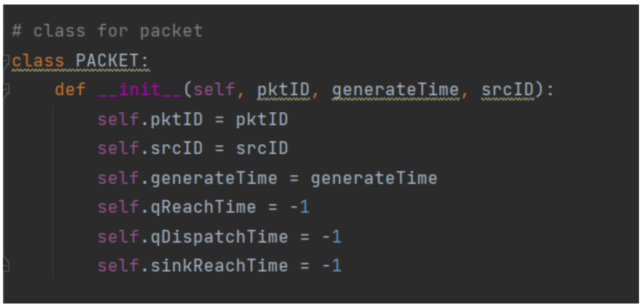
Assignment-2

1901CS75

Siddharth Sanskritayan

Class Definitions:

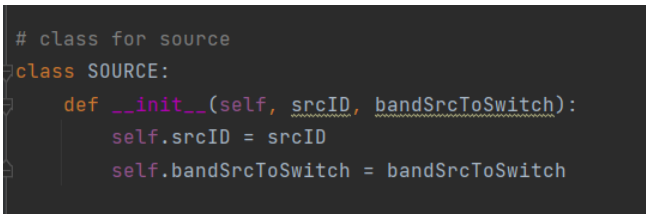
PACKET



```
# class for packet
class PACKET:
    def __init__(self, pktID, generateTime, srcID):
        self.pktID = pktID
        self.srcID = srcID
        self.generateTime = generateTime
        self.qReachTime = -1
        self.qDispatchTime = -1
        self.sinkReachTime = -1
```

Each packet has a Packet ID (starting from 0), a Source ID (also starting from 0), time of generation from the source, time at which it reaches queue (present in the switch), time at which it begins dispatching from queue and time at which it reaches the sink.

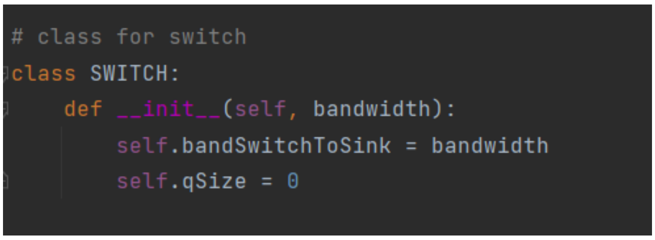
SOURCE



```
# class for source
class SOURCE:
    def __init__(self, srcID, bandSrcToSwitch):
        self.srcID = srcID
        self.bandSrcToSwitch = bandSrcToSwitch
```

Each source has a source id and a bandwidth from that source to switch (kept same for all sources).

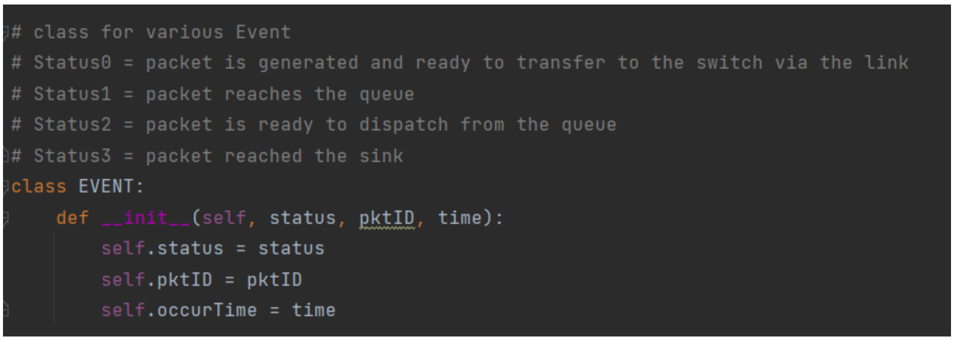
SWITCH



```
# class for switch
class SWITCH:
    def __init__(self, bandwidth):
        self.bandSwitchToSink = bandwidth
        self.qSize = 0
```

Only one switch is present which consists of queue size (number of packets currently in queue) and bandwidth from switch to sink.

EVENT DESCRIPTIONS:



```
# class for various Event
# Status0 = packet is generated and ready to transfer to the switch via the link
# Status1 = packet reaches the queue
# Status2 = packet is ready to dispatch from the queue
# Status3 = packet reached the sink
class EVENT:
    def __init__(self, status, pktID, time):
        self.status = status
        self.pktID = pktID
        self.occurTime = time
```

We push the events in the priority queue (minimum occurrence time event first) with their status, packet associated with the event and event's occurrence time.

WORKING OF ARCHITECTURE:

First we initialize the sources and the switch.

At t=0 all the sources generate one packet each with status = 0. Now we process the events one by one based on the occurrence time. When status = 0, we generate new packet based on Poisson's distribution and push the event. Also we push event having status = 1 (reaching the queue) by adding transmission delay from source to switch.

At the queue, we calculate the waiting time based on current queue size and bandwidth from switch to sink.

Then we push event with status =2 (packet is ready to dispatch from the queue). After this we again add transmission delay from switch to sink thereby pushing the event with status =3 (reached the sink).

SOME COMPUTATIONS:

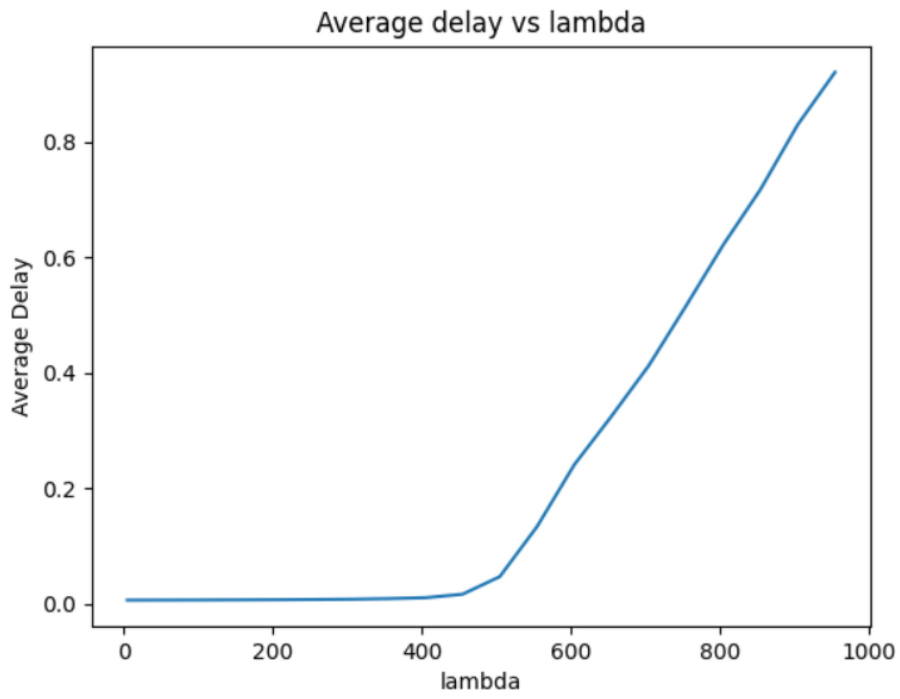
Average delay is calculated by: Delay of all packets reaching the sink / Number of packets reached the sink

Average queue size is calculated by: taking the sum of queue size at each iteration of the events divided by total number of iterations.

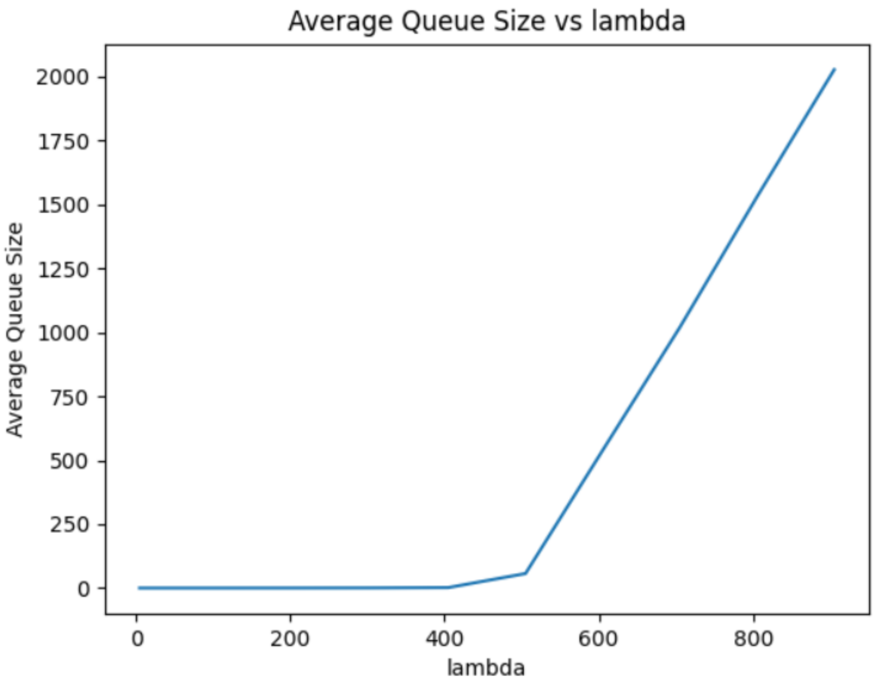
Average Packet drop rate is calculated by: total number of packets dropped (as queue had a fixed size beyond which packets start dropping) divided by total number of packets that arrived at the queue.

SCREENSHOTS OF GRAPHS OBTAINED:

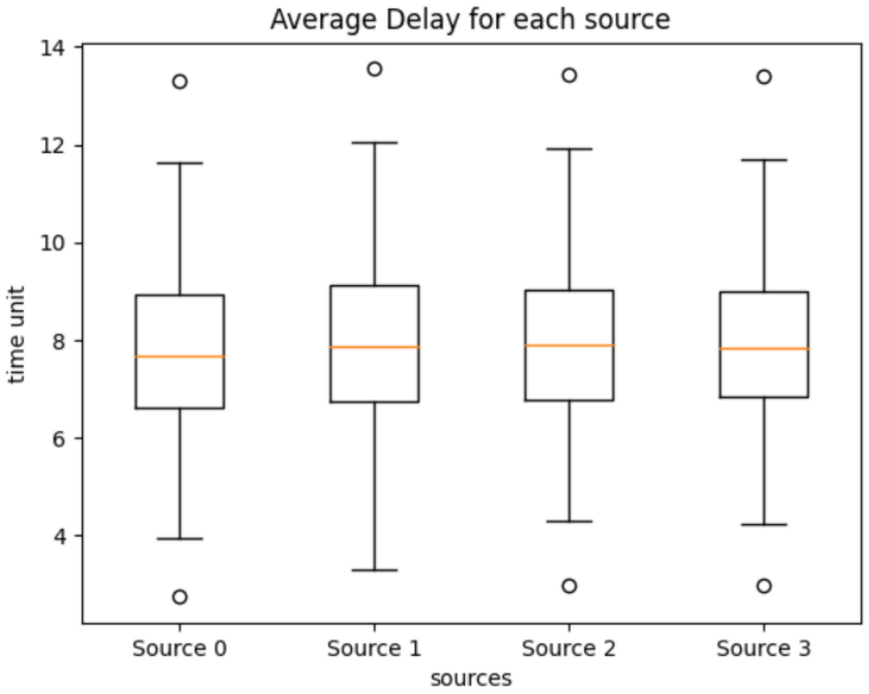
P1)



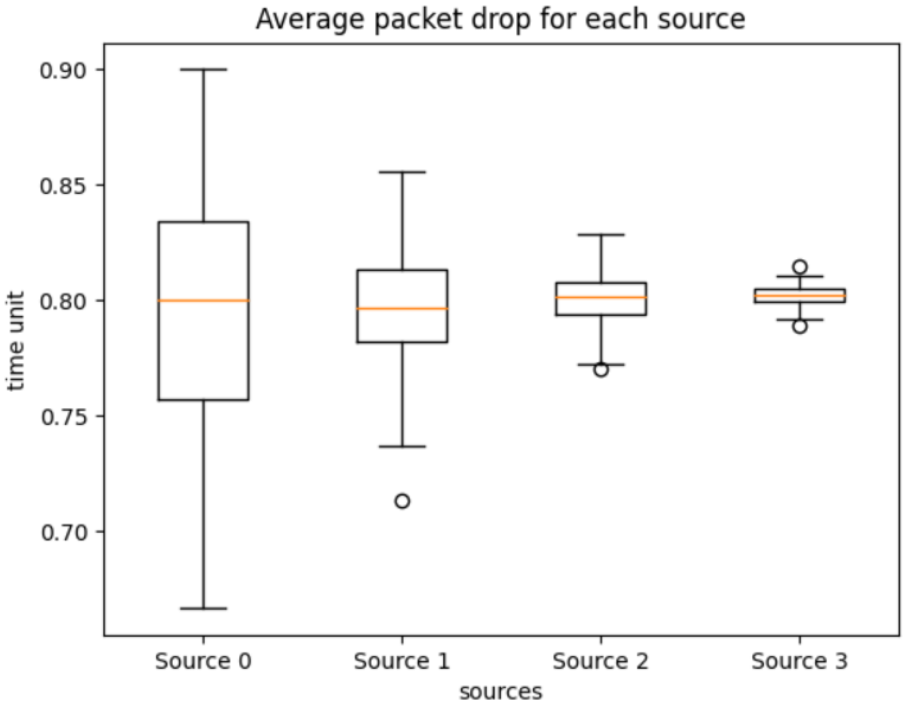
P2)



P3)


Average Delay for each source
14
12
10
8
6
4
time unit
Source 0
Source 1
Source 2
Source 3
sources

P4)


Average packet drop for each source
0.90
0.85
0.80
0.75
0.70
time unit
Source 0
Source 1
Source 2
Source 3
sources