Ex: 7
18/9/25

Practical -7

Write a program to implement flow control at data link layer using Sliding Window Protocol. Simulate the flow of frame from one node to another

AIM:-

Write a program to implement flow control at data link layer using Sliding Window Protocol. Simulate flow of frames from one node to another.

Program achive at least below given requirements. You can make it a bidirectional program where in receiver is sending its data frame with acknowledgem.

Create a sender program with follow feature.

1. Input Window size from User.
2. Input a Text message from the user.
3. Consider 1 character per frame.
4. Create a frame with Data
5. Send the frames.
6. Wait for acknowledgement from the Receiver.
7. Reader a file called Receiver_Buffer.
8. Check Ack field Acknowledge number.

4. If Acknowledgement number is as expected, sent new set of frames accordingly, Else if NACK is received, resend the frame accordingly.

create a receiver file with features.

1. Reads a file called Sender_Buffer.

2. check the frame no.

3. If frame no. are expected, write appropriate Ack no receiver_Buffer file, Else write NACK no. Receiver_Buffer file.

Student observation

Program:

```
from re import *
import time
import os
os system ('clear')
SB = open ("sender_Buffer.txt", "at")
RB = open ("Receiver_Buffer.txt", "rt")
SB . truncate(0)
RB . truncate(0)
ws = int (input ("Enter window size: "))
 S = input ("Enter input str: ")
  S = list(S)
 if (ws < len (S)):
       for i in range (0, len(S), ws):
```

```
P= S[i: i+ws]
y = S[i+ws: i+ws+ws]
Print (" sent -> " + str (p))
time. sleep (ws)
Print (" sending -> ", str (y))
 x = 0
while (x < ws):
        time.sleep (2)
        if (len (p) > x):
          Print ("Ack ~! ", P[x], "! ")
             RB. write (P[x])
         time . sleep (1)
        if (len (y) > x):
                Print (" sending -> ", y [x])
                SB. write (y[x])
          x += 1
        else:
          Print ("~> window size is too large ")
```

Sample input Output

```
Enter window size : 3
Enter input string : HELLO WORLD
sent -> ['H', 'E', 'L']
sending -> ['L', 'O', 'W']
Ack ~! H!        sending -> L
Ack ~! E!        sending -> O
Ack ~! L!        sending -> W
sent -> ['O', 'R', 'L']
sending -> ['D']
Ack ->! O!
Ack -> ! R!
Ack -> ! L!
sending -> D
```

7

Result:

hence the required data is sent using sliding window