LABORATORY 2

Gavin Austin (3279166) & Ayden Khairis (3282229)

SECTION I

Introduction:

Our objective for Laboratory 2, undertaken on Friday 4th September, was to study the behaviour of various queuing systems including M/D/1, M/M/1, and M/M/1/K. These systems were being studied to show different input traffic loads and traffic patterns and to investigate the effect of buffer size selection on the traffic throughput in an M/M/1/K queue model.

We also discovered the effect of queue size on traffic quality of service and the effect of a random number in the M/M/1 model.

This lab required calculations to figure out the load of the system, service time, and/or the interarrival time required to examine these different queuing models.

Basic Description:

The simulation model was made up of 3 main files. These files include the:

- omnetpp.ini file
- fifioNet.ned file
- General.anf file

Figure 1 shows the omnetpp.ini file code and is the main driver of the simulation models. Different variables had to be changed in order to simulate different queuing systems. For example, to simulate M/M/1, the inter-arrival time needed to be changed for differing load values and to simulate M/D/1, the "exponential" before packet length needed to be deleted. Also, the queue length needed to be infinite for M/D/1 and M/M/1 and needed to be 10 for M/M/1/K (where K is 10). The seed set was set to 100 as a default and was changed twice during some simulations to change the random number generator associated with the M/M/1 queuing system.

Figure 2 shows the fifoNet set up. This is the network description of the First In First Out network which was used to help illustrate the network simulations whilst they were running.

The General.anf file is used to collect the scalar and vector data gathered from the simulations. This data was collected, exported and graphed in the results section of this lab report.

```
[General]
network = elec3500.simulations.FIFO.fifoNet
result-dir = results  #folder where results are recorded.
sim-time-limit = 600s
seed-set = 100          #seed set value
#record-eventlog = true

**.gen.sendIaTime = exponential(0.2s)  #packet inter-arrival time
**.gen.packetLength = exponential(1000b)
**.fifo.serviceRate = 10000bps   #service rate of the fifo
**.fifo.queueLength = 10        # queue length. put -1 for an infinite queue
```

Figure 1

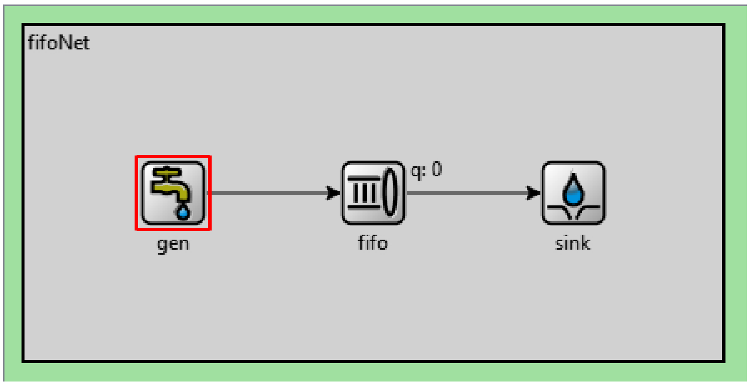


Figure 2

Results:

Figure 1:

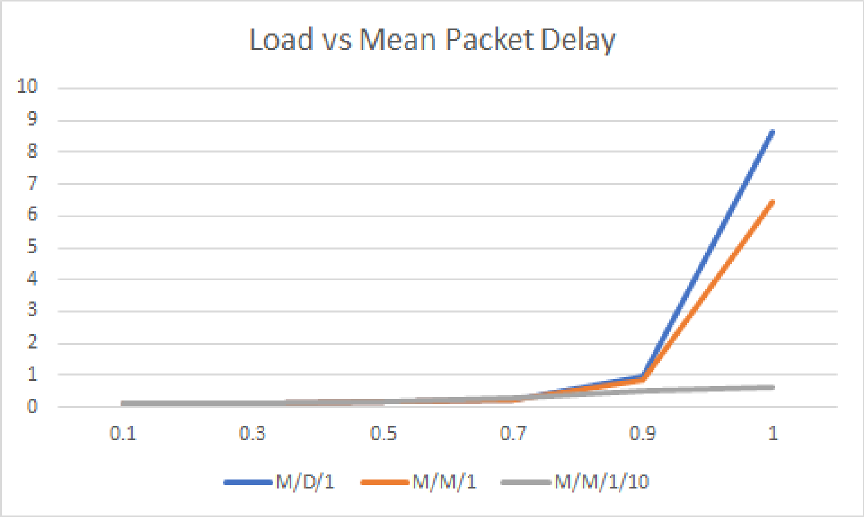


Figure 2:

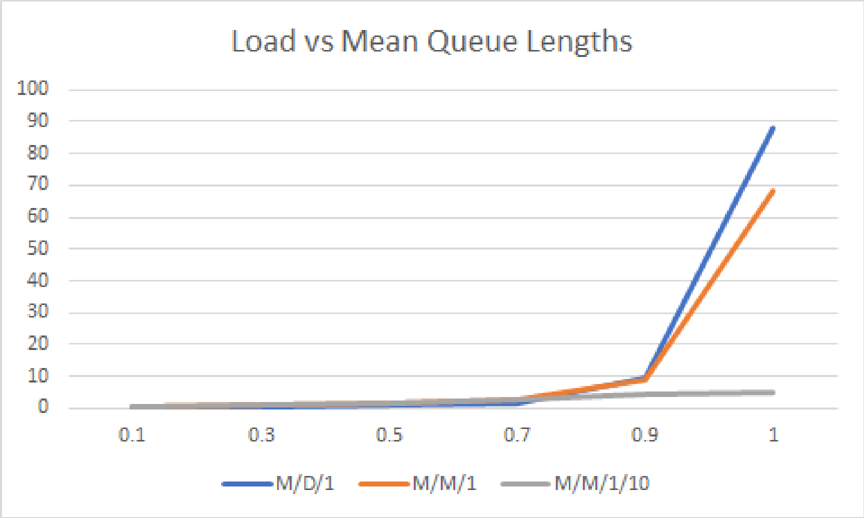


Figure 3:

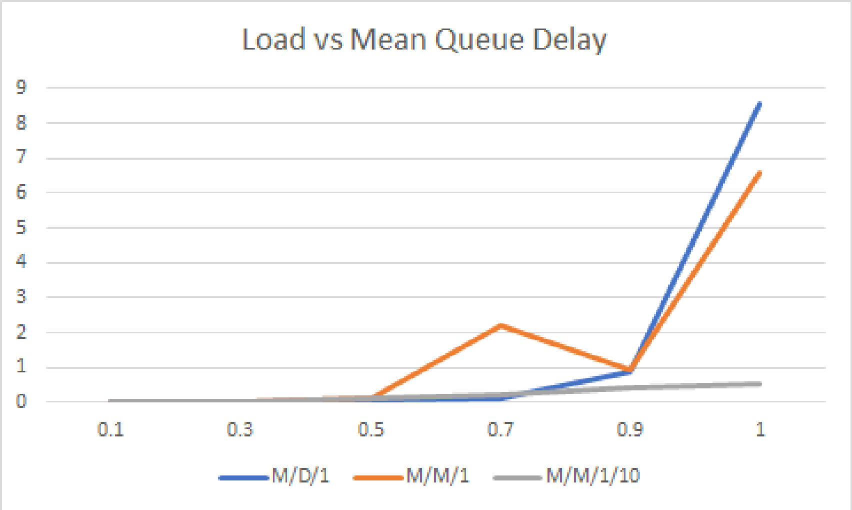


Figure 4:

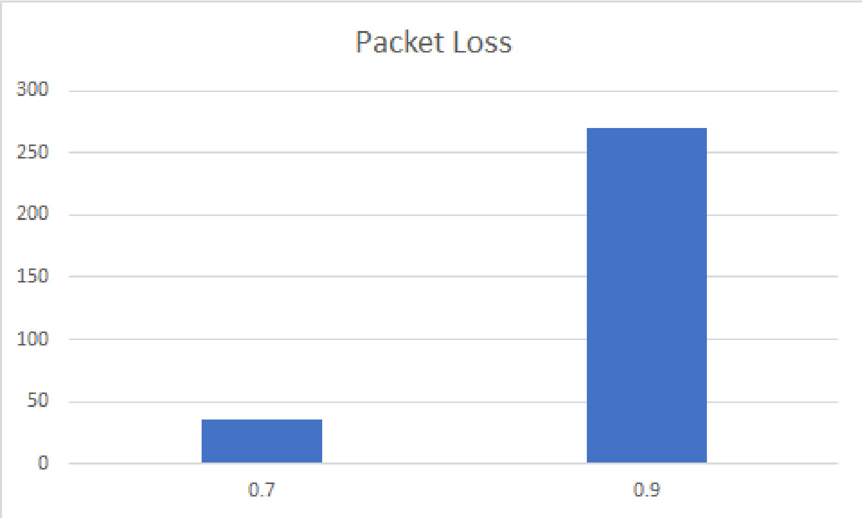
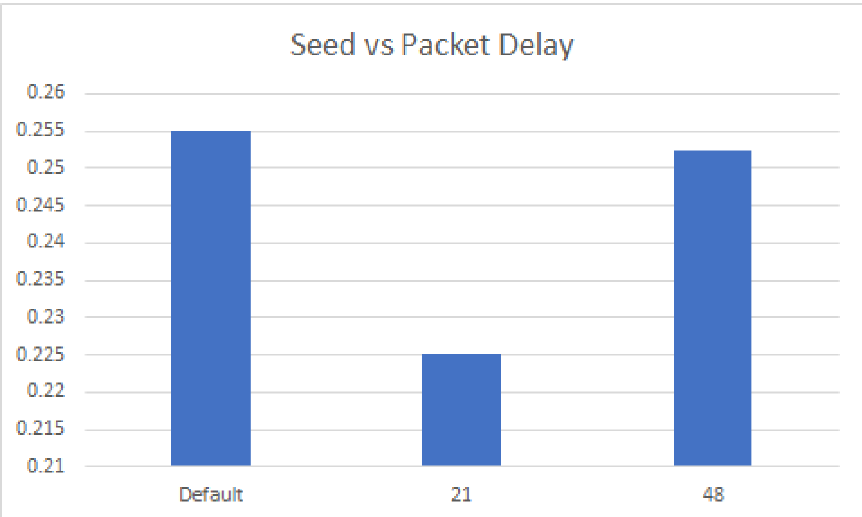


Table 1:

Queue Length	Packets Lost	End to End Delay (S)
0	2910	0.097985159
5	773	0.333959354
10	443	0.583037328
15	276	0.790093926
20	186	0.967071117
25	142	1.159477414
30	113	1.332752261
35	88	1.489581444
40	64	1.628844238
45	52	1.66825309
50	46	1.703383817
55	33	1.723503484
60	29	1.738635878
65	19	1.746799552
70	16	1.751308982
75	11	1.757468088

Figure 5:



Analysis:

When looking over figures 1, 2, & 3, we can see that delays caused by the loads tend to be similar. When experiencing low loads, M/M/1 & M/D/1 systems act very similar to M/M/1/10 systems (systems with a limited queue). This is because the incoming traffic is below the critical point that causes the data to 'back up', e.g. output < input. This can be modelled well with the probability of overflow equation for a M/M/1/K system. However, at bigger loads, the flow in and out of the system are not equal, and the incoming packets get added to the queue. For the M/M/1/10 simulation, this causes an overflow of the Queue, and packets get dropped. As packets enter the queue, their wait time grows relative to the total number for packets in the queue. This causes the gap we see between M/M/1, M/D/1 & the M/M/1/10 lines in figure 3. Instead of the delay growing, the packets are dropped and do not affect Queue delay.

SECTION II

Question 1:

M/M/1/10

$$P(loss) = \frac{(1-p)*p^{K}}{1-p^{K+1}} \rightarrow \frac{(1-0.5)*0.5^{10}}{1-0.5^{10+1}} \ \& \ \frac{(1-0.7)*0.7^{10}}{1-0.7^{10+1}}$$

$$For\ p = 0.5, \qquad P(loss) = \frac{0.5}{0.9995} \times \frac{1}{1024} \rightarrow \frac{125}{255872}$$

$$For\ p = 0.7, \qquad P(loss) = \frac{21613}{2500000}$$

$$E[n]\ for\ p = 0.5, \quad E[n] = \frac{0.5}{0.5} - \frac{(10+1)\times 0.5^{11}}{1-\ 0.5^{11}} \ \rightarrow \ \frac{2036}{2047}$$

$$E[n]\ for\ p = 0.7, \quad E[n] = \frac{0.7}{0.3} - \frac{(10+1)\times 0.7^{11}}{1-\ 0.7^{11}} \ \rightarrow \ 2 + \frac{6276026}{56317619}$$

$$E[T]\ for\ p = 0.5, \quad E[T] = \frac{\frac{2036}{2047}}{(0.5\ \times 10)\left(1-\frac{125}{255872}\right)} = \ 0.199022484\ seconds$$

$$E[T]\ for\ p = 0.7, \quad E[T] = \frac{2+\frac{6276026}{56317619}}{(0.7\ \times 10)(\frac{21613}{2500000})} = \ 0.30426469\ seconds$$

M/M/1

$$E[T] = \frac{0.5}{0.5} \times \frac{0.5}{(0.5*10)} = 0.1\ for\ p = 0.5$$

$$E[T] = \frac{0.7}{1-0.7} \times \frac{0.7}{(0.7*10)} = 0.233\ for\ p = 0.7$$

Question 2:

The values obtained in testing are extremely close to the theoretical values acquired in Question 1. For M/M/1/10 load value 0.5, the theoretical delay was 0.19902 seconds, and for load value 0.7, the theoretical delay was 0.30426 seconds. In our experiment, we calculated the value of 0.200 seconds for load value 0.5 which is extremely close to the theoretical value. For load 0.7, the difference between theoretical and calculated values are also extremely close. This can be put down to the randomness that a system simulation introduces, that the theoretical values cannot completely mimic. Figure 5 shows the effect that RNG can create, with delays varying up to 0.05s, 5x that of the difference between our theoretical and physical measurements

In the M/M/1 simulations, the expected value for load 0.5 was 0.1 seconds, and for load 0.7 was 0.233 seconds. We measured 0.102 seconds and 2.2 seconds respectively which means the delay for load 0.7 is a very large outlier. Question 3 looks further into the variance and randomness we encountered during our experiment.

Question 3:

Observing the results, we can see that M/D/1 and M/M/1 have very similar results at low to medium load levels. As the load increases, from the 0.7 to 0.9 mark, there is a small increase for both queuing systems and at the 0.9 to 1 mark, both queuing systems shoot up with M/D/1 increasing at a higher rate than M/M/1. Theoretically, M/M/1 is meant to perform worse than M/D/1 but this has not been the case given our results. A possible explanation for this could be the variance for these systems were different during the simulations. It would be difficult to process a variance difference in the early loads as they should always be very close to each other, but at the higher loads, the difference between queuing systems should always be very evident. Looking at the mean queue delay for both systems, its evident variance was present at load = 0.7 because M/M/1 delay was a lot greater than the expected value of the system at that load value.

Question 4:

The random number for different seed values adds more variance to the simulations as the seeds are meant to simulate different traffic and network operating conditions. In our case, we used seed values 21, 48 and the default value which is 100. Given our results, the packet delay distribution is higher when the seed number is greater. A big seed number means there exists the possibility for the variance to be greater than that of a small seed number and the results are proof of that.

Question 5:

The two biggest offenders in regard to Quality of Service in networks is long delays and no responses. These two factors can negatively impact a user's experience and the network itself. In order to select the correct queue size for the network, we need to consider these two factors. Having a small queue will result in a small delay from both ends which overall creates a faster connection. This can also create bad Quality of Service as it is unreliable, making the possibility for a lot of packets to be dropped. A large queue causes less dropped packets but also creates a larger wait time. Ideally, a balance between these two would create the perfect queue size, which is impossible. Table 1 shows this relationship between long delays and no responses.

Question 6:

The changes I could make would include using different queuing systems to reduce the probability of packet losses and high end-to-end delay. Using an M/M/1/K system would reduce the probability of high end-to-end delay as K packets are processed which can result in low end-to-end delay times. To reduce the probability of packet loss, I would increase the K value in the M/M/1/K queuing system. Table 1 shows how increases to the queue length result in less packets to be dropped and vice versa. Finding a good middle ground can combat both packet loss and high end-to-end delay. For example, M/M/1/35 in Table 1 has low number of packets dropped and a decent end-to-end delay time.