14/12/2020

Leaky Bucket
writeup

Karthik Venugopal
1BM18CS043

```
class LeakyBucket
    def __init__(self, bucket_size, input_stream, output_rate):
        self.size = bucket_size
        self.queue = input_stream
        self.flow = output_rate

    def control_congestion (self):
        buffer = 0
        for packet in self.queue:
            print (f"Incoming packet: {packet}")
            x = self.size - buffer
            if packet < x:
                buffer += packet
                print (f" Packets sent: {self.flow}
                    \n Buffer: {buffer} \n")

            else:
                print (f"Packet loss: {packet - x}
                    \t packets sent: {self.flow}")
                buffer = self.size
                print (f"Buffer: {buffer}\n")
            buffer -= self.flow
        print (f"Buffer: {buffer} \t Packets sent: {self.flow}")
        ~~while buffer:~~
```

14/12/2020 Leaky Bucket Karthik Venugopal

writeup 1BM18CS043

```
    while buffer
        sent = self.flow if self.flow < buffer else buffer
        buffer -= sent
        print(f" Buffer: {buffer} H Packets sent: {sent}")

input_stream = [int(x) for x in input("Enter input stream of
                        packets: ").split(' ')]

bucket_size = int(input("Enter bucket size:"))
output_rate = int(input("Enter output data rate: "))
network = LeakyBucket(bucket_size, input_stream, output_rate)
network.control_congestion()
```