EXP NO: 07

SLIDING WINDOW PROTOCOL

DATE: 06-09-24

AIM:

Write a program to implement flow control at data link layer using SLDD 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 message: ")
def create_frames(text_message):
    frames = [(i, char) for i, char in enumerate(text_message)]
    frames.append((len(text_message), '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]
        printf "Sending frames: {window}")
        write_to_file("Sender_Buffer.txt', window))
        time.sleep(3)
        receiver_buffer = read_from_file("Reciever_Buffer.txt")
        if not reciever_buffer:
            print("No acknowledgement recieved yet.")
            continue
        ack_frame = reciever_buffer[0]
        ack_number, ack_type = int(ack_frame[0]), ack_frame[1]
        if ack_type == 'ACK':
            print(f"ACK recieved for frame {ack_number}, sending next set of frames.")
            i += window_size
        elif ack_type == "NACK":
            print(f"NACK recieved for frame {ack_number}, resending 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"Recieved Frame {frame_number}: {data}")
    if random.choice([True, False]):
        print(f"Sending ACK for frame {frame_number}")
        acks.append(f"{frame_number},ACK\n")
        frame_seen.add(frame_number)
    else:
        print(f"Sending NACK for frame {frame_number}")
        acks.append(f"{frame_number},NACK\n")
        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('Reciever_Buffer.txt', acks)
        if any(frame[1] == 'END' for frame in frames):
            print("End of transmission recieved.")
if __name__ == "__main__":
    main_receiver()
```

OUTPUT:

SENDER.PY

```
Enter window size : 2
Enter text message : hi
ACK recieved for frame 1
All frame sent successfully
```

RECIEVER.PY

```
Recieved frames [(0, 'h'), (2, 'i')]
Frames 0 recieved successfully
Frame 2 recieved successfully
```

RESULT:

Thus flow control using sliding window has been successfully implemented & output is verified.