<u>Congestion Control</u>

- Niranjan Gopal
(IMT2022543)

Working

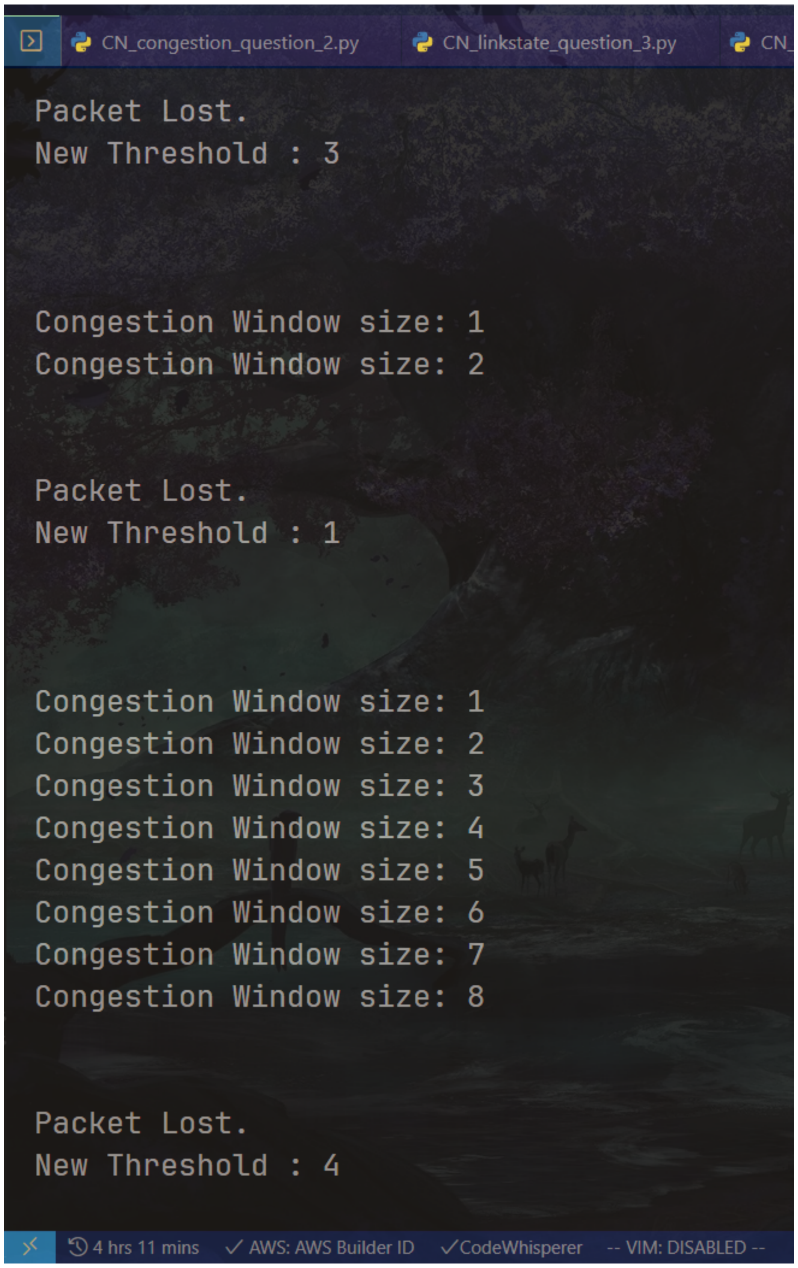
The code simulates a congestion control mechanism following Tahoe's schema. It uses a sender and a simulated channel to send data while adapting the congestion window size based on the ACKs received.

The SimulatedChannel class represents a simulated channel that can either successfully send a packet or simulate packet loss based on a specified loss rate.The CongestionControl class implements the congestion control mechanism. It keeps track of the congestion window size (congWin), the current packet being sent (packet), the channel, the threshold (`ssthresh`), the number of packets to send (packetsToSend), and the number of packets already sent (packetsSent).

The send_data method in the CongestionControl class is used to set the current packet with the given data.The send_packet method sends the current packet through the channel.
The construct_packet method constructs a packet with the given data and returns it as a dictionary.The receive_ack method checks if an ACK (acknowledgment) is received from the channel.The get_data function is a placeholder that represents the data to be sent. It returns a string of data.

The main part of the code sets up the simulation by creating an instance of the SimulatedChannel class and an instance of the CongestionControl class. It specifies the loss rate, the number of packets to send, and the initial threshold (`ssthresh`). The simulation loop runs until the number of packets sent reaches the desired number (packetsToSend). Inside the loop, the congestion window size is printed. Another loop sends packets based on the current congestion window size. It retrieves data from get_data, sends the packet, and checks if an ACK is received. If the ACK is not received, indicating packet loss, the threshold is decreased, the congestion window size is set to 1, and a message is printed. Otherwise, the congestion window size is increased based on the Additive Increase Multiplicative Decrease (AIMD) algorithm.

Output seen



```
Packet Lost.
New Threshold : 3


Congestion Window size: 1
Congestion Window size: 2


Packet Lost.
New Threshold : 1



Congestion Window size: 1
Congestion Window size: 2
Congestion Window size: 3
Congestion Window size: 4
Congestion Window size: 5
Congestion Window size: 6
Congestion Window size: 7
Congestion Window size: 8


Packet Lost.
New Threshold : 4
```

We can clearly see how the congestion window goes from slow start to congestion avoidance phase and then when a packet is lost executes AIMD.