

Page _____ Date _____

Exp: 07 Sliding Window

Aim: To write a program to implement flow control at data link layer using sliding window protocol.

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):
    frame = [(i, char) for i, char in enumerate(text_message)]
    frame.append((len(text_message), 'END'))
    return frame
def write_to_file(filename, data):
    with open(filename, 'w') as file:
        for frame in data:
            file.write(f'{frame[0]}{frame[1]}{frame[2]}\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]
        print(f'send frames: {window}')
        write_to_file('Send-buffer.txt', window)

```

Page
Date

```
time.sleep(3)
receiver-buffer = read-from-file('Receiver-buffer.txt')
if not receiver-buffer:
    send("No Acknowledgement")
    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}, sending next set of frames")
    if window-size == 0:
        if ack-type == "NACK":
            print(f"NACK Received for frame {ack-number}, resending frame from {ack-number}")
            i = ack-number
    def mainSender():
        window_size = input("Window Size")
        text_message = input("Text Message")
        frames = CreateFrames(text_message)
        send_frames(frames, window_size)
        if name == "main":
            mainSender()
```

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_frame(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():

```

Page _____
Date _____

```
while True:  
    time.sleep(3)  
    frame = socket.fromfile('Sender-Buffer.txt')  
    if not frame:  
        print("No frame to process, waiting")  
        continue  
    acks = process_frames(frame)  
    write_to_file('Receiver-Buffer.txt', acks)  
    if any(frame[i] == 'END' for i in range(len(frame))):  
        print("End of transmission Received")  
        break  
    if name == "main":  
        main_recever()
```

Output: Enter Window Size : 3
Enter text message : hii
Sending frame : [0, 'h'], [1, 'i'], [2, 'i']
Ack received from frame 2.

Receiver.py
Received frame 2 : END
Sending ACK for frame 2
End of transmission received

Result:
Thus the program is executed successfully.

6/7/20