Ex:7.
14/8/24.

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.

PROGRAM:-

```
import threading
import time
import random
import queue

sender_to_receiver_queue = queue.Queue()
receiver_to_sender_queue = queue.Queue()

def sender(window_size, message):
    frames = [message[i:i+1] for i in range
                    (0, len(message))]

    num_frames = len(frames)

    base = 0
    next_seq_num = 0
    while base < num_frames:
        print(f"Window Base: {base}")
        for i in range(base, min(base + window_
            size, num_frames)):
            frame = (i, frames[i])
```

```
sender_to_receiver_queue.put(frame)
print(f"Sending Frame no: {i}, Data: {frames
                                [i]}").

time.sleep(2).
ack_received = set()
while not receiver_to_sender_queue.empty():
    ack = receiver_to_sender_queue.get()
    if ack.startswith('ACK'):
        ack_num = int(ack.split(':')[1])
        ack_received.add(ack_num)
    elif ack.startswith('NACK'):
        nack_num = int(ack.split(':')[1])
        print(f"Received NACK for Frame no:
          {nack_num}. resending frames starting
          from {nack_num}.")
         base = nack_num
         next_seq_num = base
         break.
    else:
        base = next_seq_num + len(ack_received)
        next_seq_num = base.
    if base >= num_frames:
        print("All frames successfully sent and
                    acknowledged.")
        break
```

```
def receiver():
    expected-frame-no = 0
    while True:
        try:
            frame-no, data = sender_to_receiver_queue
                .get(timeout=5)
        except queue.Empty:
            break
        if random.random() >0.1:
            ack = f"ACK: {frame_no}"
            print(f"Received Frame no: {frame_no},
                        sending ACK. ")
            receiver_to_sender_queue.put(ack)
            if frame_no == expected_frame_no:
                expected_frame_no +=1
        else:
            ack = f"NACK: {frame_no}"
            print(f"Received frame no: {frame_no},
                    sending NACK due to error ")
            receiver_to_sender_que.put(ack)
    receiver_to_sender_queue.put("END")
```

```
def main():
    window_size = int(input("Enter window size: "))
    message = input("Enter text message: ")
    receiver_thread = threading.Thread(target
                                       = receiver).
    receiver_thread.start()
    sender(window_size, message).
    receiver_thread.join()
if __name__ == "__main__":
    main().
```

<u>output</u>:

```
python sender.py.
Enter window size : 3.
Enter text message = hello.
Sending frames: [(0,'h'), (1,'e'), (2,'l')].
Ack received for frame 1.
```

```
=> Python receiver.py:
No frames no process, waiting- sending ACK.
end of transaction.
received frame 0 : h
Sending Ack
```

```
received frame 1 : e.
Sending ACK.
received frame 2 : l.
Sending ACK.
received frame 3 : l.
Sending ACK.
receive frame 4 : o.
```

Result :

The program was successfully executed & o/p is verified.

[illegible] 18/9/24