Exp 7

Date : 6-9-24

Sliding window Protocol

AIM

To Write a program to implement Flow ~~error~~ detection and correction using Control at Data link Layer using Sliding window protocol. Simulate flow of frame from one node to other

Code - sender.py

```
import time
import os
def create_frames(text_message):
    frame = [(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(filname, '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('sender_Buffer.text', window)

        time.sleep(3)

        receiver_buffer = read_from_file
                ('Reciever_Buffer.text')

        if not receiver_Buffer:
            print("No acknowledgement
                    recieved 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}, Sender next
                set of frame.")
            i += window_size
        elif ack_type == 'nack':
            print(f"Nack {ack_number}
                resending {ack_number}")
            i = ack_number
```

```
def main-sender():
    window_size = int(input("Enter window size"));
    text_messag = input("Enter text")
    frames = create_frames(text_messag)
    Send frames =(frames, window-size)

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

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_
                number}")
        acks.append(f"{frame_number},
                ACK\n")
        frame_seen.add(frame_number)
    else:
        print(f"Sending NACK {frame_number}
                ")
        ack.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 4, sending next frame

Received frame 4 : END
Sending NACK for frame 4
End of transmission received
```

Result

Thus flow control using sliding window has been successfully implemented & o/p is verified