EXP NO-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.

PROGRAM:-

Sender.py

```
import time
import os
def input_window_size():
    return int(input("Enter window size: "))
def input_text_message():
    return input("Enter text msg: ")
def create_frames(text_message):
    frames = [(i, char) for i, char in enumerate
                                (text_msg)]
    frames.append((len(text_msg), 'END'))
    return frames
def write_to_file(filename, data):
    with open(filename, 'w') as file:
        for frame in data:
            file.write(f"{frame[0]},{frame[1]}\n")
def read_from_file(filename):
    if not os.path.exists(filename):
        return []
    with open(filename, 'r') as file:
        return [line.strip().split(',') for line in file.readlines()]
```

```
def send_frames(frames, window_size):
    i=0
    while i<len(frames):
        window = frames[i:i+window_size]
        print(f"sending frames: {window}")
        write_to_file('Send_Buffer.txt', window)
        time.sleep(3).
        receiver_buffer = read_fromfile('Receiver_Buffer.txt')
        if not receiver_buffer:
            print("NO ack received yet")
            continue.
        ack_frame = receiver_buffer[0]
        ack_number, ack_type = int(ack_frame[0]),
                               ack_frame[1])
        if ack_type == 'ACK':
            print(f'ACK received for frame
{ack_number} sending next set of frames.')
            i += window_size
        elif ack_type == 'NACK':
            print("NACK received for frame
{ack_number} sending frames from frame
{ack_number}.")
            i = ack_number
```

```
def main_sender():
    window_size = input_window_size()
    text_message = input_text_message()
    frames = create_frames(text_message)
    send_frames(frames, window_size)

if __name__ == "__main__":
    main_sender()
```

reciever.py

```
import random
import time
import os

def write_to_file(filename, data):
    with open(filename, 'w') as file:
        file.write(data)

def read_from_file(filename):
    if not os.path.exists(filename):
        return []
    with open(filename, 'r') as file:
        return [line.strip().split(',') for
line in file.readlines()]

def process_frames(frames):
    acks = []
    frame_seen = set()
    for frame in frames:
        frame_number = int(frame[0])
        data = frame[1]
```

```
if frame_number in frame_seen:
    continue
print(f"Received frame {frame_number}. Sending ACK")
if random.choice([True, False]):
    print(f"Sending ACK for frame {frame_number}")
    acks.append(f"{frame_number},NACK|")
    break
return ''.join(acks)

def main_receiver():
    while True:
        time.sleep(3)
        frames = read_from_file('Sender_Buffer.txt')
        if not frames:
            print("No frames to process, waiting...")
            continue
        acks = process_frames(frames)
        write_to_file('Receiver_Buffer.txt', acks)
        if any(frame[1] == 'END' for frame in frames):
            print("End of transmission received.")
            break

if __name__ == "__main__":
    main_receiver()
```

OUTPUT:-

python sender.py

Enter window size: 3
Enter text msg : hello
Sending frame: [(0,'h'), (1,'e'), (2,'l')]
ACK received for frame 0, sending rest of frames
Sending frames: [(3,'l'), (4,'o'), (5,'END')]
ACK received for frame 3, sending rest of frames

python receiver.py

Received frame 0: h
Sending ACK for frame 0
Received frame 1: e
Sending ACK for frame 1
Received frame 2: l
Sending NACK for frame 2
Received frame 3: l
Sending ACK for frame: 3
Received frame 4: o
Sending ACK for frame 4
Received frame 5: END
Sending ACK for frame 5
End of transmission received

RESULT:-

Thus program for sliding window is successfully executed.