ECE466 Lab 3 Report

Lizhou Wang 998327733

Qiuyi Guan 998373175

Part 1. FIFO Scheduling

Exercise 1.1 Traffic generator for compound Poisson traffic

We modified the traffic generator to be able to generate faster poisson according to parameter N. The following is Poisson traffic rate obtained from our experience:

N = 1 Rate = 0.9Mbps (Theoretical rate = 1Mbps)
N = 5 Rate = 4.5 Mbps (Theoretical rate = 5 Mbps)
N = 9 Rate = 8 Mbps (Theoretical rate = 9Mbps)

In conclusion, the practical rate is a slightly lower than theoretical rate.

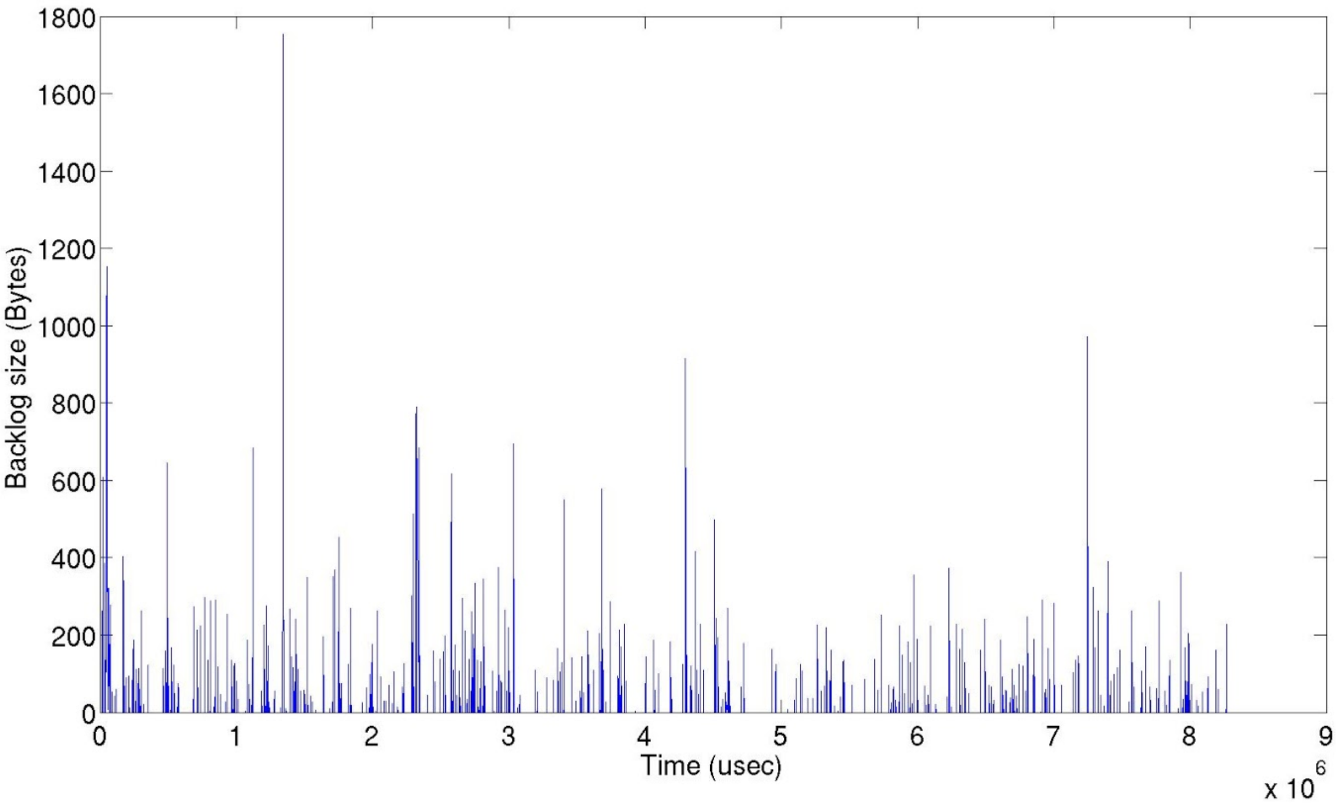
Exercise 1.2 Implement a FIFO Scheduler

We implement the FIFO scheduler with a single FIFO buffer and with no traffic tagging and traffic classification. Please refer to code in part1/ directory.

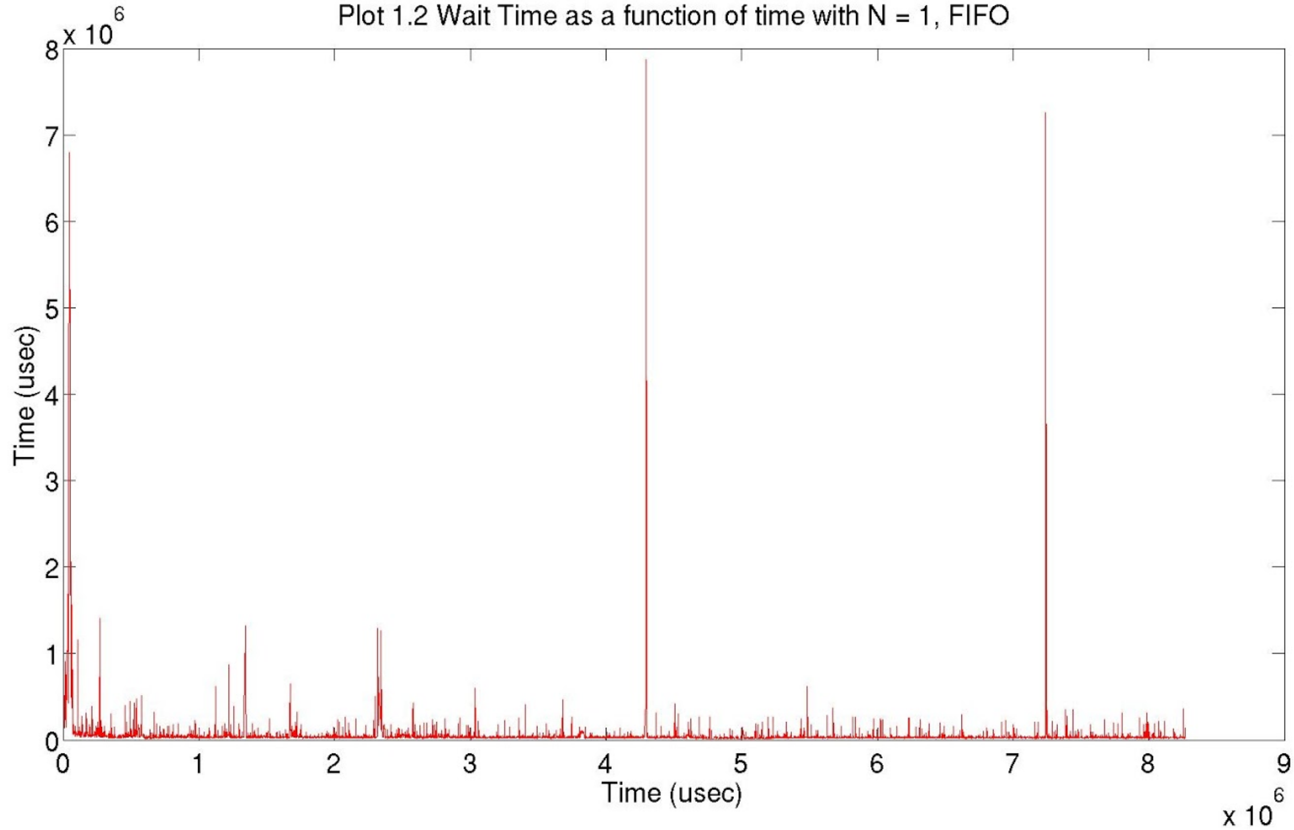
Exercise 1.3 Observing a FIFO Scheduler at different loads

Case1: Low load, with N = 1

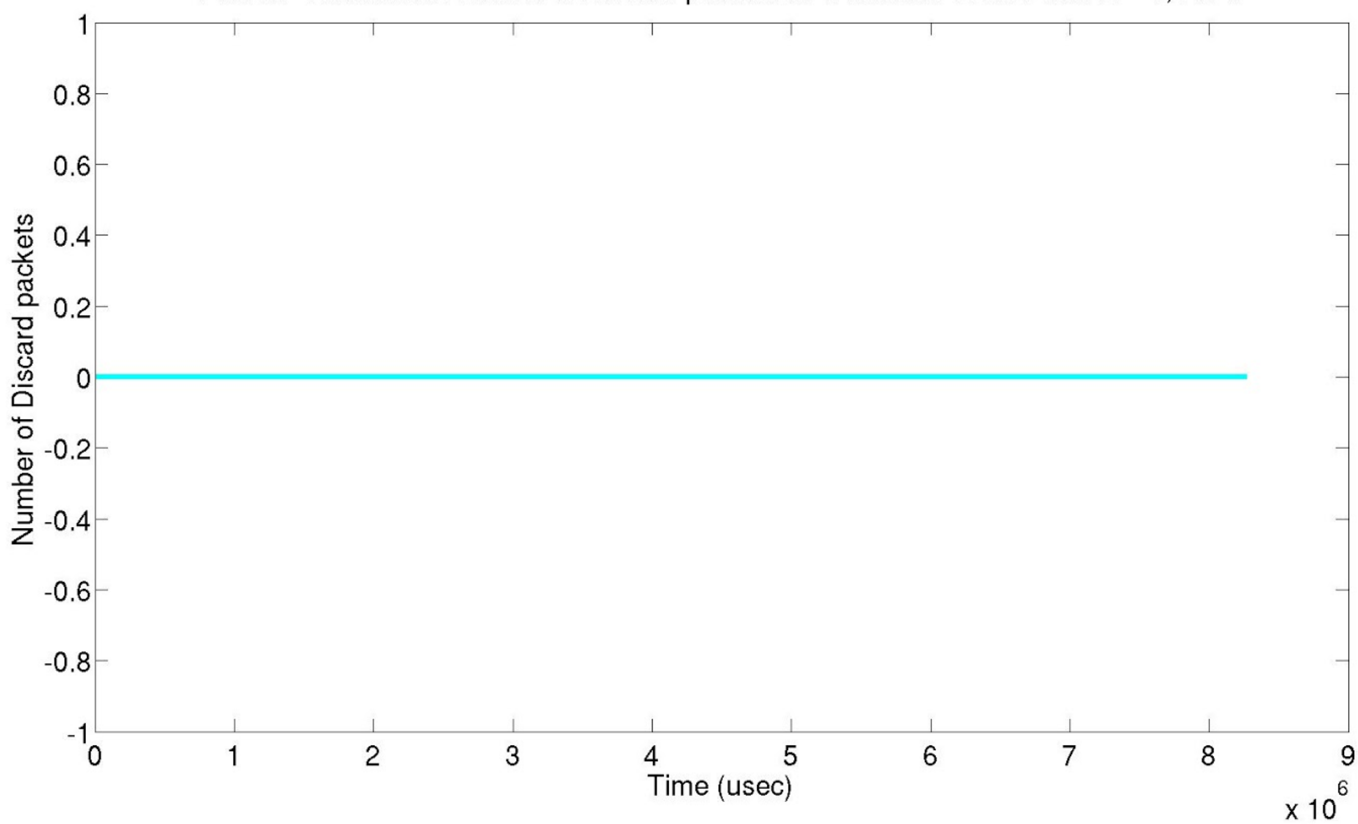
Plot 1.1 System backlog as a function of time with N = 1, FIFO



Plot 1.2 Wait Time as a function of time with N = 1, FIFO



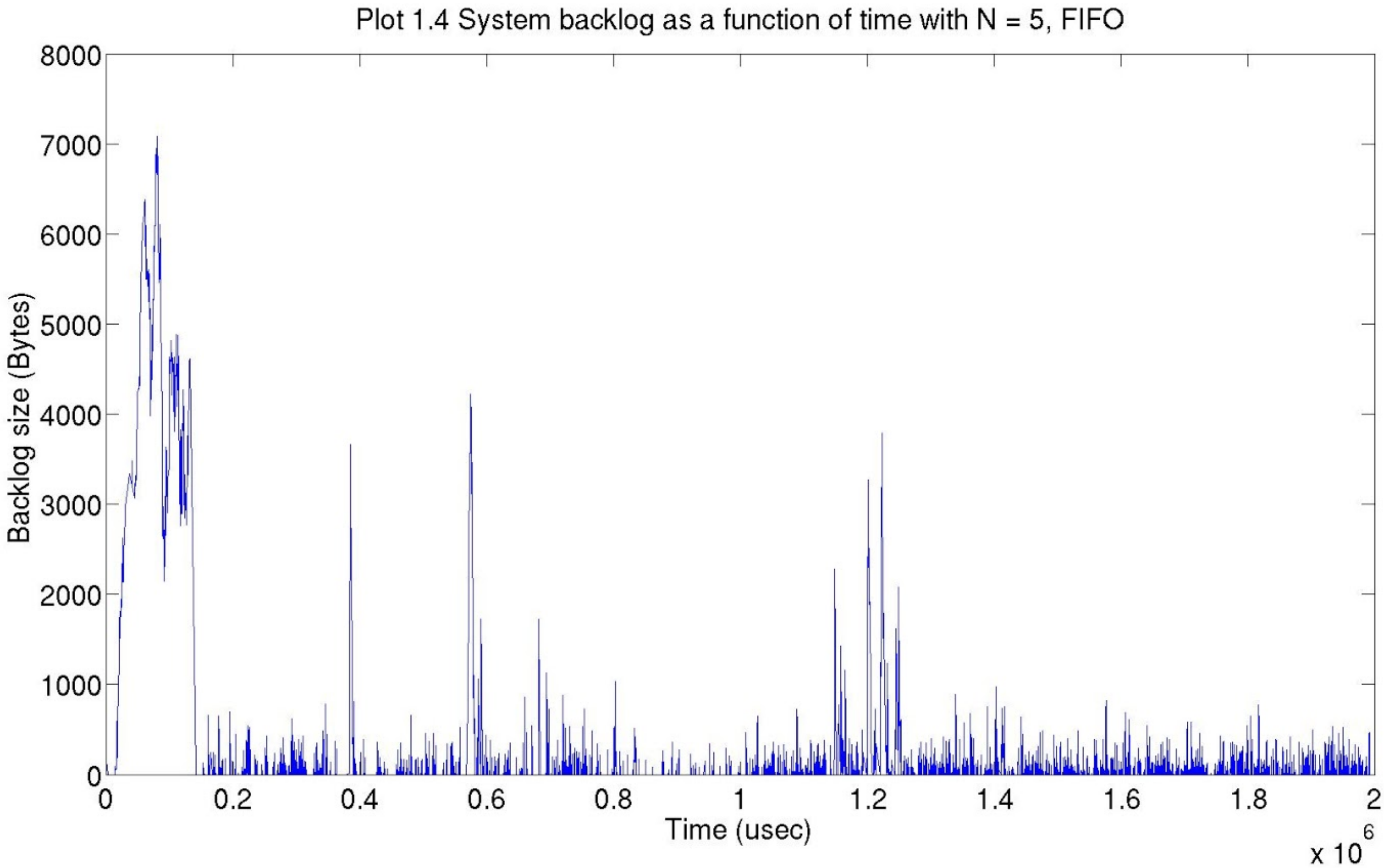
Plot 1.3 Cumulative Number of Discard packets as a function of time with N = 1, FIFO

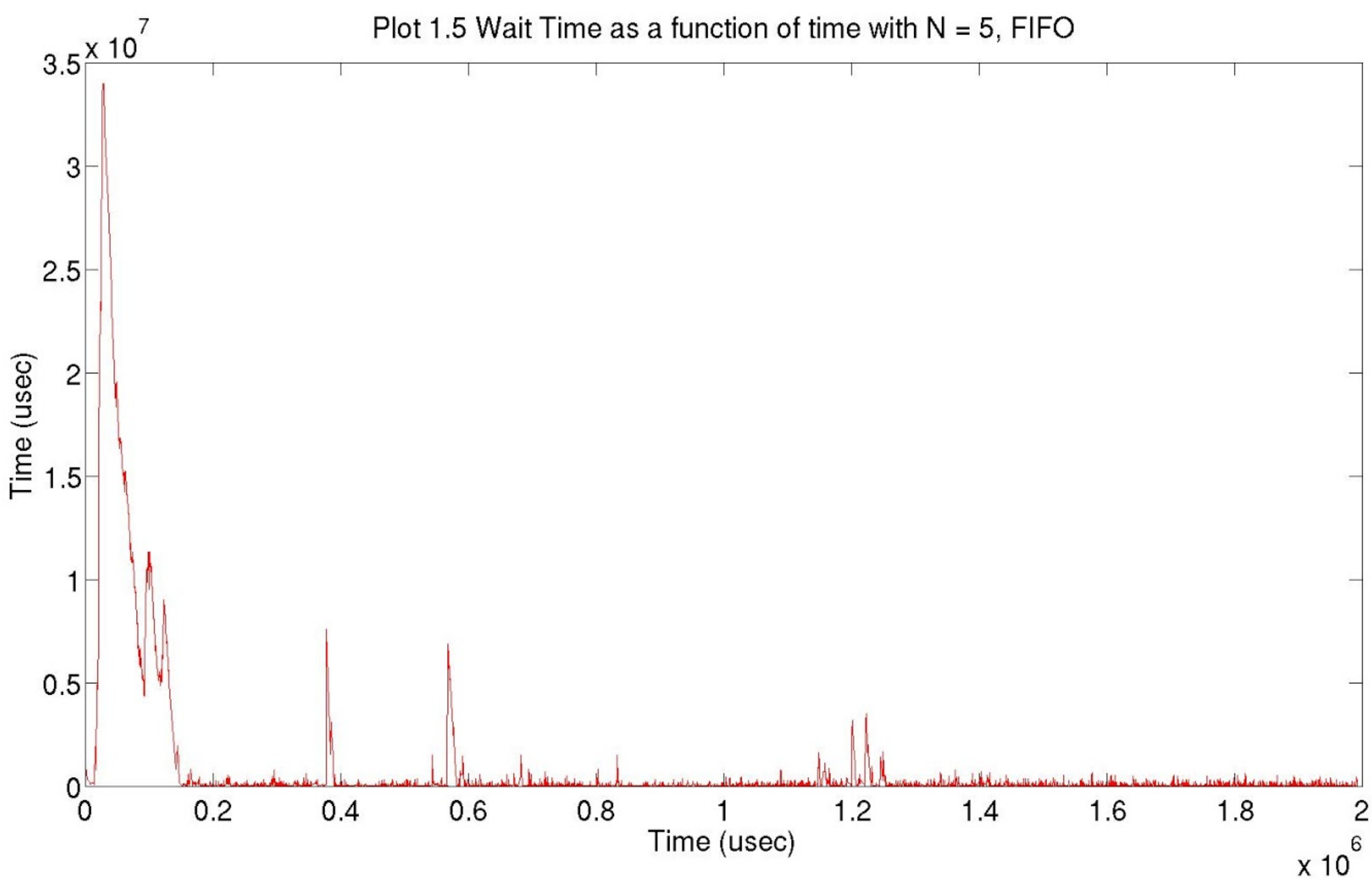


Observation:

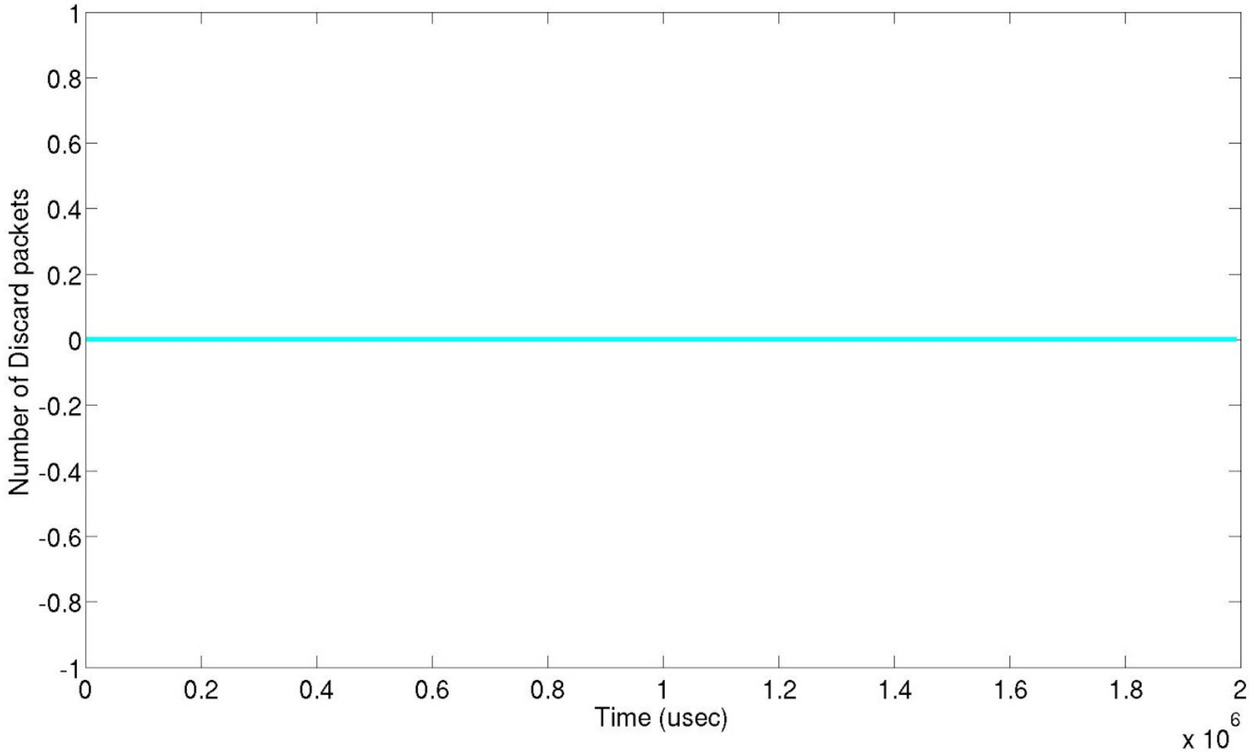
According to plot 1.3, there is no packets drop when N = 1 (source traffic rate 1Mbps). This is because the buffer link has a capacity of 10 Mbps which is significantly bigger than source rate.

Case2: Medium load, with N = 5





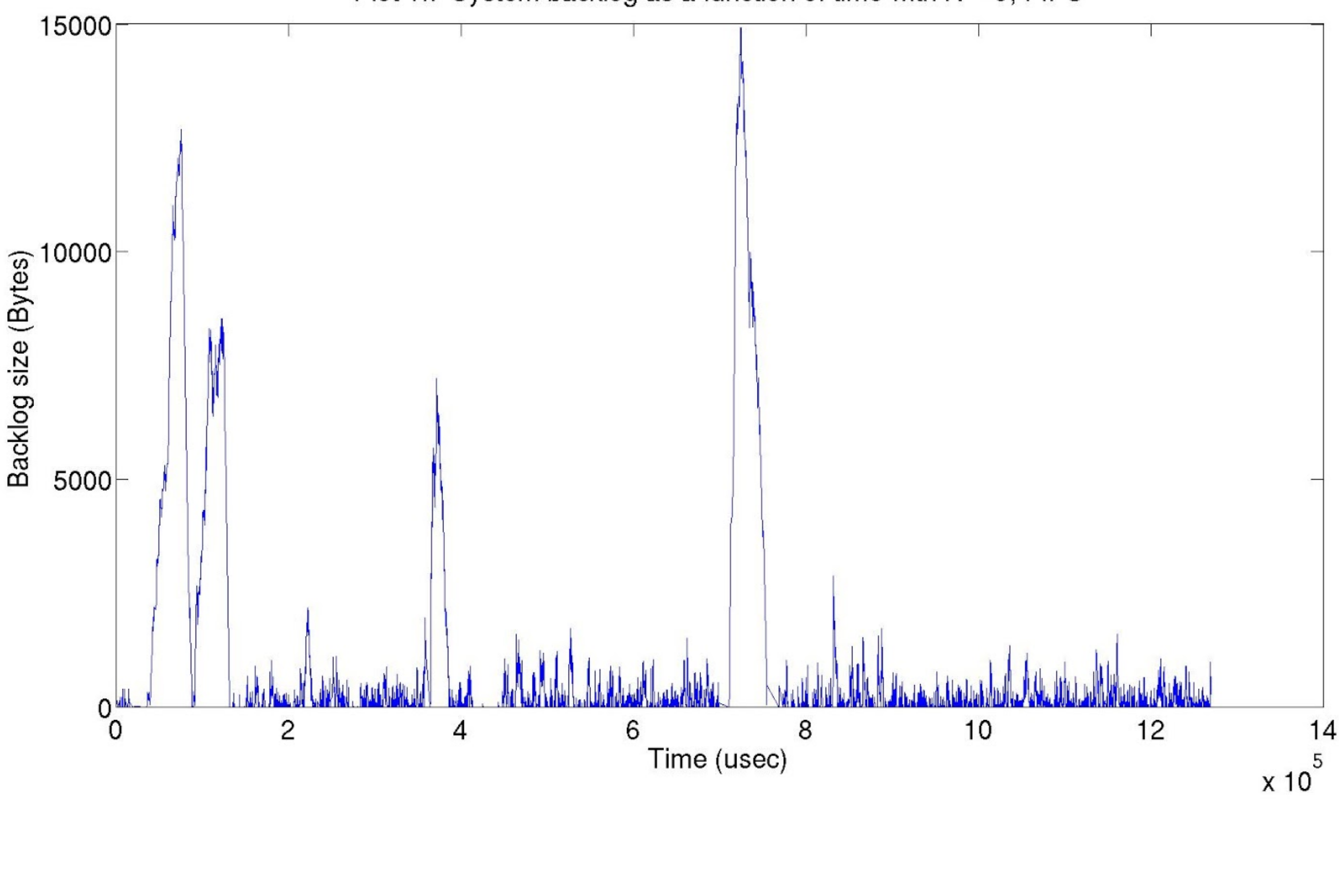
Plot 1.6 Cumulative Number of Discard packets as a function of time with N = 5, FIFO



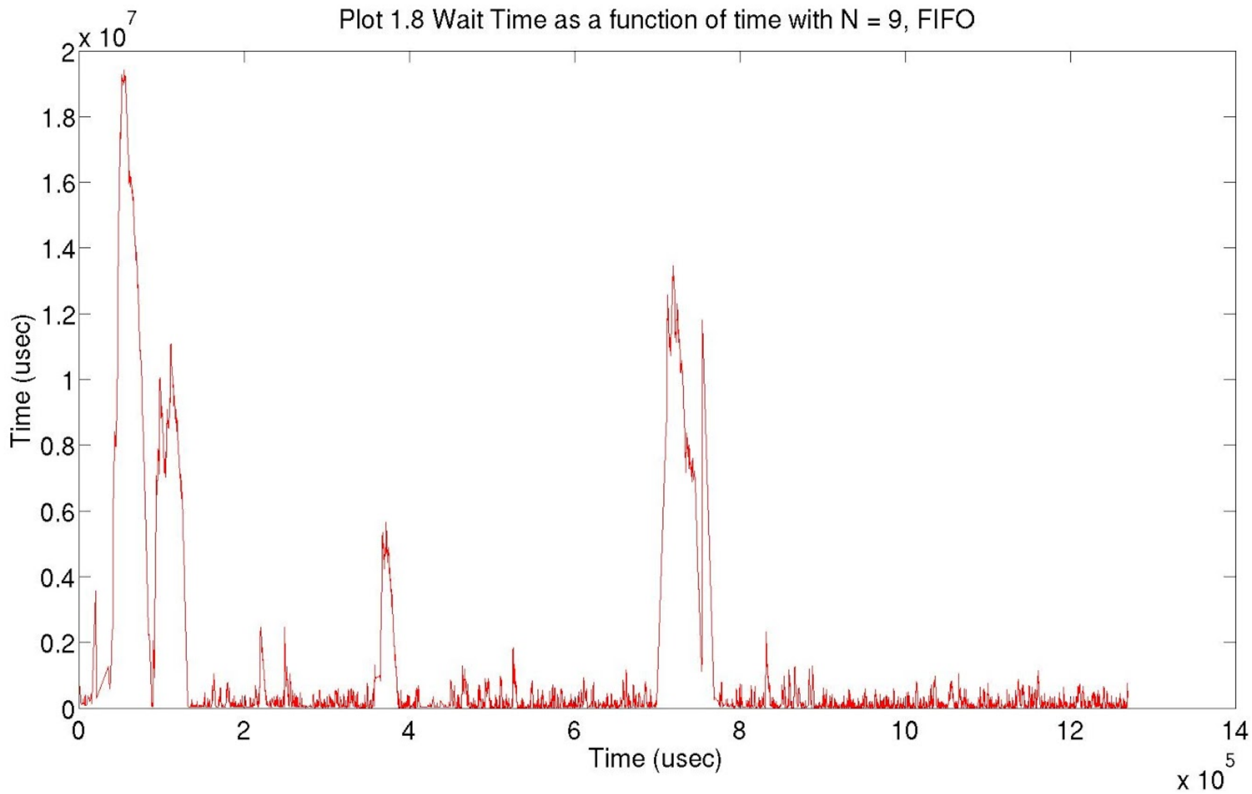
Observation:
Since the capacity of buffer link is 10Mbps which is bigger than poisson traffic rate (4.5 Mbps in this case), there is no packet drop. Comparing this case with N = 1 case, we got more backlog and longer waiting time due to bigger poisson traffic rate.

Case3: High load, with N = 9

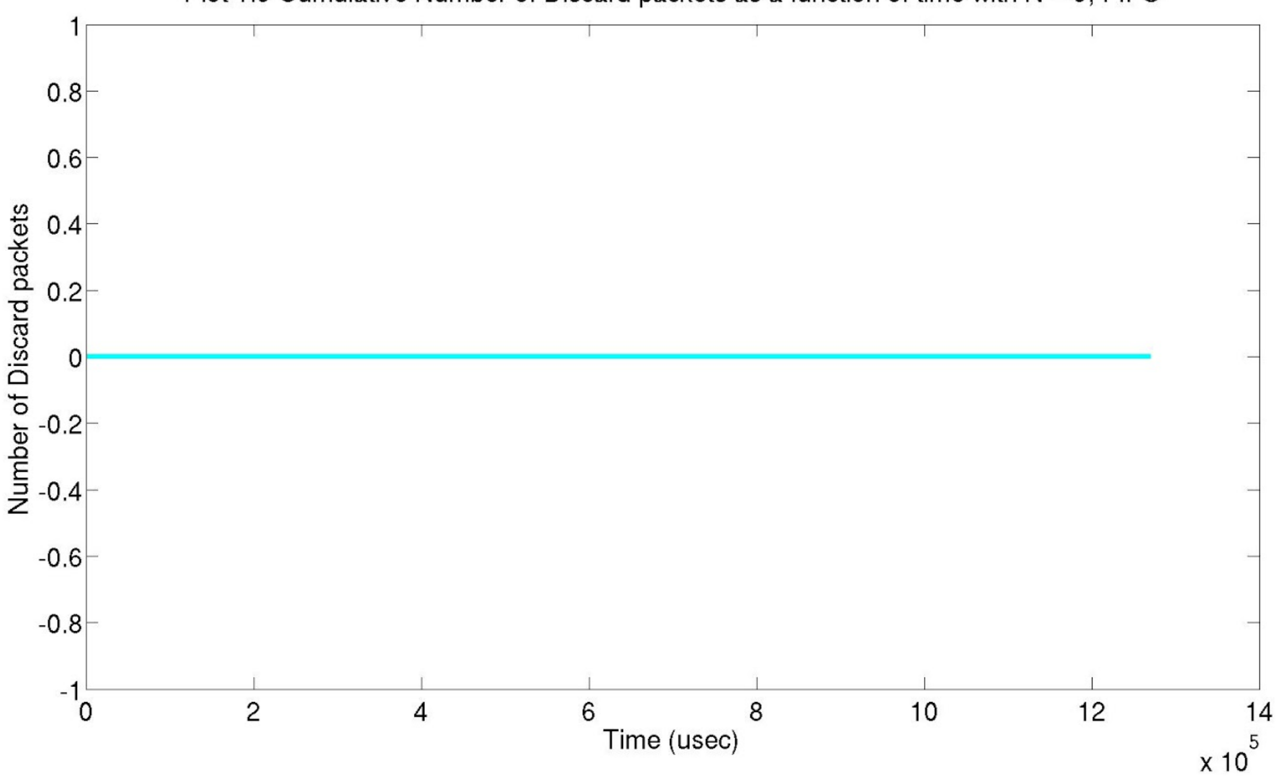
Plot 1.7 System backlog as a function of time with N = 9, FIFO



Plot 1.8 Wait Time as a function of time with N = 9, FIFO



Plot 1.9 Cumulative Number of Discard packets as a function of time with N = 9, FIFO



Observation:

Again, there is no packets drop due to bigger buffer link capacity. Comparing with N=1 and N=5 cases, we got even more backlog and longer waiting time due to bigger poisson traffic rate.

Exercise 1.4 (Optional, 5% extra credit) Unfairness in FIFO

C = 1 Mbps

N parameters	Throughputs for each source
N1 = 5 N2 =1	N1 rate = 0.85 Mbps N2 rate = 0.15 Mbps
N1 = 5 N2 =5	N1 rate = 0.55 Mbps N2 rate = 0.45 Mbps
N1 = 5 N2 =10	N1 rate = 0.1Mbps N2 rate = 0.9 Mbps

Formulation:
C = link capacity
r1 = source 1 rate
r2 = source 2 rate

Throughput 1 = C * r1 / (r1+r2)
Throughput 2 = C * r2 / (r1+r2)

Please refer to code to b2/ directory

Part 2. Priority Scheduling

Exercise 2.1 Transmission of tagged packets

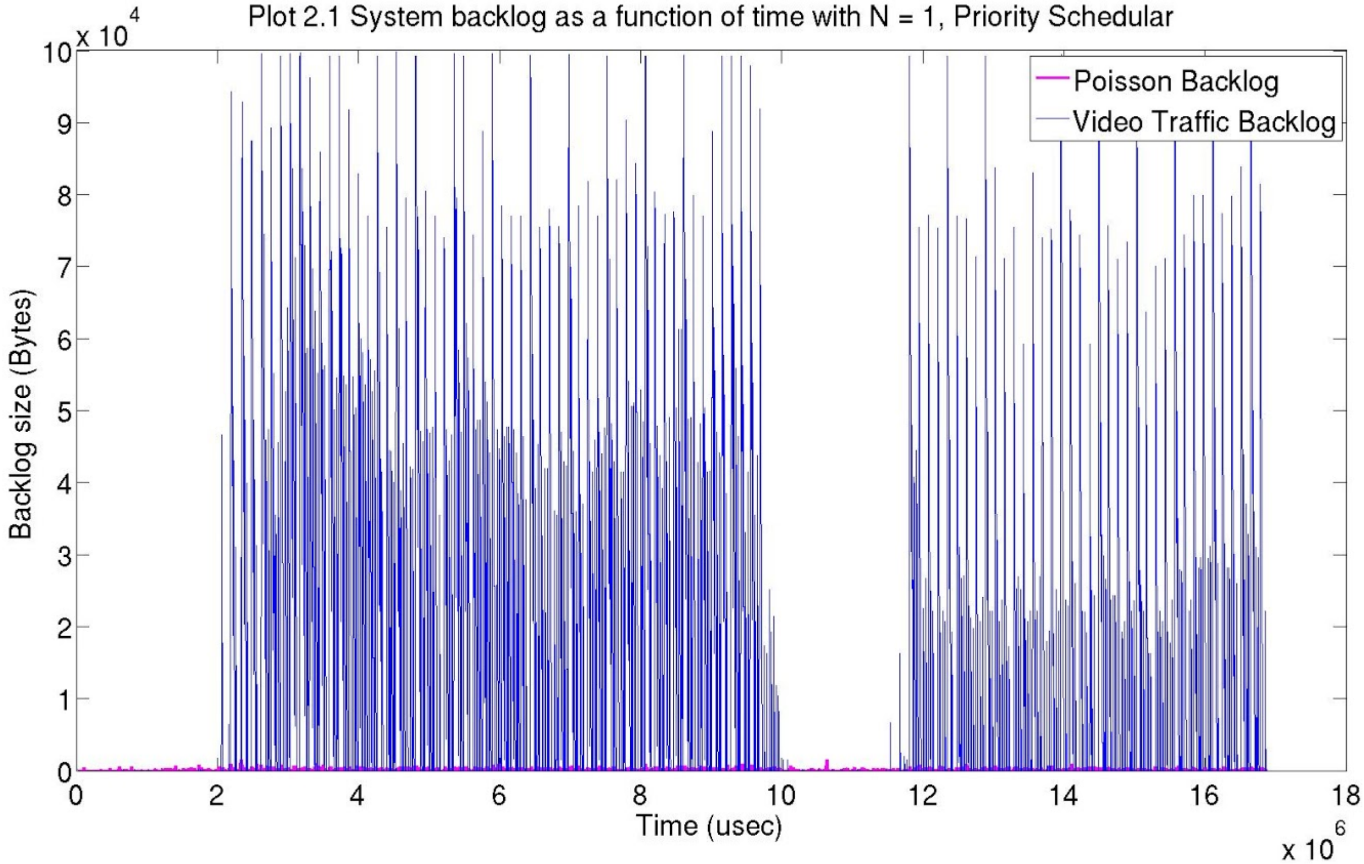
Our traffic generator can tagged the incoming packet with high and low priority by setting first byte in buffer. Please refer to code in part2/ directory.

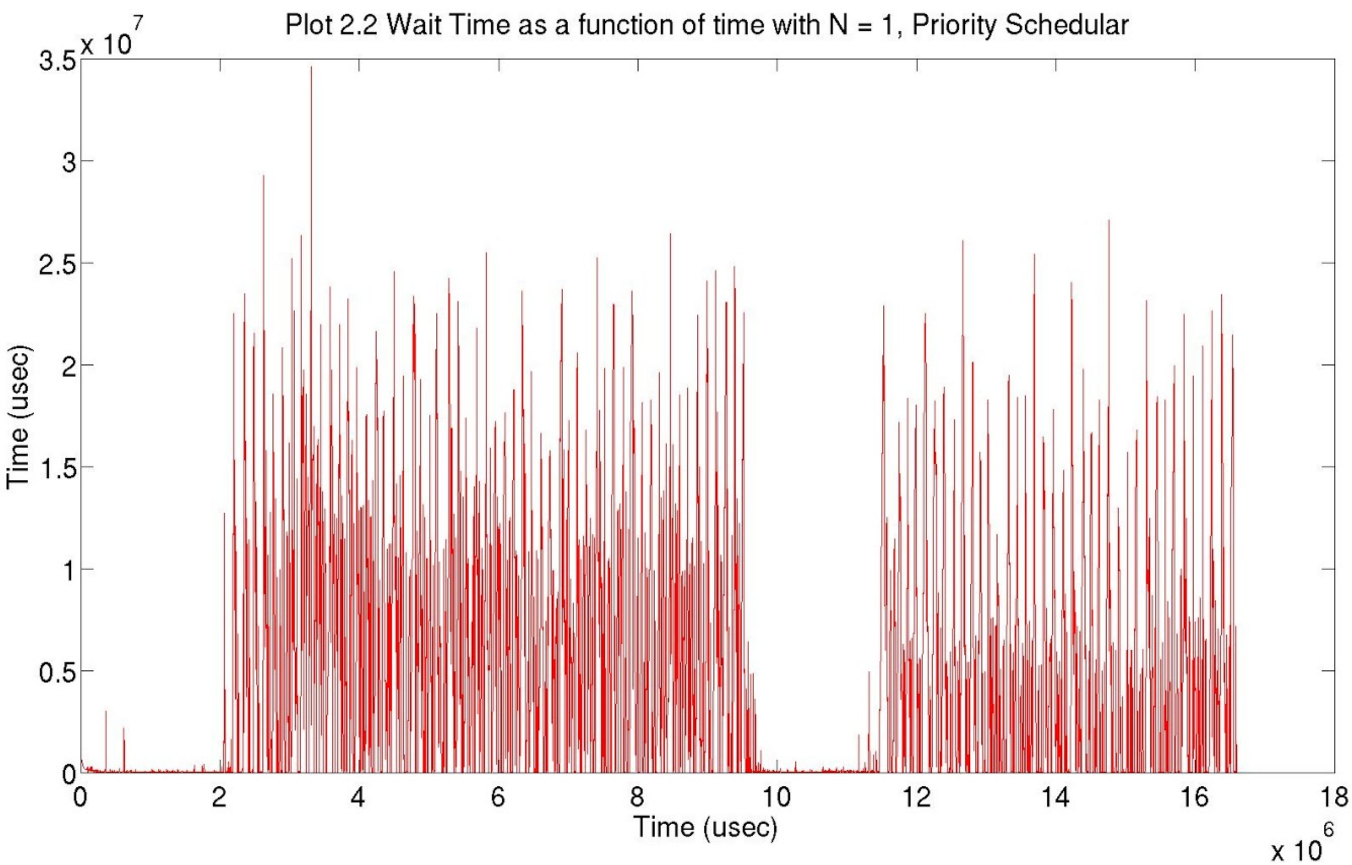
Exercise 2.2 Packet classification and priority scheduling

The priority scheduler will always transmits a high-priority packet, if the high priority FIFO buffer contains a packet. Low priority packets are selected for transmission only when there are no high priority packets.

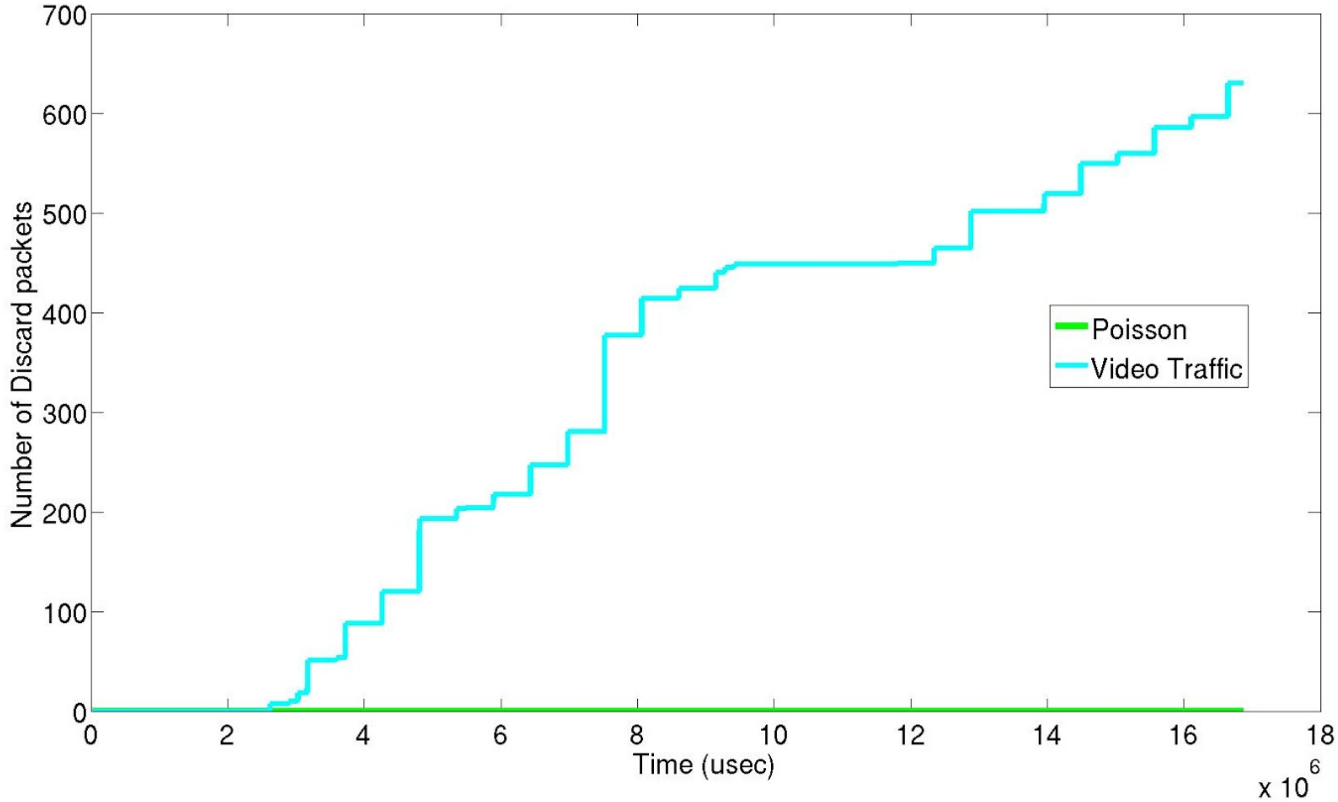
Exercise 2.3 Evaluation of the priority scheduler

Case1: Low load, with N = 1



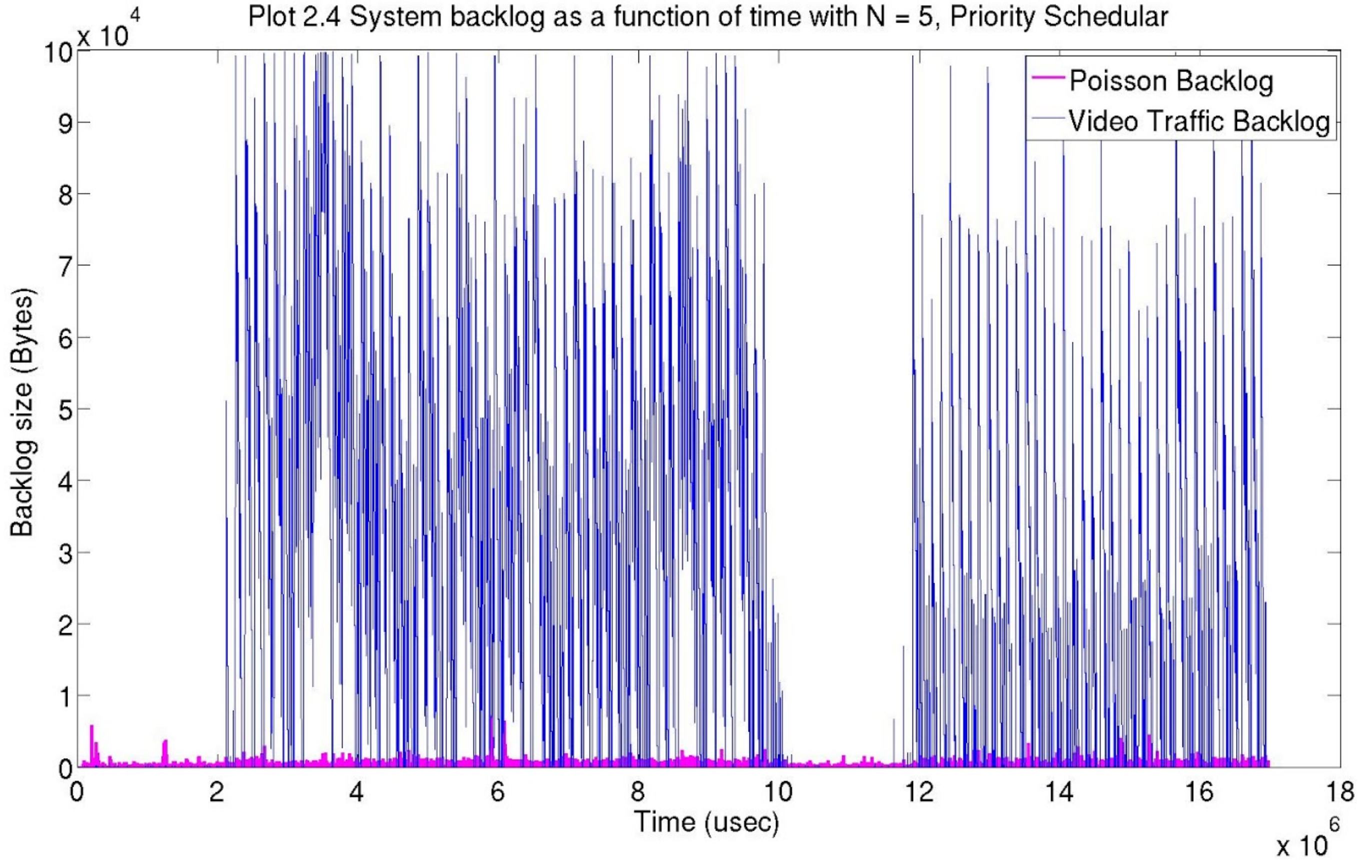


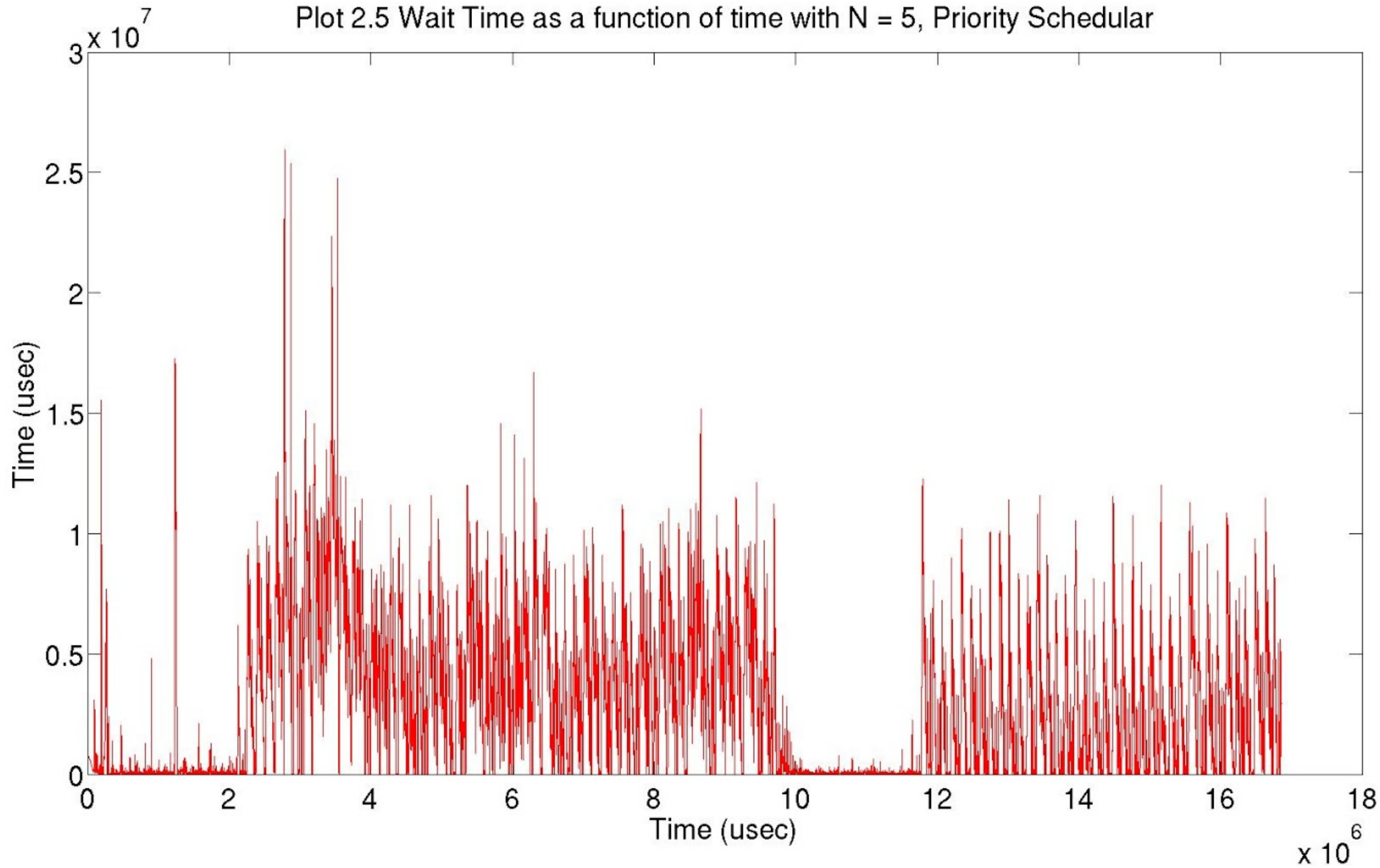
Plot 2.3 Cumulative Number of Discard packets as a function of time with N = 1, Priority Schedular

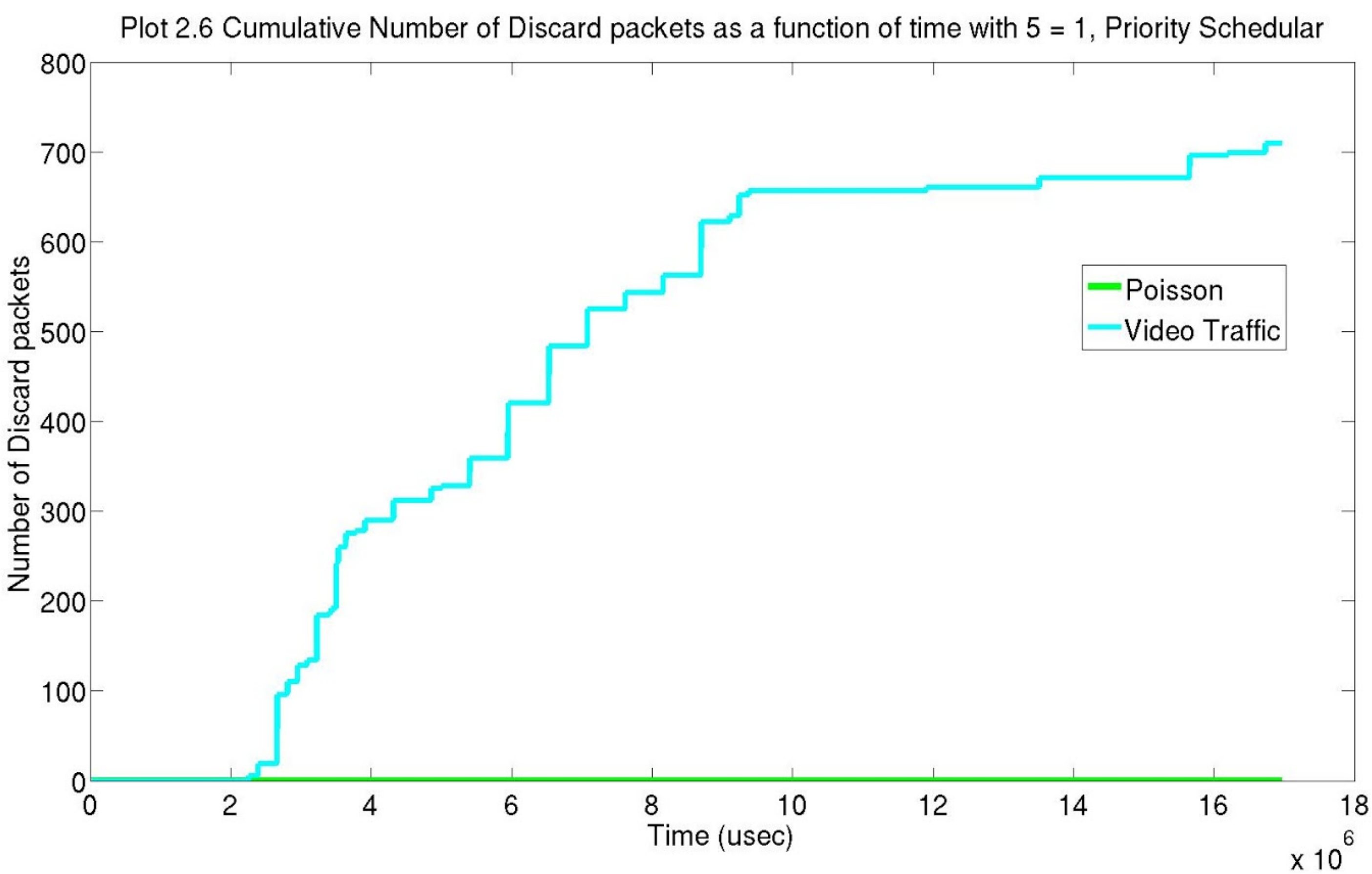


Case2: Medium load, with N = 5

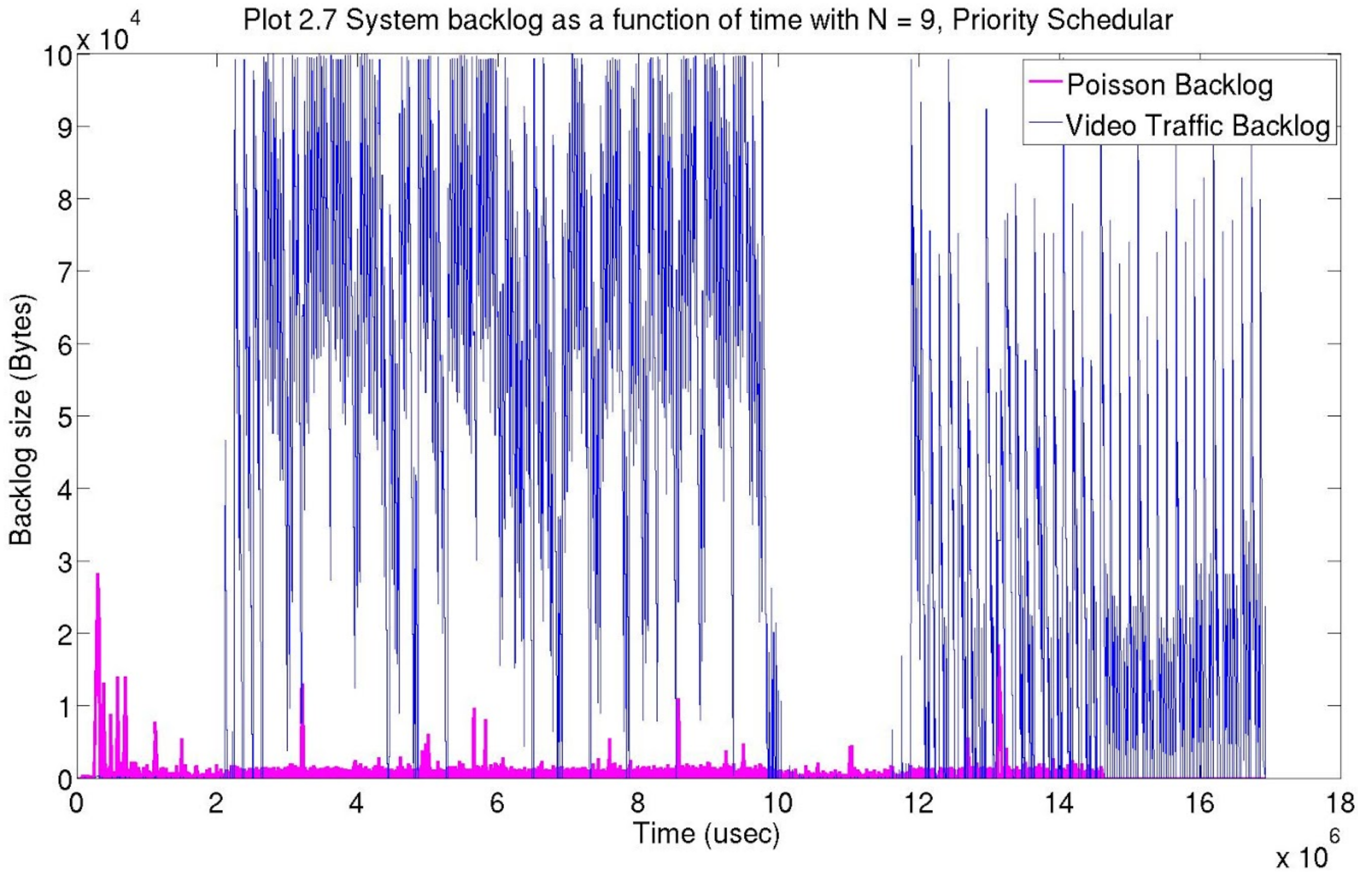
Plot 2.4 System backlog as a function of time with N = 5, Priority Schedular

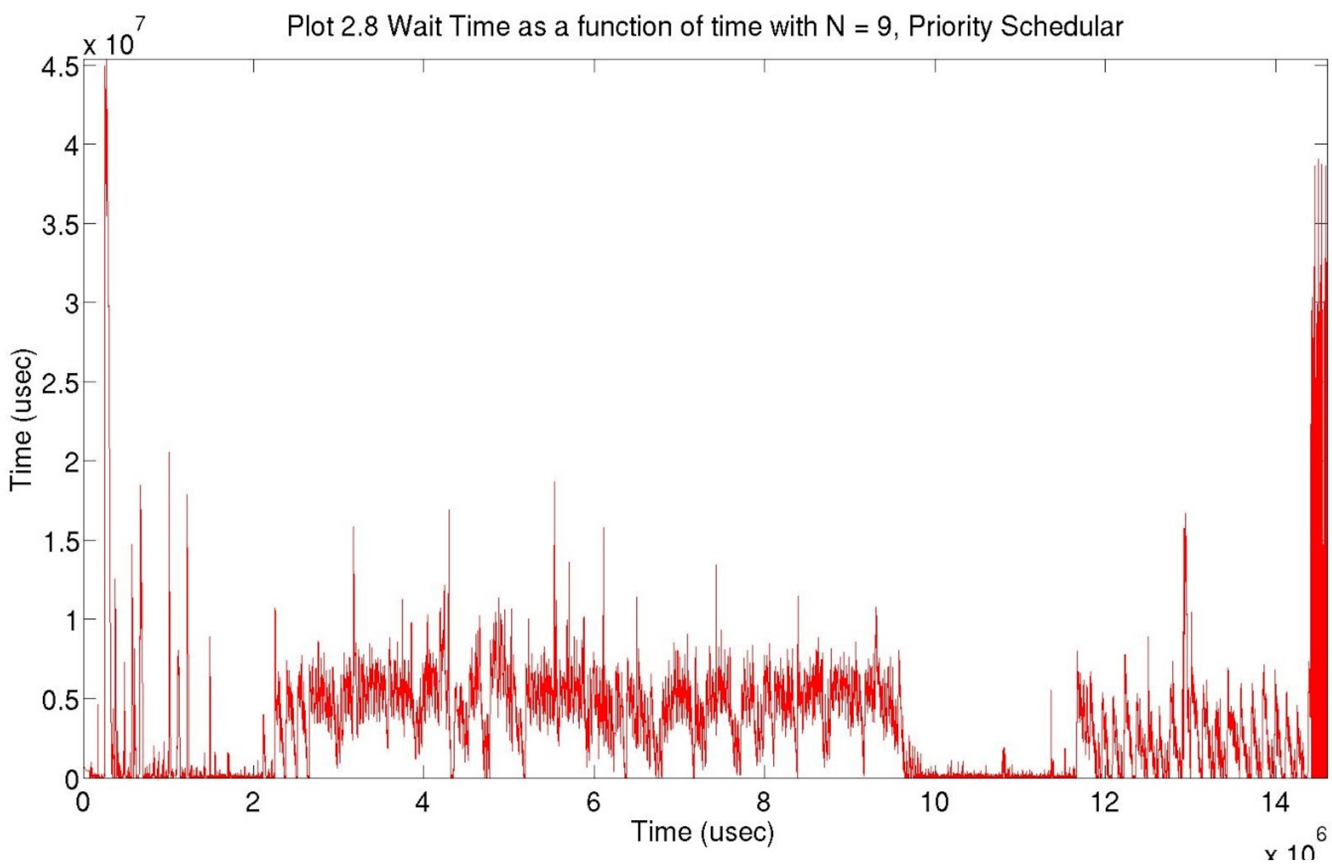


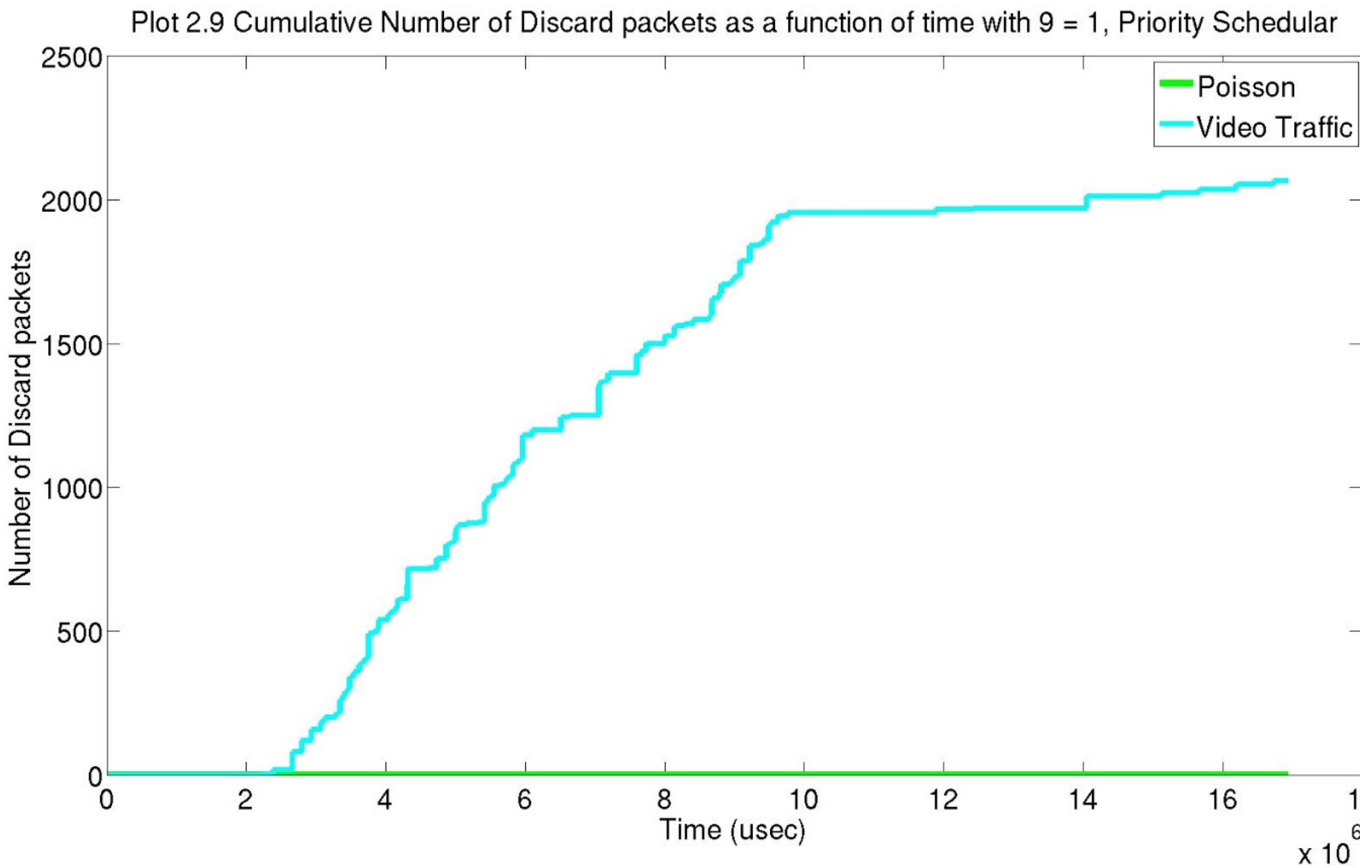

Plot 2.5 Wait Time as a function of time with N = 5, Priority Schedular
x 10^7
3
2.5
2
1.5
1
0.5
0
Time (usec)
0
2
4
6
8
10
12
14
16
18
Time (usec)
x 10^6


Plot 2.6 Cumulative Number of Discard packets as a function of time with 5 = 1, Priority Schedular
800
700
600
500
400
300
200
100
0
Number of Discard packets
Poisson
Video Traffic
0
2
4
6
8
10
12
14
16
18
Time (usec)
x 10^6

Case3: High load, with N = 9







Observation:
In this experiment, we set the Poisson traffic as the high priority traffic, and video traffic as low priority.

With all three rates (N = 1,5,9), we were able to ensure that no high priority packets was descard. Similar with part 1, as N increases, more backlog was created. Since priority scheduler starves the low priority traffic, packet drops occurs on video data, and as N increase, more packet drops occur. Furthermore, we observed that number of dropped packets are similar when N=1 and N=5, but significantly more packets were dropped when N=9. This is because the video traffic rate is 15 Mbps, and total source traffic rate didn’t exceed the buffer link capacity (20 Mbps). However, the source traffic rate is 15+9 = 24Mbps when N =0 which exceeded buffer link capacity (20 Mbps) and resulted more dropped packets.

Comparing with exercise 1.3, we got packet drops. This is because of two reasons:

1) The FIFO queue is 100KB which is smaller than many packet size from video traffic, therefore packet drops occurred (max packet size is 600KB from video traffic).
2) The source traffic rate is bigger than buffer link capacity(20 Mbps), therefore exhausted FIFO queue and packet drops occurred. This scenario only happened to N = 9, because the total source traffic rate = 9 Mbps + 15 Mbps = 24 Mbps.

Exercise 2.4 (Optional, 5% extra credit) Starvation in priority schedulers

C= 10 Mbps

N parameters	Throughputs for each source
N1 = 5 N2 =1	N1 rate = 4.5 Mbps N2 rate = 0.9 Mbps
N1 = 5 N2 =5	N1 rate = 4.5 Mbps N2 rate = 4 Mbps
N1 = 5 N2 =9	N1 rate = 1.8 Mbps N2 rate = 8.2 Mbps

In 1.4, we had only one FIFO, so higher traffic rate source consumed more link capacity and starve low rate traffic. In this case, the scheduler will alway serve N2 before N1, so when N2 has higher traffic rate, it will stave N1.

Please see code in b2/ directory.

Part 3. Weighted Round Robin (WRR) Scheduler

Exercise 3.1 Build a WRR scheduler

In this experiment, we built a WRR scheduler with three poisson sources.

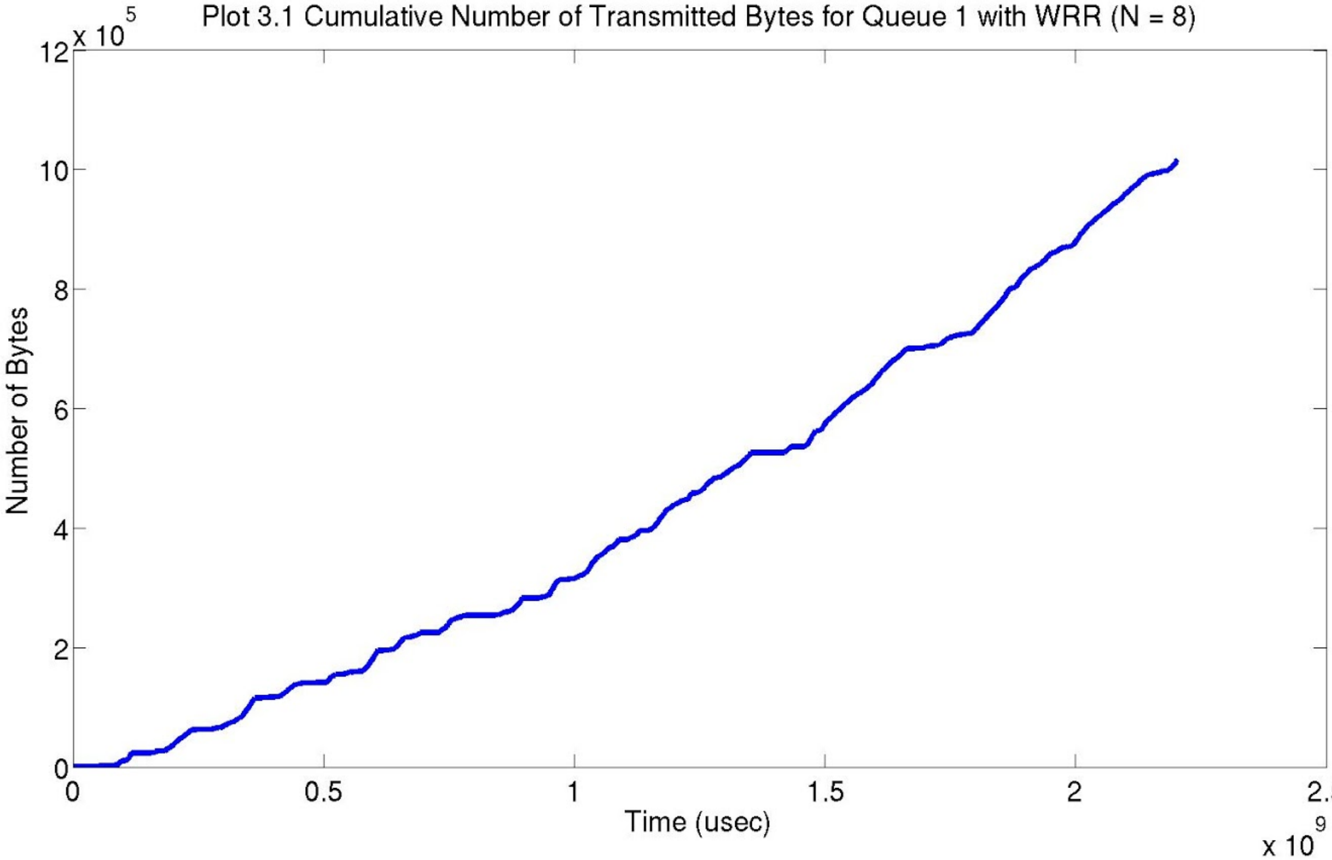
Exercise 3.2 Evaluation of a WRR scheduler: Equal weights

N=8 in for the first source, N=6 for the second source and N=2 for the third source, resulting in traffic arrival rates of 8 Mbps, 6 Mbps, and 2 Mbps.
Weights of the queues: w1 = w2 = w3 = 1.
By running WRR algorithm, the calculated packet_per_round is one for all queues and ratio is **1:1:1**. In other word, each queue sent only one packet each round. On the other hand, the calculated GPS ratio is **4:4:2.**
In theory, as N increases, less time will be used to send same amount of packets. Our plots matched this theoretical expectation:
N = 8 total time = 2.25 x 10^9 usec
N = 6 total time = 2.6 x 10^9 usec
N =2 total time = 5.2 x 10^9 usec

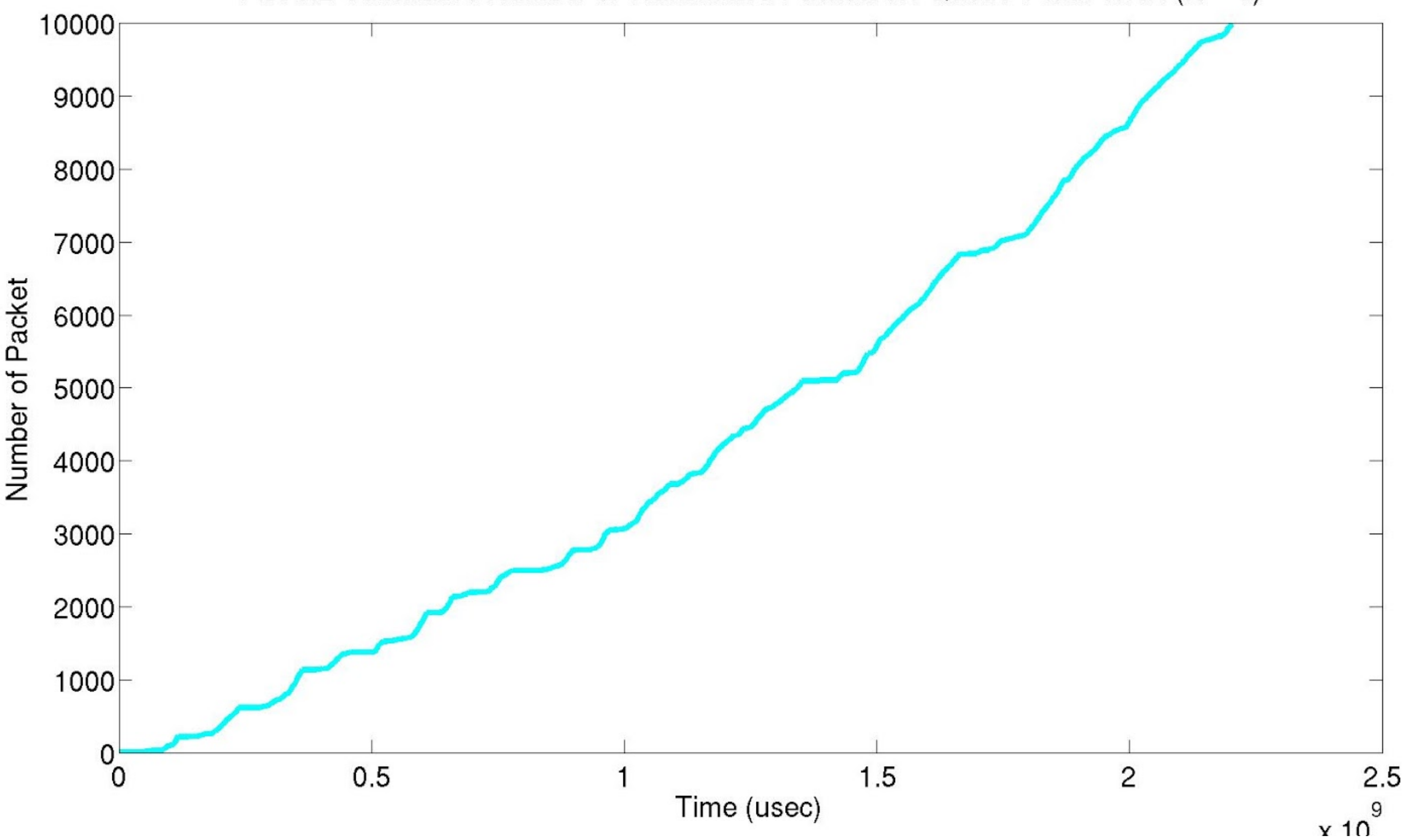
For PS, since the scheduler always send packets from the higher priority queues, therefore we expect to see “flat period” in queue 2 and queue 3 (N= 6 and N=2) plots. The “flat period” is when no packets are transmitted by the scheduler because the scheduler is serving higher priority queues. In contrast, the WRR sent 1 packet per round in this case, therefore no “flat period” in our plots.

Source 1: N=8

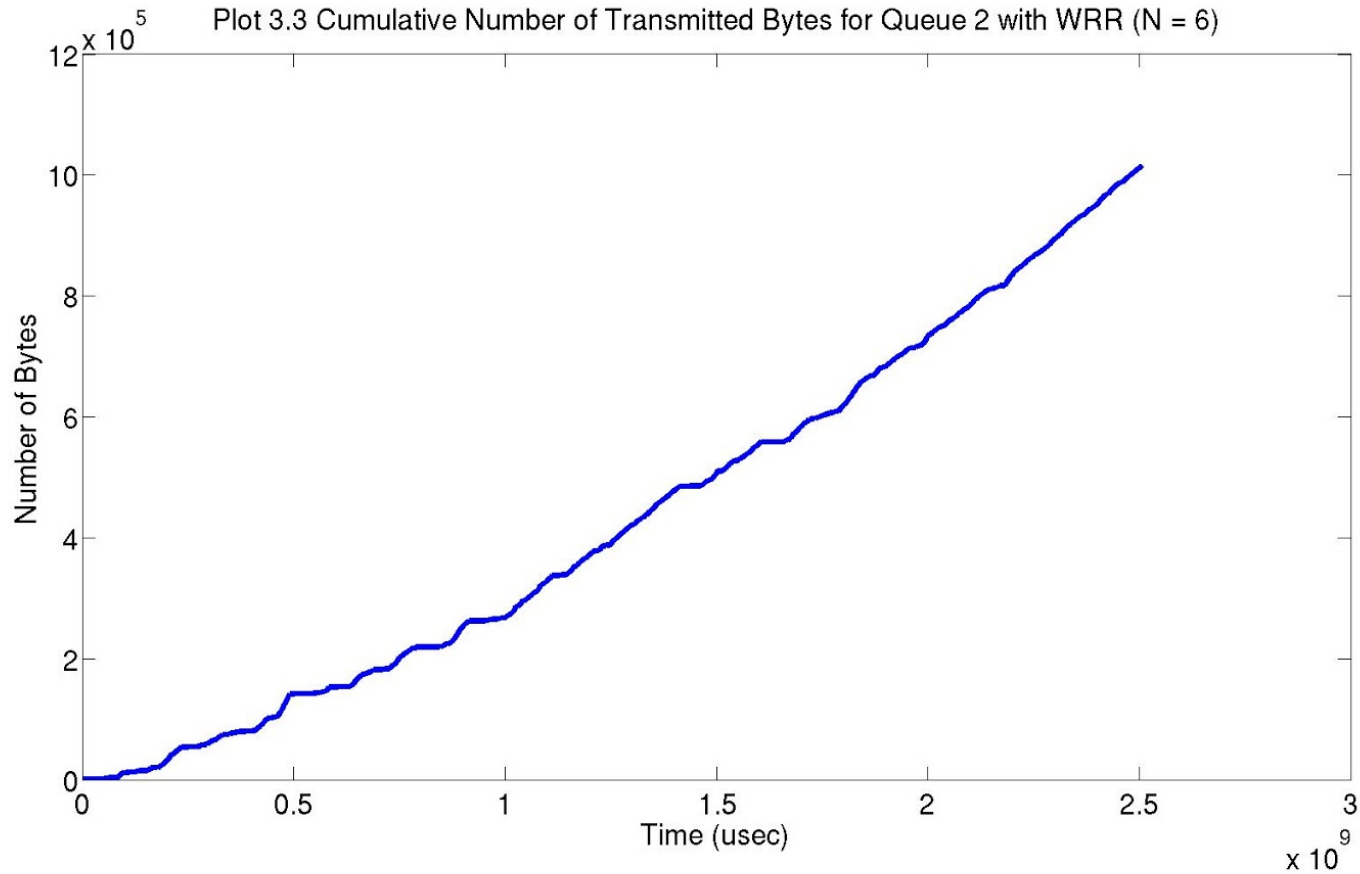
Plot 3.1 Cumulative Number of Transmitted Bytes for Queue 1 with WRR (N = 8)



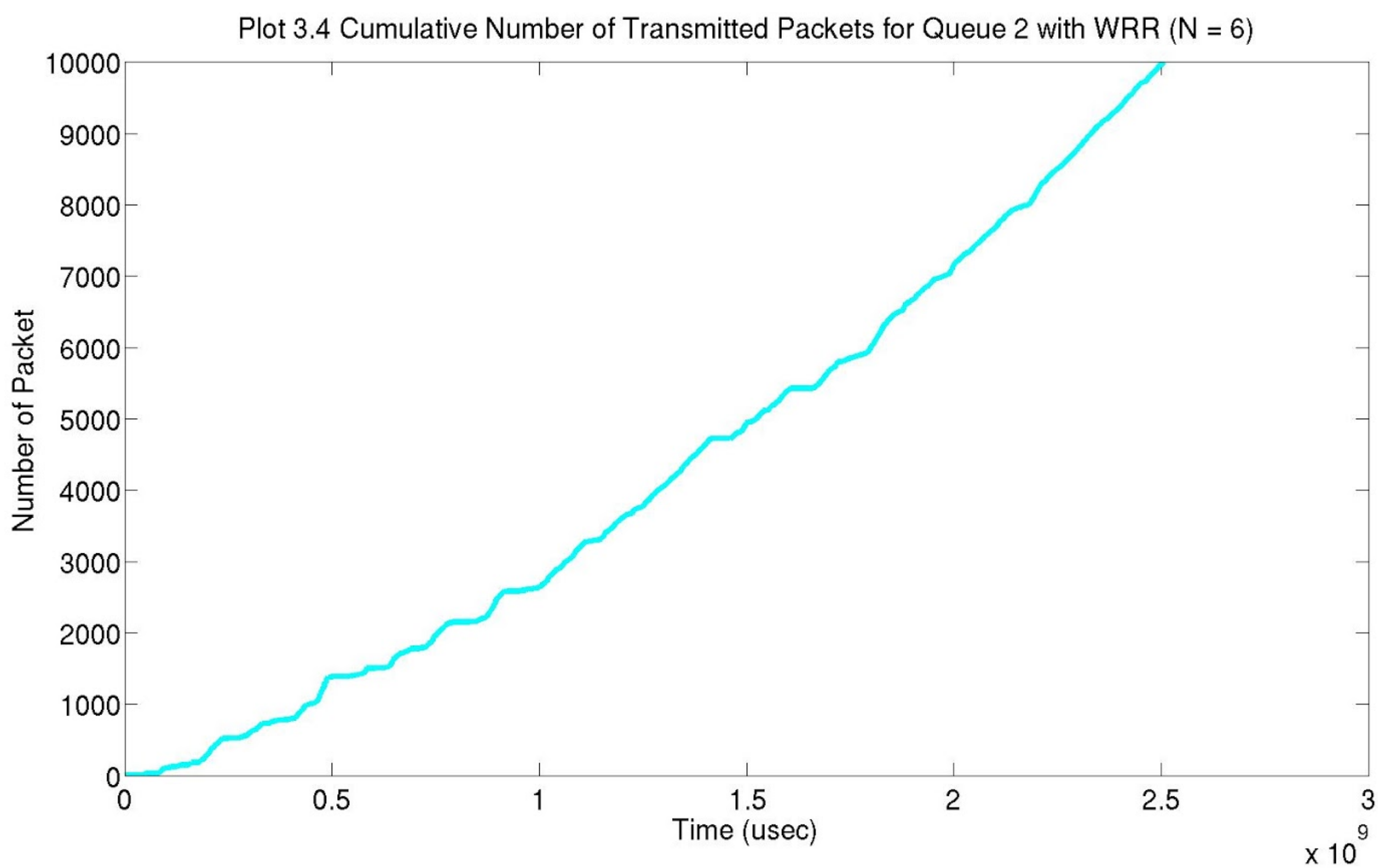
Plot 3.2 Cumulative Number of Transmitted Packets for Queue 1 with WRR (N = 8)



Source 2: N = 6

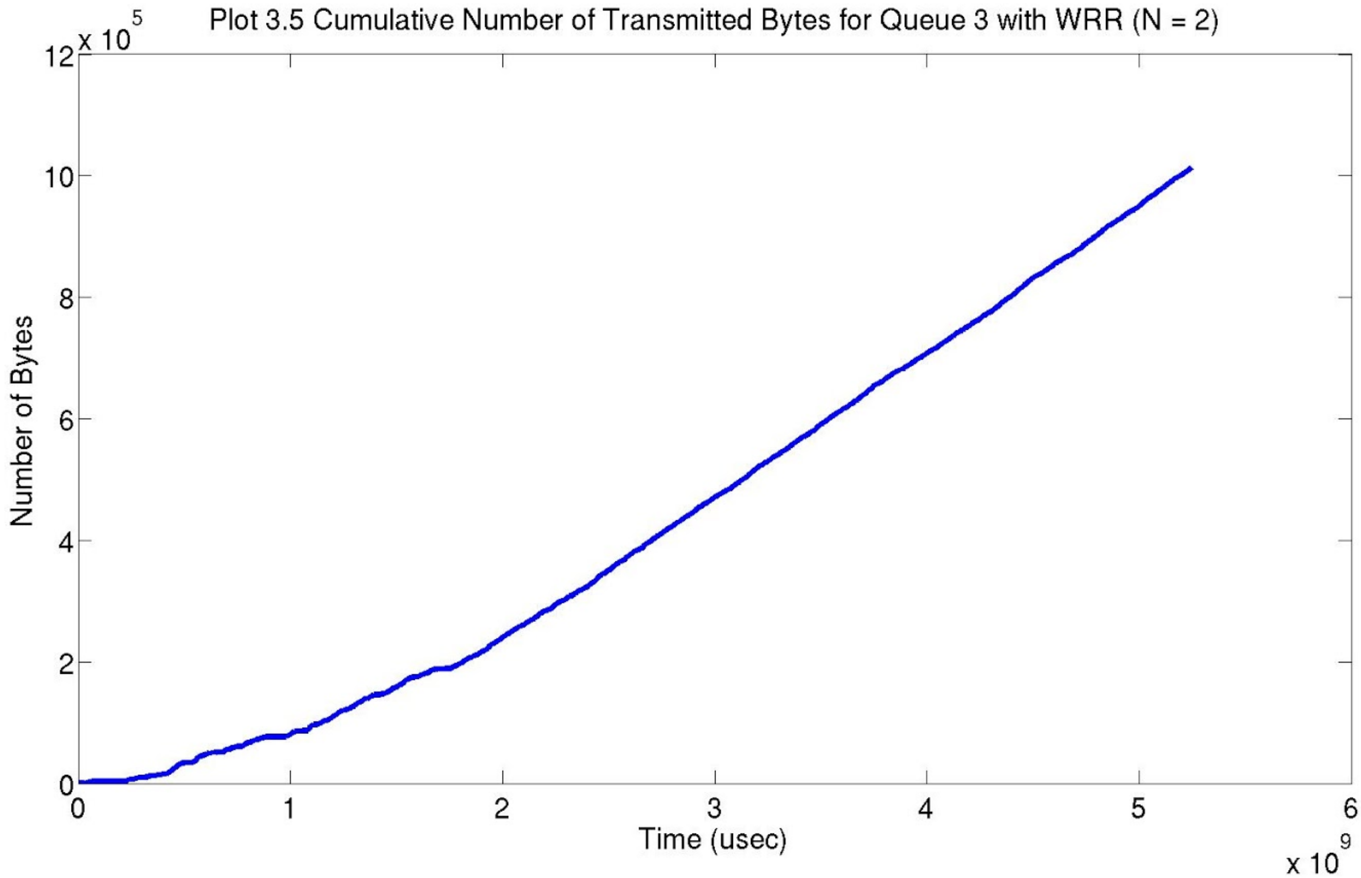


Plot 3.3 Cumulative Number of Transmitted Bytes for Queue 2 with WRR (N = 6)
x 10^5
12
10
8
6
4
2
0
Number of Bytes
0
0.5
1
1.5
2
2.5
3
Time (usec)
x 10^9

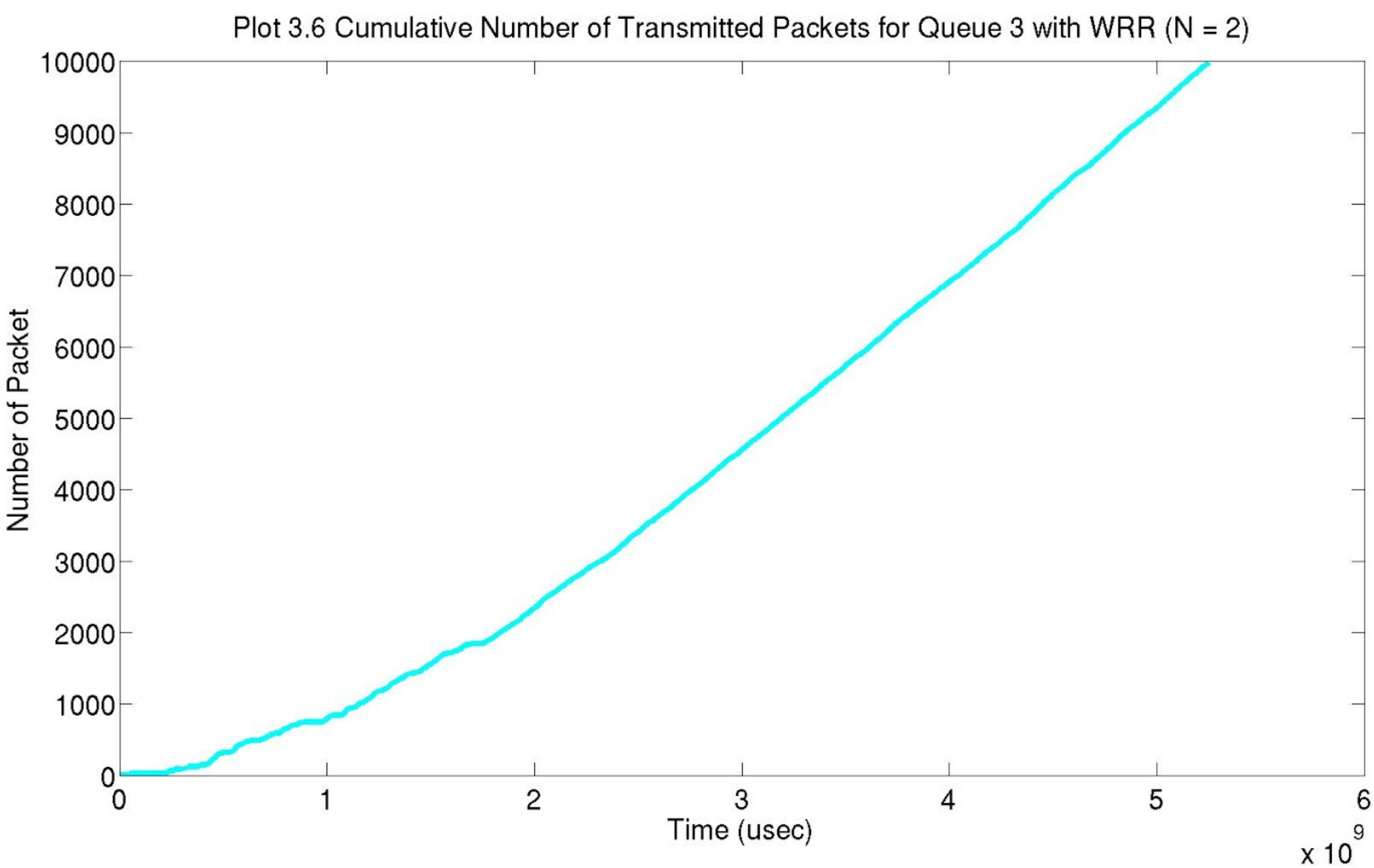


Plot 3.4 Cumulative Number of Transmitted Packets for Queue 2 with WRR (N = 6)
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Number of Packet
0
0.5
1
1.5
2
2.5
3
Time (usec)
x 10^9

Source 3: N = 2



Plot 3.5 Cumulative Number of Transmitted Bytes for Queue 3 with WRR (N = 2)
x 10^5
12
10
8
6
4
2
0
Number of Bytes
0
1
2
3
4
5
6
Time (usec)
x 10^9



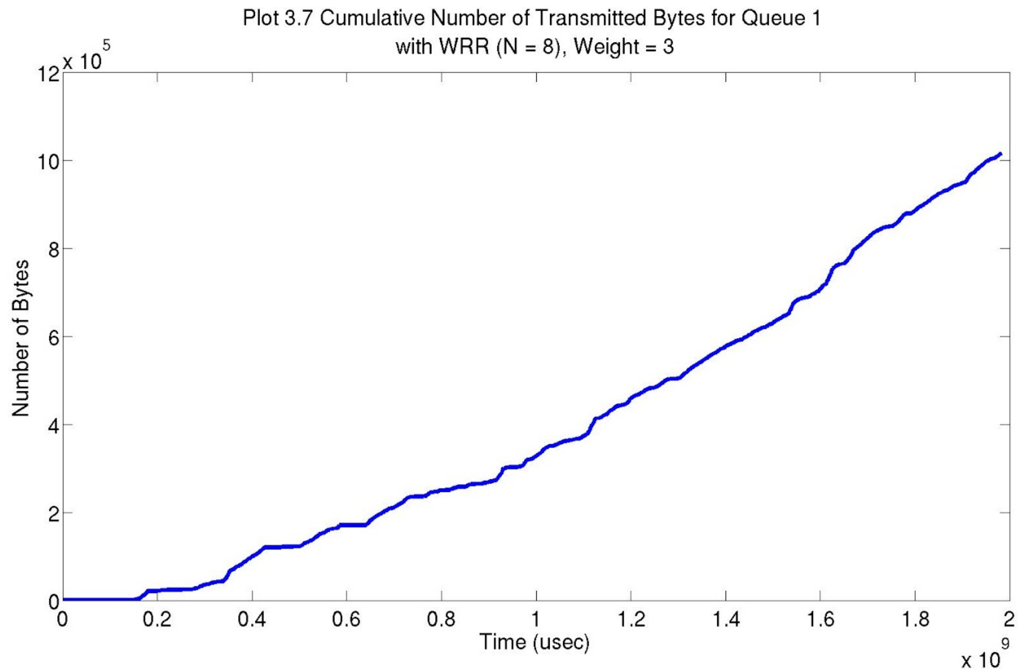
Plot 3.6 Cumulative Number of Transmitted Packets for Queue 3 with WRR (N = 2)
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Number of Packet
0
1
2
3
4
5
6
Time (usec)
x 10^9

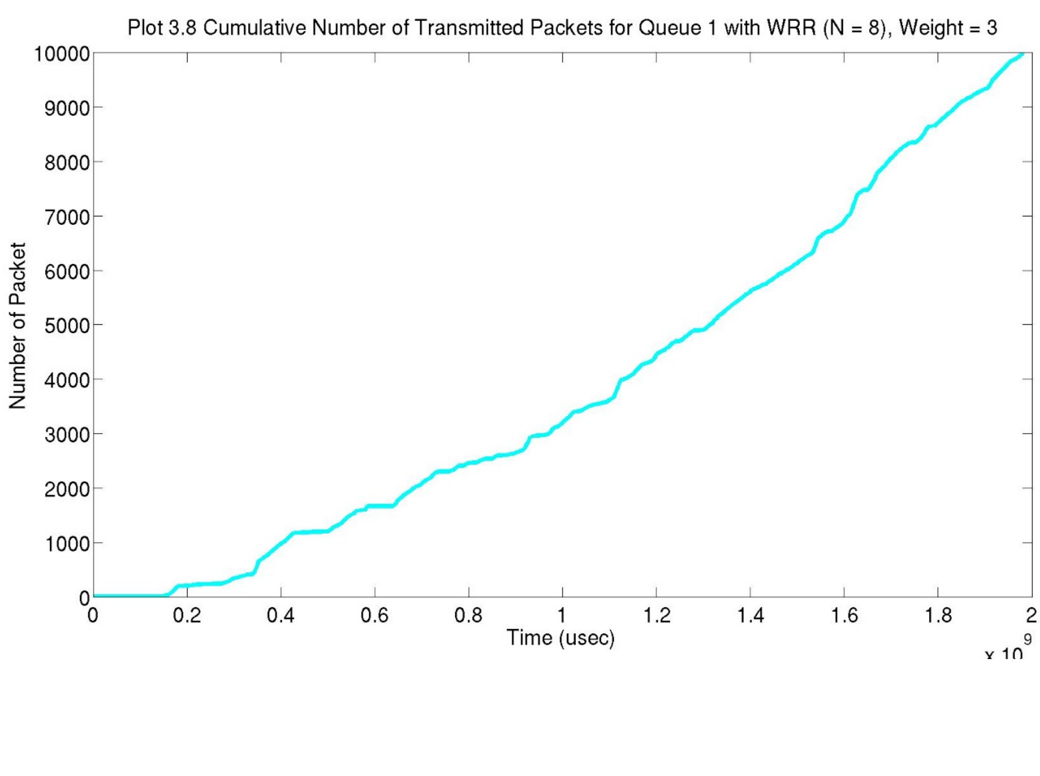
Exercise 3.3 Evaluation of a WRR scheduler: Different weights

N=8 in for the first source, N=6 for the second source and N=2 for the third source. Resulting in traffic arrival rates of 8 Mbps, 6 Mbps, and 2 Mbps.
Weights of the queues to w1=3 and w2=w3=1.

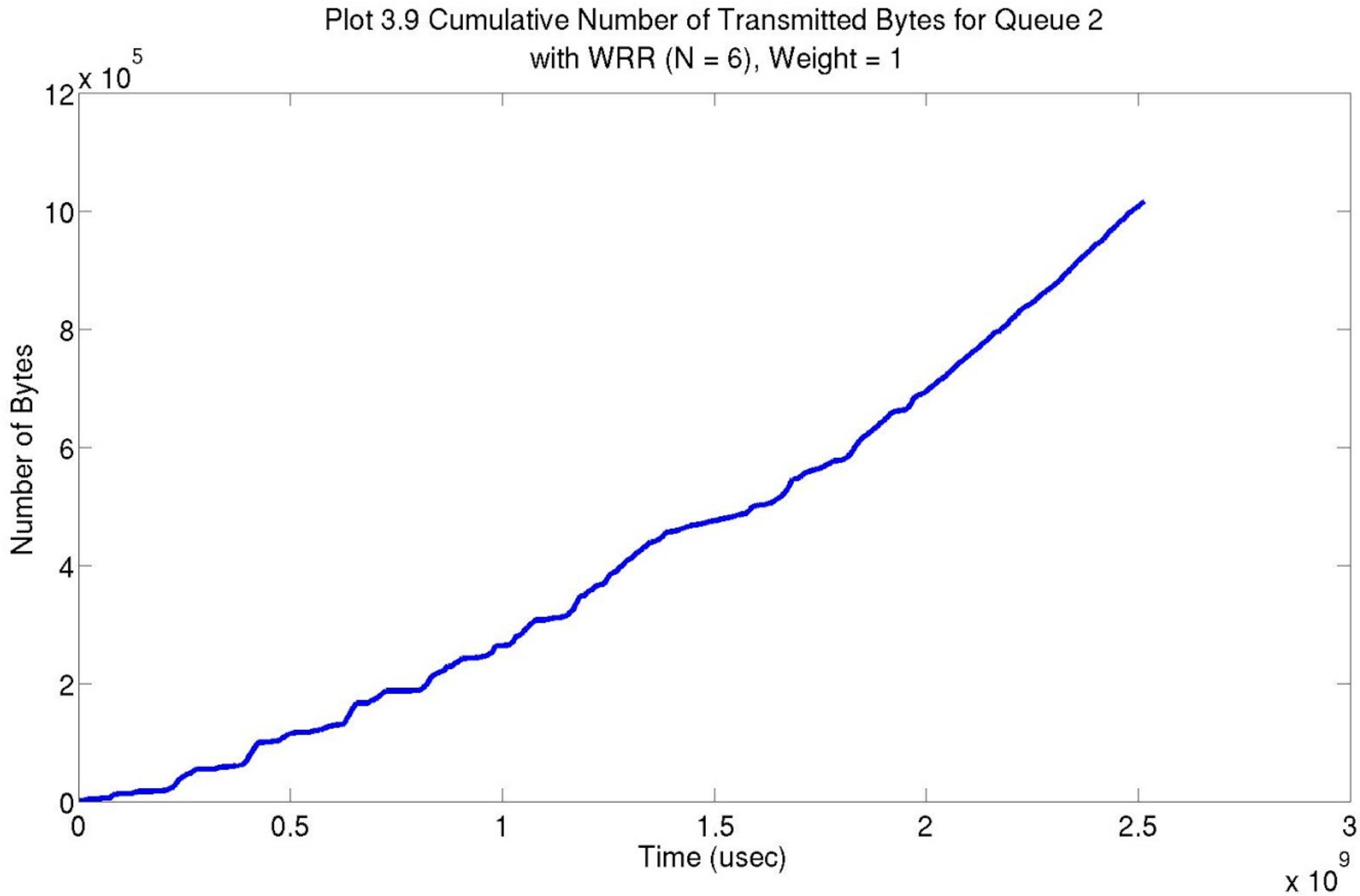
By running WRR algorithm, the calculated packet_per_round is three for queue 1 (N=8), and one for the rest two queues and ratio is **3:1:1**. In other word, queue 1 (N=8) sent three packets each round while the other two queue sent one packet respectively each round. On the other hand, the calculated GPS ratio is **6:2:2**. Although GPS sends more packets in each round, the ratio is same as WRR ratio which is 3:1:1. Therefore, we expect the GPS plots have the similar shape to WRR plots in this case.

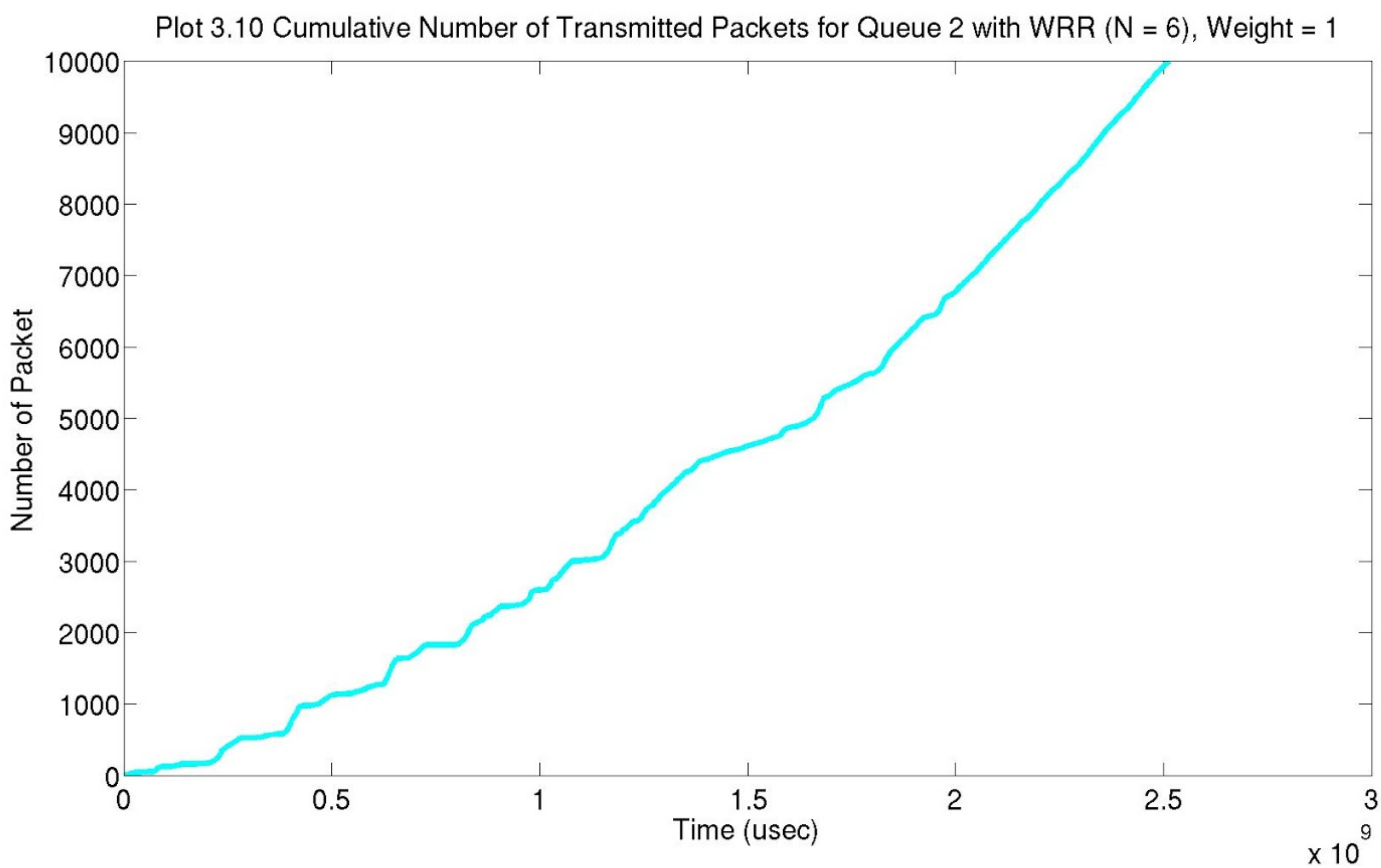
Source 1: N = 8



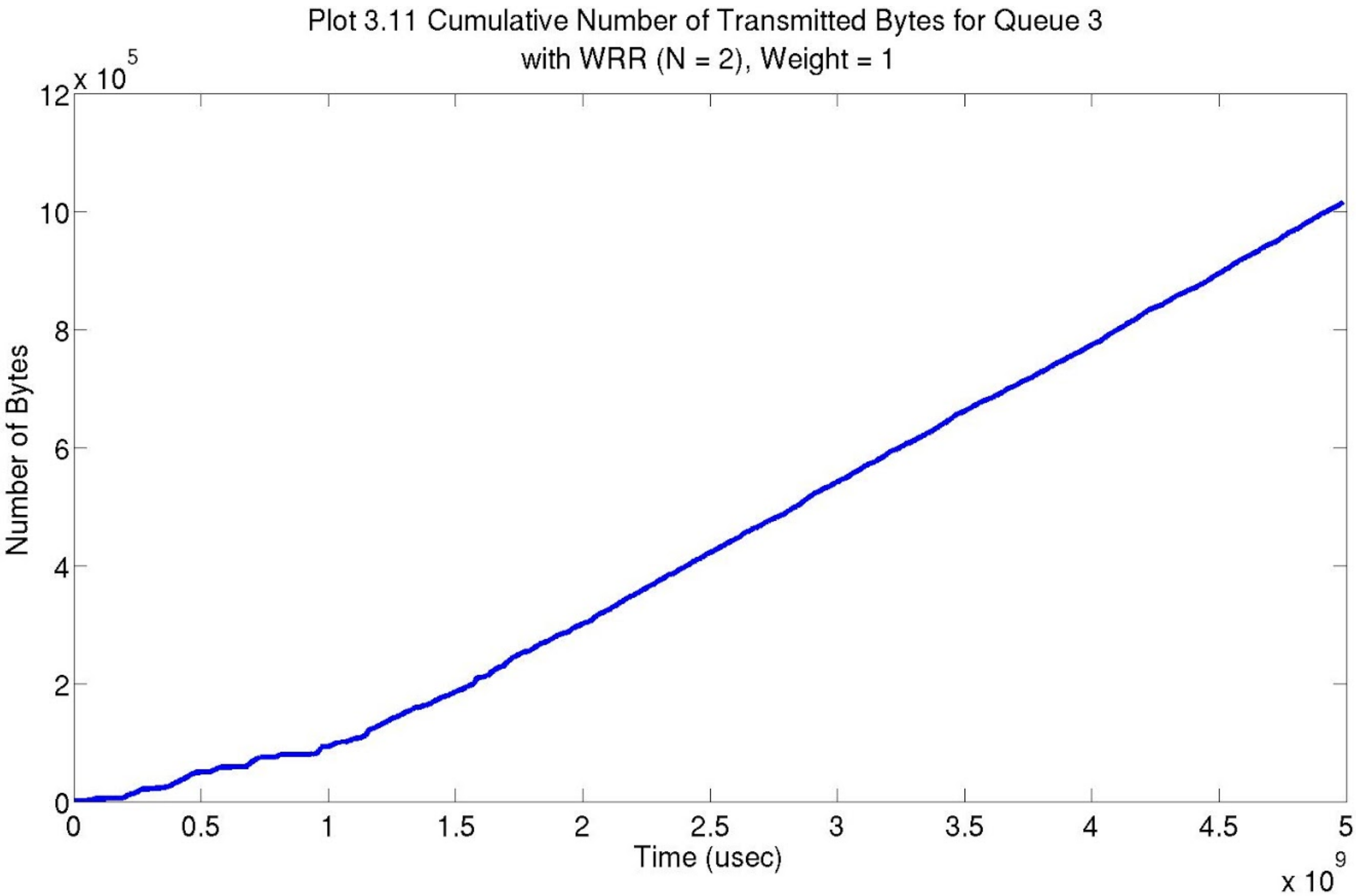


Source 2: N = 6

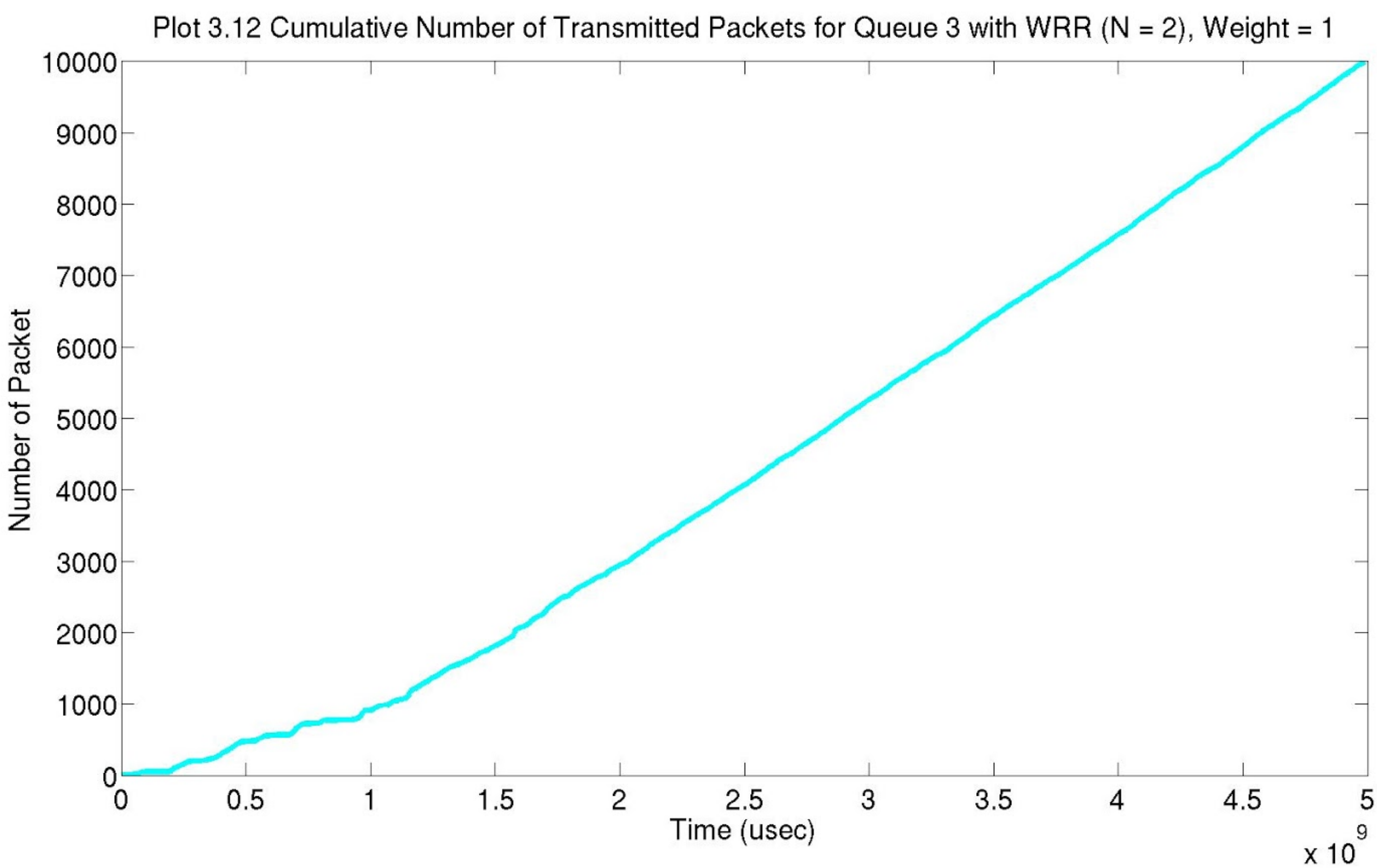

Plot 3.9 Cumulative Number of Transmitted Bytes for Queue 2
with WRR (N = 6), Weight = 1
x 10^5
12
10
8
6
4
2
0
Number of Bytes
0
0.5
1
1.5
2
2.5
3
Time (usec)
x 10^9


Plot 3.10 Cumulative Number of Transmitted Packets for Queue 2 with WRR (N = 6), Weight = 1
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Number of Packet
0
0.5
1
1.5
2
2.5
3
Time (usec)
x 10^9

Source 3: N = 2



Plot 3.11 Cumulative Number of Transmitted Bytes for Queue 3
with WRR (N = 2), Weight = 1
x 10^5
12
10
8
6
4
2
0
Number of Bytes
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Time (usec)
x 10^9



Plot 3.12 Cumulative Number of Transmitted Packets for Queue 3 with WRR (N = 2), Weight = 1
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Number of Packet
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Time (usec)
x 10^9

Exercise 3.4 (Optional, 5% extra credit) No Unfairness and no Starvation in WRR

C= 10 Mbps

N parameters	Throughputs for each source
N1 = 5 N2 =5	N1 rate = 4.5 Mbps N2 rate = 4.5 Mbps
N1 = 5 N2 =9	N1 rate = 4.5 Mbps N2 rate = 4.7 Mbps
N1 = 5 N2 = 15	N1 rate = 4.4 Mbps N2 rate = 5 Mbps

Since the two sources have the same average length and weight, the packet_per_round was calculated to be one. In other word, the scheduler sent one packet from each FIFO queue per round. Therefore, we got almost same throughputs for both sources for all cases. Comparing to 1.4 and 2.4, WRR doesn't starve any queue.

Please refer to code in b3/