Management and content delivery for smart network

Davide Macario s315054
Alessandro Redi s310471
Federico Volponi s309709

April 5th 2023

Code: https://github.com/MRVSmartNetworks/management_and_content_delivery_labs

1 Report 1: Network system simulation

1.1 Introduction

During this laboratory, the output link of a router is being simulated as a queuing system. The results are evaluated in different cases, changing input parameters and configuration to understand the system response at every variation.

In the simulation, router packets are simulated by the customer's arrival rate, while transmissions are modeled as services. When a customer arrives at the queue and all the servers are busy it will wait in the waiting line, which represents the router buffer.

This model allows to make changes easily and study the queuing system under many different conditions, paying special attention to the 'limit cases', which are typically analyzed when such systems are designed.

1.2 System Performance

An interesting analysis of the system can be done by changing the arrival rate of customers. In this way, by running different simulations, it is possible to evaluate the variation of important metrics in the system, like the loss probability of packets when the queue is full.
Also, confidence intervals can be evaluated, to analyze the steady-state behaviors of the system variables.

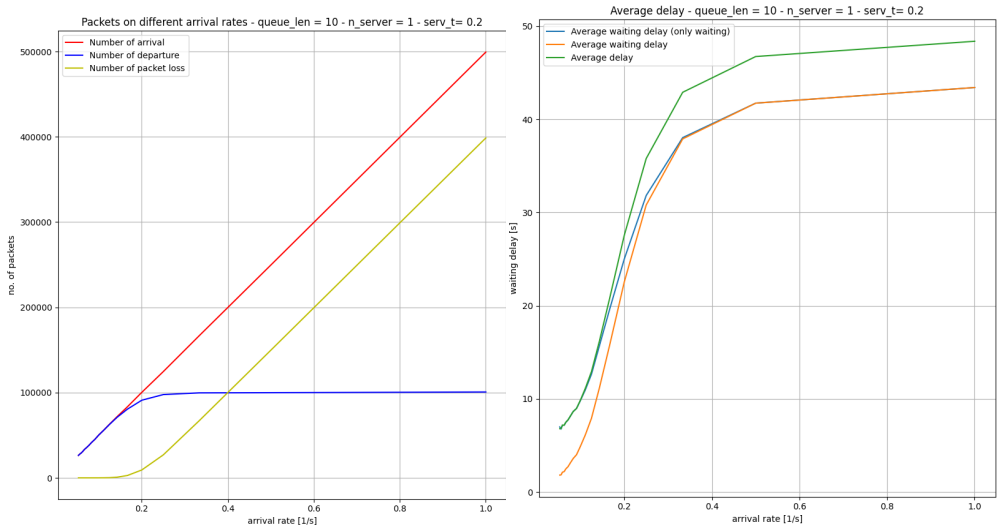
1.2.1 Case with different arrival rate

In this section the system is analyzed under a range of arrival rates from 0.05 to 1, keeping the service time value fixed.
As it is possible to see in figure 1.1a, the number of arrivals is directly proportional to the arrival

rate and, when it becomes higher than the service rate (0.2 in figure 1.1), the system is not ergodic anymore.
Image 1.1b shows the averages delay for the packets in the system. There are three major types of delay to consider;

- The green line is the total average delay of any packets from input to output
- The orange line is the average queuing delay
- The blue is the average queuing delay of packets that actually experienced some waiting delay

Looking carefully at figure 1.1b it is possible to see that the average waiting delay of the packets that actually had to wait in the queue overlaps the total average delay with a low arrival rate. This happens because there are few customers that are arriving and the majority of them are instantly served by the server. Furthermore, when the arrival rate is high the queue will be always full, then the blue line will overlap the average queuing delay, as all clients will have to wait before their service.



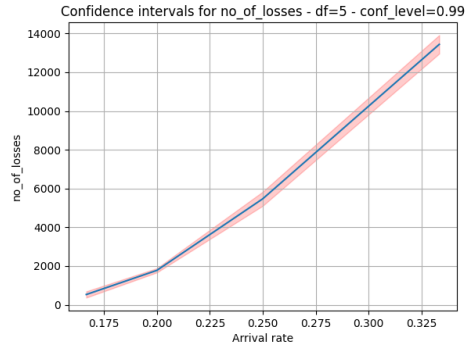
(a) Number of packets: arrived, departed, lost *(b)* Average delay of the packets in different cases

Figure 1.1: Results using one server, a queue length = 10 and a service time = 5

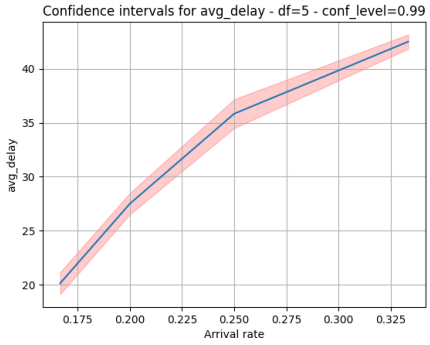
1.2.2 Confidence levels

Confidence intervals are useful tools to understand the statistical significance of obtained results since they are an estimation of the error made when the average value of a quantity is evaluated. During this analysis, a number of samples $n = 6$ and a $\alpha = 0.99$ confidence level have been chosen to find the confidence intervals; moreover, in order to obtain larger confidence intervals the simulation time has been decreased to 100000 time units. Since $n < 30$ a Student's T distribution with $n - 1$ degrees of freedom has been used, instead of a Gaussian distribution.

In figure 1.2 are reported the confidence intervals for the number of losses and for the average delay in the system varying the packets arrival rate from $\lambda = 0.143$ to $\lambda = 0.33$. If the number of samples n and the simulation time are increased then as a consequence the confidence intervals



(a) Confidence interval for the number of losses



(b) Confidence interval for the average delay

Figure 1.2: Results using one server, a queue length = 10 and a service time = 5 with a 99% confidence level and a number of iterations $n = 6$

will be narrower, as can be seen from formula 1, where $\bar{x}$ is the mean of the observations, s is the sample standard deviation and z is the t-score.

$$I = (\bar{x} - z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{n}}, \bar{x} + z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{n}}) \tag{1}$$

1.2.3 Comparison with theoretical results

Comparing the simulated behavior of the system with respect to the theoretical result it is easy to see that the achieved results confirm the expected ones. For example, regarding system stability, the load must satisfy $\rho < 1$, and in particular for M/M/1 queue $\lambda < \mu$. In the simulation shown in figure 1.1 the ergodicity is achieved for an arrival rate lower than 0.2 s^{-1}; for example, by looking at the plot in figure 1.1, the number of departures remains almost constant while the number of losses increases after surpassing the stability condition, which means that the simulation is working as expected.

1.3 Multi-server system

The following analysis will consider the case of a multi-server system, in particular, an M/M/2 according to Kendall's notation, highlighting the differences with the M/M/1 queue. The multi-server system will be studied on several buffer length sizes.

Note that the 'first-order' server policy is used for the following considerations, and the simulations have been run with a maximum buffer size of 10, an arrival rate of 0.17, and a service rate of 0.2.

1.3.1 M/M/1 versus M/M/2

The main difference between the two queueing systems can be seen in table 1: the multi-server one is much more reliable. Indeed, the loss probability of the M/M/1 is around 3% while on the other

case is negligible given the total number of arrived packets. Moreover, the multi-server is more efficient as can be seen by looking at the average waiting delay which is less than half compared to the single-server system. In the single-server case, the average time spent in the system is very close to the average waiting delay. This means that most of the time, a packet remains in the buffer waiting to be served. The distributions of the waiting delay are shown in figure 1.3 and 1.4: it is evident how, in the 2-server case, the waiting delay generally assumes lower values.

The better performance of M/M/2 is given by the fact that, unless one of the two servers is idle, the actual service rate is doubled since there are two servers working in parallel.

	no. losses	Average delay	Average waiting delay	Average buffer occupancy
MM1	2536	20.33	19.2	2.46
MM2	15	6.10	6.87	0.18

Table 1: Comparison between MM1 and MM2

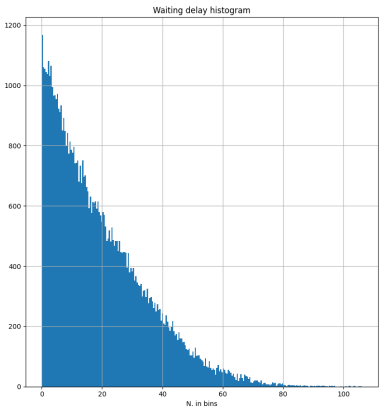


Figure 1.3: Waiting delay distribution for M/M/1

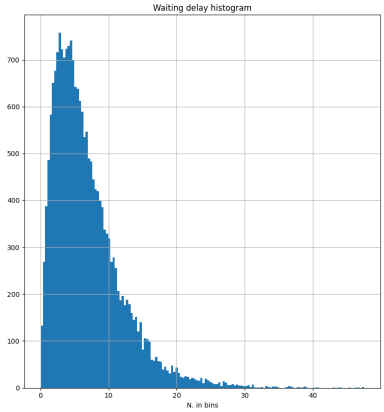


Figure 1.4: Waiting delay distribution for M/M/2

1.3.2 Case with different buffer size

In figures 1.5 and 1.6 it can be seen how the system behaves for different buffer sizes.

Regarding the number of packets in the queue shown in figure 1.5, it is easy to see the symmetric behavior of the number of losses and the number of departures: as the buffer size increases, more customers enter the waiting line without being discarded thus the number of departures grows.
Figure 1.6 shows that, as the buffer increases, more packets are able to enter the waiting line, and as a consequence, the waiting delay, which is lower for shorter queues, grows influencing the average delay of the whole system. Predictably, the average queuing delay follows the same shape as the waiting delay, and the values of the delay tend to stabilize as the queue becomes long enough, since the system is ergodic (the overall service rate considering both servers is higher than the arrival rate).

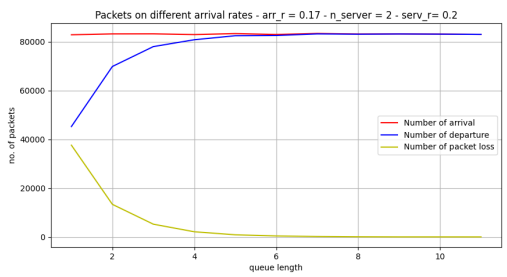


Figure 1.5: Number of packets over different buffer sizes, M/M/2

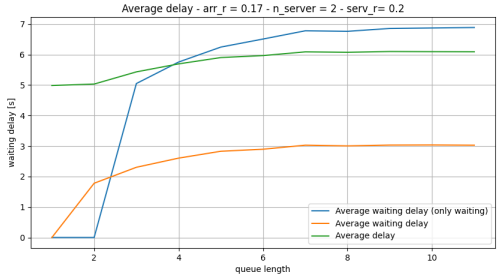


Figure 1.6: Delays over different buffer sizes, M/M/2

1.4 Load distribution techniques

An interesting analysis concerning the multi-server case is that of trying different policies for assigning the clients (packets) to each server. This can be a viable approach to model systems in which the different servers may have different capabilities in terms of service time.

The baseline scenario considered in this part consists of a queuing system composed of 5 servers, having a maximum capacity of 20 and with a fixed packet arrival rate $\lambda = 10$. The analysis has been carried out by varying the remaining parameters, namely the service rate μ and the server policy. All the different scenarios have been dimensioned to have a utilization $\rho < 1$.

1.4.1 Equal service rates, 'first idle' policy

The first considered model assigns equal service rates $\mu = 3$ to all 5 servers. The selected policy is a 'first idle' one, in which clients are assigned to the first free server (ordered according to their ID). This means that once a new packet has to be served, the transmitter which it will be assigned to will be chosen as the first one which is free, starting from server 1, up to server 5.

As a result, we can expect the utilization of the 'first' servers to be higher, as displayed in figure 1.7. By increasing further the service rate to $\mu = 8$, servers are faster, and, as highlighted in figure 1.8, it will be more likely that the first server will be idle once a new client has to be served, hence most of the packets will be assigned to the first server.

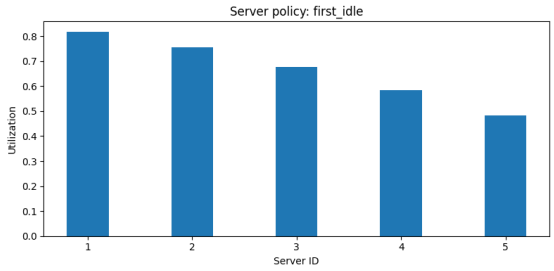


Figure 1.7: Server utilization, 'first idle', $\mu = 3$

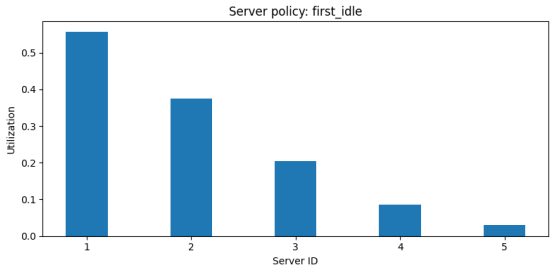


Figure 1.8: Server utilization, 'first idle', $\mu = 8$

1.4.2 Equal service rates, 'round-robin' policy

The next considered model still keeps the same service rate for all servers, but now the assignment of the different clients is done in the following way. Supposing the previous client was assigned to server i, the next packet will be assigned to server $i+1$, if free, else to server $i+2$, and so on.

This approach ensures that all servers have the 'same chance' to be assigned a client, which is the opposite of what is described in section 1.4.1.

Figure 1.9 shows how, for $\mu = 3$, the work is equally split between the different servers.

This time, by increasing the service rate to $\mu = 8$, the only change is in the average utilization of all servers. Indeed, by making each server faster, it is possible to see how the server utilization plot in 1.10 reports lower values overall, while still the different servers are active for approximately the same amount of time.

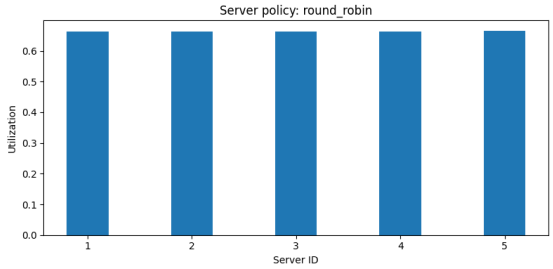


Figure 1.9: Server utilization, 'round robin', $\mu = 3$

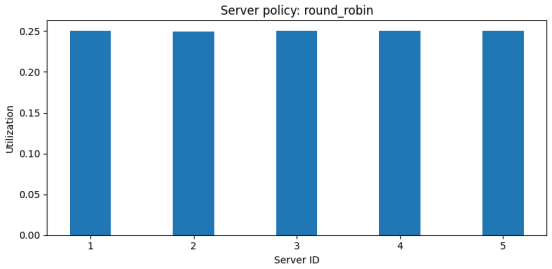


Figure 1.10: Server utilization, 'round robin', $\mu = 8$

1.4.3 Different service rates, 'round-robin' policy

Next, while keeping the 'round-robin' server selection strategy, the different servers were assigned different service rates: $\mu_1 = 10$ $\mu_2 = 7$ $\mu_3 = 5$ $\mu_4 = 2$ $\mu_5 = 1$

The result of this choice is that 'more lucky' packets will be processed by faster servers, while others will be served by slower ones. In this case, once a server ends its service, the clients which are currently being served will not change servers. In the previous cases, there was no distinction between this approach and changing the assigned servers, since the service times were all following the same exponential distribution, and thus the two strategies were equivalent (the distribution of the residual time is the same as the service time itself).

Figure 1.11 shows the utilization of the servers in this scenario. It can be noted how, since all servers have the same opportunity to be chosen, as highlighted in section 1.4.2, the ones which are busier on average are, intuitively, the ones which take longer time to serve clients, i.e., which have lower service rate μ_i.

1.4.4 Different service rates, 'faster first' policy

While still keeping the same service rates as in section 1.4.3, the server selection policy has now been set to favor faster transmitters. This means that whenever it is necessary to assign a server,

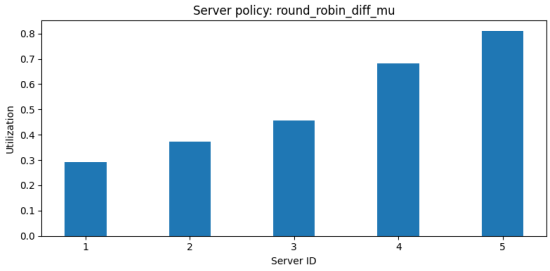


Figure 1.11: Server utilization, 'round robin', different μ_i

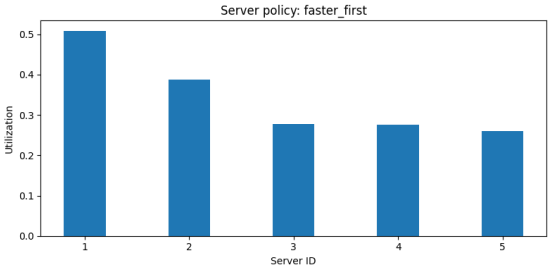


Figure 1.12: Server utilization, 'faster first', different μ_i

the faster available one will be chosen, making faster servers more utilized, as seen in figure 1.12.

For what concerns the slower servers, instead, the utilization rate does not reduce much, since their slow service time makes them busy for longer periods compared to faster ones, so the low number of times they are chosen is compensated by longer service periods on average.

1.5 Service time distribution

The next analysis concerns a single-server queue, in which different service times distributions have been tried. The considered queuing system now becomes, according to Kendall's notation, an M/G/1 queue (G stands for 'generic'). In any case, the different service times are independent instances of identically-distributed random variables.

The baseline system consists in a single server queue with finite capacity (20) in which the inter-arrival times are exponential with parameter 10, i.e., with mean value 0.1 time units. The simulation time used is 50000 time units.

1.5.1 Constant service times

The first considered scenario makes use of constant service times of value 0.05 time units. In this case, the server is able to process the clients at an overall rate which is higher than their arrival rate. Indeed, supposing the server is always busy, it will output a packet every 0.05 time units, which yields an output rate of $1/0.05 = 20$ $[packets/time\ unit]$.

Figure 1.13 reports the distribution of the queuing delay, while figure 1.14 includes the distribution of the waiting time (buffering), without considering packets that were directly served (whose service time is 0).

It can be noted how the majority of packets experience a total queuing delay which is equal to 0.05 time units (the main peak). This happens because, as highlighted previously, the service rate is still higher than the arrival rate, so most clients will directly be served upon their arrival, and will end up staying in the queue only for their service. For what concerns the waiting delay, instead, it turns out that most packets have to wait less than 0.05 time units. This can be due to the fact that, also thanks to the fast operation of the server, packets which have to wait will generally only have one client in front of them in the queue, making their waiting time at most equal to the service time.

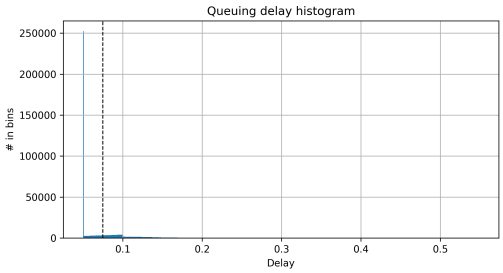


Figure 1.13: Queuing delay distribution, constant service rate

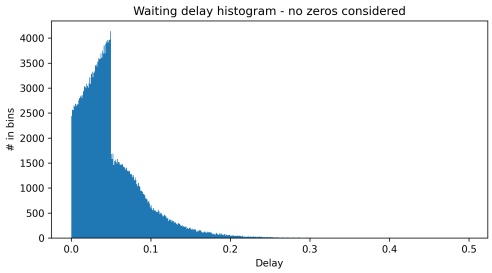


Figure 1.14: Waiting delay distribution, constant service rate

1.5.2 Uniformly-distributed service times

Another considered case is that of uniformly-distributed service times, with mean 0.05 time units, i.e., a uniform random variable between 0 and 0.1. Also in this case the resulting system has an overall output rate which on average is higher than the arrival rate.

Figure 1.15 displays the queuing delay distribution for this system, while figure 1.16 contains the waiting delay histogram (without considering clients which did *not* wait).

Looking at the queuing delay distribution, it is possible to distinguish two regions: the one related to delays lower than 0.1 time units, and the one with delays greater than 0.1. The left side of the histogram is clearly influenced by the uniformly-distributed service time, whose values are in the same range. Indeed, most clients will not even have to wait and their queuing time will fall in this range. The other portion of the distribution is instead made up only by clients which have to wait before their service. As stated before, their number is low since the overall service rate is higher than the arrival rate.

The second picture highlights again the same trend, following different 'shapes' depending on the delay being lower or greater than 0.1. Indeed, in general, a buffered client will have to wait typically for one other client being served, as proven by the average number of clients in the whole system being 0.835, and the average number of buffered clients being 0.334.

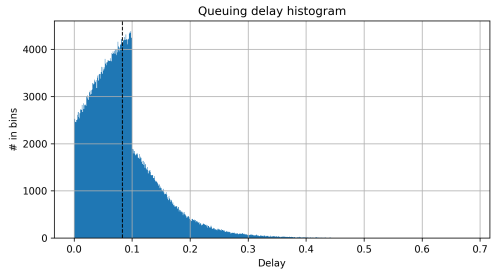


Figure 1.15: Queuing delay distribution, uniform service rate

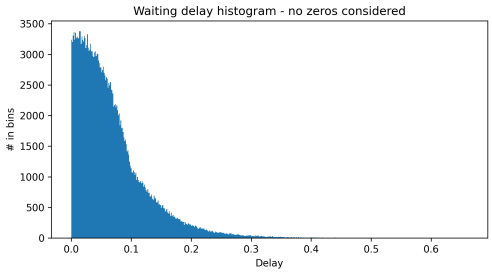


Figure 1.16: Waiting delay distribution, uniform service rate

1.6 Conclusions

During this laboratory experience, different systems have been analyzed, showing different routing conditions and some ways to deal with them. It has also been interesting to observe how different arrival rates and load distribution techniques change the behavior of the system, how having big buffers is not the most viable solution in general, and how the multi-servers technique is the best solution to reduce the number of losses of the system in case of high arrival rates.

2 Report 2: Industrial Internet of Things system simulation

2.1 Introduction

During this laboratory, a portion of an IIoT (Industrial Internet of Things) system composed of a Micro Data Center (MDC) and a Cloud Data Center (CDC), both modeled as queuing systems, is simulated. The two queues are organized in a simple tandem, in which packets arrive at the MDC and, if needed, are then forwarded to the CDC, else they exit the system. In general, it has been assumed that the CDC can make use of a greater amount of resources being located in the cloud. Data packets coming into the system can belong to two different classes, which are associated to different policies: class A, composed of high-priority tasks processed by the edge nodes only, if possible, and class B, composed of low-priority tasks that need more complex processing and are always forwarded to the Cloud Data Center. Whenever the first queue would drop a packet because of a full buffer, that packet is forwarded to the second server, even if the packet is of type A.
The parameter f is set to determine the fraction of type B packets and the transmission time between the first and the second queue is fixed to 0.2 'time units' during all simulations.

2.2 Cloud Data Center analysis, average waiting delay

Initially, the system has been simulated with the two queuing systems having only one server. In this part, the inter arrival time is an exponential random variable with average 2 time units, while the service times are exponential with average 3 time units for the first queue and 6 time units for the second one. Furthermore, queue 1 has a buffer length equal to 10, while queue 2 can contain at most 20 packets. The fraction of type B packets, f, has been set to 0.5 and the system has been simulated for 100000 time units.

This first analysis concerns the average waiting delay of the second queue. The choice of the parameters results in an overwhelmed first queue, which causes not just type B packets to be forwarded to the second queue, as it may happen that incoming A packets are directly sent to the second queue since the buffer of the first one is full. Additionally, being the server of the second queue slow, it's possible to anticipate that the second queue will fill up easily.

Figure 1.1 shows the waiting delay distribution of the second queue in this case (without distinction of packet type). In this figure, also the mean value (over the whole simulation) was represented

and, as anticipated, it is possible to see that in general the values are distributed around 100 time units.

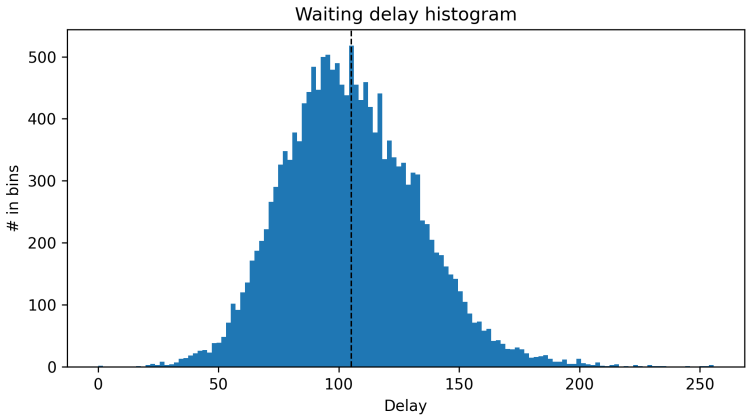


Figure 2.1: Distribution of the waiting delay, Cloud Data Center

2.2.1 Warm-up transient analysis

By analyzing the average waiting delay evolution in time, it is possible to observe the initial transient of the system. During this period, the system is not working at steady-state yet and the system variables are in the process of stabilizing in value.

Figure 2.2 shows the behavior of the average waiting delay at the CDC in time (specifically, it is the average waiting delay evaluated after each sample of this quantity). It is evident the distinction between the initial transient and the steady state behavior.

In order to evaluate more precisely the steady-state behavior, the selected approach is that of comparing the evolution of the mean value in time with the overall mean value of the waiting delay for the whole simulation. By considering the average waiting delay in time $\mu_W(t)$ and the overall mean $\hat{\mu}_W$, the relative variation of the former with respect to the latter is defined as:

$$R_W(t) = \frac{\mu_W(t) - \hat{\mu}_W}{\hat{\mu}_W} \tag{2}$$

Then, the end of the initial transient can be observed as the instant in which the value of the relative variation settles below a given percentage, chosen, for example, empirically to correspond with the knee of the curve (as in figure 2.2), or fixed in order to require a specific precision. In this case, the chosen percentage has been set to 10%, which yields an initial transient time of approximately 1200 time units. Figure 2.3 shows the plot of the relative variation in which the end of the transient has been highlighted.

In order to remove the initial transient from the analysis it is possible to simply 'ignore' the results up to the evaluated time instant. An important aspect is that, since the two queue have different parameters, the initial transient length will be different for the two systems, hence the transient removal should be repeated in the same way for the first queue.
The alternative option is that of running very long simulations, so that the initial transient does not

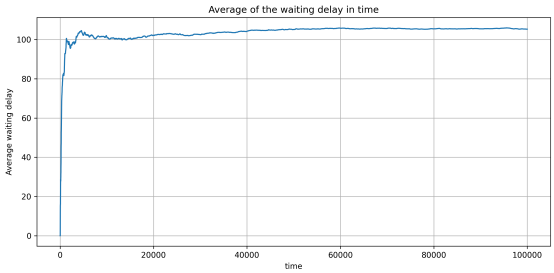


Figure 2.2: Average waiting delay, evolution in time - Cloud Data Center

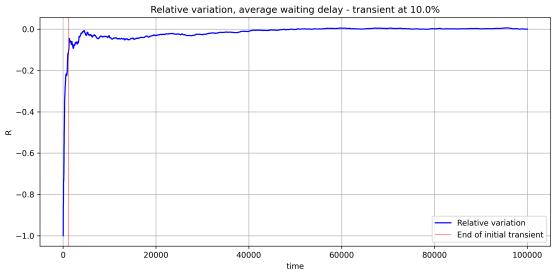


Figure 2.3: Relative variation of average waiting delay with transient end

impact the measurements in a relevant way. This kind of approach, however, is generally avoided, since the simulation cost increases with the simulation time and excessively-long simulations require a high amount of resources.

It is also important to remark that the initial transient duration is affected by the system parameters, which are numerous in this case, and therefore its length will change depending on the specific values used.

2.3 System performance: Micro and Cloud Data Center

Next, the performance of the whole system has been analyzed, concentrating on the impact of some system parameters, namely the buffer sizes and f, the fraction of 'B' packets. In order to study the behavior of the system, the average number of users in the queues was considered.

The baseline parameters are: exponential inter-arrival times with mean 10 time units, exponential service times with mean 10 time units for the first queue and 15 for the second one. Again, the system is overwhelmed, as the arrival rate is equal to the service rate in queue 1. This will help in obtaining significant plots for the system performance in terms of average number of queued packets.

2.3.1 Buffer impact

First, the system has been simulated by changing the buffer size of the first queue (Micro Data Center), while keeping the one of queue 2 equal to 20 and $f = 0.5$. The different values used are: 1, 2, 4, 5, 8, 10, 12, 15, 18, 20, 25 [packets].

Figure 2.4 contains the values of the average number of users in both queues depending on the queue size. Predictably, when the buffer size is small, most packets arriving at the first queue will need to be forwarded towards the Cloud Data Center, as likely there will not be free space in the queue. This makes the second queue more overwhelmed, as seen by the higher values for the average number of users, and causes a high number of losses.

Secondly, the system has been simulated by changing the queue size of the Cloud Data Center, using the same values used for the Micro Data Center previously, while the buffer size of queue 1 is now set to 10. The current results can be seen in figure 2.5. The plot shows how in this case the Micro Data Center is not influenced by the change in queue size of the CDC, as it is placed before

in the tandem. On the other hand, the second queue will have an increasing average number of packets, as the increasing buffer size will allow it.

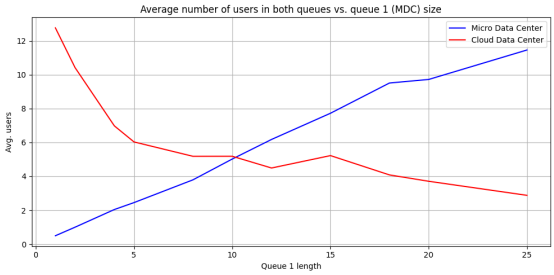


Figure 2.4: Average number of users in both queues depending on MDC buffer size

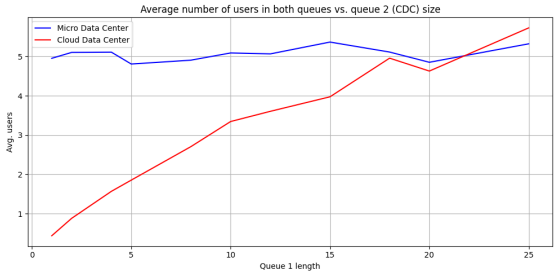


Figure 2.5: Average number of users in both queues depending on CDC buffer size

2.3.2 Input data impact

Next, the system is analyzed for different values of the fraction of 'B' packets, f. The chosen values are: 0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.

The average arrival time is 10 time units, while the service rate of the Micro Data Center is 10 time units and the one of the Cloud Data Center is 15 time units, making the system load critical.

In this case, the loss probabilities at each queue are considered, keeping in mind that, for queue 1, losses correspond to packets being forwarded directly to the second queue whenever the buffer is full, while losses occurring in queue 2 correspond to packets leaving the system unprocessed.

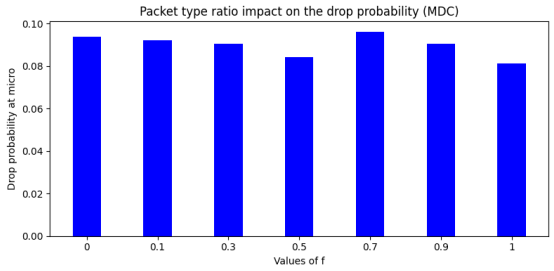


Figure 2.6: Loss probabilities as a function of f, MDC

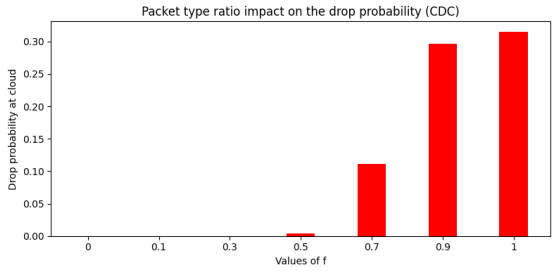


Figure 2.7: Loss probabilities as a function of f, CDC

Figures 2.6 and 2.7 contain the values of the loss probabilities, evaluated as number of losses divided by total number of arrivals, for the Micro Data Center and the Cloud Data Center, respectively. While for the MDC the loss probability results basically unaffected by the values of f, since the overall arrival rate is constant and the losses occur regardless of the packet type, for the cloud data center the loss probability increases as the packets of type 'B' increase in ratio. This happens because 'B' packets are the ones requiring processing at both data centers, hence in the limit case of $f = 1$, i.e., when all packets are of type B, the queuing network will have all packets exiting queue 1 going into queue 2 and, given the choice of the parameters the second queue will be overwhelmed by

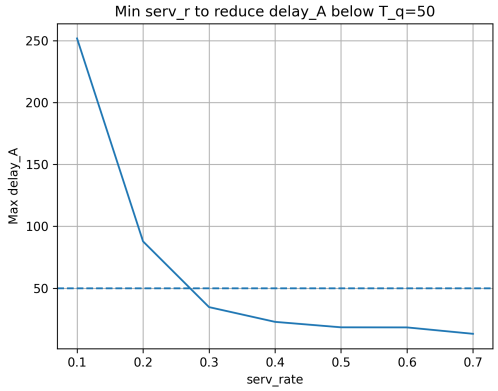


Figure 2.8: Max delay of type A packets versus service rate of the edge node

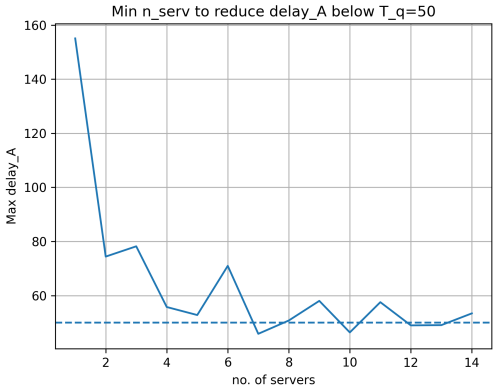


Figure 2.9: Max delay of type A packets versus the number of edge nodes

the arrivals, occurring at a higher rate than service. Indeed, it is possible to roughly approximate the arrival rate at the second queue (supposing no 'losses' occur at the first one, so infinite buffer)

2.4 Queuing time threshold

In this section, the variation of the queuing delay of type A packets has been analyzed in two different scenarios: varying the service rate of the edge node in a single-server system and increasing the number of nodes on the micro data center side. The goal is to find the minimum service rate and the number of servers to keep the queuing time of type A packets under a threshold $T_q = 50$ time units. Figures 2.8 and 2.9 show the outcomes of the simulation: the minimum service rate to keep the maximum delay below 50 time units is $\mu = 0.3$, while the minimum number of servers is $N_{serv} = 7$ given a service rate of $\mu = 0.125$ for all. It is important to underline the fact that the results are not deterministic; indeed, changing the random seed of the simulation also changes the outcome. This affects especially the number of servers in the second scenario which is very variable. However, the outcomes are consistent with what expected: the higher the service rate or the number of servers on the edge side, the lower the time spent in the system by packets A. From the plots, it is easy to see how the average delay decreases exponentially with the service rate; then to obtain satisfying results there is no need to use a configuration with extremely high performance (and high cost, for what discussed in section 2.5), but is important to find the right trade-off based on the purpose and on the requirements of the application.

2.5 Servers operational costs

In the following sections, both Micro and Cloud Data Center are assigned 4 servers each with service rate $\mu = 0.25$. For each Cloud Data Server, an operational cost is assigned, based on the service rate of the specific server, so that the faster it is in processing the packet, the higher will be its cost.

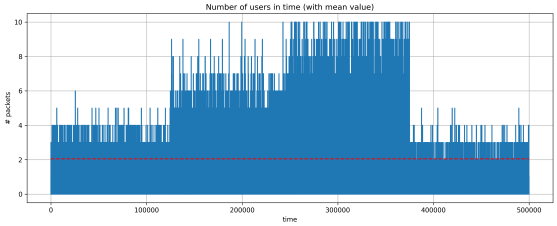


Figure 2.10: Number of packets on a simulated day traffic scenario. In red is the average number of packets.

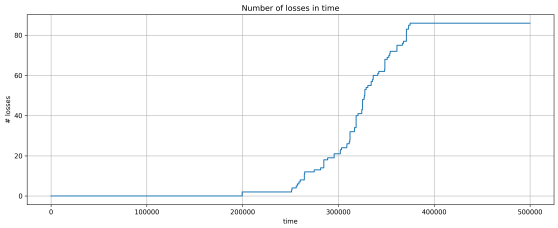


Figure 2.11: Number of lost packets packets on a simulated day traffic scenario.

2.5.1 Packet arrival-rate change over the simulation

The simulation has been divided into four different time intervals each characterized by a different arrival rate: $\lambda_1 = 0.125$, $\lambda_2 = 0.33$, $\lambda_3 = 0.5$, and $\lambda_4 = 0.08$. In this way, it is possible to model arrivals in an empirical way, for which the rate of incoming packets is different at different moments of the day, i.e. very low load during the night and higher in the daytime.

Figure 2.10 shows the number of queued packets versus the simulation time for the Micro Data Center; the change between one zone and the other is clearly visible and happens every 125000 time units.

For what concerns the performance of the system, the number of dropped packets has been considered. Figure 2.11 shows the number of losses versus the simulation time. Even if the system is still ergodic, in the third interval it is possible to observe the highest amount of losses because of the higher arrival rate.

Taking into account the Cloud Data Server side the outcomes related to the number of users is almost the same, with a slightly lower average with respect to the edge side. This happens because the edge node "sees" all the packets regardless of their type, while at the cloud on type B packets arrive. For this simulation, the fraction of type B packets has been set to $f = 0.5$, but if it is increased then the opposite occurs. Instead, for what concerns the number of dropped packets the cloud side does not experience any loss.

2.5.2 Operational cost threshold

The values set for the following analysis are $f = 0.75$, the fraction of type B packets, $T_c = 25000$, the threshold on the operational cost, and $T_q = 50$, the queuing delay threshold for type A packets. In order to respect both the constraints on the queuing delay and the operational cost the following configuration has been adopted on the cloud side: two servers with service rate $\mu = 0.143$ and two servers with $\mu = 0.25$. Table 2 reports the simulation's outcomes. The four micro data center servers have a high service rate $\mu = 0.33$ thus no type A packets are dropped and they are all processed on the edge side without needing the cloud center's help. On the cloud side, due to the constrain on the operational cost slower and cheaper servers have been chosen causing some losses of type B packets; moreover, most of the packets are of type B, due to the value of the fraction, putting more effort on the cloud servers.

Installing half of the server on the cloud side while keeping the same configuration as above it is

not feasible to respect both the thresholds on the queuing delay and on the operational cost. The best service rates to stay below the maximum cost are $\mu_1 = 0.33$ and $\mu_2 = 0.252$; however, the maximum queuing delay of A packets is $d_A = 74.63$ time units. Confronting the results with the previous simulation also the loss probability of type B packets increases to $p_{lb} = 0.11$, while that of A packets remains to zero.

MDC operational cost	16288.00
CDC operational cost	7625.21
Max queuing delay packets A	49.01
Max queuing delay packets B	50.41
Loss probability packets A	0.0
Loss probability packets B	0.03

Table 2: Simulation results of a four-server system and trade-off between operational cost and queuing delay

2.6 Conclusion

This laboratory experience has provided an interesting insight into the simulation of a complex system, composed of sensors, actuators, and the edge and cloud nodes that have been simulated. In particular, the interaction between the two queue systems has been analyzed in order to see how the behavior of one system affects the other and how the whole network responds to different traffic conditions in time. Finally, operational costs have been assigned to the servers to investigate real scenarios where it is fundamental to find the best trade-off between cost and performance depending on the application use case, thus providing some insights on the complex decisions which have to be made in real-life applications of this kind.