06/09/24

EXPERIMENT NO : 07

SLIDING WINDOW PROTOCOL (FLOW CONTROL AT DATA LINK LAYER)

Aim:

To write a program to implement flow control at datalink 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):
    frames = [(i, char) for i, char in enumeric(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_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("sending frames: {window}")
        write_to_file('sender_Buffer_text', window)
        time.sleep(3)
        receiver_buffer= read_from_file('Receiver_Buffer_text')
```

```
        if not receiver_Buffer:
            print (" No acknowledgement received yet ")
            continue

        ack_frame = receiver_buffer[0]
        ack_number, ack_type = int(ack_frame[0]) ack_frame[0]

        if ack_type == 'ACK':
            print (f" ACK received for frame {ack_number},
                    sender next set of frame.")
            i += window_size
        else 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_message = input (" Enter text")
    frames = create_frames (text_message)
    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.text')
        if not frames:
            print("No frames to process, waiting")
            continue
        acks = process_frames(frames)
        write_to_file('receiver_buffer.text', acks)
        if (acks(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: bell

Sending frame: [(0,'b'),(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 12, sending next frame

Received frame 4: END

Sending NACK for frame 4

End of transmission received.

Result:

Thus flow control uses sliding window has been successfully implemented and output is verified.

6/9/24