Ex:7
Date: 6/9/2024

Sliding Window Protocol

Aim:

To write a Program to implement Flow control at Data link Layer using Sliding window Protocol. Simulat flow of frame from one node to other

Code: Sender.Py

```
import timei
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.readlin
```

```
def send-framel(framel, window_size):
    i=0
    while i < len(framel):
        window = framel[i:i+ window_size]
        Print (f"Sending framel:{window}")
        write-to-file('Sender-Buffer.txt', window)

        time.sleep(3)

        receiver_buffer = read_from_file('Receiver_Buffer.txt')
        if not receiver-buffer:
            print("No acknowledgement received yet.")
            Catinue

       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 framel.")
            i+=window_size
      elif ack_type=='NACK':
           Print (f"NACK received for frame {ack_number},
             resending framel from frame {ack-number}.")
           i = ack-number

def main-sender():
    window_size = input-window-size()
    text-mellage= input-text-mellage()
    framel = create-framel(text-mellage)
    send-framel(framel, window_size)

if_name_=="_main_":
   main_sender()
```

code receiver.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}:{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('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:

```
Enter window size: 2
Enter text message: hell
Sending frame: [(0,'h'), (1,'e')]
ACK received for frame, sending next frame
Sending frame: [(2,'l'), (3,'l')]
ACK received for frame sending next frame
Sending frame: [(4, 'END')]
ACK received for frame 1,2, sendne next frame
Recieved frame 4: end
Sending NACK for frame 4
End of transmission recieved
```

Result:

that flow control using sliding window has been successfully implemented & O/P is verified