

EX.NO: 7

SLIDING WINDOW PROTOCOL

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 CODE:

```

Sender.py
import os

def create_frames(message):
    frames = []
    for i, char in enumerate(message):
        frames.append(i, frame_no)

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: {frame_no + i} frames")
                with open(sender_buffer, 'a') as f:
                    f.write(f'{frames[frame_no + i][0]}')

```



```

    {frames[frame-no+i][i] "\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(f"ACK received for frame: {ack_no}")
        frame_no = ack_no + 1
    else:
        print(f"NACK received for frame: {frame_no}")
        resending...
if __name__ == "__main__":
    window_size = int(input("Enter window size"))
    message = input("Enter message")
    sender(window_size, message)

```

Receiver.py

```

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(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},
```

```
        data: {data}")
```

```
        with open(receiver_buffer, 'w') as f:
```

```
            f.write(str(expected_frame_no - 1))
```

```
if __name__ == "__main__":
```

```
    receiver()
```


OUTPUT

sender.py
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 received 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 received for frame: 0, resending.

sending frame: [0, 'h']

sending frame: [1, 'e']

sending frame: [2, 'l']

sending frame: [3, 'l']

sending frame: [4, 'o']

ACK received for frame: 4

receiver.py

Unexpected frame: 2, expected: 0

Unexpected frame: 2, expected: 0

Unexpected frame: 3, expected: 0

RESULT:

Thus, the sliding window protocol program is studied and observed.