

COMPUTER NETWORKS ASSIGNMENT-1

NETWORK ALGORITHMS PRACTICAL LIST

1. SimulateCyclicRedundancyCheck(CRC)errordetectionalgorithmfor noisychannel.
2. Simulateandimplementstopandwaitprotocol fornoisychannel.
3. Simulateandimplementgobacknslidingwindowprotocol.
4. Simulateandimplement selectiverepeatslidingwindowprotocol.
5. Shortest Path algorithm.

Simulate Cyclic_Redundancy_Check(CRC) Error detection Algorithm for noisy channel.

HARSH YADAV
20/458

```

C crc1.c ×
C crc1.c > ...
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, m;
6      int i;
7
8      printf("\n\nEnter the size of message: ");
9      scanf("%d", &n);
10
11     int arr[n], temp[10];
12
13     printf("\n\nEnter the message (in binary): ");
14
15     for (i = 0; i < n; i++)
16         scanf("%d", &arr[i]);
17
18     for (i = 0; i < n; i++)
19         temp[i] = arr[i];
20
21     printf("\nMessage entered is: ");
22     for (int i = 0; i < n; i++)
23         printf("%d", arr[i]);
24
25     printf("\n\nEnter the size of generator: ");
26     scanf("%d", &m);
27
28     int arr1[m];
29
30     printf("\n\nEnter the message generator(in binary): ");
31     for (int i = 0; i < m; i++)
32         scanf("%d", &arr1[i]);
33
34     printf("\nMessage after redundant bits is: ");
35     for (int i = n; i < (n + m - 1); i++)
36         arr[i] = 0;
37
38     for (int i = 0; i < (n + m - 1); i++)
39         printf("%d", arr[i]);
40
41     for (int i = 0; i < n; i++)
42

```

```

C crc1.c ×
C crc1.c > main()
40
41     for (int i = 0; i < n; i++)
42     {
43         if (arr1[0] == arr[i])
44         {
45             for (int j = 0, k = i; j < m; j++, k++)
46             {
47                 if (!(arr[k] ^ arr1[j]))
48                     arr[k] = 0;
49                 else
50                     arr[k] = 1;
51             }
52         }
53     }
54
55     printf("\n\nCRC bits at sender's side are: ");
56     for (int i = n; i < (n + m - 1); i++)
57         printf("%d", arr[i]);
58
59     int arr2[n + m];
60     printf("\nEnter the message( with CRC's bits) at reciever's end: ");
61
62     for (int i = 0; i < (n + m - 1); i++)
63         scanf("%d", &arr2[i]);
64
65     printf("\nMessage recieved is: ");
66     for (int i = 0; i < (n + m - 1); i++)
67         printf("%d", arr2[i]);
68
69     for (int i = 0; i < n; i++)
70     {
71         if (arr1[0] == arr2[i])
72         {
73             for (int j = 0, k = i; j < m; j++, k++)
74             {
75                 if (!(arr2[k] ^ arr1[j]))
76                     arr2[k] = 0;
77                 else
78                     arr2[k] = 1;
79             }
80         }
81     }
82
83     printf("\n\nCRC bits at reciever's side are: ");
84     for (int i = n; i < (n + m - 1); i++)
85         printf("%d", arr2[i]);
86
87     return 0;
88 } // main()

```

HARSH YADAV
20/458

OUTPUT

Enter the size of message: 10

Enter the message (in binary): 1

1
0
1
0
1
1
1
1
1
1

Message entered is: 1101011111

Enter the size of generator: 5

Message entered is: 1101011111

Enter the size of generator: 5

CRC bits at sender's side are: 0010

Enter the message(with CRC's bits) at reciever's end: 1

1
0
1
0
1
1
1
1
1
0
0
1
0

Message recieved is: 1101011110010

CRC bits at reciever's side are: 0000

HARSH YADAV
20/458

Simulate and Implement Stop_and_Wait Protocol for Noisy channel.

HARSH YADAV
20/458

C crc1.c

G+ stopandwait.cpp X



G+ stopandwait.cpp > ...

```

1  #include <iostream>
2  #include <time.h>
3  #include <cstdlib>
4  #include <ctime>
5  #include <unistd.h>
6  #include <iomanip>
7  using namespace std;
8  class timer
9  {
10 private:
11     unsigned long begTime;
12
13 public:
14     void start()
15     {
16         begTime = clock();
17     }
18     unsigned long elapsedTime()
19     {
20         return ((unsigned long)clock() - begTime) / CLOCKS_PER_SEC;
21     }
22     bool isTimeout(unsigned long seconds)
23     {
24         return seconds >= elapsedTime();
25     }
26 };
27 int main()
28 {
29     int frames[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
30     unsigned long seconds = 5;
31     srand(time(NULL));
32     timer t;
33     cout << "Sender has to send frames : ";
34     for (int i = 0; i < 10; i++)
35         cout << frames[i] << " ";
36     cout << endl;
37     int count = 0;
38     bool delay = false;
39     cout << endl;
40     cout << "Sender\t\t\t\t\tReceiver" << endl;
41     do
42     {
43         bool timeout = false;
44         cout << "Sending Frame : " << frames[count];
45         cout.flush();
46         cout << "\t\t";
47         t.start();
48         if (rand() % 2)
49         {
50             int to = 24600 + rand() % (64000 - 24600) + 1;
51             for (int i = 0; i < 64000; i++)

```

C crc1.c

G+ stopandwait.cpp X



G+ stopandwait.cpp > ...

```

42     } while (count != 10);
43     return 0;
44
45     cout << "Sending Frame : " << frames[count];
46     cout.flush();
47     cout << "\t\t";
48     t.start();
49     if (rand() % 2)
50     {
51         int to = 24600 + rand() % (64000 - 24600) + 1;
52         for (int i = 0; i < 64000; i++)
53             for (int j = 0; j < to; j++)
54             {
55             }
56     }
57     if (t.elapsedTime() <= seconds)
58     {
59         cout << "Received Frame : " << frames[count] << " ";
60         if (delay)
61         {
62             cout << "Duplicate";
63             delay = false;
64         }
65         cout << endl;
66         count++;
67     }
68     else
69     {
70         cout << "---" << endl;
71         cout << "Timeout" << endl;
72         timeout = true;
73     }
74     t.start();
75     if (rand() % 2 || !timeout)
76     {
77         int to = 24600 + rand() % (64000 - 24600) + 1;
78         for (int i = 0; i < 64000; i++)
79             for (int j = 0; j < to; j++)
80             {
81             }
82         if (t.elapsedTime() > seconds)
83         {
84             cout << "Delayed Ack" << endl;
85             count--;
86             delay = true;
87         }
88         else if (!timeout)
89             cout << "Acknowledgement : " << frames[count] - 1 << endl;
90     }
91 } while (count != 10);
92 return 0;

```

HARSH YADAV
20/458

OUTPUT

Sender has to send frames : 1 2 3 4 5 6 7 8 9 10

Sender	Receiver
Sending Frame : 1	Received Frame : 1
Acknowledgement : 1	
Sending Frame : 2	Received Frame : 2
Delayed Ack	
Sending Frame : 2	Received Frame : 2 Duplicate
Acknowledgement : 2	
Sending Frame : 3	Received Frame : 3
Delayed Ack	
Sending Frame : 3	Received Frame : 3 Duplicate
Acknowledgement : 3	
Sending Frame : 4	Received Frame : 4
Acknowledgement : 4	
Sending Frame : 5	Received Frame : 5
Delayed Ack	
Sending Frame : 5	Received Frame : 5 Duplicate
Acknowledgement : 5	
Sending Frame : 6	---
Timeout	
Sending Frame : 6	Received Frame : 6
Acknowledgement : 6	
Sending Frame : 7	Received Frame : 7
Delayed Ack	
Sending Frame : 7	Received Frame : 7 Duplicate
Delayed Ack	


```

Delayed Ack
Sending Frame : 7          Received Frame : 7 Duplicate
Delayed Ack
Sending Frame : 7          Received Frame : 7 Duplicate
Delayed Ack
Sending Frame : 7          ---
Timeout
Delayed Ack
Sending Frame : 6          ---
Timeout
Sending Frame : 6          ---
Timeout
Delayed Ack
Sending Frame : 5          Received Frame : 5 Duplicate
Acknowledgement : 5
7
Acknowledgement : 7
Sending Frame : 8          ---
7
Acknowledgement : 7
Sending Frame : 8          ---
Timeout
Sending Frame : 8          ---
Timeout
Sending Frame : 8          ---
Timeout
Sending Frame : 8          Received Frame : 8

```

```

Sending Frame : 8          Received Frame : 8
Acknowledgement : 8
Sending Frame : 9          Received Frame : 9
Acknowledgement : 9
Sending Frame : 10         Received Frame : 10
Delayed Ack
Sending Frame : 10         Received Frame : 10 Duplicate
Delayed Ack
Sending Frame : 10         Received Frame : 10 Duplicate
Delayed Ack
Sending Frame : 10         Received Frame : 10 Duplicate
Acknowledgement : 33268

```


Simulate and Implement Go_Back_and_Sliding Window Protocol for Noisy channel.

HARSH YADAV
20/458

stopandwait.cpp

gobackandslide.cpp



gobackandslide.cpp > main()

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<time.h>
4
5  int main()
6  {
7      int nf,N;
8      int no_tr=0;
9
10     srand(time(NULL));
11
12     printf("Enter no. of frames: ");    scanf("%d",&nf);
13     printf("\nEnter the Window size: ");    scanf("%d",&N);
14     printf("\n\n");
15
16     int i=1, j;
17     while(i<=nf)
18     {
19         int x=0;
20         for(j=i;j<i+N && j<=nf;j++,no_tr++)    printf("Sent frame %d ",j);
21
22         for(j=i;j<i+N && j<=nf;j++)
23         {
24             int flag = rand() % 2;
25
26             if(!flag)
27             {
28                 printf("Acknowledgment for Frame %d\n",j);
29                 x++;
30             }
31             else
32             {
33                 printf("Frame %d NOT Received\n", j);
34                 printf("Retransmitting Window\n");
35                 break;
36             }
37         }
38         printf("\n");
39         i += x;
40     }//while
41
42     printf("Total number of transmissions : %d\n",no_tr);
43
44     return 0;

```



HARSH YADAV
20/458

OUTPUT

Enter no. of frames: 5

Enter the Window size: 4

Sent frame 1 Sent frame 2 Sent frame 3 Sent frame 4 Frame 1 NOT Received
Retransmitting Window

Sent frame 1 Sent frame 2 Sent frame 3 Sent frame 4 Frame 1 NOT Received
Retransmitting Window

Sent frame 1 Sent frame 2 Sent frame 3 Sent frame 4 Frame 1 NOT Received
Retransmitting Window

Sent frame 1 Sent frame 2 Sent frame 3 Sent frame 4 Acknowledgment for Frame 1
Frame 2 NOT Received
Retransmitting Window

Sent frame 2 Sent frame 3 Sent frame 4 Sent frame 5 Frame 2 NOT Received
Retransmitting Window

Sent frame 2 Sent frame 3 Sent frame 4 Sent frame 5 Acknowledgment for Frame 2
Frame 3 NOT Received
Retransmitting Window

Sent frame 3 Sent frame 4 Sent frame 5 Acknowledgment for Frame 3
Acknowledgment for Frame 4
Frame 5 NOT Received
Retransmitting Window

Sent frame 5 Acknowledgment for Frame 5

Total number of transmissions : 28

Simulate and Implement Selective_Repeat Sliding Window Protocol for Noisy channel.

HARSH YADAV
20/458

selectiverepeat.cpp > sel_repeat

```
1  /*
2   | Selective repeat sliding window protocol
3   */
4
5  #include<iostream>
6  #include<cstdlib>
7  #include<ctime>
8  #include<cmath>
9  using namespace std;
10
11  #define TOT_FRAMES 500
12  #define FRAMES_SEND 10
13
14  class sel_repeat
15  {
16  private:
17      int fr_send_at_instance;
18      int arr[TOT_FRAMES];
19      int send[FRAMES_SEND];
20      int rcvd[FRAMES_SEND];
21      char rcvd_ack[FRAMES_SEND];
22      int sw;
23      int rw; //tells expected frame
24
25  public:
26
27      void input();
28      void sender(int);
29      void receiver(int);
30
31  };
32
33  void sel_repeat::input()
34  {
35
36      int n; //no. of bits for the frame
37      int m; //no. of frames from n bits
38      int i;
39      cout<<"\nEnter the no. of bits for the sequence no. : ";
40      cin>>n;
41      m=pow(2,n);
42
43      int t=0;
44
```

selectiverepeat.cpp > sel_repeat

```
45      fr_send_at_instance=(m/2);
46
47      for(i=0;i<TOT_FRAMES;i++)
48      {
49
50
51          arr[i]=t;
52
53          t=(t+1)%m;
54
55      }
56
57      for(i=0;i<fr_send_at_instance;i++)
58      {
59
60
61          send[i]=arr[i];
62
63          rcvd[i]=arr[i];
64
65          rcvd_ack[i]='\n';
66
67      }
68
69      rw=sw=fr_send_at_instance;
70
71      sender(m);
72
73      }
74
75      void sel_repeat::sender(int m)
76      {
77
78
79          for(int i=0;i<fr_send_at_instance;i++)
80
81      {
82
83          if(rcvd_ack[i]=='\n')
84
85          cout<<"\nSENDER : Frame "<<send[i]<<" is sent\n";
86
87      }
88
```

selectiverepeat.cpp > sel_repeat

```
89      receiver(m);
90
91      }
92
93      void sel_repeat::receiver(int m)
94      {
95
96          time_t t;
97
98          int f;
99          int j;
100          int f1;
101
102          int a1;
103
104          char ch;
105
106          srand((unsigned)time(&t));
107
108          for(int i=0;i<fr_send_at_instance;i++)
109          {
110
111              if(rcvd_ack[i]=='\n')
112              {
113
114                  f=rand()%10;
115
116                  //if f=5 frame is discarded for some reason
117
118                  //else frame is correctly recieved
119
120                  if(f!=5)
121                  {
122
123                      for(int j=0;j<fr_send_at_instance;j++)
124
125                      if(rcvd[j]==send[i])
126
127                      {
128
129
```

HARSH YADAV
20/458

```

selectiverepeat.cpp > sel_repeat
133 cout<<"\nreciever:Frame "<<rcvd[j]<<" recieved correctly\n";
134
135 rcvd[j]=arr[rw];
136
137 rw=(rw+1)%m;
138
139 break;
140
141 }
142 int j;
143 if(j==fr_send_at_instance)
144
145 cout<<"\nreciever:Duplicate frame "<<send[i]<<" discarded\n";
146
147 a1=rand()%5;
148
149 //if a1==3 then ack is lost
150
151 //else recieved
152
153 if(a1==3)
154
155 {
156
157 cout<<"\n(acknowledgement "<<send[i]<<" lost)\n";
158
159 cout<<"\n(sender timeouts-->Resend the frame)\n";
160
161 rcvd_ack[i]='n';
162
163 }
164
165 else
166
167 {
168
169 cout<<"(acknowledgement "<<send[i]<<" recieved)\n";
170
171 rcvd_ack[i]='p';
172
173 }
174
175 }

```

```

selectiverepeat.cpp > sel_repeat
else
{int ld=rand()%2;

//if ==0 then frame damaged

//else frame lost

if(ld==0)
{
cout<<"\nRECEIVER : Frame "<<send[i]<<" is damaged\n";
cout<<"\nRECEIVER : Negative Acknowledgement "<<send[i]<<" sent\n";
}
else
{
cout<<"\nRECEIVER : Frame "<<send[i]<<" is lost\n";
cout<<"\n (SENDER TIMEOUTS-->RESEND THE FRAME)\n";
}
}
rcvd_ack[i]='n';
}
}

for(int j=0;j<fr_send_at_instance;j++)
{
if(rcvd_ack[j]=='n')
break;

```

```

selectiverepeat.cpp > sel_repeat
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<<"\nWant to continue('Y' or 'y' to continue): ";
cin>>ch;
cout<<"\n";
if(ch=='y')
| sender(m);
else
| exit(0);
}

int main()
{
sel_repeat sr;;
sr.input();

return 0;
}

```

HARSH YADAV
20/458

OUTPUT

```
Enter the no. of bits for the sequence no. : 3
SENDER : Frame 0 is sent
SENDER : Frame 1 is sent
SENDER : Frame 2 is sent
SENDER : Frame 3 is sent
reciever:Frame 0recieved correctly
(acknowledgement 0 recieved)
reciever:Frame 1recieved correctly
(acknowledgement 1 recieved)
RECEIVER : Frame 2 is lost
(SENDER TIMEOUTS-->RESEND THE FRAME)
RECEIVER : Frame 3 is lost
(SENDER TIMEOUTS-->RESEND THE FRAME)
Want to continue('Y' or 'y' to continue): y

SENDER : Frame 4 is sent
SENDER : Frame 5 is sent
SENDER : Frame 6 is sent
SENDER : Frame 7 is sent
```

```
reciever:Frame 4recieved correctly
(acknowledgement 4 recieved)
reciever:Frame 5recieved correctly
(acknowledgement 5 recieved)
reciever:Frame 6recieved correctly
(acknowledgement 6 recieved)
reciever:Frame 7recieved correctly
(acknowledgement 7 recieved)
Want to continue('Y' or 'y' to continue): y

SENDER : Frame 0 is sent
SENDER : Frame 1 is sent
SENDER : Frame 2 is sent
SENDER : Frame 3 is sent
reciever:Frame 0recieved correctly
(acknowledgement 0 recieved)
RECEIVER : Frame 1 is damaged
RECEIVER : Negative Acknowledgement 1 sent
reciever:Frame 2recieved correctly
(acknowledgement 2 recieved)
reciever:Frame 3recieved correctly
```



```
reciever:Frame 2recieved correctly  
(acknowledgement 2 recieved)  
  
reciever:Frame 3recieved correctly  
(acknowledgement 3 lost)  
  
(sender timeouts-->Resend the frame)  
  
Want to continue('Y' or 'y' to continue): y  
  
  
SENDER : Frame 4 is sent  
  
SENDER : Frame 5 is sent  
  
SENDER : Frame 6 is sent  
  
SENDER : Frame 7 is sent  
  
reciever:Frame 4recieved correctly  
(acknowledgement 4 recieved)  
  
reciever:Frame 5recieved correctly  
(acknowledgement 5 recieved)  
  
reciever:Frame 6recieved correctly  
(acknowledgement 6 recieved)  
  
reciever:Frame 7recieved correctly  
(acknowledgement 7 recieved)  
  
Want to continue('Y' or 'y' to continue): n
```

THANK YOU

- HARSH YADAV

(20/458)