

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.

Sender Program:

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.
6. Wait for the acknowledgement from the Receiver.
7. Reader a file called Receiver_Buffer.
8. Check Ack field for the Acknowledgement number.
9. If the Acknowledgement number is as expected, send new frames (set) accordingly. [overwrite the Sender_Buffer with new frames]. Else if No Ack is received, resend the frames accordingly. [Overwrite the Sender_Buffer with old frame].

Import time

import os

window_size = int(input("Enter Window Size:"))

message = input("Enter the message to send:")

frames = []

for i, ch in enumerate(message):

frames.append({"Frame_No": i, "DATA": ch})

def send_frames(start, end):

to_send = frames[start:end]

with open("Sender_Buffer.txt", "w") as f:

for frame in to_send:

line = f"Frame: {frame['Frame_No']} DATA: {frame['DATA']}\n"


```
print("Sending->", line.strip())  
f.write(line)
```

```
print("Frames written to Sender-Buffer.txt")
```

```
def wait_ack():
```

```
    print("Waiting for Ack/NACK from Receiver...")  
    time.sleep(3)
```

```
    if not os.path.exists("Receiver-Buffer.txt"):
```

```
        print("No Receiver-Buffer found. Exiting...")
```

```
        return None
```

```
    with open("Receiver-Buffer.txt", "r") as f:
```

```
        ack_msg = f.read().strip()
```

```
        print("Receiver says:", ack_msg)
```

```
        return ack_msg
```

```
base = 0
```

```
while base < len(frames):
```

```
    send_frames(base, base + window_size)
```

```
    ack_msg = wait_for_ack()
```

```
    if ack_msg is None:
```

```
        break
```

```
    if ack_msg.startswith("Ack"):
```

```
        ack_no = int(ack_msg.split()[1])
```

```
        if ack_no == base + window_size:
```

```
            print("Ack received. Sending next set...\n")
```

```
            base += window_size
```

```
        else:
```

```
            print("Unexpected Ack. Resending...")
```

```
    elif ack_msg.startswith("NACK"):
```

```
        nack_no = int(ack_msg.split()[1])
```

```
        print(f"No Ack for frame {nack_no}. Resending from there...")
```

```
        base = nack_no
```


else:

```
print("Unknown Response. Resending same window...")
```

Enter Window Size: 3

Enter the message to send: Helloworld

Sending → Frame: 0 DATA: H

Sending → Frame: 1 DATA: e

Sending → Frame: 2 DATA: l

Frames written to Sender-Buffer.txt

Waiting for Ack/NACK from Receiver...

Receiver says: Ack 3

Ack received. Sending next set...

Receiver Program:

```
import time
```

```
import random
```

```
import os
```

```
def read_frames():
```

```
    if not os.path.exists("Sender-Buffer.txt"):
```

```
        print("Sender Buffer not found.")
```

```
        return []
```

```
frames = []
```

```
with open("Sender-Buffer.txt", "r") as f:
```

```
    for line in f:
```

```
        parts = line.strip().split()
```

```
        if len(parts) >= 4:
```

```
            frame_no = int(parts[1])
```

```
            data = parts[3]
```

```
            frames.append((frame_no, data))
```

```
return frames
```



```
def send_ack(msg):
    with open("Receiver_Buffer.txt", "w") as f:
        f.write(msg + "\n")
    print(f"Sent to Sender: {msg}")
```

```
def receiver():
    expected_frame = 0
    while True:
        time.sleep(3)
        frames = read_frames()
        if not frames:
            continue

        print("\n---Receiver Reading Frames---")
        for fr in frames:
            print(f"Received => Frame: {fr[0]} DATA: {fr[1]}")

        if random.randint(0,4) == 0:
            nack_no = expected_frame
            send_ack(f"NACK {nack_no}")
        else:
            expected_frame += len(frames)
            send_ack(f"Ack {expected_frame}")
```

---Receiver Reading Frames---

Received → Frame: 0 DATA: H

Received → Frame: 1 DATA: e

Received → Frame: 2 DATA: l

Sent to Sender: Ack 3

Result:

Hence, Sliding Window Protocol is implemented successfully.

9/10/22