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 should achieve at least the done given requirement. You can make it is bidirectional program wherein receiver is sending its data frames with acknowledgement (piggybacking)

Create a sender program with following features:

1) Input window size from the user
2) Input a test message from the user.
3) consider, character perframe.
4) make a frame with following fields: [Frame no, DATA]
5) Send the frame. [Print the output on screen and save it in a file called sender-Buffer].
6) wait for the acknowledgement from the Reciever [Induce delay in the program].
7) reader a file named Reciver-Buffer
8) check ACK field for the Acknowledgement number.
9) If the acknowledgement number is as expected send new set of frames accordingly.

Create a reciever file with following features:

1) Reader a file called sender-Buffer.
2) check the Frame no
3) If the Frame no is as expected, write the appropriate ACK no in the Reciever-Buffer file.

Student observation

1) Sender

```
import time
import os
def sender(window_size, message):
    sender_buffer = "sender_buffer.txt"
    reciver_buffer = "reciever_buffer.txt"
    frame_no = 0
    frames = [[i, message[i]] for i in
              range(len(message))]
    while frame_no < len(frames):
```

```
for i in range (window_size):
    if frame_no + i < len(frames):
print(f"sending frame : {frames
                [frame_no + i]}"):
with open(sender_buffer, 'a') as f:
f.write(f"{frames[frame_no + i]}
                \n")
time.sleep() #simulate delay

while True:
if os.path.exists(reciever_buffer):
with open(reciever_buffer, 'r') as f:
    ack_no = int(f.read(), strip())
    os.remove(reciver_buffer)
    break

if ack_no >= frame_no:
print(f"ack recieved for frame:
                {ack_no}")
    frame_no = ack_no + 1
else
print(f"NACK recieved for frame: {frame
                _no}
                resending...")
```

```
if __name__ == "__main__":
window_size = int(input("Enter window
                size:"))
message = input("Enter message:")
Sender(window_size, message)
```

(iii) Reciever

```
import time
import os
def reciever():
    sender_buffer = "Sender_Buffer.txt"
    reciever_buffer = "Reciever_Buffer.txt"
    expected_frame_no = 0
while True:
if os.path.exists(sender_buffer):
    with open(sender_buffer, 'r') as f:
        lines = f.readlines()
    os.remove(sender_buffer)

for line in lines:
    frame = line.strip().split()
    frame_no = int(frame[0])
    data = frame[1]
if frame_no == expected_frame_no:
    print(f"recieved frame: {frame_no}")
```

```
data = {data}")
with open(reciever_buffer, 'w') as f:
    f.write(str(frame_no))
expected_frame_no += 1
else:
    print(f"unexpected frame: {frame_no}
        expected: {expected_frame_no}")
with open(reciever_buffer, 'w') as f:
    f.write(str(frame_no))
    expected_frame_no += 1
else
    print(f"unexpected frame: {frame_no}
        expected: {expected_frame_no}")
with open(reciever_buffer, 'w') as f:
    f.write(str(expected_frame_no - 1))
if __name__ == "__main__":
    reciever()
```

o/p:

```
Enter window size: 5
Enter message: hello
Sending frame: [0, 'h']
Sending frame: [1, 'e']
Sending frame: [2, 'l']
Sending frame: [3, 'l']
Sending frame: [4, 'o']
NACK recieved for frame: 0, resending....
Sending frame: [0, 'h']
Sending frame: [1, 'e']
Sending frame: [2, 'l']
Sending frame: [3, 'l']
Sending frame: [4, 'o']
NACK recieved for frame: 0, resending....
Sending frame: [0, 'h']
Sending frame: [1, 'e']
Sending frame: [2, 'l']
Sending frame: [3, 'l']
Sending frame: [4, 'o']
```

reciever o/p

```
unexpected frame: 2 | expected: 0
unexpected frame: 2, expected: 0
unexpected frame: 3, expected: 0
~~unexpected~~ recieved frame, data: h
~~unexpected~~
Recieved frame, data: e
Recieved frame: 2, data: 1
Recieved frame: 3, data: l
Recieved frame: 4, data: o
```

Result:

Thus the sliding window for Sender and Reciever file is executed successfully and o/p is verified