Politecnico
di Torino

Politecnico di Torino
Management and Content Delivery
Laboratory report

Luca Barotto - s329657
Alessandro Caramia - s322228
Federico Villata - s319922

Academic Year 2023-2024

Contents

1 Lab 1: Drone-assisted communication system simulation

1.1 Introduction

The purpose of this laboratory is to simulate the operation of a portion of an urban Radio Access Network (RAN) where terrestrial Base Stations are supported by renewable powered Unmanned Aerial Vehicles (UAVs), that are periodically sent to fly over a given area to provide additional capacity during peak periods, to offload the on-ground nodes and enable a better Quality of Service for the communication network. The network is modeled as a queuing system, and the objectives are to examine how well it performs in a variety of configurations, comprehend how various parameter settings may influence the behavior of the system, and determine how changing network system parameters may affect performance metrics. All simulations have a duration of 500000 time units. Realistic time units in this case could be milliseconds. To have reproducibility of the script a seed is set.

1.2 M/M/1

The first analyzed case is the MM1 queue, a queuing system with a single server. It is examined in both the infinite and finite buffer scenarios. Customers are served in the same order in which they arrive, inter-arrival times between them and service times are modeled as exponentially distributed random variables, and the system is assumed to operate under steady-state conditions. The performance metrics that have been considered (and that will be considered for all the other cases) are :

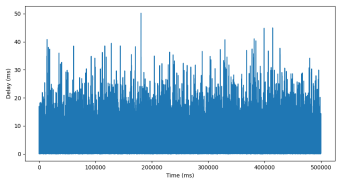
- Number of users in the queue at the end of the simulation;
- Number of arrivals and number of departures;
- Average number of users in the system;
- Average delay;
- Number of lost packets;
- Busy time.

The number of users in the queue at the end of the simulation depends on the arrival rate, the higher the arrival rate, the larger the queue at the end of the simulation. The number of arrivals and departures depends on the arrival rate and on the service rate. The system's average number of users is affected by service and arrival rates. The average delay represents the total time spent by a client in the system. This includes both queuing time and service time. The number of lost packets is the number of dropped clients due to a full buffer. The busy time is the total time during which the server is busy serving clients.

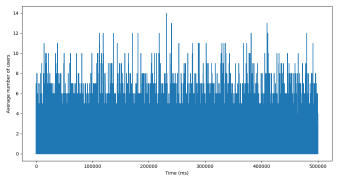
1.2.1 Infinite buffer

This system consists of a single server and an infinite buffer that can accommodate any number of clients that are waiting to be served without losing any packets. Fig. 1a and fig. 1c show the average time a client spends inside the system. In the first figure, where service time equals 2.5 ms, we notice that the average time spent in the system is considerably less than in the second figure where service time equals 4.5 ms. This is because in the second case, since the server is more overloaded, it is slower to dispose of clients and therefore the queue size increases, also increasing the waiting time. Also, we note that we are in the presence of an ergodic system and since the buffer is infinite there are no losses, however, periodically packets tend to accumulate thus increasing the delay. In the case where the load is greater than 1 the number of packets in the queue, as well as the delay, would continue to increase. The increase in queue size is also shown in figures 1b and 1d in which the number of users within the system is seen with time. Again in the figure for the most overloaded system (service time = 4.5), we notice a significant increase in users in the queue. Fig. 1e shows the confidence interval for the average time spent in the system.

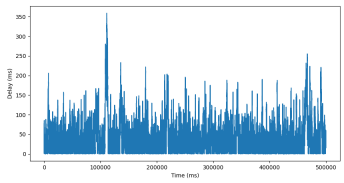
In tab. 1 Three different situations were analyzed, one in which service time is equal to 2.5 ms, one with service time = 3.5 ms, and one with service time = 4.5 ms. The arrival time is always equal to 5 ms.



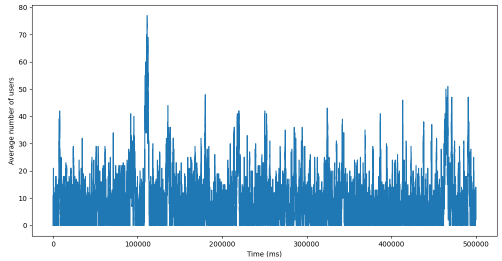
(a) Average time a client spends in the system with service time = 2.5



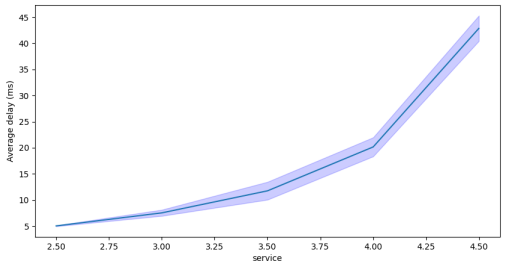
(b) Average number of users in the system with service time = 2.5



(c) Average time a client spends in the system with service time = 4.5



(d) Average number of users in the system with service time = 4.5



(e) C. I. for average delay (conf. level=0.95)

Table 1: MM1 with infinite buffer measurements

Quantity	Service time = 2.5 ms	Service time = 3.5 ms	Service time = 4.5 ms
no. of users in the queue at the end of the simulation	1	2	10
no. of arrivals	100033	100071	99756
no. of departures	100032	100069	99746
average number of users in the system	1.01	2.32	8.59
average delay	7.49 ms	11.62 ms	43.05 ms
busy time	300060.86 ms	350676.73 ms	448117.29 ms
no. of lost packets	0	0	0

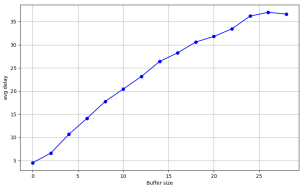
1.2.2 Finite buffer

This system consists of a single server and a finite buffer that can accommodate a limited number of clients, when the buffer is full the following clients are dropped.

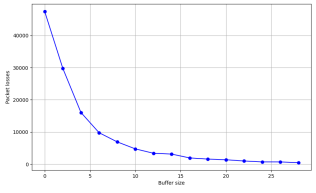
Fig. 2a shows the average time a client spends in the system as the buffer size varies, by increasing the buffer size, packets tend to stay longer in the queue, thus increasing the average delay. This is consistent with queuing theory, which predicts an increase in delay as the buffer increases since packets are more likely to be queued rather than lost. The average delay starts to stabilize at about buffer size = 25. The reason for this is that the average delay mainly represents the service time and server capacity once the buffer is large enough to manage the flow of packets efficiently. It then reaches a stable value that does not increase as the buffer grows.

Fig. 2b shows the number of packets lost as a function of buffer size. There is an exponential decrease in the number of packets lost as the buffer size increases. For very small buffers, losses are high, but increasing buffer capacity reduces losses until they stabilize.

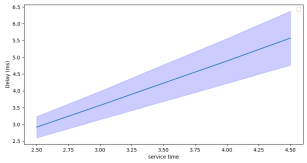
Fig 2c shows the delay as a function of service time with the confidence interval. With a buffer limited to 2 places, the queue quickly becomes congested as the load increases. A higher service time, with constant arrival time, increases the probability that the buffer will fill up and packets will have to wait longer because more packets arrive per unit of time. The average delay rises as a result of this increase in service time. The system becomes less predictable and more prone to fluctuations under high service time, as indicated by the growing confidence interval, which also reflects the increased variability of delay.



(a) Average time a client spends in the system with buffer size



(b) Number of packets lost with buffer size



(c) C.I of delay values concerning the service time with buffer size = 2, conf. level=0.95

Figure 2: Comparison of different performance metrics with varying buffer sizes

Table 2: MM1 with finite buffer measurements, buffer size = 4

Quantity	service time = 2.5 ms	service time = 3.5 ms	service time = 4.5 ms
no. of users in the queue at the end of the simulation	1	0	3
no. of arrivals	99934	99496	100167
no. of departures	96675	90932	84082
average number of users in the system	0.84	1.32	1.79
average delay	4.39 ms	7.25 ms	10.67 ms
busy time	241386.19 ms	318630.1 ms	377512.79 ms
no. of lost packets	3258	8564	16082

1.2.3 Comparison with theoretically expected values

The results of the analysis are consistent with theoretically expected values, both for the MM1 with finite buffer and for the MM1 infinite buffer. How the various delays, losses, number of users vary, and busy times are all under the theoretical aspects.

1.3 M/M/2

The M/M/2 queuing model is an extension of the M/M/1 that considers a system with two parallel servers. Both finite buffers and infinite buffers are considered. The server selection is done first by checking if the first server is free, if busy check the second server. If both servers are busy the packet will go to the queue.

1.3.1 MM2 with finite and infinite buffer

Table 3 shows three different situations, one in which the buffer has infinite size and both servers have service time = 9 ms, one in which the buffer size = 2 and both servers have service time = 9 ms, and one in which the buffer size = 2 and both servers have service time = 5 ms. The arrival time is always 5 ms.

Quantity	**Buffer size = infinite**	**Buffer size = 2**	**Buffer size = 2**
service time (per server)	9	9	5
arrival time	5	5	5
no. of users in the queue at the end of the simulation	0	1	0
no. of arrivals	99893	100163	99684
no. of departures from server 1	50763	43249	56445
no. of departures from server 2	49129	38499	39190
average number of users in the system	9.16	1.64	0.75
percentage of usage of server 1	91.94%	78.58%	56.25%
percentage of usage of server 2	88.48%	69.45%	38.96%
average delay in server 1	47.55 ms	12.39 ms	5.74 ms
average delay in server 2	49.03 ms	12.74 ms	6.06 ms
number of losses	0	18412	4048

Table 3: MM2 with finite and infinite buffer measurements

1.4 2/M/M/1

The 2/M/M/1 queuing model is an extension of the M/M/1 that considers a system with two parallel queues with two corresponding servers.

Table 4 shows three different situations, one in which the buffer has infinite size and both servers have service time = 9 ms, one in which the buffer size = 2 and both servers have service time = 9 ms, and one in which the buffer size = 2 and both servers have service time = 5 ms. The arrival time is always 5 ms.

Quantity	**Buffer size = infinite**	**Buffer size = 2**	**Buffer size = 2**
service time (per server)	4.5	4.5	2.5
arrival time	5	5	5
no. of users in the queue 1 at the end of the simulation	0	0	2
no. of users in the queue 2 at the end of the simulation	1	1	0
no. of arrivals	99942	87628	95334
no. of departures from queue 1	49855	43823	47733
no. of departures from queue 2	50086	43804	47599
average number of users in the system	0.82	0.52	0.29
average delay in queue 1	8.25 ms	5.93 ms	3 ms
average delay in queue 2	8.23 ms	5.89 ms	3.02 ms
number of losses in queue 1	0	6227	2465
number of losses in queue 2	0	6240	2519

Table 4: 2MM1 with finite and infinite buffer measurements

1.4.1 Comparison between MM2 and 2MM1

The M/M/2 queue loses more packets and has a higher average delay mainly due to the single shared queue that can lead to higher congestion and slower servers than the 2/M/M/1 model. Separate load management in two MM1 queues allows for a more balanced distribution and a reduction in average wait time. Approximately for the same load, since is not that simple to compute the load for finite buffer, the M/M/2 system structure with a single queue may introduce more variability and longer wait times due to a longer average service time and a less efficient load-balancing mechanism. The two M/M/1 queues, with a shorter average service time and a more balanced load distribution, may offer better performance in terms of average delay and packet loss.

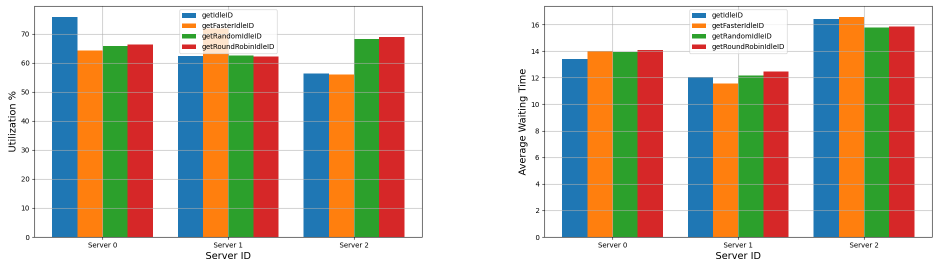
1.5 M/M/m

In the M/M/m queuing model we now consider a generic number of m of parallel servers. The analysis focuses on investigating the load distribution among servers depending on the different client scheduling algorithms.

Fig. 3 shows 2 graphs, In the first, the percentage of each server's total simulation time that is spent actively serving clients to various scheduling algorithms is displayed, while the second shows the typical amount of time a client must wait in line before a server attends to them in light of the various scheduling algorithms.

In the analysis of the approaches used, we consider an M/M/3 system with service time = 10 ms for server 1, service time = 8 ms for server 2, and service time = 12 ms for server 3. The arrival time is 10 ms.

The first analyzed approach is the "getIdleID", the idea is to scroll through the list of servers orderly and assign the client to the first free server encountered. As shown in figure 3a, the most used server is the first, while the least used is the last. The second algorithm is "getFastestIdleId", the client is associated with the fastest server if free, if busy it is associated with the second fastest server and so on. Here the most used server is the second one since it is the one with the highest service rate among the three. The third scheme is "getRandomIdleId", when a client is to be served, only free servers are considered among them the choice is random. As is shown in figure 3a with this method, server utilization is definitely more equitable despite the different service rates of different servers. A very similar result is obtained with the "getRoundRobinIdleId" approach. In this case, the idea is to assign to each server a specific position in a circular list. The subsequent request is associated with the server after it has been processed. This procedure guarantees that an equal number of requests are sent to each server. As for the figure, the choice of scheduling algorithm influences the average waiting time only in small part, what influences more is the different service times of the servers. Moreover, it is possible to observe that the "getFasterIdleId" approach performs significantly better when considering queuing delay because the servers that are busiest are also the most efficient. However there might be applications where the workload needs to be split evenly among the servers, and in that case, "getRandomIdleId" and "getRoundRobiIdleID" perform better.



(a) Server utilization by selection method for each server

(b) Average waiting time by selection method for each server

Figure 3: Comparison of different server selection methods

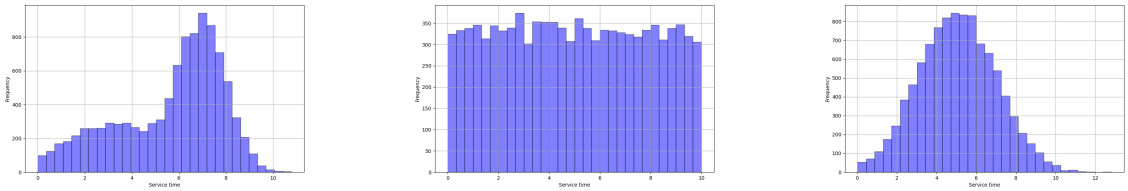
1.6 M/G/1

The service time distribution is varied to study the M/G/1 queuing system, and how the system performance changes are observed.

Fig. 4 shows three different distributions of service time, the first distribution is a composition of two Gaussian distributions, it can simulate a situation in which two different

types of packets must be handled. The other considered distributions are the uniform and the Gaussian. Fig. 5 shows the distribution of the total time packets spent in the system for the different distributions of service time, which includes both queuing and service times. This can provide insights into the overall performance of the system, particularly in terms of delays experienced by packets.

Based on both the tail length and the peak frequency, we can conclude that the third histogram (Fig. 5c) represents the distribution that performs best in terms of average time in the system. The intermediate peak frequency and the shorter tail indicate a better average time in the system compared to the other two histograms. In particular, this distribution effectively balances a high frequency of quickly completed events with a significant reduction in events that require extremely long times. Therefore, we can state that, among the analyzed distributions, the third one is the most efficient and balanced for optimizing the average time in the system.

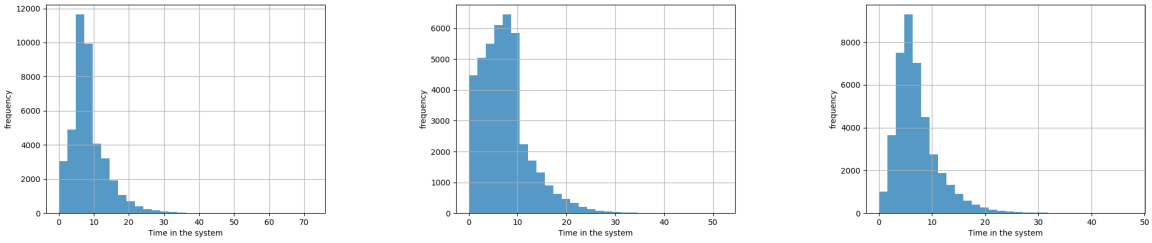


(a) distribution of two Gaussians composed for service time

(b) Uniform distribution for service time

(c) Gaussian distribution for service time

Figure 4: distribution of service time



(a) Time in the system given the two composed Gaussians distribution for service time

(b) Time in the system given the uniform distribution for service time

(c) Time in the system given the Gaussian distribution for service time

Figure 5: Distribution of the total time that packets spend in the system

1.7 Conclusion

This report underscores the importance of buffer management, server selection strategies, and service time distribution in optimizing the performance of UAV-assisted urban RANs.

The simulations provide critical insights for enhancing network efficiency and quality of service in real-world implementations.

2 Lab 2: Drone-assisted communication system simulation, Taking into account drone autonomy

2.1 Introduction

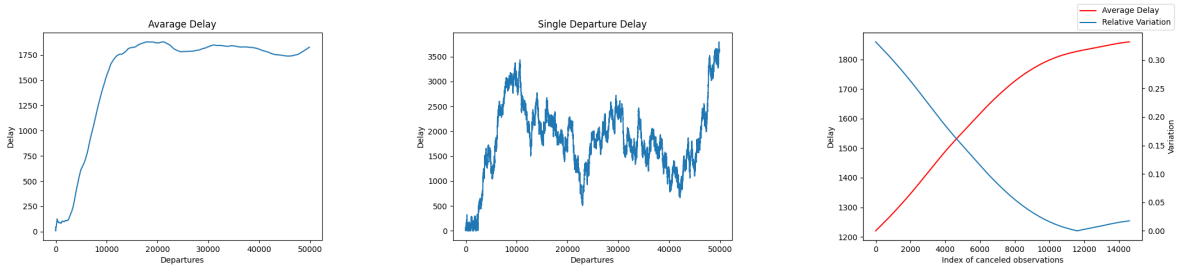
The laboratory's purpose is to describe a drone-assisted communication system in an urban area, where baseline mobile coverage is provided by on-ground Base Stations (BSs). During peak traffic, up to N drones equipped with BSs can be deployed to provide extra capacity and offload some traffic from terrestrial BSs. Our focus is on the aerial nodes, specifically the Unmanned Aerial Vehicles (UAVs). Each UAV is outfitted with multiple antennas and utilizes Multi-User Multiple Input-Multiple Output (MU-MIMO) technology. The MU-MIMO base station is represented as a queuing system with m servers, where each server corresponds to a single MIMO antenna. In this queuing system, the data packets arriving at a drone are the customers, while the service is the transmission of data by the UAV-mounted base station.

The lab activity involves implementing scheduling strategies to determine the optimal times to deploy drones to manage mobile traffic, considering system configurations and autonomy constraints, over a time window such as 8 a.m. to 8 p.m. in a business area.

2.2 Warm-up transient

The first point of the lab involves observing a situation with a single drone hosting a BS and equipped with a power supply, so a drone with an MM1 queue with infinite buffer size. The aim is to analyze how the warm-up transient affects the system. This represents the primacy effect of the early training examples: the time taken by the system to reach steady-state. In our case, to identify the transient, we used the variation of the average delay of packets during the entire simulation 6b. As shown in the figure 6a, the average delay calculated at each observation tends to rise until it stabilizes around a certain value; furthermore, it can be seen that the delay of individual packets varies until it stabilizes upon reaching steady-state (larger delays).

Based on the formula: $\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j$ for calculating the warm-up transient, we identified the point closest to the transient by observing the system with 20,000 samples. We then analyzed the relative variation of the average delay using: $\bar{R}_k = \frac{\bar{x}_k - \bar{x}}{\bar{x}}$. Where x_k is the average delay of packets calculated over a shorter simulation $\bar{x}_k = \frac{1}{n-k}\sum_{j=k+1}^{n} x_j$. The graph 6c was constructed with a range of values (samples) smaller than the entire simulation because, after the recorded point, the relative variation deteriorated due to the excessively small number of samples.



(a) distribution of the average delay of the system (b) distribution of the delay of single samples (c) Average delay over Relative variation

Figure 6: Warm-up transient

2.3 Single Drone in a business scenario

Considering the situation described in the previous point, we would precisely want to study a situation consisting of a drone being sent over a business area to handle some of the mobile traffic. In particular, the system presents an arrival rate of $1/10s^{-1}$ and a service rate of $1/10s^{-1}$ (both the drone and the base station). Considering that the drone is equipped with a battery that lasts up to 25 minutes, a recharging time of 60 minutes, and a buffer capacity of 10 packets, we want to analyze how the drone handles the traffic and how it can help limit the load on the base station.

2.3.1 Different Scheduling Strategies

The first requirement to be analyzed is to study how different scheduling strategies perform over that given area. Considering that the union of two Gaussian represents the traffic, and has two peaks around 10 a.m. and 3 p.m. of the working day, as it is shown in figure 7, the first strategy used was to send the drone continuously, even when the packets' arrival rate was perfectly manageable by the base station. The second one we analyzed was to send the drone into activity only when the packet arrival rate exceeds a certain threshold (in our case 16). Then, the third strategy we studied was to start the drone not only in the case that the arrival rate exceeded a certain threshold but also when the packet arrival trend approached the peaks. Thus, by observing the slope of the curve. The comparison of the performances is shown in table 5.

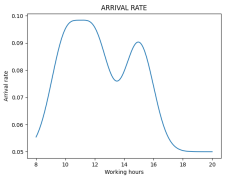


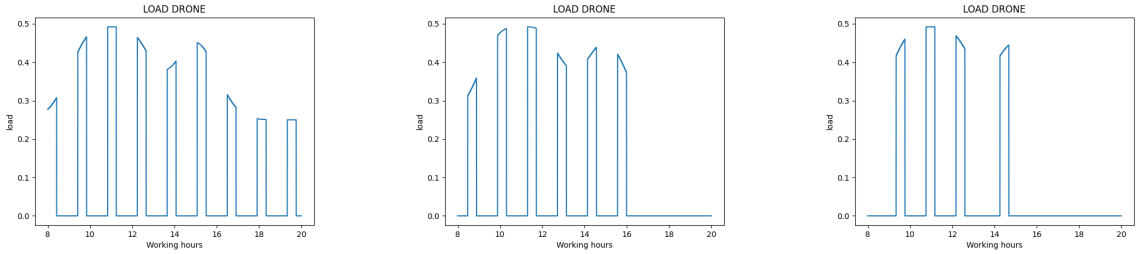
Figure 7: Arrival time distribution

Strategy	Average delay	N° of delivered packets	Drone losses	N° of take off
First	13.9244	528	6	9
Second	11.6328	404	3	6
Third	15.5690	300	2	4

Table 5: Strategies performances

Finally in the figure 8 we show the different loads of the drone,

depending on the chosen strategy. We observe that the different peaks constitute the time when the drone is in the air.



(a) First strategy (b) Second strategy (c) Third strategy

Figure 8: Load distribution of different strategies

2.3.2 Traffic Volume

To determine which strategy was best, we considered as traffic volume, the ratio of packets served by the drone to total packets served:

$$\frac{nofpacketsdeliveredbythedrone}{totalnofpacketsdeliveredbythesystem} \tag{1}$$

The table 6 shows the different traffic volumes and we observe that the most efficient strategy is the third one since, if we divide the number of packets served by the number of times that the drone was in service, we observe that 75 % of packages was served (compared to: 58.7 % of first and 67.3 % of the second). This is due to the fact that the drone only operates when traffic is at its highest.

Strategy	**Traffic Volume**
First	16.33%
Second	12.51%
Third	9.28%

Table 6: Traffic Volumes

2.3.3 Fixed number of cycles

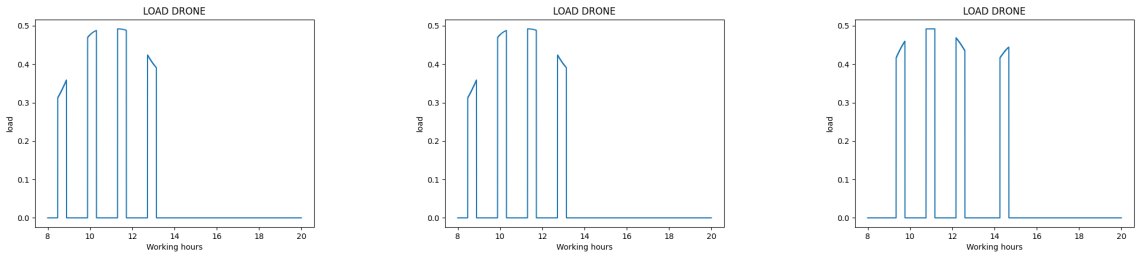
In the third request we analyzed the behavior of the three strategies with a fixed number of cycles for the drone (take off + recharge), 4 in our case. We observed that the third strategy continues to be the best in terms of the average number of packets served per cycle and traffic volume.

The different performances are shown in the table 7.

Furthermore, we note in figure 9 that the load of the drone (arrival rate/ service rate) varies depending on the strategy used and we observe that only in the last case is the drone in service during both peaks (10 a.m. , 3 p.m.).

Strategy	Average delay	N° of delivered packets	Drone losses	Traffic volume
First	13.2164	270	3	8.35%
Second	11.5994	281	3	8.68%
Third	15.5690	300	2	9.28%

Table 7: Strategies performances



(a) First strategy (b) Second strategy (c) Third strategy

Figure 9: Load distribution of different strategies

2.4 Drone equipped with PV panel

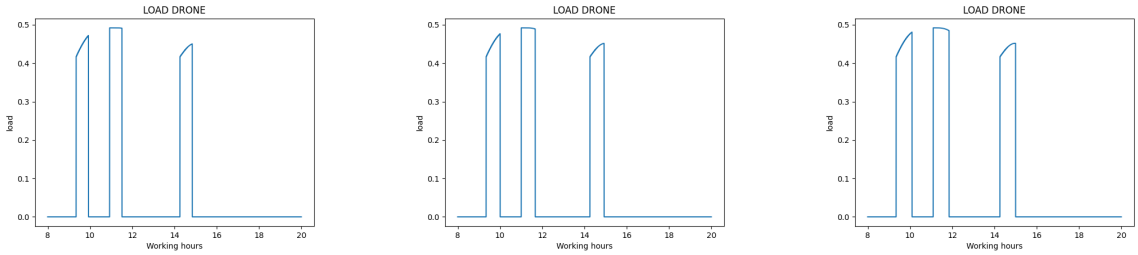
In the third point of the laboratory, we analyzed a scenario consisting of a drone, with the same characteristics as the previous points, but equipped with an additional PV panel. This panel adds autonomy to the drone based on the capacity (power) encapsulated inside it; a 45 W panel gives the drone an autonomy of 35 minutes, with 65 W of power it lasts 40 minutes, and with 75 W it reaches 45 minutes of autonomy.

The addition of the panel increases the duration of the drone's flight but, to avoid damage to the panel, after 4pm (after the second traffic peak), the drone's battery life returns to 25 minutes. So, under this condition, we studied the performances of the drone using the third schedule strategy, shown in the table 8.

Pv capacity	Average delay	N°of packets delivered	N° of take off	Drone losses	Traffic Volume
45 W	14.4830	313	3	1	9.69%
65 W	13.6431	352	3	2	10.89%
75 W	12.6846	387	3	2	11.97%

Table 8: PV panel performances

We have observed that thanks to the increase in the drone's flight time it is possible to significantly reduce the load managed by the base station, making the system more uniformly distributed and increase the number of packets delivered per cycle, making the system more efficient during peaks. It is possible to observe the increase in the flight time in the graph of the drone's load in the three cases in figure 10.



(a) Pv of 45 (b) PV of 65 (c) Pv of 75

Figure 10: Load distribution with different Pv panel

2.5 Scenario with N available drones

In the last point of the laboratory, still considering the business area, we studied a scenario consisting of N drones operating together. The analyzed scenario involves the use of drones with three possible configurations:

- Type A: the drone is powered by a battery only; its BS is equipped with 1 antenna featuring service rate 2µ, and the buffer size is equal to s_A;
- Type B: the drone is powered by a battery and a PV panel with a small capacity (45 W); its BS is equipped with 2 antennas featuring the same service rate µ, and the buffer size is equal to s_A;
- Type C: the drone is powered by a battery and a PV panel with a larger capacity; its BS is equipped with 1 antenna featuring service rate µ, and the buffer size is equal to $2s_A$.

where μ is equal to $1/20\text{s}^{-1}$ and s_A is equal to 10. To define which was the best configuration in terms of traffic volume handled during peak periods, we adopted the third strategy, as the departure schedule (arrival rate greater than $1/16^{-1}$ and slope close to the peak). We observed three different types of scenarios: ABC, ABB, and CCC.

1. ABC:

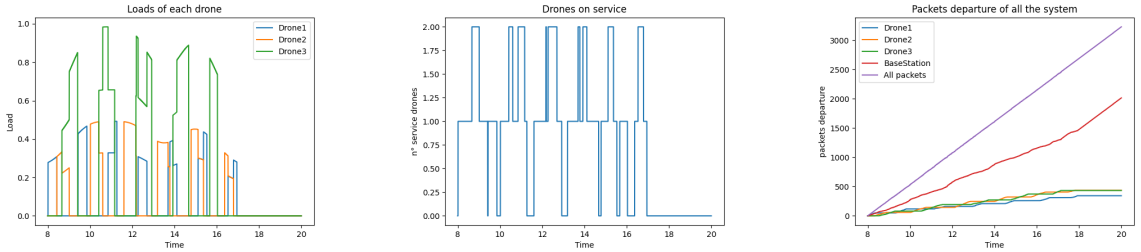
   The first configuration consists of having a scenario characterized by three different drones: i) typology A, ii) typology B, iii) typology C.

   The analysis of this scenario give the performances that are reported in the table 9. In this case, the losses linked to the third drone refer to the moment in which it was

Drone	Average delay	N°of packets delivered	Drone losses	Traffic volume	Total traffic Volume
D1	10.8880	344	2	10.64%	
D2	21.8392	438	2	13.55%	37.59%
D3	150.844	433	6	13.40%	

Table 9: ABC performances

charging. This is due to the fact that being slow to serve, all packets not served during the working period are lost. Instead, figure 11 shows the behaviour of the drones of this first configuration.



(a) Load of drones (b) N° of working drone (c) N° of packets delivered

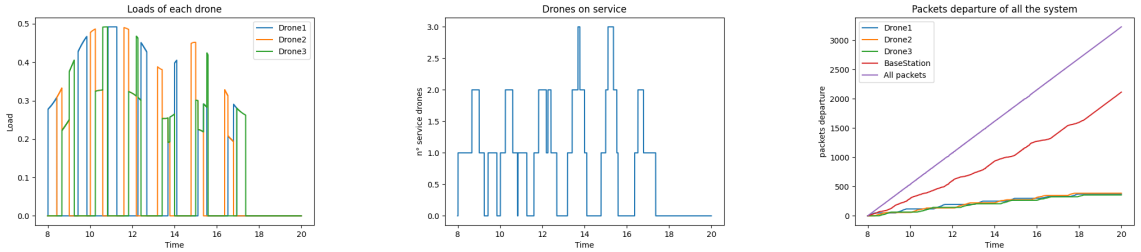
Figure 11: Behaviour of the drones with ABC network typology

2. ABB:

The third configuration consists of having a scenario with two different drones: one of typology A and two drones of typology B. The corresponding results are shown in the table. 10 Again, the behavior of the drones is shown in the figure 12

Drone	Average delay	N°of packets delivered	Drone losses	Traffic volume	Total traffic Volume
D1	12.1399	372	3	11.58%	
D2	23.3770	387	1	11.98%	34.54%
D3	32.2725	357	1	11.05%	

Table 10: ABB performances



(a) Load of drones (b) N° of working drone (c) N° of packets delivered

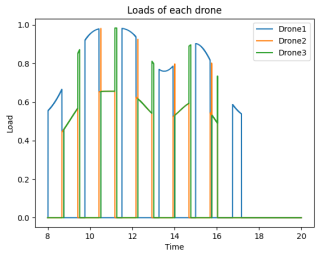
Figure 12: Behaviour of the drones with ABB network typology

3. CCC:

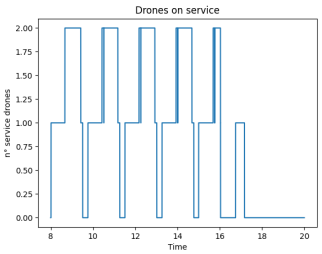
The last configuration consists of having a similar scenario as the second one but the drones of the third typology C. The analysis carried out with this type of network reports the following results in the table 11. The behavior of the drones is shown in the figure13

Drone	Average delay	N°of packets delivered	Drone losses	Traffic volume	Total traffic Volume
D1	102.630	597	3	18.47%	
D2	36.4491	323	2	9.99%	40.03%
D3	82.0496	374	1	11.57%	

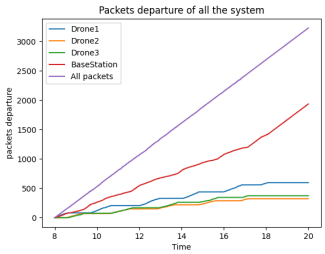
Table 11: CCC performances



(a) Load of drones



(b) N° of working drone



(c) N° of packets delivered

Figure 13: Behaviour of the drones with CCC network typology

Looking at the performances reported in the various tables, we note that the third configuration (CCC) is the most efficient one because it manages to serve more packets, thanks to the higher battery level which allows drones to remain in flight for longer during peaks. However, if we consider the quality of service (avg delay) the ABB configuration appears to be the best but is the worst in terms of traffic volume. Therefore we believe that the CCC configuration is the best compromise between quality and traffic volume, even though the load of the drones reaches the maximum value multiple times deteriorating the health of them.

2.6 Conclusion

The aim of this laboratory was to determine, at each step, which scheduling or drone configuration strategy was the most efficient. In fact, we have underlined how the system (drone + base station) changed in terms of: average delay or packets, number of packets delivered, losses, and traffic volume depending on the choices made. In particular, we focused on the study of drones characterized by an MMm queue in which the load, used to show the results of the different scheduling strategies, was referred to the simplified case $\frac{\lambda}{m\mu}$. Thanks to the load we were able to determine which strategy was the best depending on the case.

Therefore, determining which network system performs best is fundamental in real cases, because having an efficient system means having excellent infrastructure.