

Practical - 7

Aim:

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

Sender program:

- 1) Input window size and text message.
- 2) Create frames [Frames no, DATA]
- 3) Print + save frames in sender-Buffer
- 4) Wait(delay) then read Receiver-Buffer
- 5) If ACK = expected \rightarrow 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 \rightarrow write ACK in Receiver-Buffer
- 4) If incorrect \rightarrow 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-Buffer.txt", "rt")
SB.truncate(0)
RB.truncate(0)

WS = int(input("Enter window size: "))
S = input("Enter input string: ")
S = list(S)
    
```

(31)

```

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], "!")
                R.B.write(p[x])
            time.sleep(1)
            if (len(y) > x):
                print("Sending → ", y[x])
                g.B.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
sent → ['O', 'R', 'L']
sending → [D]
ACK ~!, O!
ACK ~!, R!
ACK ~!, L!
Sending → D

```

Result:

Hence the required data is sent using sliding window.