Exp:No:7
19/9/25

Sliding Window Protocol.

Aim:

Write a program to implement flow control at data link layer using sliding window protocol simulate the flow of frames from one mode to another.

Sender program:

1) Input window size and text message.
2) Create frames [Frame no, DATA]
3) Print + save frames in sender-Buffer
4) Wait (delay) then read Receiver-Buffer
5) If ACK = expected → send next frames(overwrite sender_Buffer)
6) If NACK = resend old frames (overwrite sender_Buffer)

Receiver Program:

1) Read Sender-Buffer
2) Check frame numbers
3) If correct → write ACK in Receiver-Buffer
4) If incorrect → write NACK in Receiver-Buffer.

Program:

```
from re import *
import time
~~import os~~
~~os.system('clear')~~
SB= open("sender-Buffer.txt", "at")
RB = open("Receiver-Buffertxt", "rt")
SB.truncate(0)
RB.truncate(0)
WS = int(input("Enter window size:"))
```

```
s =input("Enter input string:")
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("~> The 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!
    ACK ~!E!
```

```
ACk~!L!
Sending -> L
Sending -> O
Sending -> W
Set -> ['o', 'R'', 'L']
Sending -> ['D']
ACk ~!O!
Ack ~!R!
ACk ~!L!
Sending -> D
```

Result!

Hence the required data is sent using sliding window