GOA COLLEGE OF ENGINEERING

ASSIGNMENT NO: 1 **DATE:**08/05/2021

SIMULATE THE SLIDING WINDOW PROTOCOL

(I) GO BACK N

```
# include <iostream>
# include <conio.h>
# include <stdlib.h>
# include <time.h>
# include <math.h>
using namespace std;

# define TOT_FRAMES 500
# define FRAMES_SEND 10

class gobkn
{
 private:
  int fr_send_at_instance;
  int arr[TOT_FRAMES];
  int arr1[FRAMES_SEND];
  int sw;
  int rw; // tells expected frame
 public:
  gobkn();
  void input();
  void sender(int);
  void reciever(int);
};

//constructor
gobkn :: gobkn()
{
 sw = 0;
 rw = 0;
}

void gobkn :: input()
{
 int n;  // no of bits for the frame
 int m;  // no of frames from n bits

 cout << "Enter the no of bits for the sequence no ";
 cin >> n;

 m = pow (2 , n);
```

```
 int t = 0;

 fr_send_at_instance = (m / 2);

 for (int i = 0 ; i < TOT_FRAMES ; i++)
 {
  arr[i] = t;
  t = (t + 1) % m;
 }

 sender(m);
}

void gobkn :: sender(int m)
{
 int j = 0;

 for (int i = sw ; i < sw + fr_send_at_instance ; i++)
 {
  arr1[j] = arr[i];
  j++;
 }

 for (int i = 0 ; i < j ; i++)
  cout << " SENDER   : Frame " << arr1[i] << " is sent\n";

 reciever (m);
}

void gobkn :: reciever(int m)
{
 time_t t;
 int f;
 int f1;
 int a1;
 char ch;

 srand((unsigned) time(&t));
 f = rand() % 10;

   // if = 5 frame is discarded for some reason
   // else they are correctly recieved

 if (f != 5)
 {
  for (int i = 0 ; i < fr_send_at_instance ; i++)
  {
   if (rw == arr1[i])
```

```
 {
  cout << "RECIEVER : Frame " << arr1[i] << " recieved correctly\n";
  rw = (rw + 1) % m;
 }
 else
  cout << "RECIEVER : Duplicate frame " << arr1[i] << " discarded\n";
}
a1 = rand() % 15;

 // if a1 belongs to 0 to 3 then
 //     all ack after this (incl this one) lost
 // else
 //     all recieved

if (a1 >= 0 && a1 <= 3)
{
 cout << "(Acknowledgement " << arr1[a1] << " & all after this lost)\n";
 sw = arr1[a1];
}
else
 sw = (sw + fr_send_at_instance) % m;
}
else
{
f1 = rand() % fr_send_at_instance;

 // f1 gives index of the frame being lost

for (int i = 0 ; i < f1 ; i++)
{
 if (rw == arr1[i])
 {
  cout << " RECIEVER : Frame " << arr1[i] << " recieved correctly\n";
  rw = (rw + 1) % m;
 }
 else
  cout << " RECIEVER : Duplicate frame " << arr1[i] << " discarded\n";
}

int ld = rand() % 2;
  // ld == 0 frame damaged
  // else frame lost
if (ld == 0)
 cout << " RECIEVER : Frame " << arr1[f1] << " damaged\n";
else
 cout << "           (Frame " << arr1[f1] << " lost)\n";

for (int i = f1 + 1 ; i < fr_send_at_instance ; i++)
```

```
    cout << " RECIEVER : Frame " << arr1[i] << " discarded\n";

  cout << " (SENDER TIMEOUTS --> RESEND THE FRAME)\n";

  sw = arr1[f1];
 }
 cout << "Want to continue...";
 cin >> ch;

 if (ch == 'y')
  sender(m);
 else
  exit(0);
}

int main()
{
 gobkn gb;
 gb.input();
 getch();
}
```

(II) SELECTIVE REPEAT

```
// SIMULATE SELECTIVE REPEAT PROTOCOL
# include <iostream>
# include <conio.h>
# include <stdlib.h>
# include <time.h>
# include <math.h>
using namespace std;

# define TOT_FRAMES 500
# define FRAMES_SEND 10

class sel_repeat
{
 private:
  int fr_send_at_instance;
  int arr[TOT_FRAMES];
  int send[FRAMES_SEND];
  int rcvd[FRAMES_SEND];
  char rcvd_ack[FRAMES_SEND];
  int sw;
  int rw; // tells expected frame
 public:
  void input();
```

```
  void sender(int);
  void reciever(int);
};

void sel_repeat :: input()
{
 int n;  // no of bits for the frame
 int m;  // no of frames from n bits

 cout << "Enter the no of bits for the sequence number ";
 cin >> n;

 m = pow (2 , n);

 int t = 0;

 fr_send_at_instance = (m / 2);

 for (int i = 0 ; i < TOT_FRAMES ; i++)
 {
  arr[i] = t;
  t = (t + 1) % m;
 }

 for (int i = 0 ; i < fr_send_at_instance ; i++)
 {
  send[i] = arr[i];
  rcvd[i] = arr[i];
  rcvd_ack[i] = 'n';
 }

 rw = sw = fr_send_at_instance;

 sender(m);
}

void sel_repeat :: sender(int m)
{
 for (int i = 0 ; i < fr_send_at_instance ; i++)
 {
  if ( rcvd_ack[i] == 'n' )
    cout << " SENDER   : Frame " << send[i] << " is sent\n";
 }
 reciever (m);
}

void sel_repeat :: reciever(int m)
{
```

```
time_t t;
int f;
int f1;
int a1;
char ch;

srand((unsigned) time(&t));

for (int i = 0 ; i < fr_send_at_instance ; i++)
{
 if (rcvd_ack[i] == 'n')
 {
  f = rand() % 10;

  // if = 5 frame is discarded for some reason
  // else frame is correctly recieved

  if (f != 5)
  {
      int j;
   for ( j = 0 ; j < fr_send_at_instance ; j++)

    if (rcvd[j] == send[i])
    {
     cout << "RECIEVER : Frame " << rcvd[j] << " recieved correctly\n";
     rcvd[j] = arr[rw];
     rw = (rw + 1) % m;
     break;
    }

   if ( j == fr_send_at_instance)
    cout << "RECIEVER : Duplicate frame " << send[i] << " discarded\n";

   a1 = rand() % 5;

    // if a1 == 3 then ack is lost
    //           else recieved

       if (a1 == 3)
       {
          cout << "(Acknowledgement " << send[i] << " lost)\n";
          cout << " (SENDER TIMEOUTS --> RESEND THE FRAME)\n";
          rcvd_ack[i] = 'n';
       }
       else
       {
          cout << "(Acknowledgement " << send[i] << " recieved)\n";
          rcvd_ack[i] = 'p';
```

```
      }
   }
   else
   {
    int ld = rand() % 2;

    // if = 0 then frame damaged
    // else frame lost

    if (ld == 0)
    {
     cout << "RECIEVER : Frame " << send[i] << " is damaged\n";
     cout << "RECIEVER : Negative acknowledgement " << send[i]  << " sent\n";
    }
    else
    {
     cout << "RECIEVER : Frame " << send[i] << " is lost\n";
     cout << " (SENDER TIMEOUTS --> RESEND THE FRAME)\n";
    }
    rcvd_ack[i] = 'n';
   }
  }
 }
int j;
 for (  j = 0 ; j < fr_send_at_instance ; j++)
 {
  if (rcvd_ack[j] == 'n')
   break;
 }

 int i = 0 ;

 for (int k = j ; k < fr_send_at_instance ; k++)
 {
  send[i] = send[k];

  if (rcvd_ack[k] == 'n')
   rcvd_ack[i] = 'n';
  else
   rcvd_ack[i] = 'p';

  i++;
 }

 if ( i != fr_send_at_instance )
 {
  for ( int k = i ; k < fr_send_at_instance ; k++)
  {
```

```
   send[k] = arr[sw];
   sw = (sw + 1) % m;
   rcvd_ack[k] = 'n';
  }
 }
 cout << "Want to continue...";
 cin >> ch;
 cout << "\n";

 if (ch == 'y')
  sender(m);
 else
  exit(0);
}

Int main()
{
 sel_repeat sr;
 sr.input();
 getch();
}
```