

Practical - 07

Exp no: 1
Date: 21/07/24

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.

CREATE A SENDER PROGRAM WITH FOLLOWING FEATURES:

- 1) Input window size from the user.
- 2) Input a text message from the user.
- 3) consider 1 character per frame.
- 4) Create a frame with following fields [frame no., DATA]
- 5) Send the frames. [Print the output on screen and save it in a file]
- 6) Wait for the acknowledgement from the receiver.
- 7) Read a file (receiver.py).
- 8) Check ACK field for the acknowledgement.
- 9) If the acknowledgement number is expected, send new set of frames accordingly.

CREATE A RECEIVER FILE WITH FOLLOWING FEATURES:

- 1) Reader a file called sender Buffer
- 2) Check the frame no.

3) If the frame no. are as expected, write the appropriate ACK no. in the Buffer file.
Else write NACK no. in the Receiver-Buffer file.

CODE: sender.py:

```
import time  
import os  
  
def sender(window_size, message):  
    sender_buffer = "Sender-Buffer.txt";  
    receiver_buffer = "Receiver-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'{frame_no+i}[0] {frames[frame_no+i][1]}\n')
 time.sleep(1)

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

```

if ack_no >= frame_no:
    print("ACK received for frame: {ack-no}")
    frame_no = ack_no + 1
else:
    print("NACK received for frame: {frame-no},\nresending ...")
if name == "main":
    window_size = int(input("Enter window\nsize: "))
    message = input("Enter message: ")
    sender(window_size, message)

```

Receiver.py:

```

import time
import os
def receiver():
    sender_buffer = "Sender-Buffer.txt"
    receiver_buffer = "Receiver-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 ("Received frame: {} frame-no: {}",
          data: {} data)
    with open (receiver-buffer, 'w') as f:
        f.write (str (frame-no))
    expected-frame-no += 1
else:
    print ("Unexpected frame: {} frame-no: {}",
          expected: {} expected-frame-no)
    with open (receiver-buffer, 'w') as f:
        f.write (str (expected-frame-no - 1))

if __name__ == "__main__":
    receiver()

```

OUTPUT:

sender side	Receiver side
Enter window size : 3	
Enter message : hello	
Sending frame : [0, 'h']	
ACK received for frame: 0	
Sending frame: [1, 'e']	
Sending frame: [2, 'l']	
ACK received for frame: 0	
Sending frame: [1, 'e']	
Sending frame: [2, 'l']	
Sending frame: [3, 'l']	
ACK received for frame: 3	

ending frame : [4, 6]

sending frame : [5, 6]

ACK received for frame: 4

Ex: 8a
Date: 10/10/2023

AIM: a)

RESULT:

Thus the program is executed and

the output is verified

8/10
28/10/23