PRACTICAL NO : 4

NAME : KRISH PAROTHI

SECTION: A4

BATCH: B3

ROLL NO. : 49

SUBJECT : COMPUTER NETWORKS

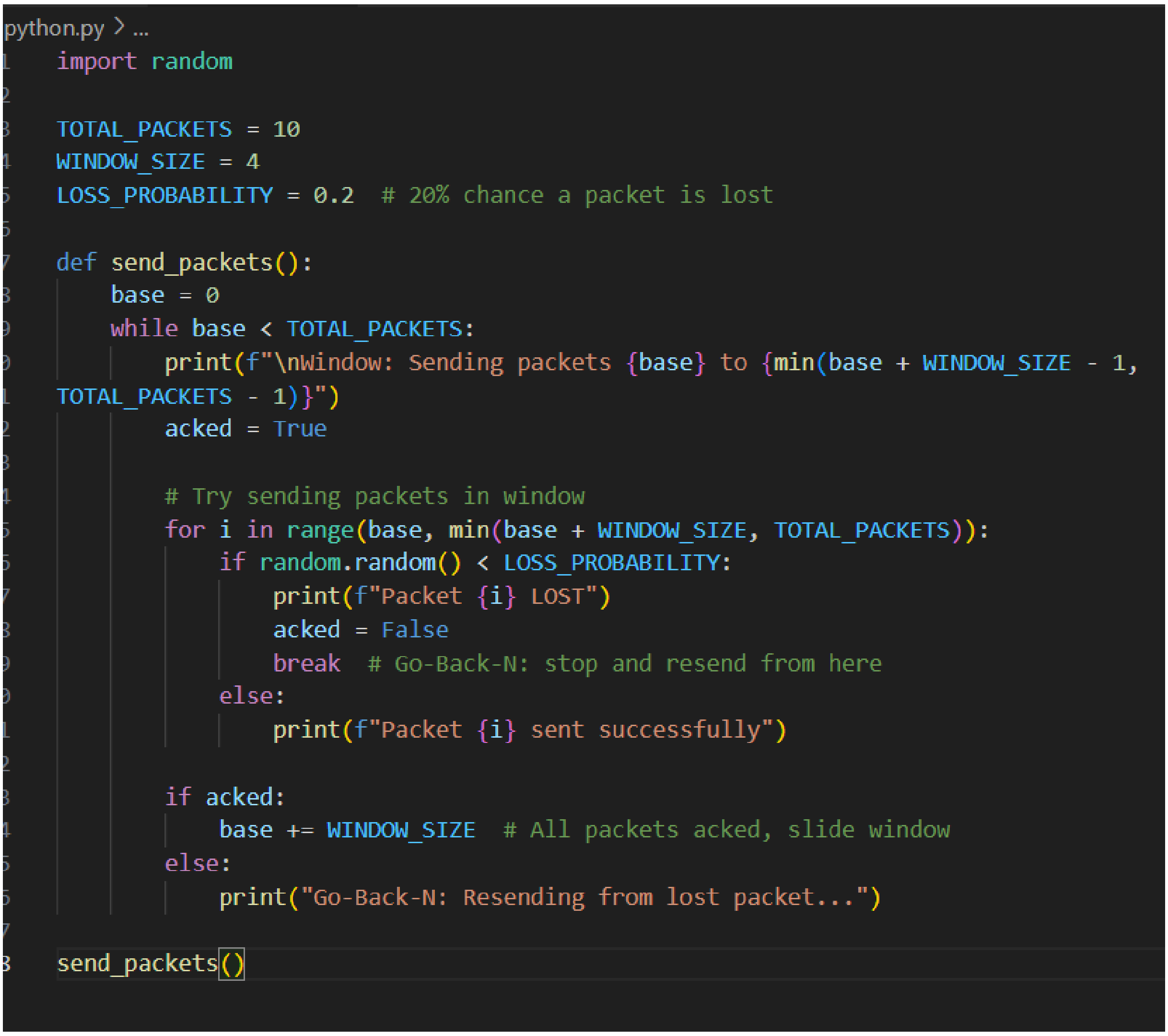
AIM : To Implement Data link Layer flow control mechanism Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism:-
A small file (set of data packets) needs to be transmitted over a channel that may lose or reorder packets. To manage the flow and recover from lost packets, the sender and receiver implement the sliding window protocol with Go-Back-N (window size = 4). If a packet acknowledgment is not received within a fixed timeout, the sender should retransmit all packets from the unacknowledged one.

CODE :

```
import random

TOTAL_PACKETS = 10
WINDOW_SIZE = 4
LOSS_PROBABILITY = 0.2  # 20% chance a packet is lost

def send_packets():
    base = 0
    while base < TOTAL_PACKETS:
        print(f"\nWindow: Sending packets {base} to {min(base +
WINDOW_SIZE - 1,
TOTAL_PACKETS - 1)}")
        acked = True

        # Try sending packets in window
        for i in range(base, min(base + WINDOW_SIZE, TOTAL_PACKETS)):
            if random.random() < LOSS_PROBABILITY:
                print(f"Packet {i} LOST")
                acked = False
                break  # Go-Back-N: stop and resend from here
            else:
                print(f"Packet {i} sent successfully")

        if acked:
            base += WINDOW_SIZE  # All packets acked, slide window
        else:
            print("Go-Back-N: Resending from lost packet...")
```

```
send_packets()
```

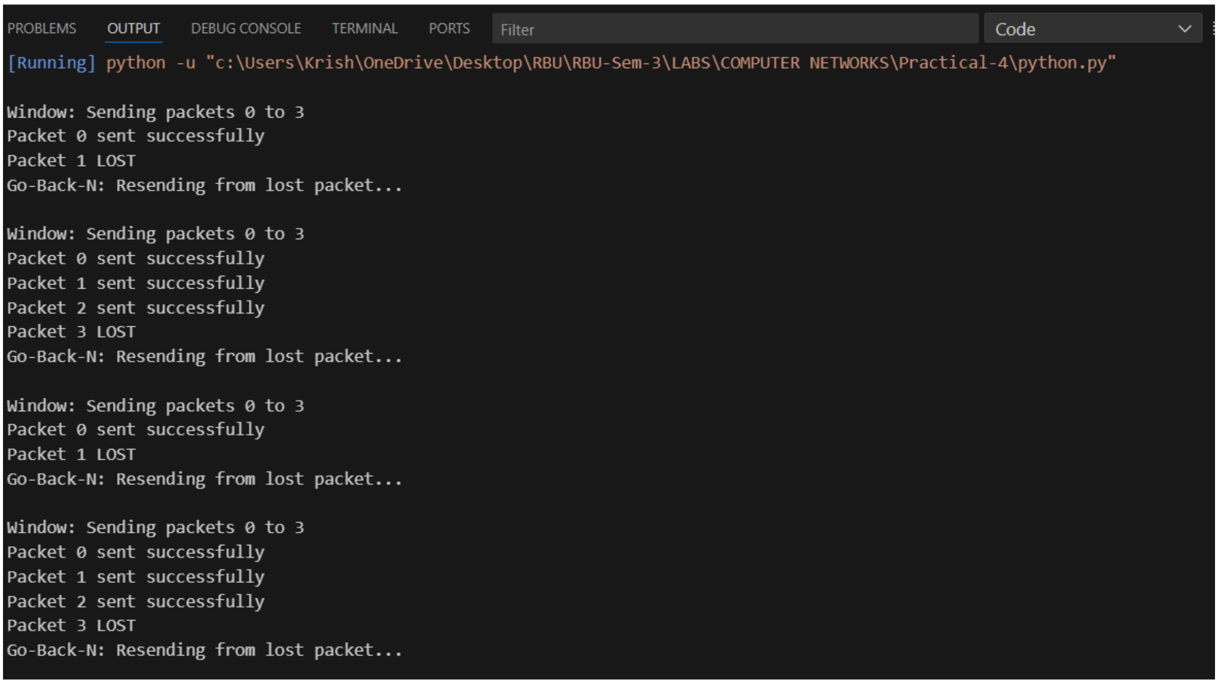
CODE SCREENSHOT:



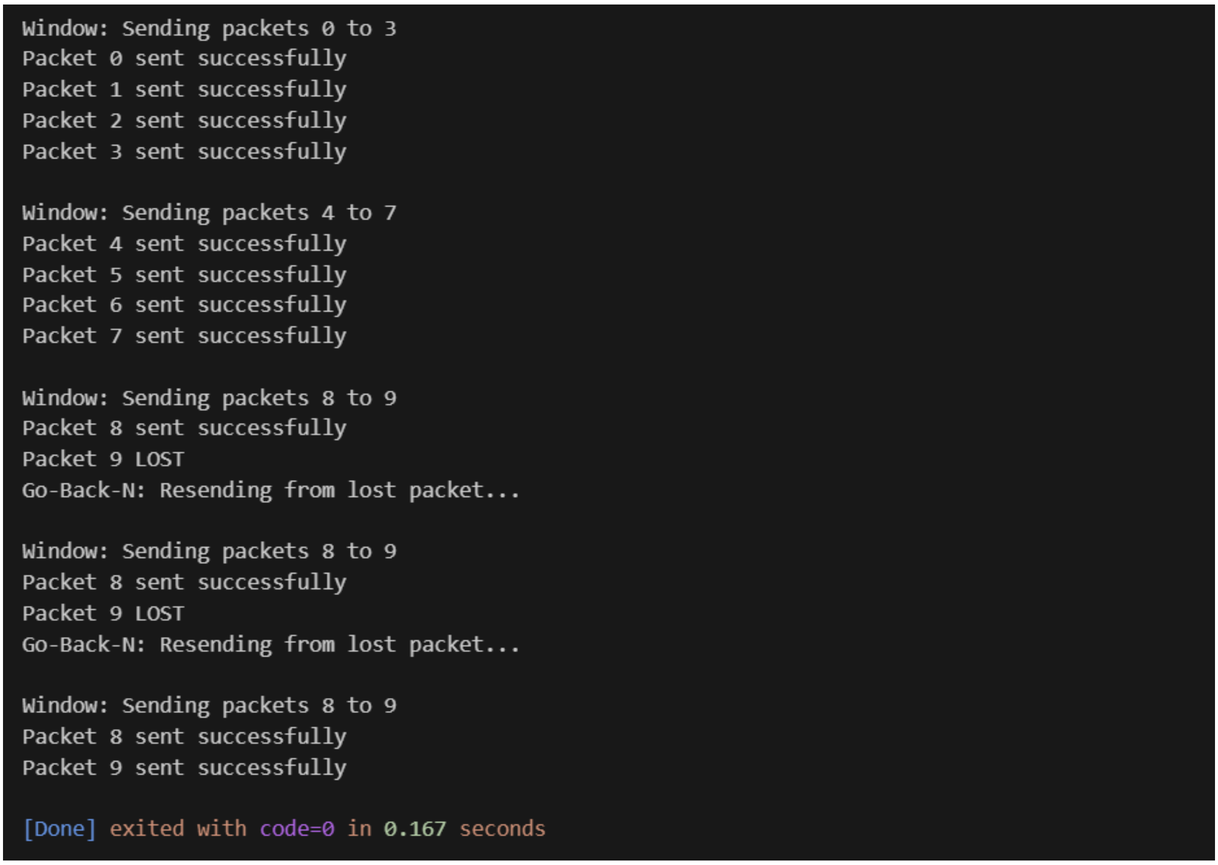
python.py > ...

```python
import random

TOTAL_PACKETS = 10
WINDOW_SIZE = 4
LOSS_PROBABILITY = 0.2  # 20% chance a packet is lost

def send_packets():
    base = 0
    while base < TOTAL_PACKETS:
        print(f"\nWindow: Sending packets {base} to {min(base + WINDOW_SIZE - 1,
TOTAL_PACKETS - 1)}")
        acked = True

        # Try sending packets in window
        for i in range(base, min(base + WINDOW_SIZE, TOTAL_PACKETS)):
            if random.random() < LOSS_PROBABILITY:
                print(f"Packet {i} LOST")
                acked = False
                break  # Go-Back-N: stop and resend from here
            else:
                print(f"Packet {i} sent successfully")

        if acked:
            base += WINDOW_SIZE  # All packets acked, slide window
        else:
            print("Go-Back-N: Resending from lost packet...")

send_packets()
```

OUTPUT :



```
[Running] python -u "c:\Users\Krish\OneDrive\Desktop\RBU\RBU-Sem-3\LABS\COMPUTER NETWORKS\Practical-4\python.py"

Window: Sending packets 0 to 3
Packet 0 sent successfully
Packet 1 LOST
Go-Back-N: Resending from lost packet...

Window: Sending packets 0 to 3
Packet 0 sent successfully
Packet 1 sent successfully
Packet 2 sent successfully
Packet 3 LOST
Go-Back-N: Resending from lost packet...

Window: Sending packets 0 to 3
Packet 0 sent successfully
Packet 1 LOST
Go-Back-N: Resending from lost packet...

Window: Sending packets 0 to 3
Packet 0 sent successfully
Packet 1 sent successfully
Packet 2 sent successfully
Packet 3 LOST
Go-Back-N: Resending from lost packet...
```



```
Window: Sending packets 0 to 3
Packet 0 sent successfully
Packet 1 sent successfully
Packet 2 sent successfully
Packet 3 sent successfully

Window: Sending packets 4 to 7
Packet 4 sent successfully
Packet 5 sent successfully
Packet 6 sent successfully
Packet 7 sent successfully

Window: Sending packets 8 to 9
Packet 8 sent successfully
Packet 9 LOST
Go-Back-N: Resending from lost packet...

Window: Sending packets 8 to 9
Packet 8 sent successfully
Packet 9 LOST
Go-Back-N: Resending from lost packet...

Window: Sending packets 8 to 9
Packet 8 sent successfully
Packet 9 sent successfully

[Done] exited with code=0 in 0.167 seconds
```