Management and Content Delivery for Smart Networks: Algorithms and Modeling

LAB 1 and LAB 2

Professor: Michela Meo

Participants:

Setareh Pourgholamali (309064)

Hossein Zahedi Nezhad (309247)

Politecnico di Torino

Academic Year 2022-2023

LAB1

INTRODUCTION:

This lab aims to simulate the output link of a router by modeling it as a queuing system. Its purpose is to analyze the system's performance under different configurations, understanding the impact of parameter settings on system behavior and performance metrics. In this queuing system, packets arriving at the output link are considered as customers, while the transmission on the channel is represented as the service. The waiting line, on the other hand, represents the buffer where packets are stored before transmission.

TASK 1

We investigated the system performance under different arrival rates while assuming an infinite queue capacity. We analyzed the relevant performance metrics to understand the impact of varying arrival rates on the system under these conditions.

We consider a single server with an infinite waiting line. We vary the arrival rate and keep the average service rate fixed. We focused on the different performance metrics shown in Table 1 with the results given.

Arrival Rates	Number of arrivals	Average number of packets	Number of dropped packets	Loss probability	Busy time
0.1	4281	0.097	0	0.0 %	0.726 %
0.2	8576	0.208	0	0.0 %	3.012 %
0.3	12756	0.347	0	0.0 %	6.546 %
0.4	16806	0.511	0	0.0 %	11.254 %
0.5	21589	0.74	0	0.0 %	18.427 %
0.6	25432	1.048	0	0.0 %	26.314 %
0.7	29638	1.504	0	0.0 %	35.74 %
0.8	34264	2.193	0	0.0 %	46.916 %
0.9	38467	3.394	0	0.0 %	59.034 %

Table 1. Performance Metrics in different Arrival rate

As it shown in table 1, the number of arrivals will directly depend on the arrival rate (λ). Therefore, as it increases, the number of arrivals will also increase. By rising λ, the average number of packets in the system which represents the average number of packets in the queue will also increases, leading to longer busy times for the server.

In a system with an infinite queue, there are no dropped packets since the queue can accommodate an unlimited number of packets. Therefore, the number of dropped packets remained constant at zero regardless of the arrival rate. Consequently, Loss probability which represents the probability that an arriving packet is lost (dropped) will be zero. This growth leads to longer busy times for the server.

Average Delay:

With an infinite queue, the average delay metric provides insights into the queuing behavior and service time. As shown in the Figure below, the delay experienced by packets increases with increasing arrival rate. This is because of imbalance situation between customer arrival and service rates, leading to longer queues and increased wait times for customers so the packets must wait longer in the buffer before they can be served.

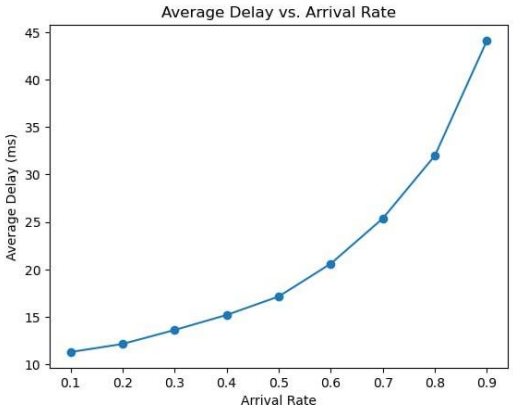


Figure 1. Average delay changes in M/M/1 system

TASK 2

In this part we want to test different buffer size in M/M/1 queue and M/M/2 queue:

By comparing the number of losses for different buffer sizes at different arrival rate figure.2 and figure.3, we will be able to observe how the buffer size affects the likelihood of lossing customers as the arrival rate increases. Smaller buffer sizes will generally lead to higher losses as the system can hold fewer waiting customers, while larger buffer sizes can accommodate more customers and result in lower loss.

It is obvious that in smaller arrival rate there is a significant different in number of losses for example in buffer size 5 and 10 but when arrival rate increases more and more the differentiation in losses become smaller as any arrival packet will be dropped because of the full buffer.

For comparing the performance of the system with one and two servers against each other: As the arrival rate increases and approaches or exceeds the service rate (μ), the queue starts to build up, and customers may have to wait in the queue after that when the queue reaches its maximum capacity (buffer size), any additional arriving customers will be rejected (lost) as there is no space in the queue to accommodate them. In an M/M/1 queuing system, there is a single server available to serve incoming customers, in an M/M/2 queuing system, there are two identical servers available to serve incoming customers, with two servers in the system, it can handle a higher arrival rate compared to the M/M/1 system. Each server can serve customers independently, and the system can process more customers simultaneously. As the arrival rate increases, both servers are likely to be busier, but the overall capacity of the system is increased due to the presence of the second server. By referring to figures.2 and figure.3 we can see that in the same buffer size, for

example buffer size (5) and same arrival rate (1.5) the number of losses for M/M/1 is about (14000) and for M/M/2 is about (500). Another point Based on our results which they are showing that buffer size can have a significant impact on number of losses in comparing these two systems in other words in buffer size (1) there is not major difference between one and two servers Since the buffer becomes full with one extra packet in both models but in buffer size (10) the variation is so high.

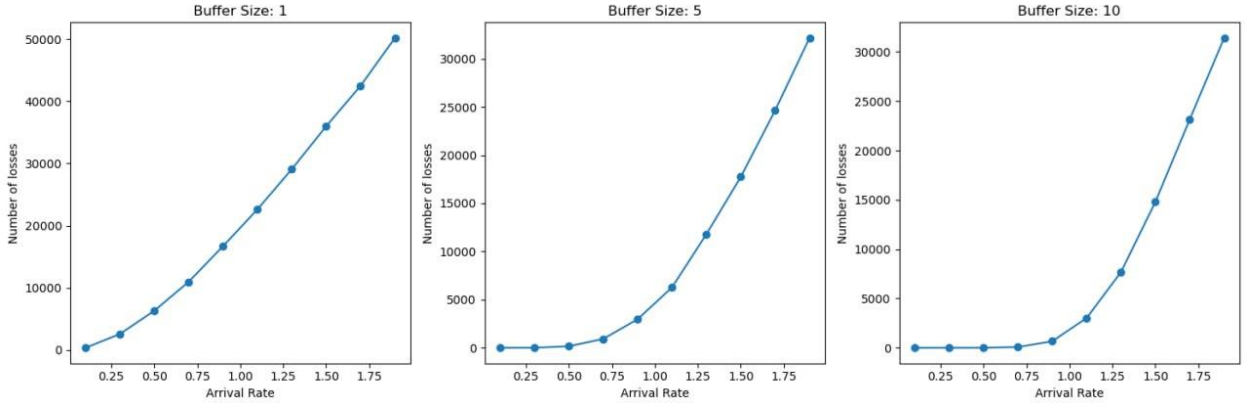


Figure 2. Impact of change in the buffer size on the Number of losses in M/M/1 queue

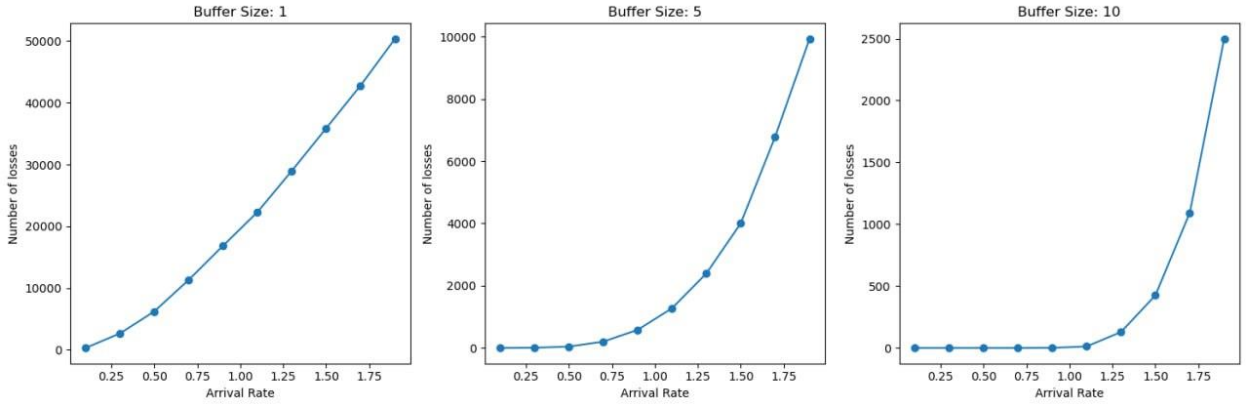


Figure 3. Impact of change in the buffer size on the Number of losses in M/M/2 queue

By comparing the average delay for different buffer sizes at different arrival rates (figure.4 and figure.5), we will be able to observe how the buffer size affects this metric in M/M/1 and M/M/2 queuing system. The M/M/2 queuing system with two servers typically offers a lower average queuing delay compared to the M/M/1 queuing system with one server. This is mainly due to the increased capacity and faster service times provided by the additional server in the M/M/2 system. The M/M/2 system can handle higher customer loads and is more efficient in reducing queuing delays during busy periods. For example, in buffer size (10) under arrival rate (1.5) in system with one server we experience approximately 75ms average delay while with two servers this metric is about 16ms.

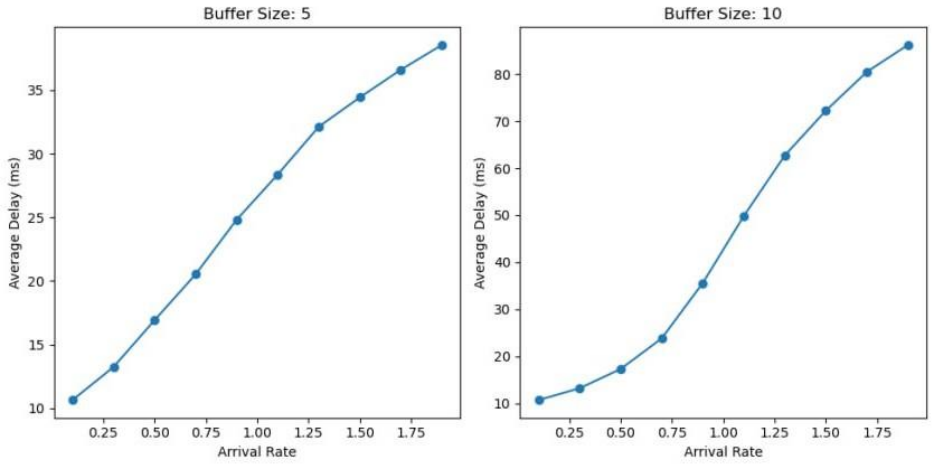


Figure 4. Impact of change in the buffer size on the average delay (ms) in M/M/1 queue

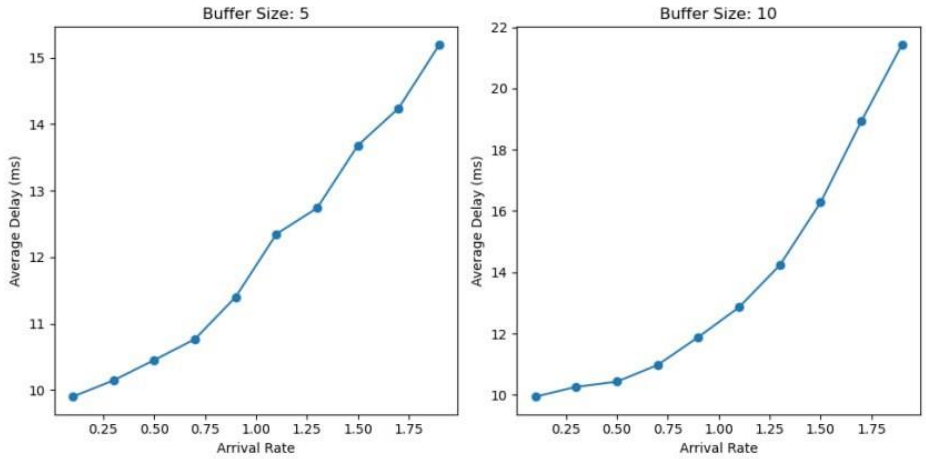


Figure 5. Impact of change in the buffer size on the average delay (ms) in M/M/2 queue

Now we want to compare these performances with respect to infinite buffer size (M/M/2):

As we discuss in task 1 when we have infinite buffer size, we do not have any dropped packet since the queue can accommodate an unlimited number of packets. In the M/M/2 queuing system with finite buffer, the queuing delay will increase, and once the buffer is full, there will be a sudden spike in the queuing delay due to blocking or loss of additional arriving customers.

In the M/M/2 queuing system with infinite buffer, the queuing delay will also increase, but it will grow more gradually since there is no blocking or loss of customers. The queue will accommodate all arriving customers, even though it may take longer to serve them all.

TASK 3

In a multi-server scenario, we want to investigate the load distribution among servers by testing different assignment algorithms. Several algorithms can be used to assign packets to the servers. In this task we will compare random assignment, round-robin assignment, and assignment to the fastest servers based on service rates.

1. Random Assignment: Each new request is randomly assigned to an available server. Load distribution may be random and uneven, leading to imbalanced workloads among servers.
2. Round-Robin Assignment: New requests are assigned to servers sequentially in a rotating fashion. This algorithm ensures a fair distribution of workload among servers, resulting in a relatively balanced load distribution.
3. Assignment to the Fastest Servers: Requests are assigned to the fastest servers based on their service rates. This approach aims to distribute workload efficiently, potentially resulting in an imbalanced load distribution favoring the fastest servers.

Now, we are going to analyze how these algorithms effect on the functions of two different metrics: utilization and loss probability

Difference in utilization:

Based on the obtained Results (figure.6) which is related on utilization, In the random method, the utilization of each server can vary significantly due to the random assignment since Some servers might be underutilized, wasting their processing capacity, while others might be overloaded that is why random assignment shows us the largest and sharpest change (varying from 1 to 1.4). Using round robin algorithms will result in balanced utilization of all servers, as each server gets an equal share of the traffic. And in the fastest method, Servers with faster processing capabilities might have higher utilization, while slower servers may remain underutilized that is why the total utilization of servers is not changing significantly- from 1 to 1.030.

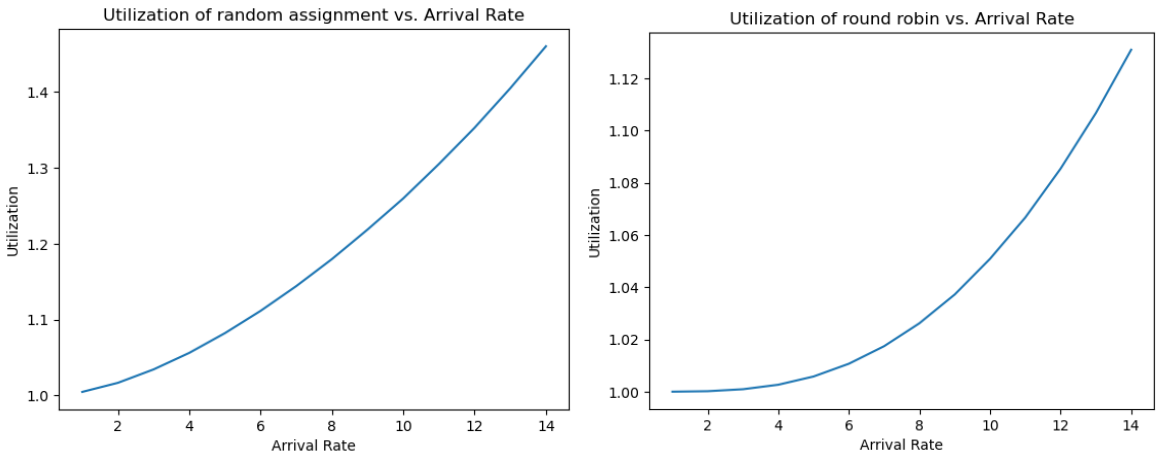
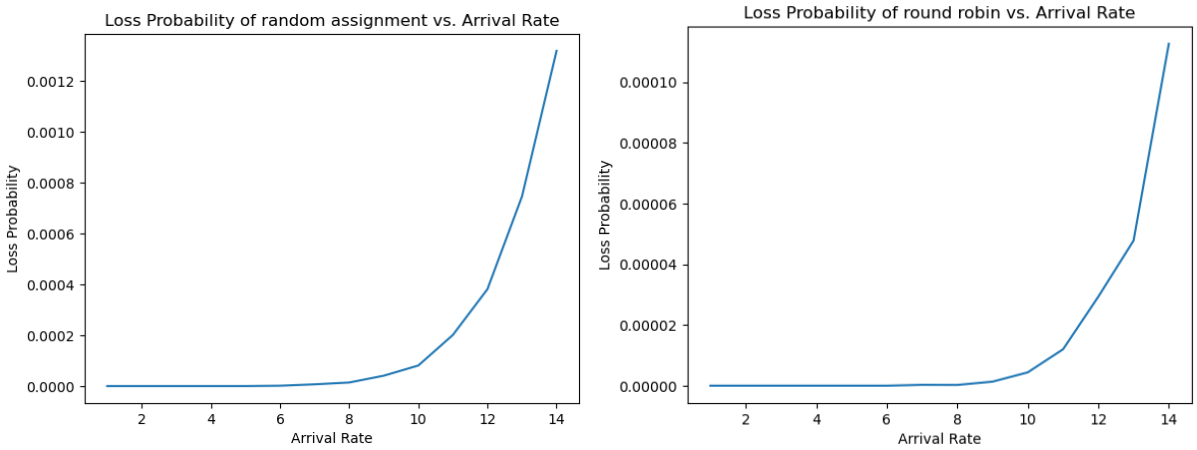


Figure 6. Comparison of utilization in multi-server system using three different algorithms

Difference in Loss Probability:

In this range of arrival rate based on our results figure.7, for Random assignment the probability of loss can vary depending on how evenly or unevenly the requests are distributed among the servers. If one server is overloaded while others have spare capacity, the overall loss probability might be higher due to the potential buffer overflow in that overloaded server. In Round Robin the loss probability is generally low since requests are evenly distributed among servers, reducing the chance of buffer overflow and rejections. Assignment to the Fastest Servers the loss probability might be lower compared to random assignment, especially if the fastest servers can handle incoming requests more efficiently. However, it may not be as low as in the round-robin approach if the fastest servers are overloaded while other servers have available capacity.



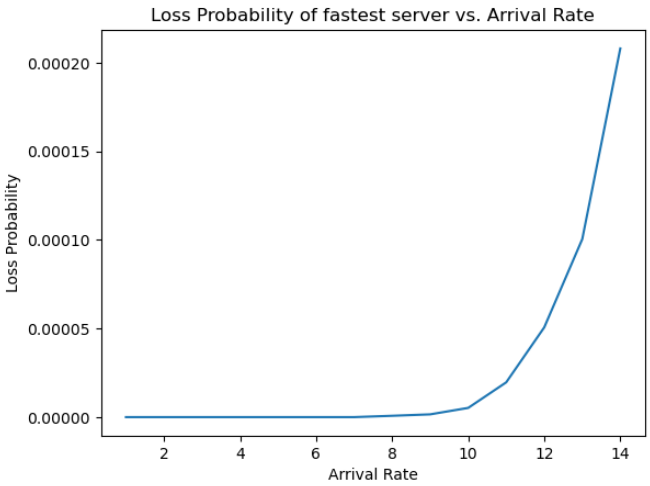


Figure 7. Comparison of loss probability in multi-server system using three different algorithms

TASK 4

In the M/G/1 queuing system, the service time is assumed to be exponentially distributed. We can try to vary the distribution of the service time by considering the case of M/G/1, and observe how the system performance changes, assuming one or more different distribution types for the service time instead of exponential distribution. We test two different distribution types:

Normal distribution and Uniform distribution

Let's examine the changes in the utilization and loss probability for both the normal and uniform service time distributions as the arrival rate increases over time:

According to the obtained Results (figure 8 and 9), as the arrival rate increases, the system utilization also increases. Initially, as utilization rises, the loss probability also increases as the system gets more congested, and packets may start to be dropped. As the arrival rate continues to increase, there comes a point where the load on the system becomes more balanced, and the utilization (ρ) approaches 1. At this stage, the system is close to its capacity, and there might still be packet losses due to congestion. With an increasing arrival rate beyond peak utilization, the load on the system starts to become more balanced. The server is busy handling packets with varying service times, and the variability in service times helps to distribute the load more evenly among packets and across time. This can lead to a slight decrease in utilization and, subsequently, a decrease in the loss probability.

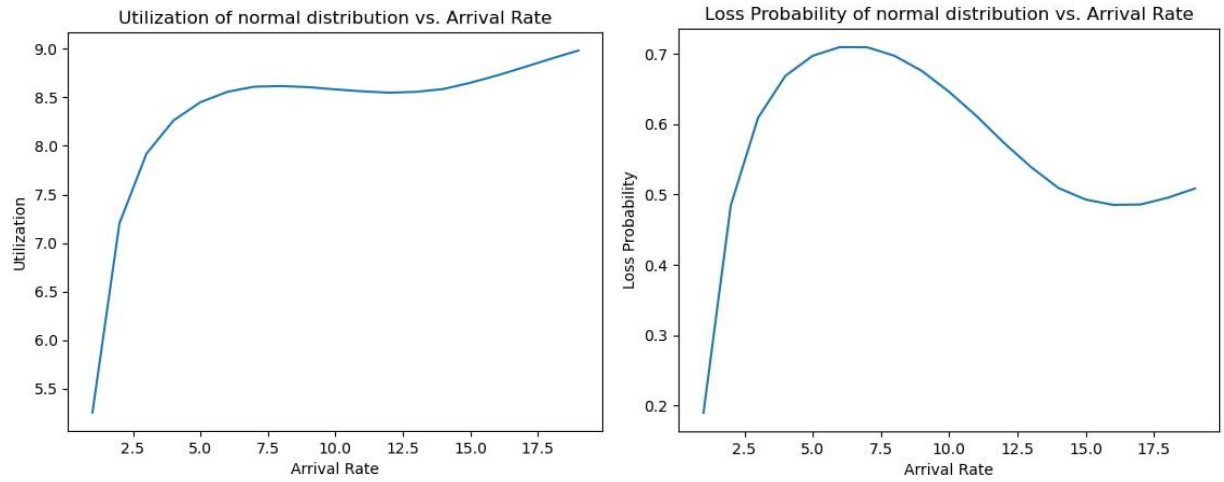


Figure 8. Normal distribution of the service time

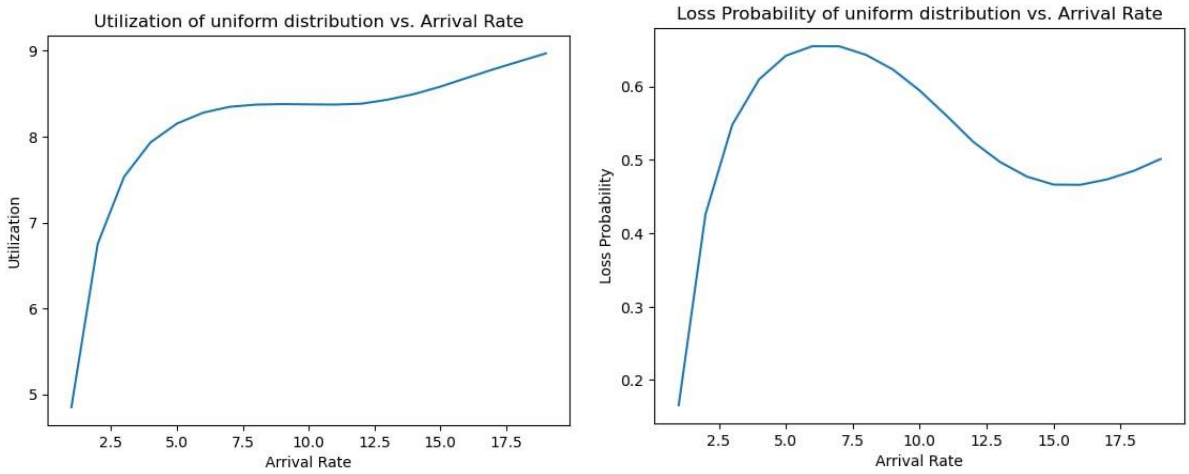


Figure 9. Normal distribution of the service time

Depicted results show that changes in the utilization and the loss probability in the both of distribution methods is almost the same. The reason is that the two distributions are very similar and the only difference between the two distributions is that the normal distribution is more likely to produce longer service times than the uniform distribution. However, this difference does not have a significant impact on the utilization and loss probability. The reason for this is that the probability of a customer arriving and finding the server busy is primarily determined by the arrival rate and the distribution of service times only plays a secondary role.

- Notice that the mean value and standard deviation are equal to 10 and 3 for the normal distribution, respectively. Also, the defined range for the uniform distribution is between 0 and 10.

CONCLUSION

In conclusion, the simulation study analyzed the performance of a queuing system representing the router's output link. The infinite queue capacity resulted in no dropped packets and zero loss probability. Smaller buffer sizes in M/M/1 and M/M/2 systems led to higher losses, but M/M/2 performed better overall. Round-Robin assignment provided balanced utilization and lower loss probability. Both normal and uniform service time distributions showed similar trends with increasing arrival rates. Overall, the study highlighted the importance of buffer size, number of servers, and assignment algorithms in optimizing queuing systems for efficient data transmission and routing.

LAB2 (version B)

INTRODUCTION

This lab simulates an Ilott queuing system to understand how different configurations and network parameters impact its performance in an Industry 4.0 scenario. The goal is to optimize the system and improve production line management.

TASK 1

The warm-up period is the initial phase of the simulation during this period, there may be fluctuations in the packet drop probability as the system adjusts to the changing traffic. The goal is to observe this transient behavior and identify the point at which the system is allowed to stabilize and reach a state that is representative of the system's behavior (steady-state). This phase is typically discarded or ignored when analyzing the results, as it might not accurately represent the long-term behavior of the system.

The transition to steady state refers to the phase of the simulation after the warm-up period, during which the system has transitioned to the steady state, where the simulation results are statistically reliable and reflect the average behavior of the system. During this phase the data collection, analysis, and performance evaluation are performed.

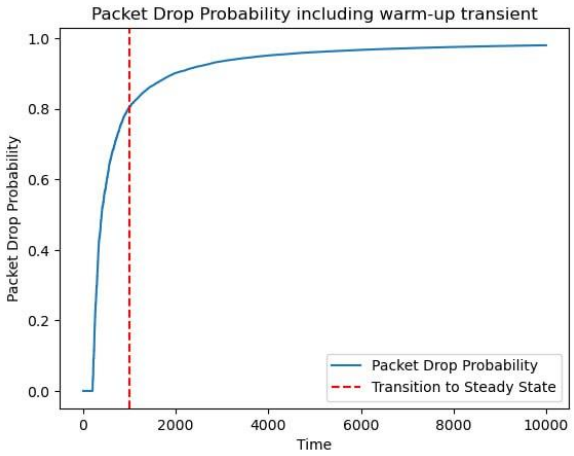


Figure 1. Showing Warm up transient in Packet Drop Probability

By analyzing the packet drop probability during the steady state, after removing the warm-up transient, we gain a better understanding of the Cloud Data Center's long-term performance. Several methods can be used to remove warm-up transient in simulations. In this task, we apply the Discarding Data method. For using this approach, it needs to run the simulation for a certain number of iterations or time steps (warm-up period) without recording any data. After the warm-up period, the simulation starts collecting data, and the transient data from the warm-up period is ignored or removed from the analysis.

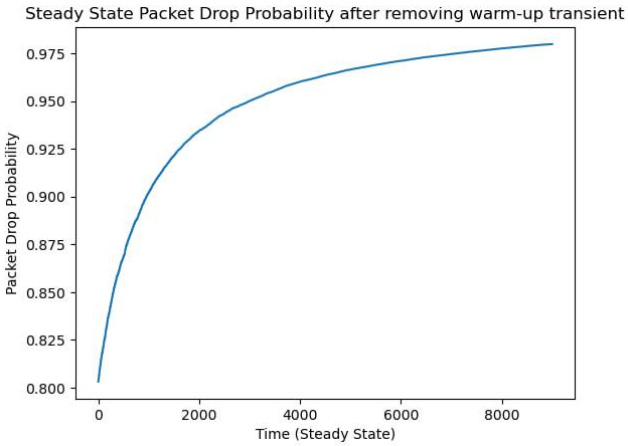


Figure 2. stabilizing system after the removing Warm up transient

TASK 2

(a)

Based on our result figure.3 Increasing the size of the buffer in the Micro Data Center can have a positive impact on the overall system performance. A larger buffer size in the Micro Data Center provides more storage capacity to temporarily hold incoming packets before processing. This can reduce the likelihood of congestion in the Micro Data Center, as there is a higher chance that incoming packets can be accommodated in the buffer. With a larger buffer in the Micro Data Center, fewer packets are forwarded to the Cloud Data Center immediately. This means the Cloud Data Center receives a smoother and more evenly distributed workload, reducing the likelihood of congestion and packet drops at the Cloud level as well. In this case changing buffer from 50 to 500 in simulation time (2000) can lead to a change in the drop probability from approximately 0.9 to 0.5.

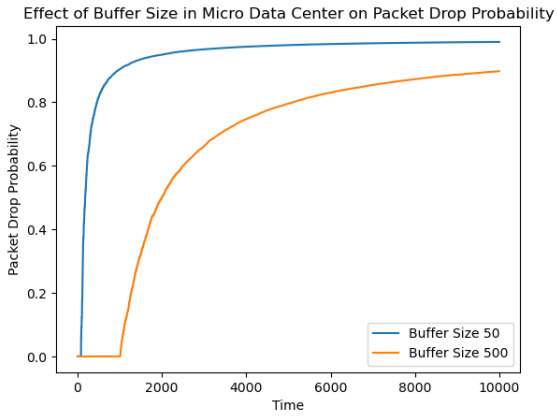


Figure.3

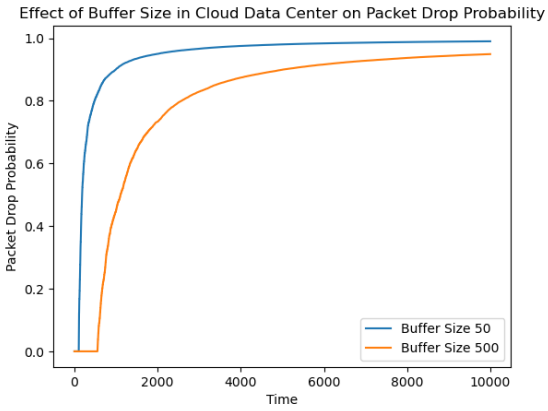


Figure .4

(b)

As figure.4 shows the impact of the buffer size in the Cloud Data Center may not be as significant as in the Micro Data Center. Since Type B packets, which require more complex processing, are forwarded to the Cloud Data Center after local pre-processing, the buffer in the Cloud Data Center primarily acts as a temporary storage before the packets are fully processed. However, having a larger buffer in the Cloud Data Center can still be beneficial in handling temporary surges in incoming packets and reducing the chances of buffer overflow. For example, changing buffer from 50 to 500 in simulation time (2000) can lead to a change in the drop probability from approximately 0.9 to 0.7 which shows that still is beneficial but with lower impact in comparison with previous case.

(c)

Figure.5 represents when "f" is low, which means that a large portion of the incoming data packets are of type A, which are high-priority tasks and can be locally processed at the edge nodes. Since type A packets can be processed locally, they experience lower queuing delays since they do not need to be forwarded to the Cloud Data Center. As a result, the overall average queuing delay will be relatively low. When "f" is high, it means that a significant portion of the incoming data packets are of type B, which are low-priority tasks requiring more complex computations. Consequently, a larger number of packets need complex processing, resulting in higher queuing delays, also these packets require forwarding to the Cloud Data Center after local pre-processing at the Micro Data Center, there will be additional propagation delays especially if the distance between the Micro Data Center and the Cloud Data Center is large. Higher propagation delays can increase the overall average queuing delay.

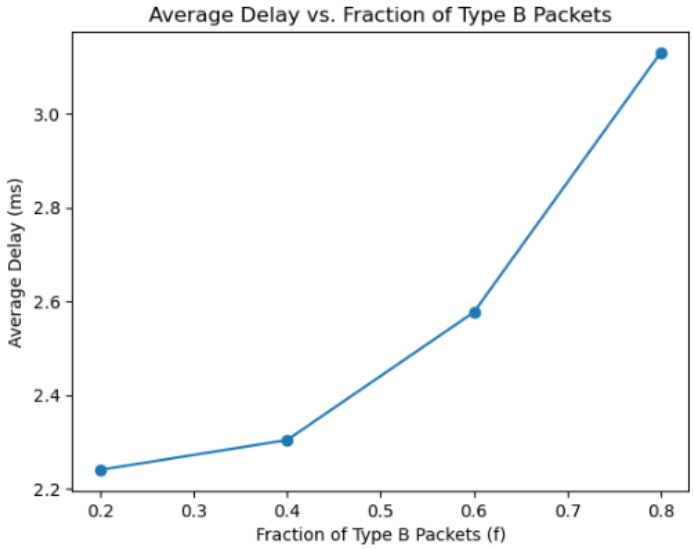


Figure 5. Impact of fraction on the Average delay

```
Average queuing delay with f=0.2: 2.240018594251395
Average queuing delay with f=0.4: 2.303708082983751
Average queuing delay with f=0.6: 2.5767437877473607
Average queuing delay with f=0.8: 3.1309506422542976
```

TASK 3

(a)

The threshold Tq represents the maximum average queuing time for type A packets that we observed during the simulation. In this section, we iteratively adjusted the threshold using a binary search algorithm to find the minimum service rate required to achieve the desired maximum queuing time. The minimum service rate for Micro Data Center value is found by using the binary search algorithm to adjust the service rate until the average queuing time for type A packets is

below the threshold Tq. Figure.6 shows how the average queuing time for type A packets changes as we vary the Micro Data Center's service rate. If the Micro Data Center's service rate is too low, the average queuing time for type A packets will be high, exceeding the desired threshold Tq. As we increase the service rate, the average queuing time for type A packets decreases until it reaches or falls below the threshold Tq. The goal is to find a balance where the service rate of the Micro Data Center is set at the minimum level required to achieve the desired threshold Tq, without overloading the system with excessive processing power. Threshold Tq: 2.33(ms) and Minimum service rate for Micro Data Center: 0.89(packets/ms), are the values computed during the execution of the code. These values are specific to the randomly generated simulation data and depend on the initial conditions (such as the number of packets (10000), fraction of type B packets (0.5), etc.) and the random number generator's behavior.

```
Threshold Tq: 2.3333333333357587 ms
Minimum service rate for Micro Data Center: 0.8976074218750001 packets/ms
```

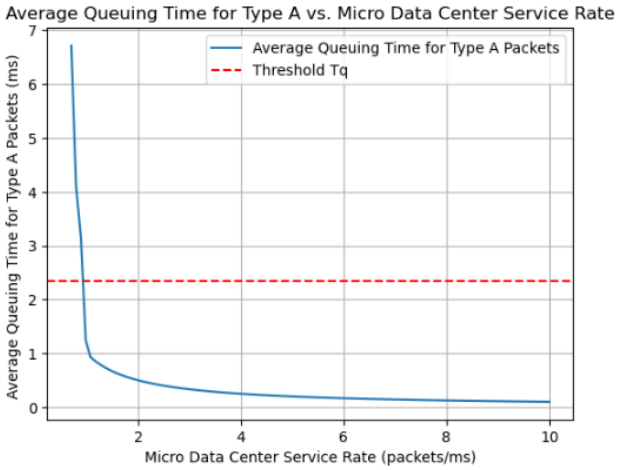


Figure .6

(b)

In this scenario for the simulation assumes a fixed average service rate of 0.6 (packets/ms) for each edge node in the Micro Data Center. This means that each edge node can process an average of 0.6 packets per millisecond. Increasing Number of Edge Nodes: As the number of edge nodes increases, the overall processing capacity of the Micro Data Center also increases. This is because each edge node can independently process incoming packets.

Based on Figure.7 the exponential decrease in the average queuing time occurs due to the Law of Large Numbers and the increased processing capacity. With more edge nodes, the arrival load can be distributed more evenly among them, reducing the queuing time. The red dashed line in the plot represents the fixed threshold Tq of 2.3333333333357587 (ms), with this line we can determine that minimum number of servers required to reduce the queuing time below the threshold Tq is 3.

The threshold is used as a reference to determine if the queuing time is acceptable. If the average queuing time falls below this threshold, it means that the Micro Data Center can handle the incoming packets efficiently. With more edge nodes available to process packets, there is a reduced likelihood of contention and waiting times at any single node. As a result, packets experience shorter queuing delays, leading to improved overall system performance. In this scenario the minimum number of edge nodes which are required to be below threshold is three nodes. We should take into account that increasing edge nodes without any rational behind it can increased costs for hardware, maintenance, and operations, heightened complexity in management and resource fragmentation, and higher communication overhead between servers. Additionally, scalability limits may be reached, fault tolerance can become more challenging, and energy consumption may rise significantly. So even if we reach a lower average delay on the other hand some disadvantages may increase.

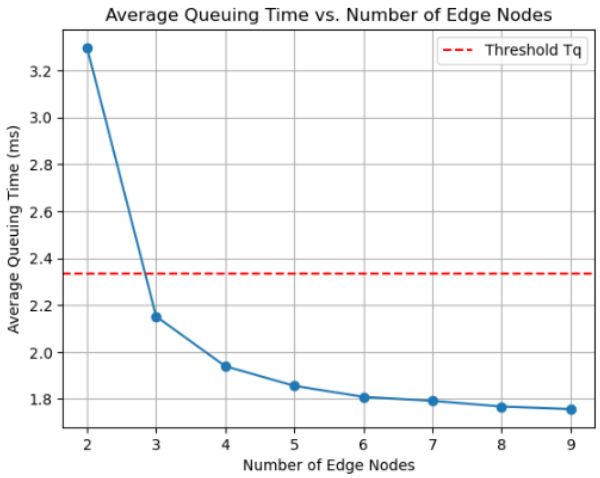


Figure .7