

AIM:

Program to implement flow control at data link layer using SLIDING WINDOW PROTOCOL.
Simulate the flow of frames from one node to another.

Program should achieve atleast below given requirements. You can make it bidirectional program where in receiver is sending its data frames with acknowledgement (piggy backing).

Create a Sender program with following features

- 1) Input window size from the user
- 2) Input a text message from the user.
- 3) Consider 1 character per frame
- 4) Create a frame with following fields.
(Frame no., DATA)
- 5) Send the frames. [Print the output on screen and save it in a file called sender_Buffer]
- 6) Wait for the acknowledgement from the Receiver [Induce delay in the program]
- 7) Reader a file called Receiver_Buffer
- 8) Check Ack field for the acknowledgement number.
- 9) If the acknowledgement number is as expected send new set of frames accordingly. [Over write the sender_Buffer file with new frames]

Else if NACK is received, resend the frames accordingly. [Over write the sender_Buffer with old frame].

- Create a receiver file with following features. 25
- 1) Reader a file called Sender_Buffer
 - 2) check the Frame no.
 - 3) If the Frame no are as expected, write the appropriate Ack no in the Receiver_Buffer file.
Else write Nack no in the Receiver_Buffer file.

NOTE: Induce Error and verify the behaviour of the program. Manually change the frame no and Ack no in the files.

CODE :

```

import time
import random

def simulate():
    time.sleep(random.uniform(0.5, 2))

def Sender():
    window_size = int(input("Enter"))
    message = input(" ")
    Sender_Buffer = [{"frame[i]: {message[i]}": i for i in range(len(message))}]
    expected = 0
    while expected < len(message):
        for i in range(expected + window_size):
            print("Sending: " + str(Sender_Buffer[i]))
            with open("Sender_Buffer.txt", "w") as f:
                f.write(str(Sender_Buffer[i]))
        simulate_delay()
        Receiver_Buffer = []
        with open("Receiver_Buffer.txt", "r") as f:
            Receiver_Buffer = f.read()
        if receiver_Buffer:
            for ack in receiver_Buffer:
                if ack == "NACK":
                    print("NACK received")
                    break
                else:
                    print("ACK received")
                    break
    print("Transmission completed")

```

```

print("")

if ack == startsumtech ("Ack")
    expected_ack = int(ack_splice[i])
else if ack starts with ("NACK")
    print("")

else
    print("")

if --name == "--main--"
    sender()

```

Receiver:

```

import time
import random
def receiver():
    sender_buffer = []
    with open("sender-buffer.mc", "r") as f:
        sender_buffer = f.reading
    expected_frames_no = 0
    for frame in sender_buffer:
        frame_no = int()
        print(f"")
        ack = f + "Ack [frame_no]"
        print(f"")
        expected_frame_no += 1
    else:
        print("Received:")
        main = f + NACK [frame_no]
        print(f"Scoring {nack}")

```

simulate-delay()

with open("receiver-buffer.tr") as f:
 f.write(f"")

if --name == "--main--":
 Receiver()

Output:

Enter window = size : 1

Enter message : Hello

Sending : frame 0+1

Received Ack : 0

Sending : Frame 1 : e

Received : Ack 1

Sending : frame 2 : L

Received : Free . Ack 2

Sending : frame 3 : L

Received : Ack 3

Sending : Frame 4 : o

Received : Ack 4

RESULT :

Thus the Sliding window protocol using python is executed.

10/10

9/10