

Exp-No: 7
DATE: 23/8/24

Practical - 07

Aims:

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 the 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
5. Send the frames
6. Wait for the acknowledgement from the receiver
7. Read a file called Receiver_Buffer.
8. Check ACK field for the acknowledgement no.
9. If the acknowledgement no is as expected send next set of frames accordingly. Else Nack is received, resend the frames accordingly.

Create a receiver file with following features:

1. Reader a file called Sender - Buffer.
 2. Check the frame id.
 3. If the frame id. are as expected
the appropriate ACK no. is the Receiver - Buffer file.

Code: (Sender)

Sender .py
import time
import as
def Sender (Cwrdon - Nj, message):
 Sender - buffer = "Sender - Buffer.txt"
 Reciever - buffer = "Reciever - buffer.txt"
 frame - me = 0
 frames = CCI, message [P] for i in range
 (len (message))

while frame - no < len(frames):
for i in range(len(frames) - 1):
 if frame_no + i < len(frames):
 print(f"Sealing frame {frame_no + i} frames[Frame {frame_no + i}]")
 with open(f"Sender-buffer_{i}.txt", "w") as f:
 f.write(f"frames[Frame {frame_no + i}]")
 for j in range(i + 1, len(frames) - 1):
 frames[j] = frames[j + 1]

time.sleep(1)

while true:

if os.path.exists('receiver-buffer'):
with open('receiver-buffer', 'r') as f:
ask_no = int(f.read().strip())
as remove('receiver-buffer')
break

if ack_no >= frame_no:
print(f"ACK received for frame: {ack_no}")
frame_no = ack_no + 1

else:

print(f"NAck received for frames
frame_no")

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

Code 2 (Receiver)

Receiver.py
comfort time
comfort as
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(f"Received frame {frame_no} data:  
        {data}")  
    with open('Receiver-buffer', 'w') as f:  
        f.write(str(frame_no))  
    expected_frame_no += 1
```

else:
 print(f"Unexpected frame {frame_no}
 expected is {expected_frame_no}
 with open('Receiver-buffer', 'w') as f:
 f.write(str(expected_frame_no - 1))

```
if __name__ == '__main__':  
    main()
```

message = input("Enter message to send")

window_size = int(input("Enter the window size"))

Sender (message, window_size)

Output :

Enter the message : hello

Enter the window size : 2

Sender : Sending frames from 1 to 2

frame 1 : 'h' sent

frame 2 : 'e' sent

Receiver : Sending frame from 3 to 4

frame 3 : 'l' sent

frame 4 : 'o' sent

Receiver : A acknowledgement Nos

Received from frames

from 3 to 4 reverting the frames

Sender : Sending frames from 3 to 4

frame 3 : 'l' sent

frame 4 : 'o' sent

Receiver : Acknowledgment received from 3 to 4

Results

Thus program is created and verified successfully.