if(i >= nh) break;
}
temp[t] = sum % 2;
   sum = 0;
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. _____

```
printf (" Parity bits p %d: %d \n", t+1, temp[t]);
}
printf (" Sender side ");
for(i = 0; i<nh; i++)
    printf ("%d ", temp[i]);
printf (" Receiver Side ");
for (i=0; i<nh; i++)
    scanf ("%d", &temp2[i]);
}
Scum = 0;
for(k=0; k<n; k++)
    t = pow(2,k) - 1;
for( i=t; i<nh; )
{
    for(j=0; j<=t; j++)
    {
        Scum = Scum + temp2[i];
        i++;
    }
    if (i > nh) break;
}
if(i > nh) break;
for( j=0; j<=t; j++)
    i++;
}
b[m] = Scum % 2
Scum = 0
printf (" No of parity bits are P%d: %d \n", t+1, b[m
m--;
}
```

Teacher's Signature _____

```
for (m=0; m<n; m++)
{
    pos = pos + b[n-m-1] * pow(2,m);
}
printf(" position of error : %d \n pos);
if (temp2[pos-1] == 0)
    temp2[pos-1] = 1;
else
    temp2[pos-1] = 0;

printf("\n hamming code : Receiver side error
        corrected ");
for (i=0; i<nh; i++)
{
    printf("%d", temp2[i]);
}
```

Expt. No. _____ Date _____

Page No. _____

(4) Implement a Simple Multicast routing Mechanism.

Sender.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <net/inet.h>
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#define Hello_port 12345
#define Hello_Group "225.0.6.37"
int main (int argc, char *argv[])
{
struct sockaddr_in addr;
int fd, cnt;
struct ip_mreq mreq;
char *message = "RVCE-CSE";
if ((fd=socket(AF_inet, SOCK_DGRAM,0))<0) {
        perror("Socket");
        exit(1);
  }
memset(&addr,0,sizeof(addr));
addr.sin_family = AF_inet;
addr.sin_addr.s_addr = inet_addr(Hello_group);
addr.sin_port = htons(hello_port);
while(1) {
  if (sendto(fd,message,sizeof(message),0,(struct
                                          sockaddr
```

Teacher's Signature _____

Output

./sender	./listener	./listener2
	RVCE-CSE	RVCE-CSE
	RVCE-CSE	RVCE-CSE
	RVCE-CSE	
	RVCE-CSE	
	RVCE-CSE	
	RVCE-CSE	

Expt. No. _____ Date _____

Page No. _____

```
&addr
,sizeof(addr)<0){
        perror("send error");
    exit(1);
}
  sleep(1);
}
 return 0;
}
```

listener.c

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<arpa/inet.h>
#define Hello_port 12345
#define hello_group "225.0.0.37"
#define Msgbufsize 25
int main (int argc, char *argv[])
 {
  struct sockaddr_in addr;
int fd, nbytes, addrlen;
 struct ip_mreq mreq;
char msgbuf[msgbufsize];
 u_int yes=1.
memset (&addr, 0, sizeof(addr));
addr.sin_addr.s_addr= htonl(INaddrAny)
```

Teacher's Signature _____

Date

Expt. No. Page No.

```
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl(INADDR_ANY);
addr.sin_port = htons(hello_port);
if (bind(fd, (struct sockaddr *)&addr, sizeof(addr)) < 0)
{
    perror("bind");
    exit(1);
}
mreq.imr_multiaddr.s_addr = inet_addr(hello_group);
mreq.imr_interface.s_addr = htonl(INADDR_ANY);
if (setsockopt(fd, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq,
        sizeof(mreq)) < 0) {
    perror("setsockopt");
    exit(1);
}
while(1)
addrlen = sizeof(addr);
if ((nbytes = recvfrom(fd, msgbuf, MSGBUFSIZE, 0,
        (struct sockaddr *)&addr, &addrlen)) {
    perror("recvfrom");
    exit(0);
}
puts(msgbuf);
}
}
```

Date _____

Expt. No. _____ Page No. _____

5 Write a program to implement concurrent chat server that allows current logged in users to communicate with each other.

In this Program, Multiple instances of Client can be created.

Server.c

```
#include <stdio.h>
#include <sys/socket.h>
#include <net_inet.h>
#include <string.h>

int Compare_strings (char a[], char b[])
{
  int c=0;
while(a[c]==b[c])
  {
    if(a[c]=='\0' || b[c]=='\0')
      break;
    c++;
  }
  if (a[c]=='\0' && b[c]=='\0')
    return 0
  else
    return -1;
}
```

Teacher's Signature _____

Output of Server:

/. server

The Socket was created
Binding Socket
Server listening.
Client 1 joined
Client 2 joined

Output of Clients:

/. client 1

Client 1 to Client 2.
Hi
Client 2 to Client 1
hello.

/. client 2.

Client 2 to Client 1
hi
hello.

Date

Expt. No. Page No.

```
int main() {
  int welcomeSocket, client1, client2;
  struct sockaddr_in serveraddr;
  struct sockaddr_storage serverStorage;
  socklen_t addr_size;
  char buffer[1024];
welcomeSocket = socket(AF_inet, SockStream, 0);
  serverAddr.sin_family = AF_inet
  serverAddr.sin_port = htons(7811)
  serverAddr.sind_addr.s_addr = inet_addr(127.0.0.01)
memset(serverAddr.sin_zero, '\0', sizeof serverAddr.sin_zero);
bind(welcomeSocket, (struct sockaddr *)&serveraddr, sizeof
                                        (serveraddr))

if(listen(welcomeSocket, 5)==0)
      printf("listening\n");
else
      printf("Error\n");

addr_size = sizeof serverStorage;
Client1 = accept(welcomeSocket, (struct sockaddr *)&serverStorage, &addr
Client2 = accept(welcomeSocket, (struct sockaddr)&serverStorage, &addr
                                                          size
int cmdExit=0.
while(cmdExit==0)
{
  recv(Client1, buffer, 1024, 0);
  printf("%s\n", buffer);
send(Client2, buffer, 1024, 0);
if(compare_strings(buffer, "exit")==0)
  cmdExit=1;
```

Teacher's Signature

```
else
{
    memset (&buffer[0], 0, sizeof(buffer));
    recv (client2, buffer, 1024, 0);

    printf ("%s\n", buffer);
    send (client1, buffer, 1024, 0);

    if (compare_strings (buffer, "exit") == 0)
    {
        cmdEXIT = 1
    }
}
}
return 0;
}
```

Client.c

```
// include headers from previous

int main () {
    int clientsocket;
    char buffer [1024]
    struct sockaddr_in serverAddr;
        socklent addr_size;
    int cmdExit = 0;
Client SOCKET = socket (PF_inet, Sock_Stream, 0);
```

Teacher's Signature _____

Date

Expt. No. Page No.

```
serverAddr.sin_family = AF_INET;
serverAddr.sin_port = htons(7891);
serverAddr.sin_addr.s_addr=inet_addr("127.0.0.1");
memset(serverAddr.sin_zero,'\0', sizeof serverAddr.sin_
                                        zero);
addr_size = sizeof serverAddr;

connect(clientSocket,(struct sockaddr*)&serverAddr,addr_size);
printf("Client 1 :")
Scanf("%[^\n]s",buffer);
send(clientSocket, buffer, sizeof buffer-1,0);
while(endexit == 0)
{ if (compare_strings(buffer,"exit")==-1)
  {
    memset(&buffer[0],0, sizeof(buffer)));
  int recvalue = recv(clientSocket,buffer, size buffer-1,0)
   if (recvalue != 1)
   {
     if (compare_strings(buffer "exit")==-1)
     { printf("Client 2 :");
       printf("%s \n", buffer);
       memset(&buffer[0], 0, sizeof(buffer));
     }
     else endExit=1;
   }
else { printf("Client 1:");
       scanf("%[^\n]s, buffer);
       send(ClientSocket  buffer, buffer-1,0);
   }}
{ else endexit = 1;
return 0;
```

Teacher's Signature

Date

Expt. No. Page No.

6 Implementation of Concurrent and iterative echo server using both connection and connectionless system calls

UDP server (connectionless)

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
void str_echo(int sockfd, struct sockaddr * cli_addr,
              int clilen).
{  int n;
   int bufsize = 1024;
   char * buffer = malloc(bufsize);
   int addrlen;
   for(;;){
     addrlen = clilen;
   n = recvfrom(sockd, buffer, bufsize, 0, cliaddress,
       &addrlen);
   sendto(sockfd, buffer, n, 0, cliaddress, addrlen);}
}
int main()
{
 int sockfd;
struct sockaddr_in serv_address, cli_address;
```

Teacher's Signature

Output

./udpc 127.0.0.1

The socket was created

hey

hey

./udps

The server is connected

Expt. No. _____ Date _____

Page No. _____

6 Implementation of Concurrent and iterative echo server using both connection and connectionless system calls

UDP server (connectionless)

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
void str_echo(int sockfd, struct sockaddr * cli_addr
                int clilen).
{  int n;
   int bufsize = 1024;
   char * buffer = malloc(bufsize);
   int addrlen;
   for(;;){
     addrlen = clilen;
   n = recvfrom(sockd, buffer, bufsize, 0, cliaddress,
       &addrlen);
   sendto(sockfd, buffer, n, &cliaddress, addrlen);}
}
int main()
{
  int sockfd;
struct sockaddr_in serv_address, cli_address;
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. _____

```
if ((sockfd = socket(AF_inet, SOCK_DGRAM, 0)) < 0).
        printf ("the socket was created").

cserv_address.sin_family = AF_INET;
serv_address.sin_addr.s_addr = INADDR_ANY;
user_address.sin_port = htons(16001);
printf ("the address before bind %s---\n", inet_ntoa(
    serv_address.sin_addr));
if (bind(sockfd, (struct addr *)&serv_address,
    sizeof(serv_address)) = 0)
    printf ("binding socket");
str_echo(sockfd, (struct sockaddr *)&cli_address,
    sizeof(cli_address));
return 0;
}
```

UDP Client-c

```
// including all the previous headers

void str_cli (FILE *fp, int sockfd, struct sockaddr
        serv_address, int servlen).

{
    int buffsize = 1024, cond;
    Char* buffer = malloc (bufsize);
    while (fgets (buffer, bufsize, fp) != NULL){
send to (sockfd, buffer, sizeof(buffer), 0, serv_address,
        servlen);
if ((cond = recvfrom(sockfd, buffer, bufsize, 0, NULL, NULL)
}
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

```
fputs(buffer, stdout);
}}
printf("\n EOF \n");
}
int main(int argc, char *argv[]).
{
int sockfd
struct sockaddr_in ser_address;.
if((sockfd = socket(AF_net, sock_DGRAM,0))>0)
  printf("the socket was created \n");
ser_address.sin_family = AF_INET;.
ser_address.sin_port = htons(16002
inet_pton(AF_INET,argv[1],&ser_address.sin_addr);
str_cli(stdin, sockfd, (struct sockaddr*) &ser
    address, sizeof(ser_address));
exit(0);
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

Connection oriented :

Server :

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <fcntl.h>

void str_echo(int connfd)
{
    int n;
    int bufsize = 1024;
    char * buffer = malloc(bufsize)

again: while ((n = recv(connfd, buffer, bufsize, 0)) > 0)
           send(connfd, buffer, n);
       if (n < 0)
           goto again;
}
int main()
{
    int con, listenfd, connfd, addrlen, addrlen2, fd
    pid; addrlen3;
    struct sockaddr_in address, cli_address;
    if ((listenfd = socket(AF_INET, SOCK_STREAM, 0)) > 0)
        printf("the socket was created");
    address.sin_family = AF_INET;
```

Teacher's Signature _____

Output.

./ic 127.0.0.2 (for iterative)

The socket was created.
The client is connecting to the server 127.0.0.2.

hey

./ic 127.0.0.1. (for Concurrent)

The socket was created..

hey
hey.

Expt. No. _____ Date _____

Page No. _____

```
address.sin_addr.s_addr = INADDR_ANY;
address.sin_port = htons(1500);
printf("The address before bind %s--\n", inet_ntoa
    (address.sin_addr));
    printf("Binding Socket");
printf("The address after bind %s--\n", inet_ntoa
    (address.sin_addr));
listen(listen_fd, 3);
printf("Server is listening \n");
for(;;){
    addrlen = sizeof(struct sockaddr_in);
    confd = accept(listenfd, (struct sockaddr*)
        &cli_address, &addrlen);
printf("The client %s is connected on port %d\n",
    inet_ntoa(cli_address.sin_addr), htons
    (cli_address.sin_port));
    str_echo(confd);
    close(confd);
}
return 0;
}
```

TCP

Client.c

```
#including all the previous header
void str_cli(FILE *fp, int sockfd);
{
    int bufsize = 1024, cont;
    char *buffer = malloc(bufsize);
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

```
while (fgets (buffer, bufsize, fp)!= NULL)
{
    send (sockfd, buffer, sizeof (buffer), 0);
    if ((cnt = recv( sockfd, buffer, bufsize, 0))>0){
        fputs (buffer, stdout);
    }
}
printf ("\nEOF\n");
}
int main (int argc, char * argv[])
{
    int create_socket;
    struct sockaddr_in address;
if ((create_socket = socket(AF_inet, SOCK_Stream, 0))>0).
    printf ("The socket is created \n");
address.sin_family = AF_INET;
address.sin_port = htons(15000);
inet_pton (AF_inet, argv[1], &address.sin_addr);

if (connect (create_socket, (struct sockaddr *)&address,
        sizeof (address))==0).
    printf ("The client is connecting to server o/s -- %s\n
                                    argv[1]);
else
printf ("error in connect\n");

str_cli (stdin, create_socket);

return close (create_socket);
}
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

7. Implementation of remote command execution using socket system calls:.

Server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <net/inet.h>
int main()
{
int sd, acpt, clnt, bytes, port;
char send[50], recev[50];
struct sockaddr_in serv, cli;
if ((sd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        printf("socket error");
        exit(0);
    }
    printf("The socket was created \n");
bzero(&serv, sizeof(serv));
serv.sin_family = AF_inet.
serv.sin_port = (15002);
serv.sin_addr.s_addr = htonl(INADDR_ANY);
printf("The address before bind d --- \n", inet_ntoa
    (serv.sin_addr));..
if (bind(sd, (struct sockaddr *) &serv, sizeof(serv)) < 0
```

Teacher's Signature _____

Output

Server side (./server)

Client Connected

① Downloads Desktop
NPS a.sql

② home

Client Side (./client)

Enter Command : ls
Enter Command pwd.

Enter Command : touch a

Expt. No. _____ Date _____

Page No. _____

```
{
printf (" Error in bind \n"); exit (0); }
      printf (Binding socket)
printf ("The address after bind is ... \n", inet_ntoa
                                    (serv.sin_addr);
if (listen (sd, 3) < 0).
  { printf ("Error in listen \n"); exit (0) }.

if ( acpt = accept (sd, (struct sockaddr *) NULL, Null)) < 0)
   printf (" accept error");
while (1)
{
bytes = recv (acpt, recin, 50, 0); recin [bytes] = '\0';
if ( strcmp (recin, " end ") == 0).
  {
      close (acpt);
      close (sd);
      exit (0);
  }
else printf ("Command received : %s" recin, system (recin)
                                    printf ("\n"));).
```

Teacher's Signature _____

Client-C

```
// including all headers from server
int main()
{
  int sd, acpt, dw, bytes, port;
  char send1[50], recv[50];
  struct sockaddr_in serv, cli;
  if ((sd = socket(AF_inet, SOCK_Stream, 0))<0).
    {
      printf("Error in socket\n");
      exit(0);
    }
  bzero(&serv, sizeof(serv));
  serv.sin_family = AF_INET
  serv.sin_port = htons(1502);
  serv.sin_addr.s_addr = htonl(INADDR_ANY);
  if (connect(sd, (struct sockaddr*)&serv, sizeof(serv))<
    {
      perror("connection error");
      exit(0);
    }
  while(1)
  {
    printf("Enter the Command");
    gets(send
    if (strcmp(send1, "end"))!=0)
      send(sd, send1, 50, 0);
      close(sd)
      break; }}}
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. _____

8 Write a program to encrypt and decrypt the data using RSA and exchange the key securely using Diffie Hellman key exchange protocol.

RSA.C

```
#include <stdio.h>
#include <stdlib.h>
#include<math.h>
#include<string.h>
int x, y, n, t, i, flag;
long int e[50], d[50], temp[50], j, m[50], en[50];
char msg[100];
int prime_key
int encryption_key();
int cd(long int);
int encrypt();
int decrypt.
int Main()
{
  printf("Enter first prime");
  scanf("%d", &x);
  flag = prime(x);
  if(flag==0)
  {
    printf("\n Invalid Input:\n"
    exit(0);
}
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. _____

```
printf("\n Enter second prime ")
Scanf("%d", &y);
flag = prime(y);
if(flag == 0 || x == y).
{
  printf("Invalid Input")
    exit(0);
}
printf("Enter a string to Encrypt");
Scanf("%s", msg).
for(i=0; msg[i] != Null; i++).
    ms[i] = msg[i].
n = x * x;
t = (x-1) * (y-1);
encryption key();
printf("The possible values of e and d are \n");
  for(i=0; i<y-1; i++).
printf("\n %d to %d", e[i], d[i]
  encrypt()
  decrypt()
  return 0;
}
int prime(long prime)
{
  int i;
  j = sqrt(pr);
  for(i=2; i<=j; i++)
  { if(pr%i == 0).
      return 0;
  }
  return 1;
}
```

Teacher's Signature _____

```
void cencryption_key().
{ int K;
  K = 0;
for (i=2; i<t; i++)
{
   if (t%i == 0)
     continue;
flag = prime.
   if (flag > 0)
   {
      d[K] = flag;
      K++;
   }
  if(K=99) break;
}
void encrypt ()
{
 long int pt, ct, key1=e[0], K, len;
  i = 0;
  len = strlen (msg);
while (i != len).
{
  pt = m[i];
  pt = pt - 96;
  K = 1
for (j = 0; j < key; j++)
  { K = K * pt;
    K = K % n;
}
```

Teacher's Signature

```
temp[i] = k;
ct = k + 96;
en[i] = ct;
i++;
}
en[i] = -1;
printf("the encrypted message is \n");
for(i=0; en[i] != -1; i++)
    printf("%c", en[i]);
}
void decrypt()
{
int i, pt, ct, key = d[0], k;
i = 0;
while(en[i] != -1)
{
ct = temp[i];
k = 1;
for(j=0; j<key; j++)
{
k = k * ct;
k = k % n;
}
pt = k + 96;
m[i] = pt;
i++;
}
m[i] = -1;
printf("\n the decrypted Message is ");
printf("%s", m);
printf("\n");
```

Page No.

Teacher's Signature

Output :

Enter first prime Number

3

Enter second prime Number

11

Enter Message or string to ENCRYPT

RVCE IS GREAT

Possible values of e and d are

7	3
13	17
17	13

The encrypted Message is YN|ULD|

The decrypted Message is RVCEISG

(ii) DH-C (diffie hellman program)

```
#include <stdio.h>
int compute (int a, int m, int n)
{
    int r;
    int y = 1;
    while (m > 0)
    {
        r = m % 2;
        if (r == 1)
            y = y(y * a) % n;
        a = a * a % n;
        m = m / 2;
    }
    return y
}
int main ()
{
    int p = 23;
    int q = 5;
    int a, b;
    int A, B;
    srand (time(0)).
    a = rand ().
    A = compute (q, a, p);
    srand (time (0));
    b = rand ();
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. _____

```
B = compute (g,b,p);

int keyA = compute (B,a,p);
int keyB = compute (A,b,p);

printf (" Alice's secret key is %d.", keyA);
printf (" Bob's secret key is %d" keyB);
```

Output : ./a.out

Alice's secret key is 3

Bob's secret key is 3.

Part B : Qualnet Simulation ..

Date

Expt. No. Page No.

Qualnet: A Brief Introduction

It is a planning, testing and training tool that mimics the behaviour of real network / communication. It is cost effective method for the development deployment and managing network - centric systems throughout their entire lifecycle. Users can test combination / that is likely to work by creating scenarios, protocols and analyzing their performance.

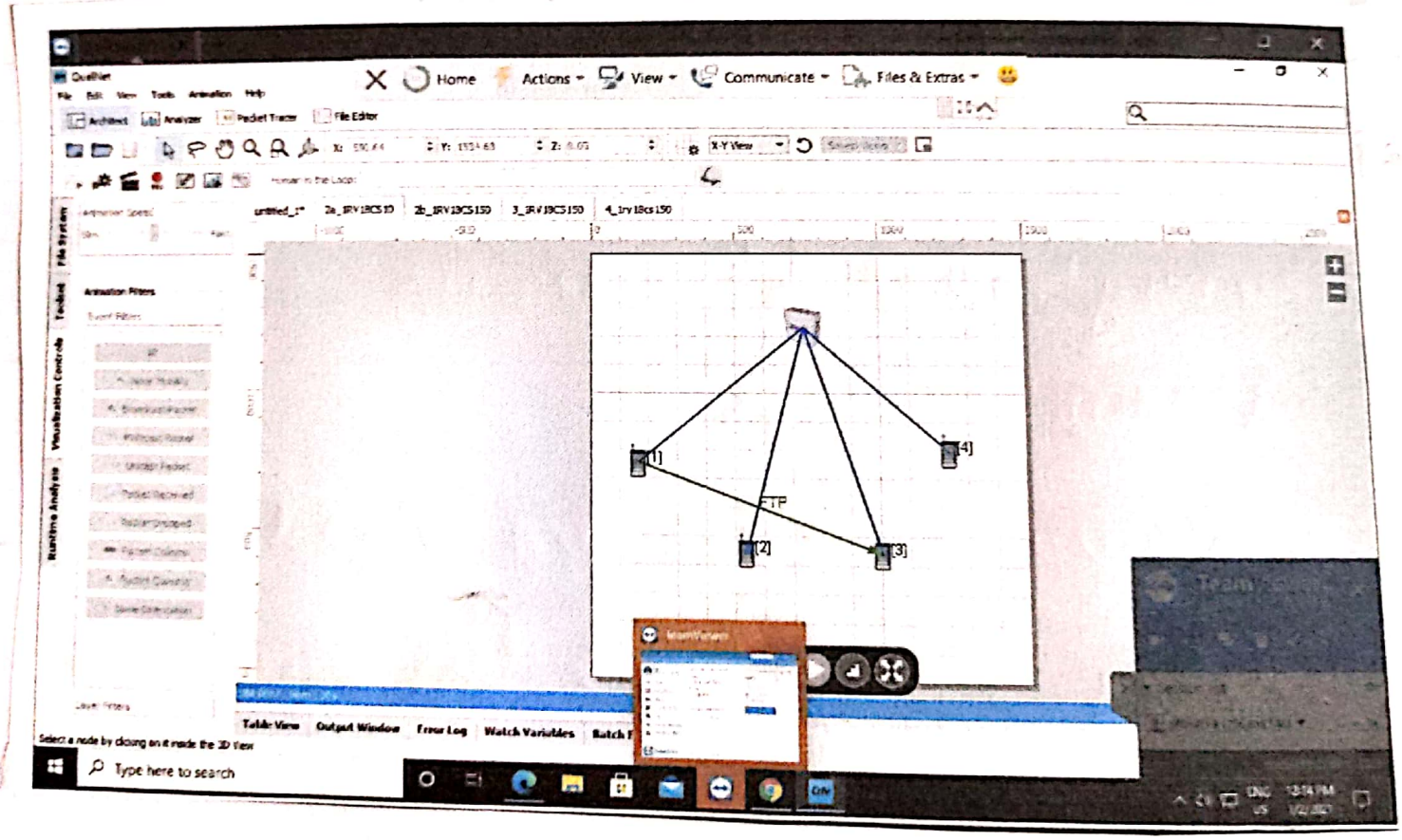
Features Of Qualnet:

1) Designing new Architecture
2) Running "What if" scenarios
3) Test New radios
4) Simulating response and impact.
5) Modelling Mobile wireless equipment.

Teacher's Signature

(Q-1) Setup an IEEE 802.3 network with a hub. Apply the file transfer protocol between nodes.

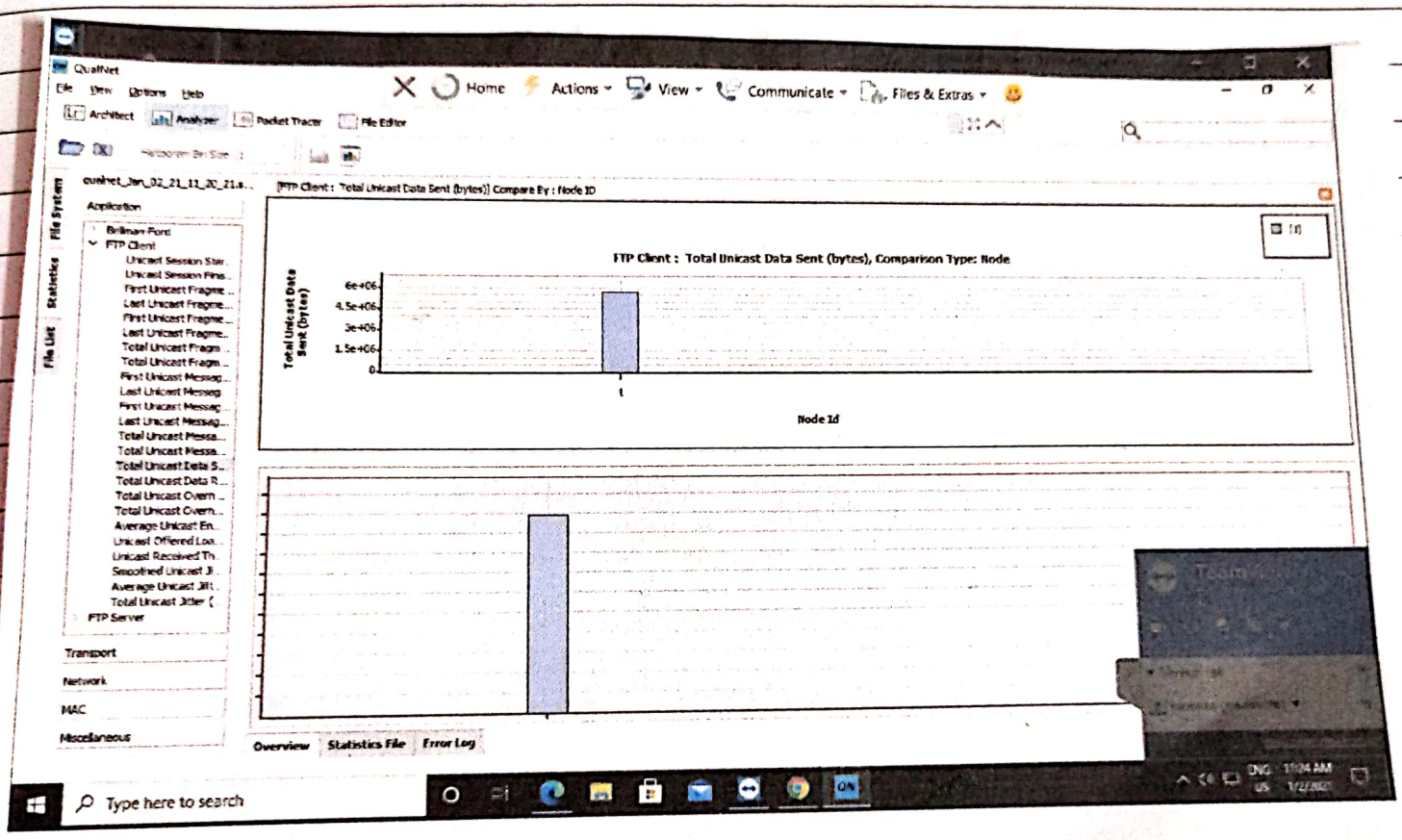
(A) The following simulation is a solution to the given day night. It consist of various devices such as hubs etc.



Expt. No. _____ Page No. _____

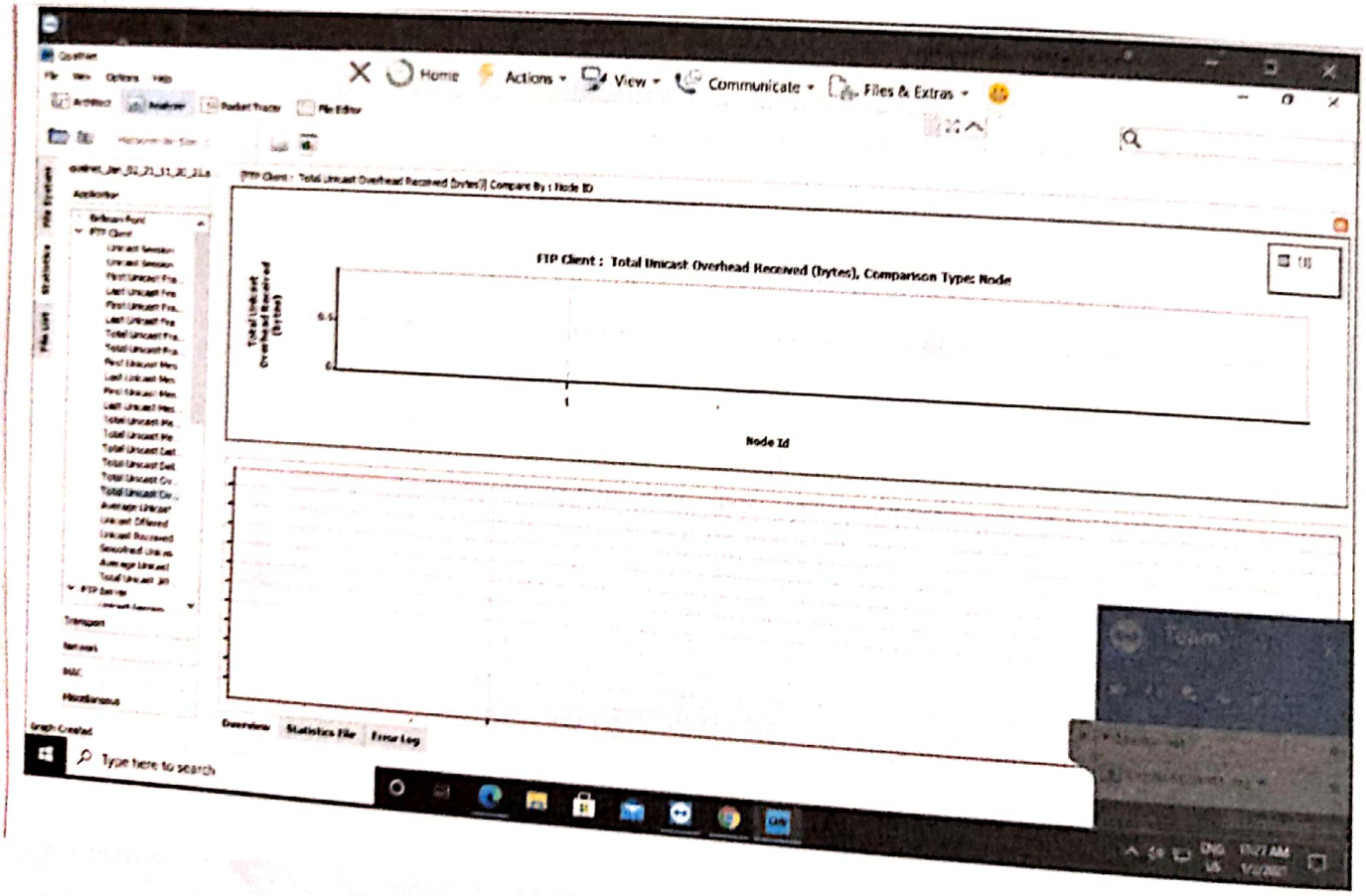
Date _____

(Q-1-A). : The following diagram tracks the changes in the FTP service tracking between nodes. This is a file transfer client and the graphs here that graph represents, the data sent the probable (per-byte) the total total unicast data bytes using or nodes set to its respective.



Teacher's Signature _____

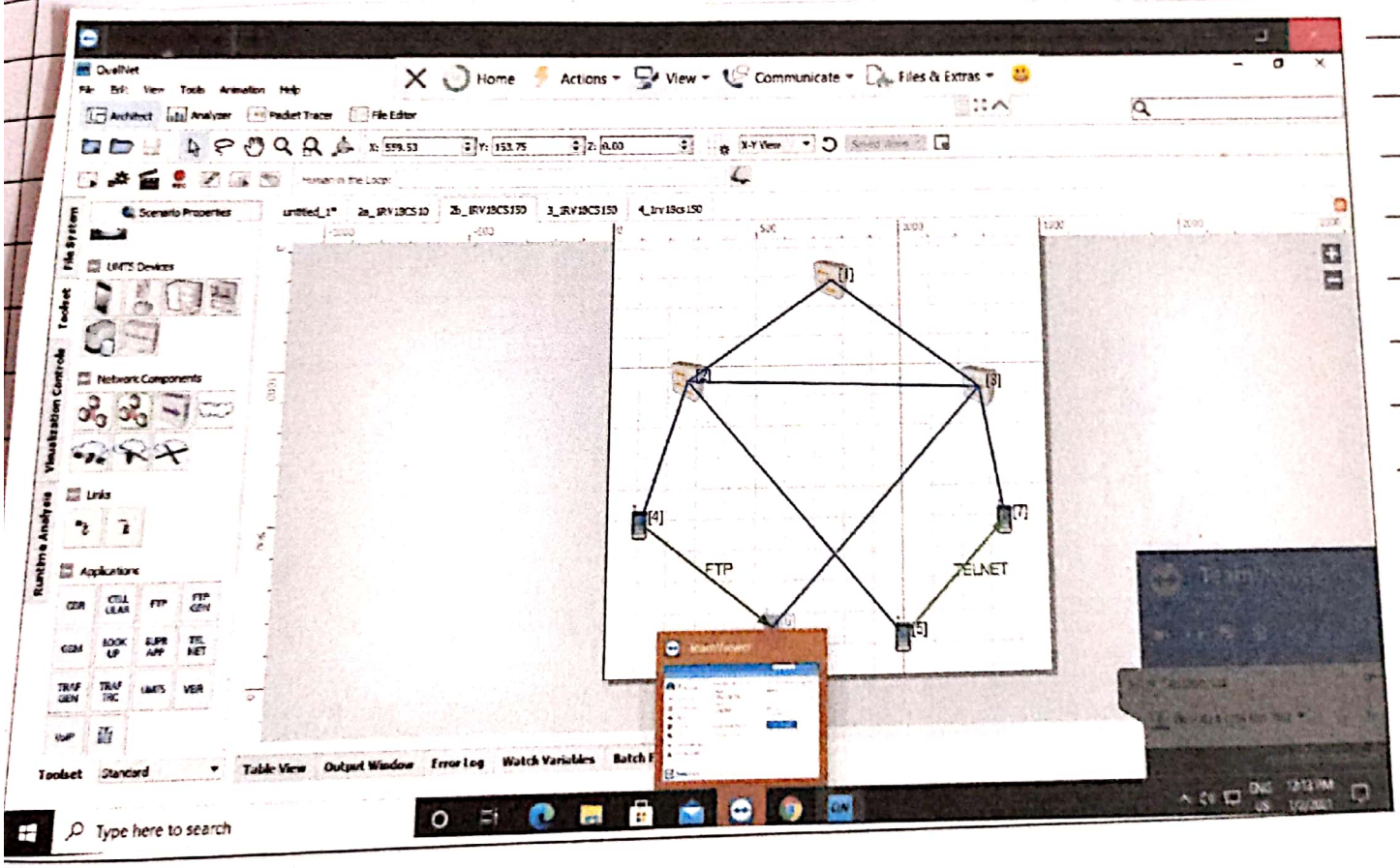
(Q-1-b) The following graphs show us another information on our question is to Map the Overhead data being Send / received while there is remode by to calculate Overhead expenses.



Expt. No. _____ Page No. _____

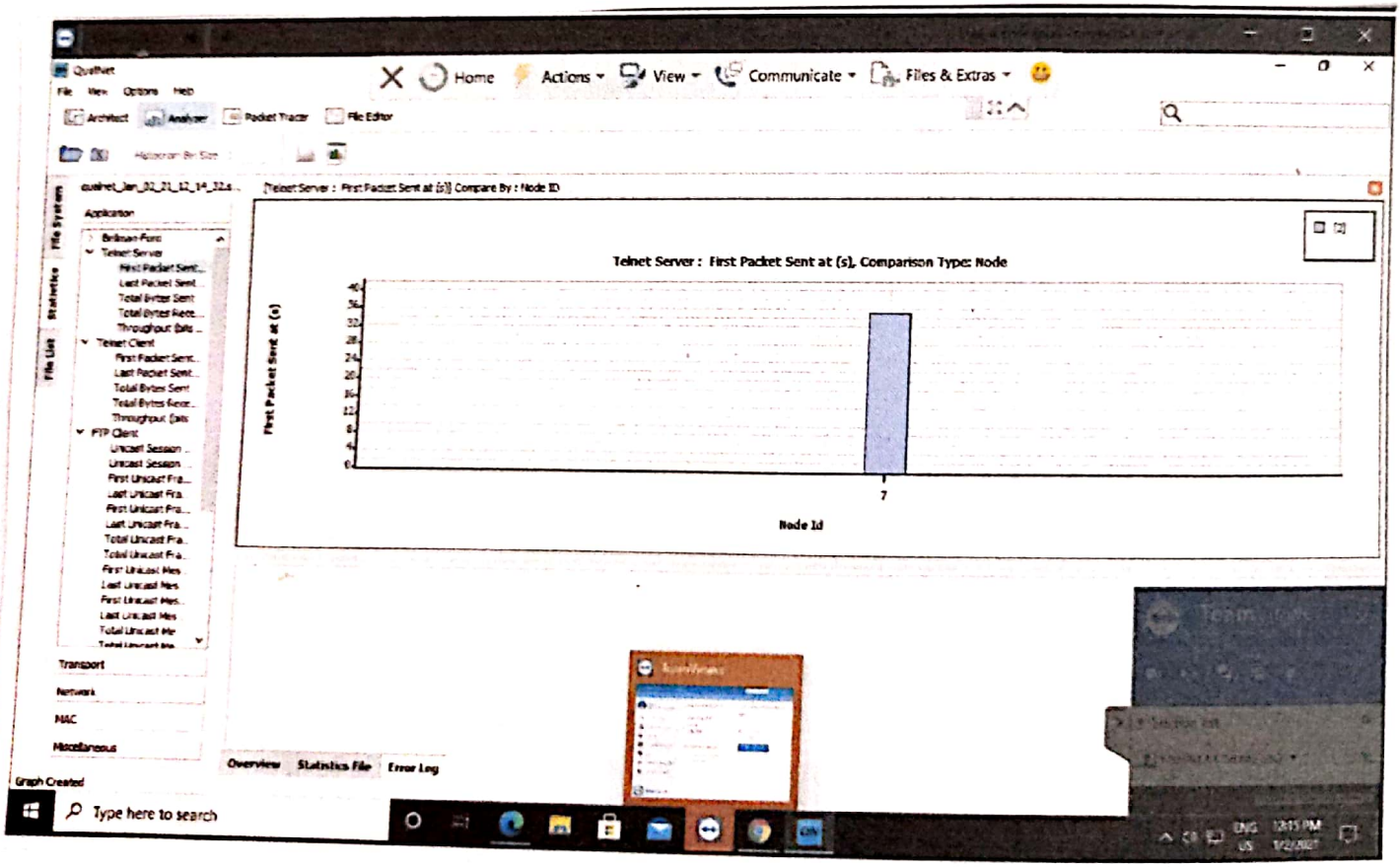
Q-2) Same question but with a different Construct (Switch & Hierachy).

According to our questions, we have made a Node on Blackboard book which explicitly states that No. of No 8. We also connected two servers Cisco .. We have made. Cisco servers as well.

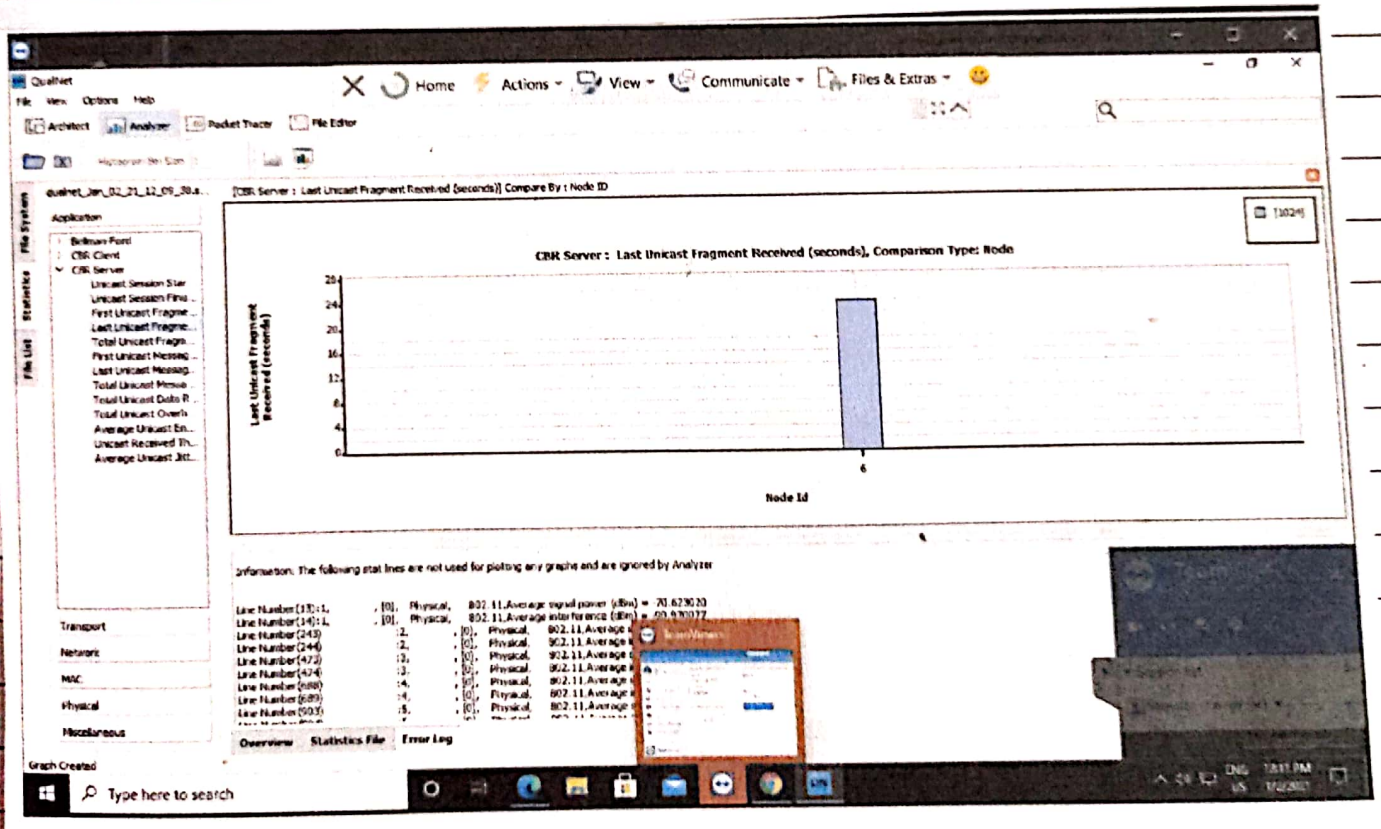


Teacher's Signature _____

(Q-2-a) The following figure, in accordance shows us. the first Packet sent by the simulator / nodes

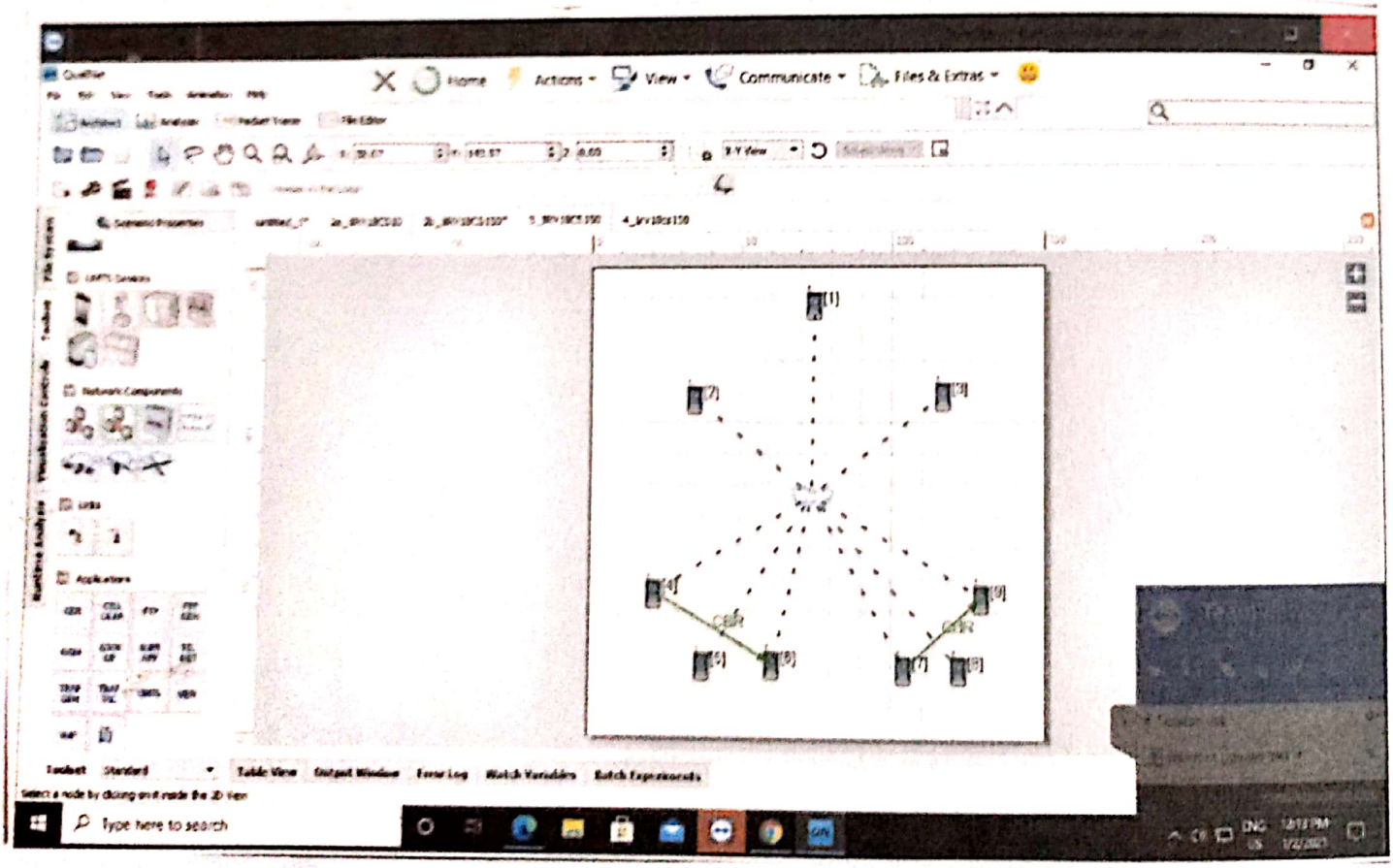


(Q-2-b) The following code shows the last fragment, in our condition this experiment model adds/ asks about and also lists Unicast CBR fragment receipt generator.



Teacher's Signature _____

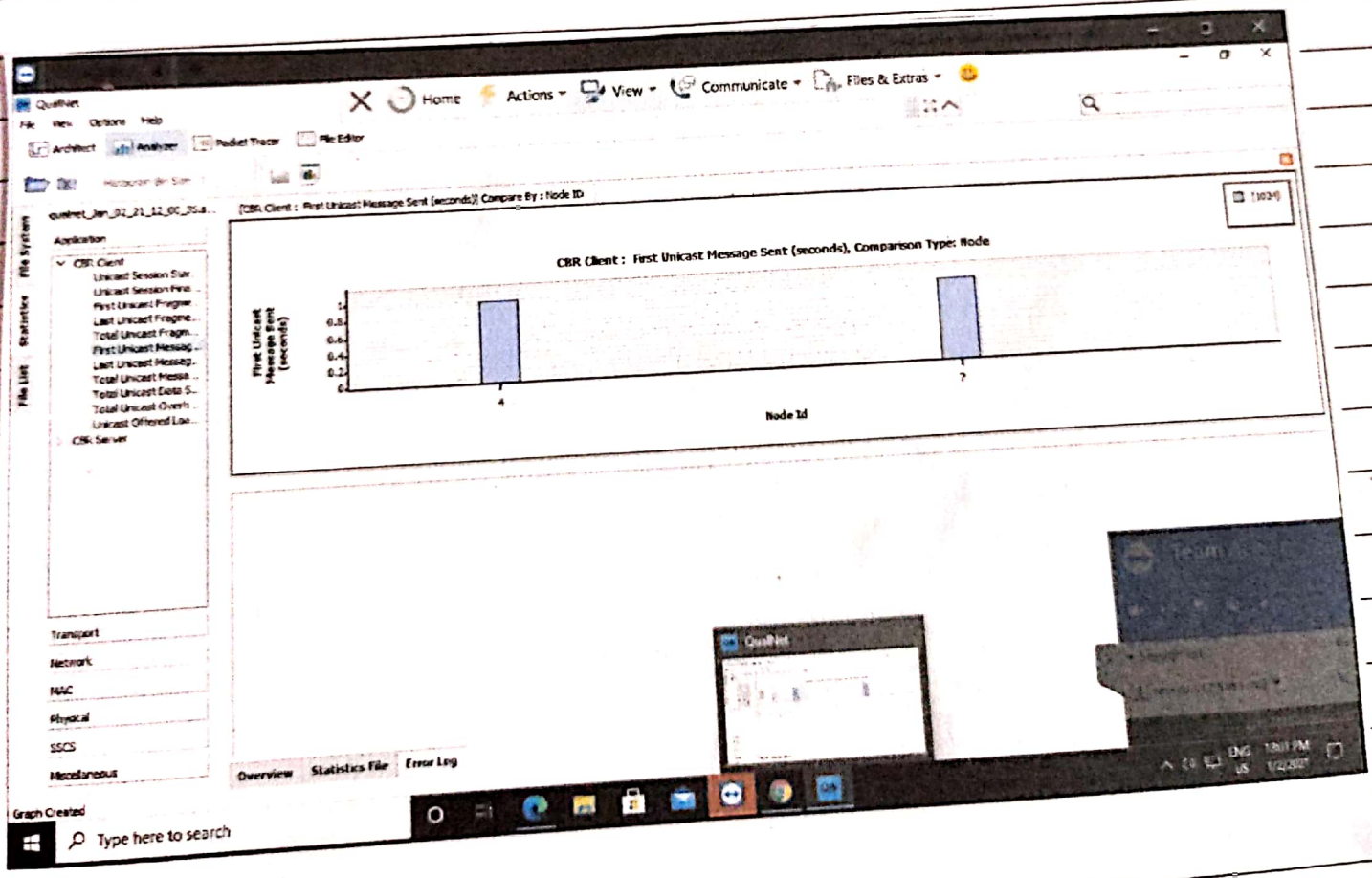
(Q-3-a) Set up a wireless sensor Network with atleast 2 devices. CSR and VBR has to be Provided. application between single nodes. coordinators in the same.



Expt. No. _____ Page No. _____

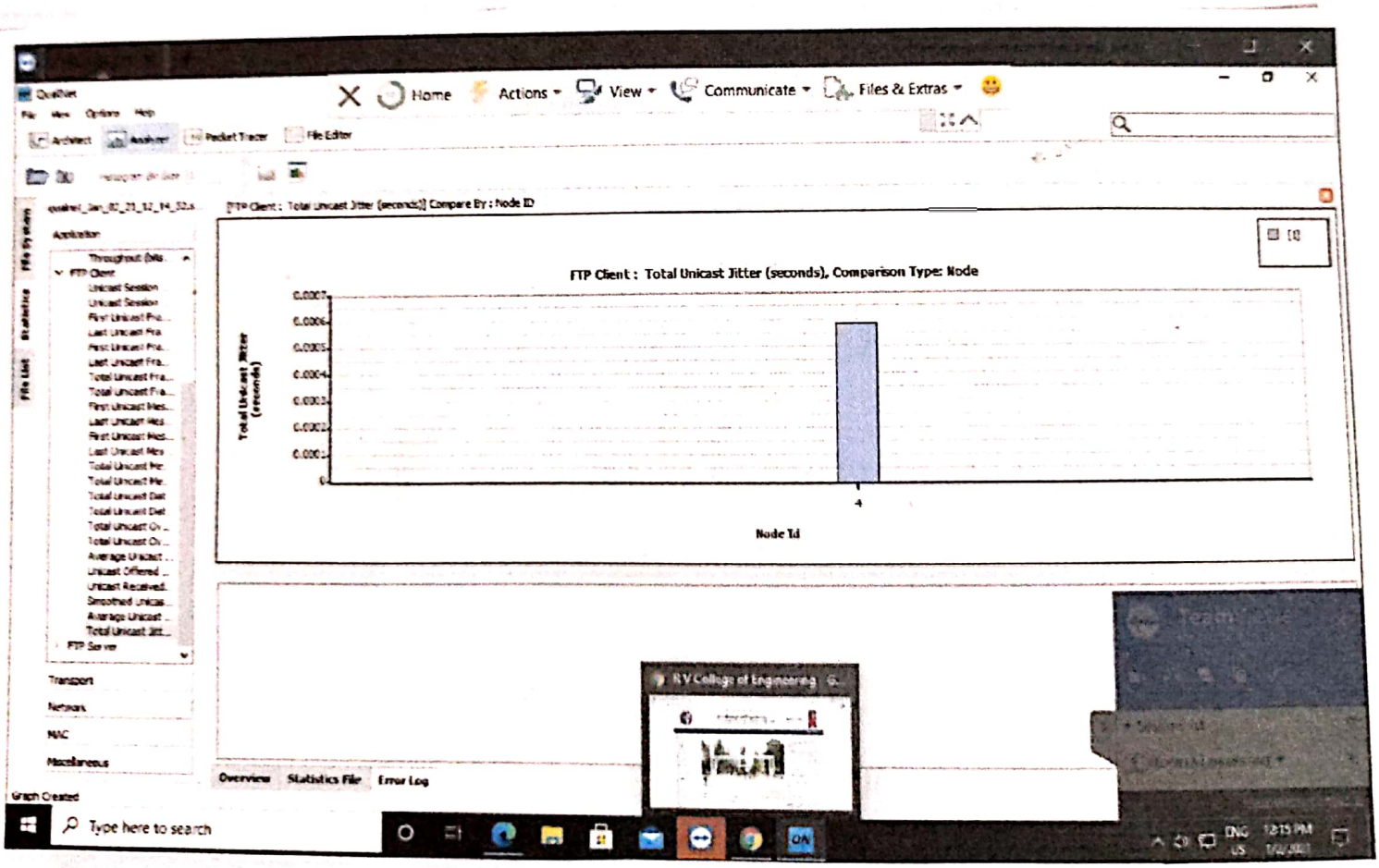
Q-3-(d) The following experiment can generate this graph, basically it helps us to coordinate of performance. It also tells us the first bit of a Encoding send.

We can use it MAC layer.



Teacher's Signature _____

(Q-3b) This stat helps us to Calculate the jitter.

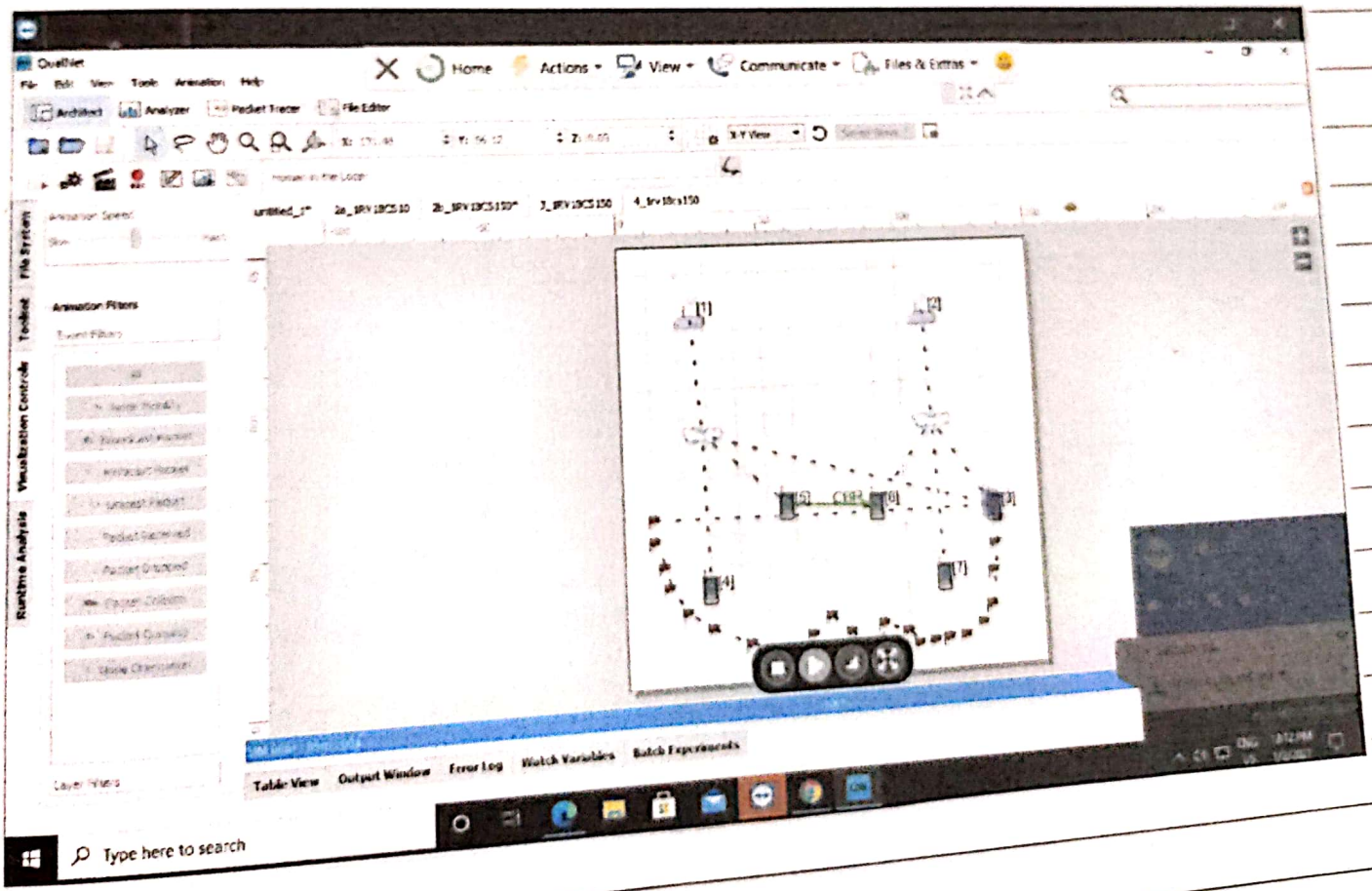


Expt. No. _____ Date _____

Page No. _____

(Q-4) Setup up IEEE 802.13 with two access point. Vary out the results.

This setup, helps us to let a device roam, infinetly through physical and MAC layers.

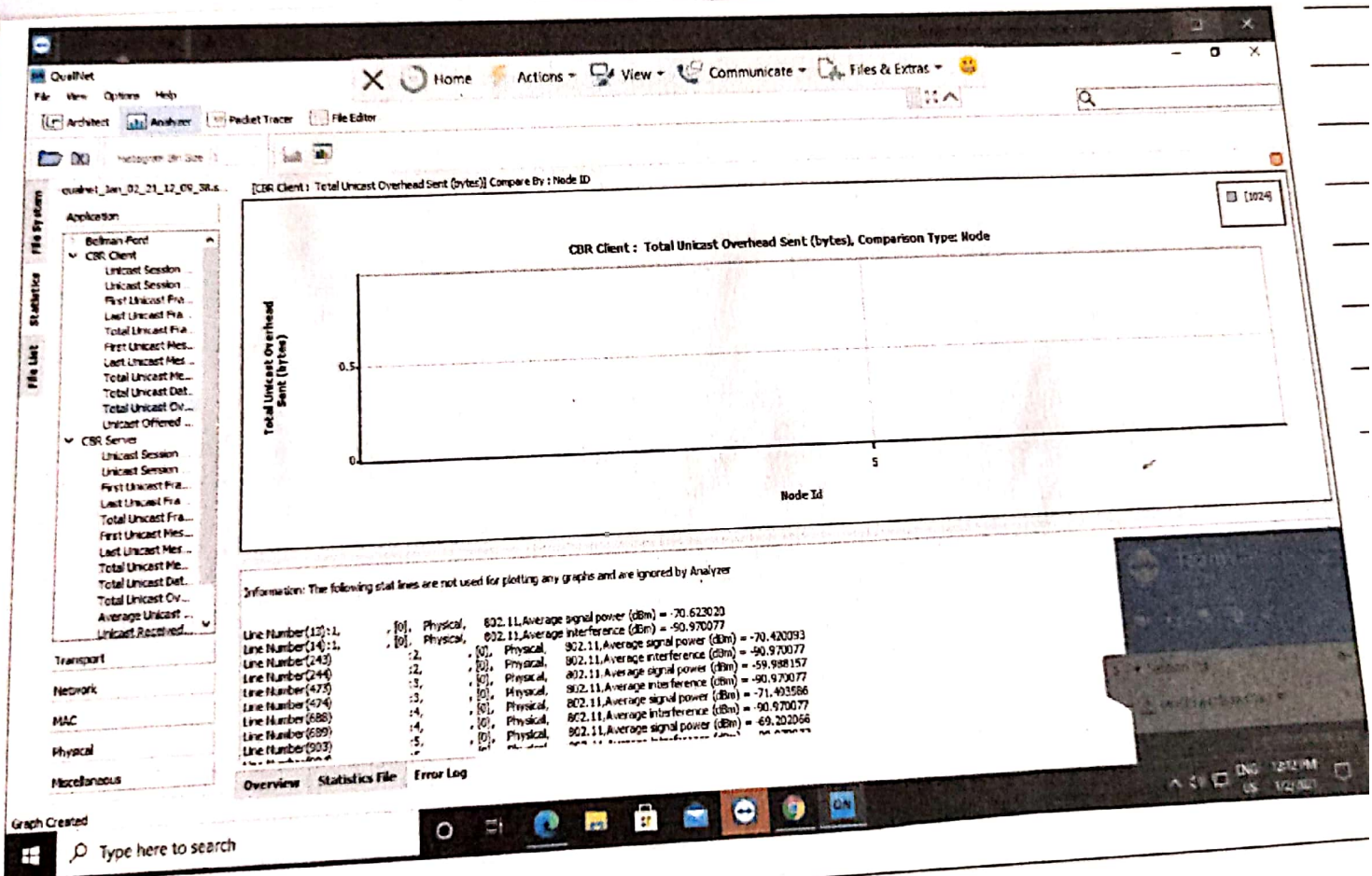


Teacher's Signature _____

(Q-4a) for every node, the stat dists but every update the packets receive through time.

(Q - 4-b) : As discussed in one of the previous questions, one may or may not have dsrx. This stat, and also def it called total upcast, Interference blw,

It helps us identify the upcast overhead.



Teacher's Signature