

Computer Network

Practicals

Ramanujan College, Delhi University

Name - Ujjawal kumar

College Roll no – 20201441

University roll no – 20020570038

Course – BSc computer science (Hons)

SUBMITTED TO – Mr. Nikhil Rajput \Mr. Arun Sir

Practical1:- Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel. **Output:**

```
← TAB _ ⋮

Enter the frame size
5

Enter frame(0's and 1's only)
10101
10110
11110
01110
11001

Enter the generator size(atleast 1 less than frame)
2

Enter generator(0's and 1's only)
10001
11100

Sender side
Frame (message): 101011011011110111011001

Generator: 100011100

0's appended in frame: 1
Frame after appending 0's: 1010110110111101110110010
CRC bits: 1
Transmitted: 1010110110111101110110011

Reciever side
Reciever frame: 1010110110111101110110011
Remainder: 1
Successful transmission of message

[Program finished]
```

Practical 2:-Simulate and implement stop and wait protocol for noisy channel.

Output:

```
← TAB _ ⋮

sender information      1
sequence no.          0
Receiver : packet received      1
Acknowledgement sent
sender information      2
sequence no.          1
Receiver : packet received      2
Acknowledgement sent
sender information      3
sequence no.          0
Receiver : packet received      3
Acknowledgement sent
sender information      4
sequence no.          1
Receiver : packet received      4
Acknowledgement sent
sender information      5
sequence no.          0
Receiver : packet received      5
Acknowledgement sent
discontinue

[Program finished]
```

Practical 3:-Simulate and implement go back n sliding window protocol.

Output:

```
← TAB _ ⋮
enter the value of m i.e no of bits allowed for sequence no. :-->
3
size of sliding window =7
enter the total no of frames you want tosend
3

1 --> go to sender
0 --> exit
enter code :--> 1

frames :- 0 1 2
sn = 0
sf = 0

1 --> send frame
2 --> send frame (lost or incorrect)
3 --> go back
:--> 1

frame F0 sent...
frames :- 0 1 2
sn = 1
sf = 0

1 --> send frame
2 --> send frame (lost or incorrect)
3 --> go back
:--> 2

frame F1 sent...
frames :- 0 1 2
sn = 2
sf = 0

1 --> send frame
2 --> send frame (lost or incorrect)
3 --> go back
:--> 1

frame F2 sent...
frames :- 0 1 2
sn = 3
sf = 0

1 --> send frame
2 --> send frame (lost or incorrect)
3 --> go back
:--> 1

**all frames sent**
frames :- 0 1 2
sn = 3
sf = 0
```

```
**all frames sent**

frames :- 0 1 2

sn = 3
sf = 0

1 --> send frame
2 --> send frame (lost or incorrect)
3 --> go back
:--> 3

1 --> go to sender
2 --> go to reciever
0 --> exit
enter code :--> 2

frames recieved :- 0
rn = 1

1 --> send acknowledgement
2 --> go back
:--> 1

acknowledgement A1 sent

frames recieved :- 0
rn = 1

1 --> send acknowledgement
2 --> go back
:--> 2

1 --> go to sender
2 --> go to reciever
0 --> exit
enter code :--> 1

frames :- 0 1 2

sn = 3
sf = 1

1 --> send frame
2 --> send frame (lost or incorrect)
3 --> go back
:--> 3

1 --> go to sender
2 --> go to reciever
0 --> exit
enter code :--> 0

[Program finished]
```

Practical 4:-Simulate and implement selective repeat sliding window protocol.

Output:

```
← TAB _ ⋮
Enter the no. of bits for the sequence no.(max :4) : 3
SENDER : Frame 0 is sent
SENDER : Frame 1 is sent
SENDER : Frame 2 is sent
SENDER : Frame 3 is sent
reciever:Frame0recieved correctly
(acknowledgement 0 recieved)
reciever:Frame1recieved correctly
(acknowledgement 1 lost)
(sender timeouts-->Resend the frame)
RECEIVER : Frame 2 is damaged
RECEIVER : Negative Acknowledgement 2 sent
RECEIVER : Frame 3 is lost
(SENDER TIMEOUTS-->RESEND THE FRAME)
Want to continue:: y-yes
y

SENDER : Frame 4 is sent
SENDER : Frame 5 is sent
SENDER : Frame 6 is sent
SENDER : Frame 7 is sent
reciever:Frame4recieved correctly
(acknowledgement 4 lost)
(sender timeouts-->Resend the frame)
reciever:Frame5recieved correctly
(acknowledgement 5 recieved)
reciever:Frame6recieved correctly
(acknowledgement 6 recieved)
reciever:Frame7recieved correctly
(acknowledgement 7 recieved)
Want to continue:: y-yes
y

SENDER : Frame 0 is sent
SENDER : Frame 1 is sent
SENDER : Frame 2 is sent
SENDER : Frame 3 is sent
reciever:Frame0recieved correctly
```

```
reciever:Frame0recieved correctly
(acknowledgement 0 recieved)
reciever:Frame1recieved correctly
(acknowledgement 1 recieved)
reciever:Frame2recieved correctly
(acknowledgement 2 recieved)
RECEIVER : Frame 3 is lost
(SENDER TIMEOUTS-->RESEND THE FRAME)
Want to continue:: y-yes
y

SENDER : Frame 4 is sent
SENDER : Frame 5 is sent
SENDER : Frame 6 is sent
SENDER : Frame 7 is sent
reciever:Frame4recieved correctly
(acknowledgement 4 recieved)
reciever:Frame5recieved correctly
(acknowledgement 5 recieved)
reciever:Frame6recieved correctly
(acknowledgement 6 recieved)
reciever:Frame7recieved correctly
(acknowledgement 7 recieved)
Want to continue:: y-yes
n

[Program finished]
```

Practical 5:-Simulate and implement distance vector routing algorithm

Output:

```
← TAB _ ⋮
Enter no of nodes.
3
Enter the distance matrix:
1
3
5
7
2
9
6
9
12
Router info for router: 1
Dest    Next Hop    Dist
1       1           0
2       2           3
3       3           5
Router info for router: 2
Dest    Next Hop    Dist
1       1           7
2       2           0
3       3           9
Router info for router: 3
Dest    Next Hop    Dist
1       1           6
2       2           9
3       3           0
[Program finished]
```


Practical 6:-Simulate and implement Dijkstra algorithm for shortest path routing.

Output:

```
← TAB _ ⋮
Vertex      Distance from Source
0           0
1           4
2          12
3          19
4          21
5          11
6           9
7           8
8          14
[Program finished]
```