UNIVERSITÀ DI PISA

Computer Engineering

Performance Evaluation of Computer Systems and Networks

Aerocom System

Project Documentation

Gaia Anastasi

Clarissa Polidori

Leonardo Poggiani

Academic Year: 2020/2021

Table of Contents

1 | Introduction

Aerocom System is a communication system between some *aircrafts* (**AC**) and a *control tower* (**CT**). ACs have N *data link* (**DL**) available for communication with the CT, and can use only one of them at a time as serving DL. Each AC communicates with a control tower (CT) generating one packet of **fixed size** every **k** seconds. Each DL has a **time-varying capacity**: periodically a new **target capacity** is selected and the capacity will **linearly change** from the current one to the target one, which is reached after **t** seconds.
The system has two modes of operation:

- The AC keeps the same serving DL for the whole simulation;
- The AC constantly monitors the service time of DLs and before each transmission it selects the DL with the **highest actual capacity**. Monitoring DL **service time** gives a **malus X** to capacity.

1.1 Objectives

The goal of this study is to evaluate the end to end delay between ACs and the CT and the queue length varying the **interarrival time of packets**, the **monitoring period**, the **number of DLs** and the **number of ACs** knowing that the **service time** is given by the packet size (Byte) over the DL's actual capacity (Byte/s), where actual capacity is the capacity of the DL at the moment of forwarding the packet, a value that varies linearly between the previous and the new one.
For each operation mode described in the previous section, we analyzed the behaviour in two different scenarios:

- *Exponential* distribution of interarrivals, *uniform* DL's capacity and *Exponential* distribution of capacity setting time.
- *Exponential* distribution of interarrivals, *uniform* DL's capacity and *lognormal* distribution of capacity setting time.

In each scenario, the response time was evaluated according to the variation of malus $\boldsymbol{X}$, the number of DLs $\boldsymbol{nDL}$ and the monitoring time $\boldsymbol{M}$.

The source code and all the files used for the subsequent analysis can be found at `https://github.com/leonardopoggiani/PECSNproject`.

2 | Model

2.1 Assumptions

- Communication channel between an aircraft and the control tower can be considered **ideal**, i.e. no delay and no BER.
- Our evaluations and analysis are made for a system composed of an aircraft and a control tower and since as we have decided to structure the model, the behavior of an aircraft does not influence the behavior of others. In particular, we want to evaluate what happens inside the aircraft from the creation of the packet until its sending to the control tower.
- We have 3 different random number generators (RNGs) for: *interarrival times*, *capacity setting time*, and new *capacity to select*.
- The capacity of a Data Link is considered as the bytes per second that a DL is able to forward. So the Service Time of our system (the time the DL spends to send a packet) it's computed in the following way:
  - Size: packet's fixed size $[byte]$
  - Capacity: DL's capacity $[byte/s]$

  $ServiceTime = Size/Capacity\ [s]$

Parameter	**Values**
System operation mode	0 or 1
New capacity generation mode	*"exponential" or "lognormal"*
Mean interarrival time of packets	k
Mean setting capacity time	t
Extremes for the target capacity extraction	dimPool
Packet size	s
Number of DLs	nDL
Number of ACs	nA
Malus of service time due to monitoring	X
Monitoring time	m

Table 2.1: System parameters.

2.2 Components

The core components of the model are Control Tower, Aircraft, Packet Generator, Link Selector, Data Link and Packet.

- **Control Tower**: It is a module that only receives the packets and drops them, it is not the focus of our evaluations.
- **Aircraft**: It is the main module of our analysis, it is a compound module composed of 3 simple modules: Packet Generator, Link Selector and Data Links. The number of Data Links is a system parameter.
- **Packet Generator**: It handles packet generation with exponential interarrival distribution, whose mean is defined by a parameter. It then forwards the packet to the link selector.
- **Link Selector**: Link Selector is represented by a queue managed in FIFO mode, whose server is responsible of forwarding the packets. In the second operation mode, the Link Selector periodically monitors the current capacity of each data link and keeps track of the data link with the highest capacity, leading to a malus on the service time. The forwarding of packets is paused for the time necessary to serve the penalty due to the malus, and then normal operation resumes. When it receives a new packet it queues it. The service consists in forwarding the packet on a data link with the highest capacity with a service time proportional to that capacity. Otherwise, if we are in non-monitoring mode, it always forwards the packet on the same link chosen at the beginning.
- **Data Link**: Each aircraft has a certain number of data links and each of them is responsible of selecting a new capacity, that will be reached after a time t. The capacity varies linearly from the old capacity to the new one.
- **Packet**: A packet is the information unit we are considering in this work. It is created by the packet generator and its size is fixed.

3 | Model Validation

In order to validate the model, we performed the following:

- **Code verification**: The simulator was been checked for bugs and memory leaks using Valgrind. In order to verify the accurancy of our code, the service time for a simulation was recorded, and a time interval was observed to detect possible issues.
- **Degeneracy test**: A configuration with 0 aircrafts and another one with 0 datalinks were tried, and the correct functioning of the system was checked.
- **Packet loss test**: Aa simulation was run in order to measure the number of packets sent and the number of packets received, expecting them to be equal.
- **Little's law test**: Performance indexes results obtained from the simulation was compared with the one computed using Little's law.
- **Continuity test**: The simulation parameters were slightly varied, expecting the output not to behave wildly.

#DL	16
Mean int-time	50ms (may vary)
Mean set-capacity time	2s
Sim-Time	400s
Repetition	10

Table 3.1: General test scenario

3.1 Preliminary parameter tuning

Stability: For simplicity, we report here only the case of the scenario with non-monitoring operation mode, in both cases mean setting capacity time *exponential* and *lognormal*. In order to find the stability condition, several simulations were performed with different mean interarrival times. For the exponential setting capacity time , it has been discovered that stability condition is achieved at $k > 8ms$ as we can see from Figure 3.1

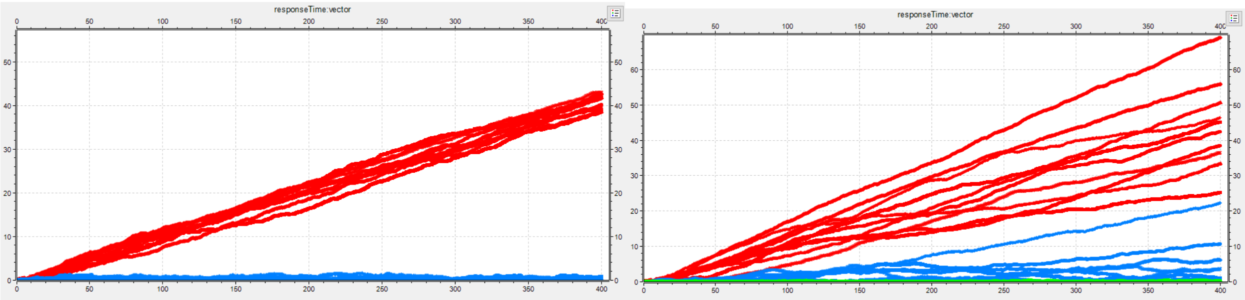

Figure 3.1: Response time around stability point, the red curve is $k \approx 7$ms, the blue curve is $k \approx 8$ms and the green curve is $k \approx 9$ms

Using the same method, we demostrated that stability is reached at $k > 8ms$. even in the *lognormal* case. Although the system is stable at $k \approx 8ms$ it behaves too wild among different repetitions (Figure 3.2), so we picked $k \approx 9ms$ as the worst stable case.

To perform early test on our system we chose the parameters of the simulation in order to have meaningful yet "debuggable" results. Therefore exponential distributed capacity setting time was used, 10 repetition combined with 400s of simulation time in order to have good coverage of various trajectories for our system.
We fixed mean setting capacity time at 1ms and we picked k = 50ms to stay very far enough from limit conditions.

3.2 Code verification via service time

In order to verify the correctness of our code, we recorded the service time for a simulation and looked at a small time interval in order to spot wrong situations.

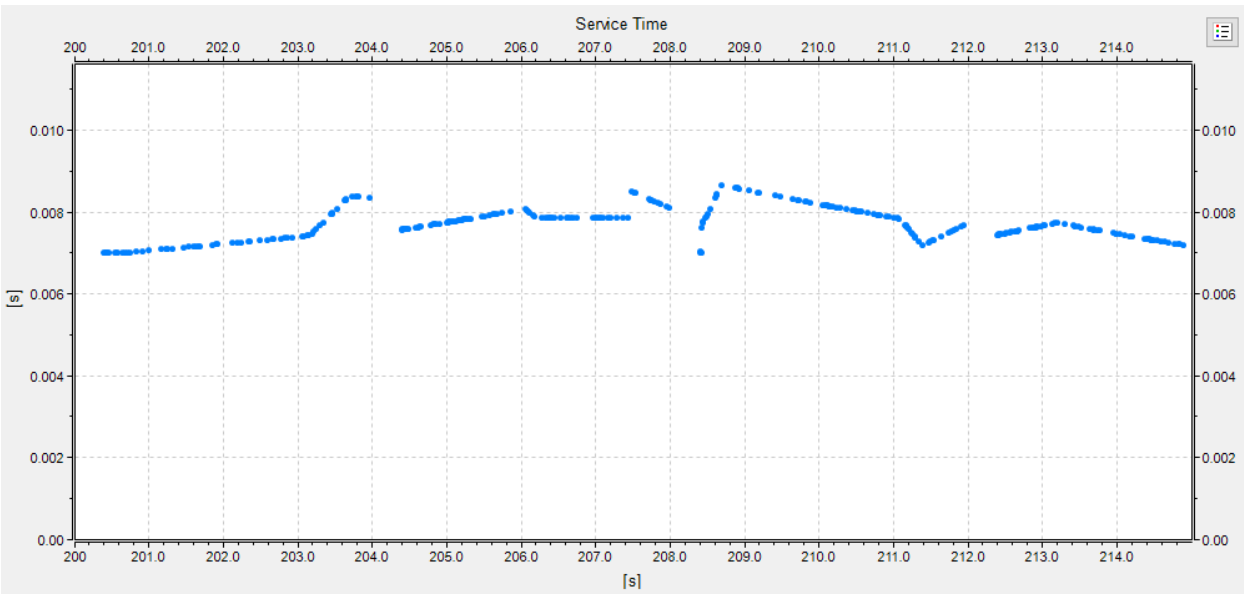

Figure 3.2: Service time during a test simulation

We were able to notice that:

- The service time has a linear shape as we expected since the service time is inversely proportional to the capacity, which has a value that varies linearly from the two target capacities.

- Graph discontinuities, where there are no emits on service time, represent the penalty time, which can be observed after a monitoring; service time correctly decreases after this period.

At this point, we are confident that the code for our simulator is correctly modelling the real world scenario, thus we can go through a more in-depth inspection by performing some validation test.

3.3 Degeneracy Test

3.3.1 Test 1

Before sending a packet generated by the packet generator, the system checks if there is at least one DL to send it to, otherwise it deletes the packet. With zero packets, all the statistics found within the Data Link cannot be calculated, those within the other modules have the following behavior:

Test scenario:

#DL	0
Mean int-time	${20,21,22,23,24}ms

Table 3.2: Degeneracy test scenario

Experiment	**Module**	**Name**	**Value**
Exponential capacity setting time	Control Tower	packetsReceived:count	0
Exponential capacity setting time	Link Selector	responseTime:vector	0.0 (Mean)
Exponential capacity setting time	Link Selector	waitingTime:vector	0.0 (Mean)
Exponential capacity setting time	Link Selector	queueLength:count	0

Table 3.3: Degeneracy test result

No trace of the packets remains in the memory, so we have no **memory leaks** and the system does not have any strange behavior.

3.3.2 Test 2

To verify the correctness of our model, we also verified if the behavior in the scenario without monitoring is equivalent to the one in the scenario with monitoring with a very high monitoring interval (M). What actually happens, as it is shown in Figure 3.3, is that the two systems are equivalent regardless of the value of k.

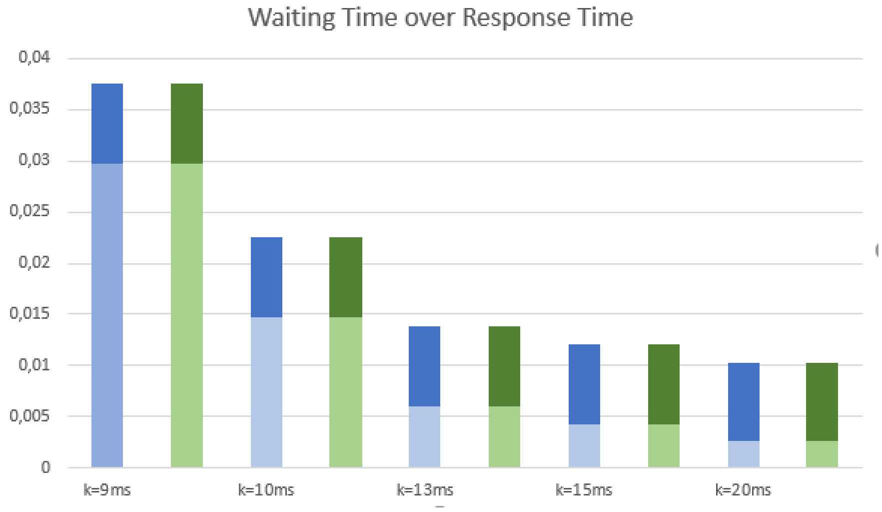

Figure 3.3: Response time and Waiting Time in Non-monitoring mode and Monitoring Mode with m=400s

3.4 Little's law test

Capacity pool	[8000-10000]

Table 3.4: Little's Law test scenario

Knowing that each Data Link will choose its next capacity from a pool with a uniform distribution, in order to simplify Little's law computations we set the DL capacity on the mean value of the pool that is 9000.

μ	$E[ts] = 1/\mu$	$\rho = \lambda/\mu$	E[N]	$E[Nq] = E[N] + \rho$	$E[R] = E[N]/\lambda$	E[W] = E[R]-[ts]
128.57	0.0078	0.1556	0.0143	0.0085	0.007	0.1361

Table 3.5: Values computed by Little's law

Rep	$E[ts] = 1/\mu$	$\rho = \lambda/\mu$	E[N]	$E[Nq]$	$E[R]$	$E[W]$
1	0.0078	0.1556	0.1704	0.0148	0.0085	0.007
2	0.0078	0.1556	0.1679	0.0124	0.0084	0.007
3	0.0078	0.1556	0.1690	0.0135	0.0085	0.007
4	0.0078	0.1556	0.1717	0.0161	0.0085	0.007
5	0.0078	0.1556	0.1684	0.0128	0.0084	0.007
6	0.0078	0.1556	0.1722	0.0166	0.0085	0.007
7	0.0078	0.1556	0.1710	0.0154	0.0085	0.007
8	0.0078	0.1556	0.1690	0.0135	0.0085	0.007
9	0.0078	0.1556	0.1695	0.0140	0.0085	0.007
10	0.0078	0.1556	0.1695	0.0140	0.0085	0.007

Table 3.6: Values obtained from measurements in the simulation

3.5 Continuity Test

Test scenario:

Mean int-time	\${50,51,52,53,54}ms

Table 3.7: Continuity test scenario

Finally, we plotted the response time (Figure 3.3) to show that the system does not behave too differently varying the interarrival time

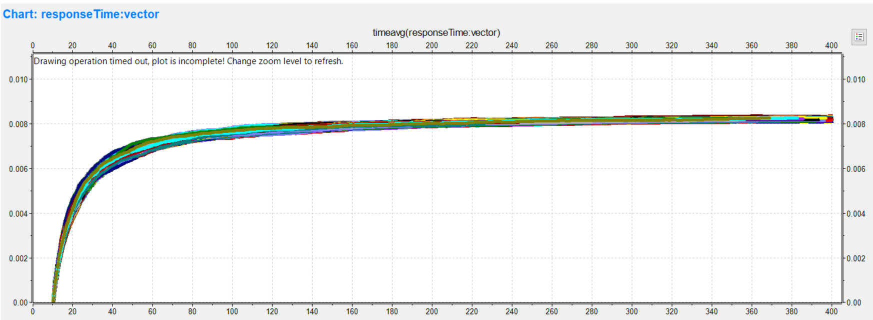

Figure 3.4: Continuity test

4 | Warm-up Analysis

In order to perform warm-up analysis, we considered the two main performance indexes of our work: **response time** and **queue lenght**. The stability of both of these indices is very load-dependent and become totally unstable if condition was not met. Under this considerations, we extimated the warm-up time for the system with a value very close to the stability limit that is k = 15ms. For simplicity we decided to report here only the scenario where the system works in **non-monitoring** mode, but we still made sure that similar things are registered in the case with monitoring . As Figure 4.1 shows, for each performance index, the case with *exponential* and *lognormal* capacity setting time. The following plot was obtained in a simulation with 10 repetiotions because we noticed that 10 repetitions were enough to verify the behavior of the system.

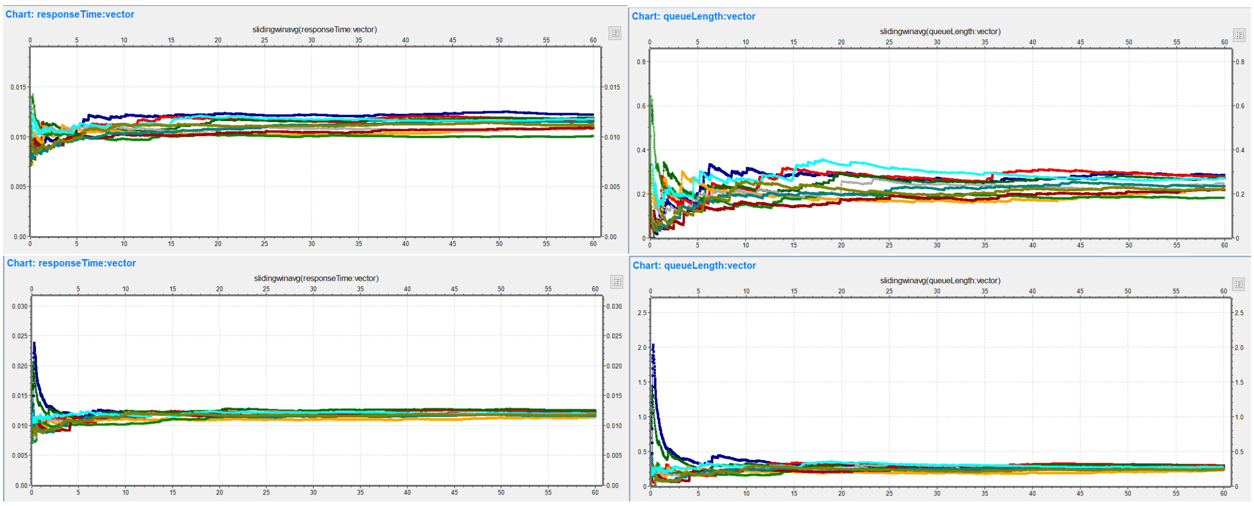

Figure 4.1: *lognormal* and exponential capacity setting time warm-up

As we can see from plots above, warm-up period can be safely extimated to be around 10s for the queueing lenght, and this holds also for response time.

5 | Validation Experiments

5.1 Ratio between X and m

Conducting experiments varying the values of the malus X and the monitoring time m, we realized the close relationship between these two and their importance for what concerns the response time. We also realized the need to establish a driving rule for our system, in order to avoid wrong and unpredictable behaviors. In fact, it is important to choose values of X always lower than m to assure the correct behavior of the system. Plus, if the difference between X and m is greater the system behavior will work better because close values lead to errors in response time detection. Obviously values of X and m had to be chosen adequately because assuming a fixed X that cannot be modified or increasing too much m makes monitoring meaningless. This aspect will be heavily discussed in the next paragraphs.

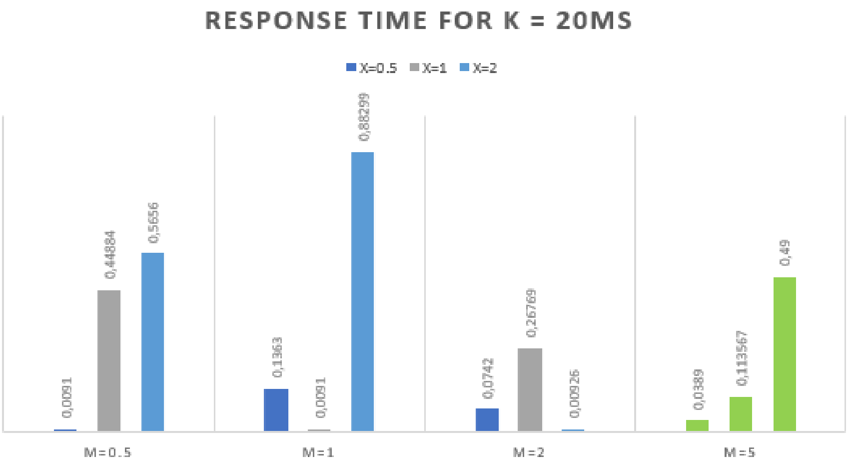

Figure 5.1: Response time for k=20ms and various values of X and m.

6 | Experiments

After the simulator validation, experiments were carried out in order to study the system performance with a particular attention on what concerns the end-to-end delay of packets delivery, and the queue length. For every parameter,100 independent simulations were run.

Parameters	**Values**	**Unit**
Mean interarrival time (k)	${9, 10, 13, 15, 20}	ms
Monitoring period (m)	${0.5,1.5,1.8,4.5,12,20}	s
Penalty time (X)	${0.01,0.05,0.5,1,2,10}	s

- We want a DL which selects its new target capacity at most every 2s, for air traffic control necessities. We analyzed k values with a lower bound of 9ms because with a lower value than this the system started to become unstable due to very high utilization ρ;
- We wanted to study how the system behave varying the monitoring interval and penalty. How these values were chosen and the relationship between them is already been described in the previous chapter.

6.1 Preliminary experiments

6.1.1 Waiting Time over Response Time

To fully understand the behavior of our system varying the workload and monitoring modes, we decided to compare the waiting time with the response time obtained using different values of mean interarrival time k and monitoring time m. In particular, what we want to study is how each components of the response time affects the latter knowing that the response time is composed by service time and waiting time. From the design of our system, we already know that the waiting is affected from both the interarrival time of packets k and the malus X due to monitoring.

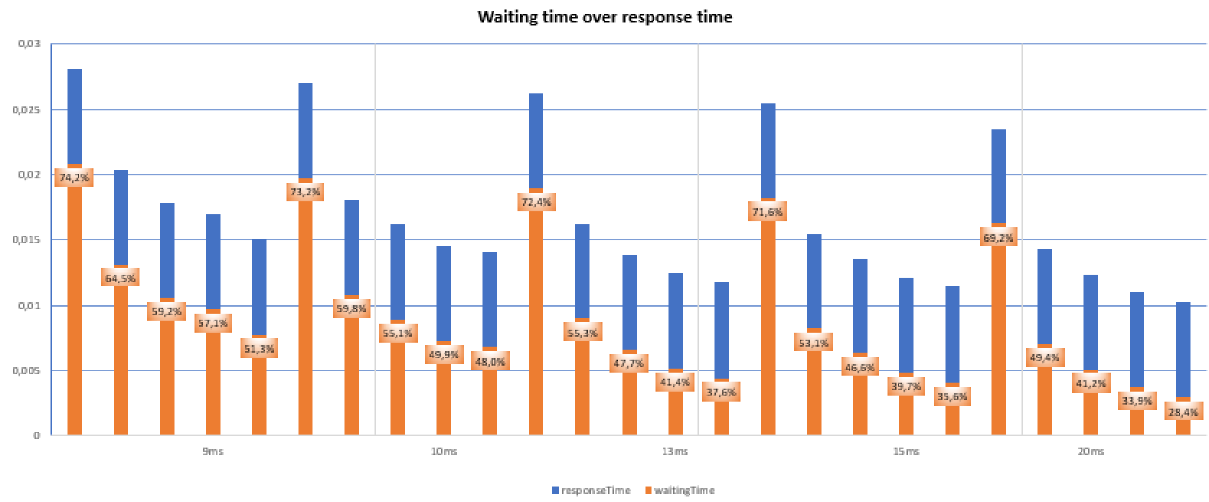

Figure 6.1: Waiting Time over Response Time

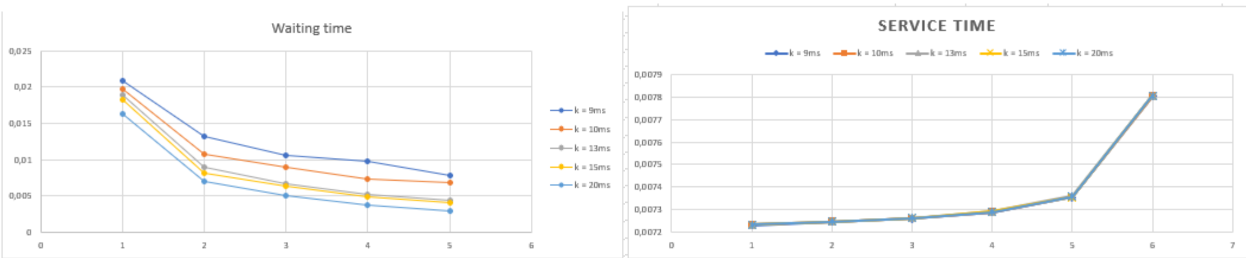

Figure 6.2: Waiting time variation and Service time variation

By analyzing the graph we assume that:

- The percentage of waiting time over response time decreases as k increases. This is due to the relationship between waiting time and k. Waiting time depends on the speed at which packets arrive so it decreases if k increases because packets arrive with a lower rate. However, the decrease in response time is smaller compared to what could be expected and this because of the small influence the queue had on response time compared to the influence of the part of waiting time due to malus.

- The percentage of waiting time decreases when an increasing of m is registered. Monitoring less often, the malus penalty to be discounted decreases and waiting time decreases in turn.

- As k increases, the queue tends to 0 and the waiting time will be due only to the penalty of the malus.

- Another main aspect is how our service time grows when m increases. This seems strange since monitoring the channel less often will not let us take advantage of monitoring and will not let us use the data link with the highest capacity to forward packets but the reason behind this is that the increase in service time due to monitoring is really negligible compared to the entire response time. Monitoring leads to an improvement in service time very negligible and, in addition, it introduces a substanctial worsening with the malus. Here we have just presented a brief overview of this topic, which will be discussed further in section 6.4.

6.1.2 Varying the number of Data Links

As shown in the following figure, when the system is working in monitoring mode the response time decreases as the number of Data Links increases, this because it increases the possibility of having DL that have chosen a very high capacity and decreases the probability that all DL will choose low or similar capacities.

However, the improvement is very negligible as the number of data links is not one of the factors that particularly affects the system. Plus, the capacities that are selected by each DL are extracted from a fairly small pool so they do not vary significantly. This aspect will be analysed in detail in paragraph 6.4. We therefore decided to set the number of data links in order to analyze the behavior by varying more significant parameters. Considering the data links as interfaces of the plane it is very unrealistic to set this number at limit values such as 2 or 400 or 1000, 16 seemed to us the right compromise.

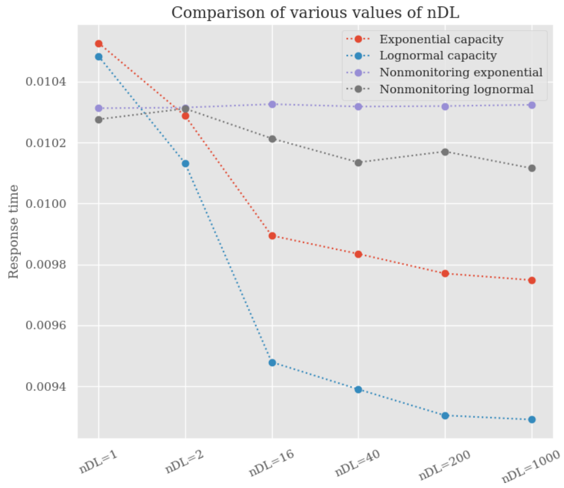

Figure 6.3: Response Time varying nDL

6.2 Non-monitoring Mode

We briefly analyzed the non-monitoring operation mode in our system. In this case, the response time behavior and the queue length are only affected by mean interarrival time, as we found that they are not affected much by the mean setting capacity time distribution. We decided to not report the graphs related to non monitoring operation mode here and to focus more on the monitoring operation mode because it offers wider reflections.

6.3 Monitoring Mode

6.3.1 Varying both monitoring period (m) and mean interarrival time (k)

First, we studied the response time behavior and the queue length varying both monitoring period and mean interarrival time. The first two graphs refer to response time in both possible distributions of t: *exponential* and *lognormal*, while the last two are about the queue length, again for both distributions of t. As we can see from the graphs the *lognormal* scenario involves a minor response time and a smaller queue. It was logical to expect, as well as visible, the behavior and the shape, that these indices assume, remain almost unchanged. We can made some considerations and apply them to both scenarios:

- When the mean interarrival time increases, the response time decreases and consequently the length of the queue decreases.
- When the monitoring period increases, the response time decreases. This is surprising because we actually expect the opposite in a system that introduces monitoring in order to increase performance. This is due to the presence of malus which is very influential on response time.

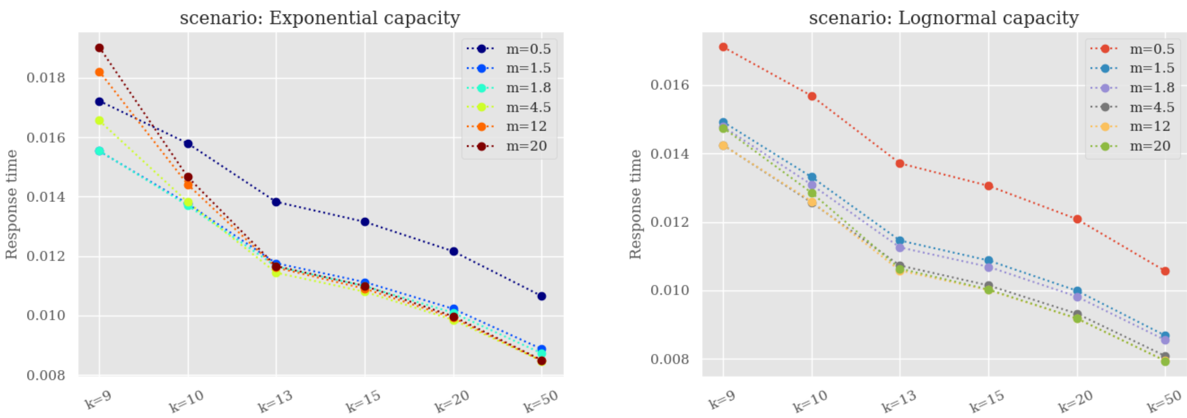

Figure 6.4: Response time in Monitoring Mode

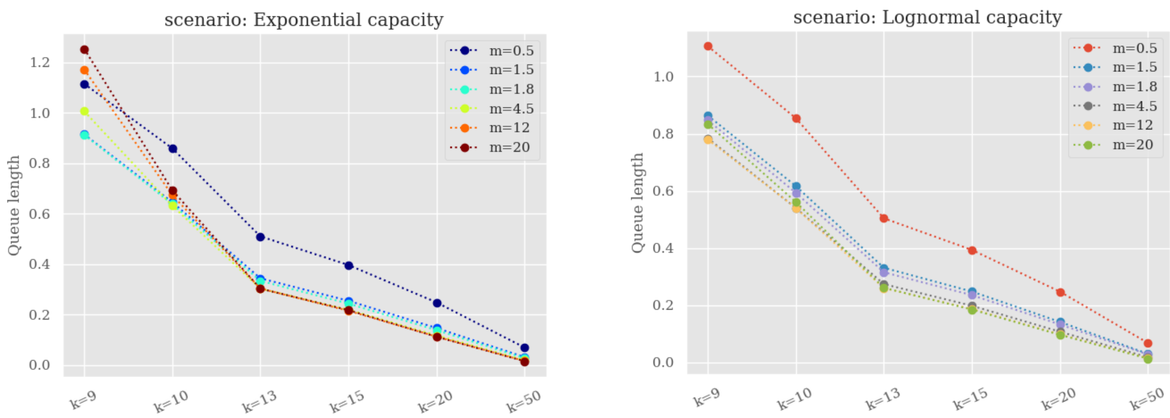

Figure 6.5: Queue Length in Monitoring Mode

6.3.2 Varying the malus penalty (X)

In this paragraph, we would vary the value of the monitoring penalty which is a very influential factor for our system, as we have abundantly demonstrated. To do this we set the monitoring period value m taking into consideration the results obtained from previous experiments. In particular, we chose m=4.5s for the case with *exponential* distribution t and m=12s for the case where the distribution t is *lognormal*. Here, for simplicity, we report only the results for the *lognormal* case, as the *exponential* one has the same behavior.

The following graphs demonstrate once again how results obtained when of X is greater than m cannot be taken into consideration. As a matter of fact, we can easily see that in the case in which X=10s there is an undefined behavior. In particular, we can see from the graph how response time and queue length decrease in this case while we expect instead the opposite behavior.

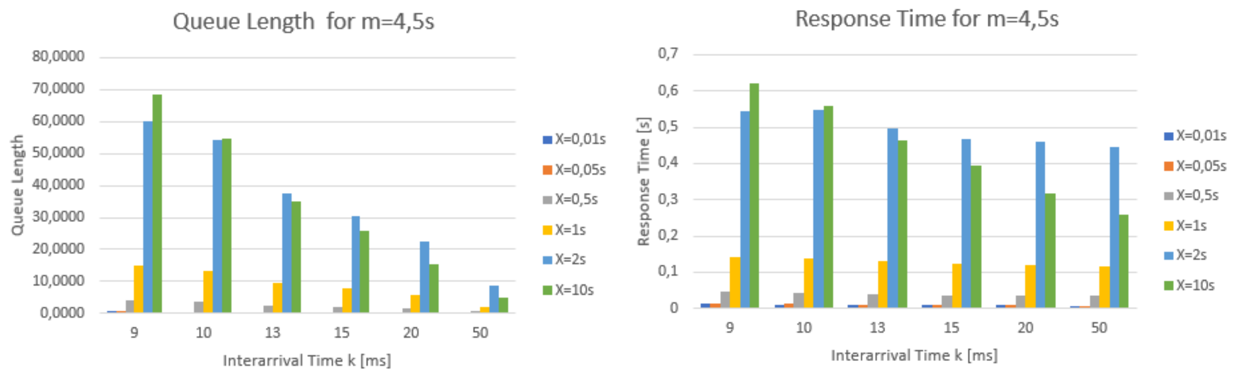

Figure 6.6: Response time and Queue Length with different value of k and X

The following graphs show how response time and queue length vary with different values of X and k. For security reasons, we have only considered values of X conforming to the assumptions made up to now, so values of X lower of m.

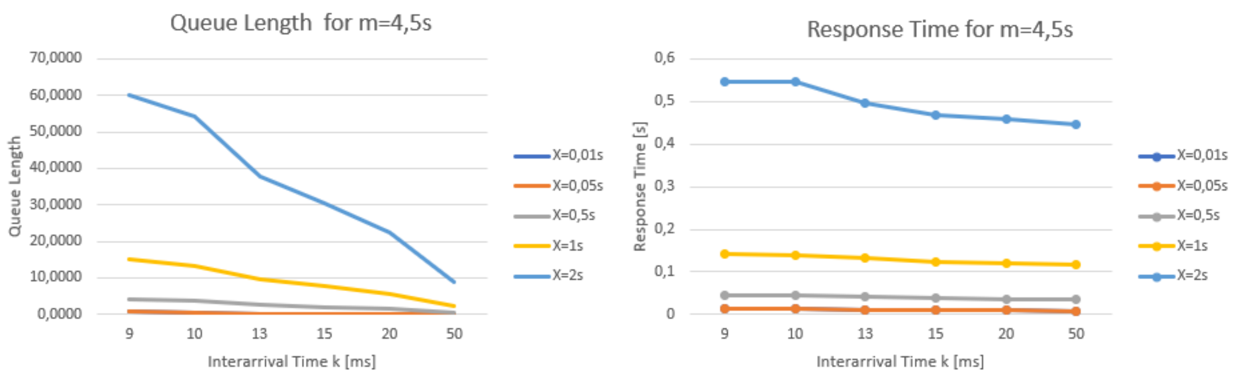

Figure 6.7: Response time and Queue Length with different value of k and X

From the graphs we can notice that when X decreases, also the response time and queue length decrease. Plus, More X moves away from m and more the decrease becomes negligible to an extent that it does not have any influence on response time and queue length anymore. Under these circumstances, queue length and response time reach very low values tending to 0, as we have reached a stage in which we can benefit of all the advantages of the monitoring scenario at the cost of a negligible penalty.

6.4 Final considerations

What we would like from our system is that the response time begins to rise to the increase of the monitoring period because it would mean that a long monitoring time leads us to keep the same DL even if the capacity of the latter comes to be very low and therefore the service time very long. What must happen is that the monitoring time is selected in order to be able to obtain the greatest possible benefit considering a fixed penalty and on which it is not possible to intervene.
In any scenario without monitoring both indices are worse than the configuration with monitoring, otherwise the choice of the latter mode of operation would be in any case not recommended.

6.5 Possible improvements

During the simulations we realized the impact of the malus penalty on the response time.
When the penalty period was greater then more packets were added to the waiting time, which turned out to be the fundamental component of the response time.
Assuming we cannot intervene on malus and we have to keep a fixed malus period due to physical problems of instrumentation or channel, it remains only another factor that has relevance on the efficiency of monitoring and on which we can intervene: the capacity.

- To simulate the varying capacity of datalinks we have chosen a pool of capacity ranging from a minimum of 8000 $\frac{byte}{seconds}$ (chosen for reasons of system stability) up to a maximum of 10000$\frac{byte}{seconds}$.The width of the pool turned out to be too small to actually take advantage of the monitoring as the fundamental benefit of enabling monitoring is the lowering of the service time following the choice of the transmission channel with higher capacity. If this benefit is too reduced compared to the penalty to be discounted we may not feel benefits and even worsen the performance of the system.

For this reason, in order to show in more detail the possible benefits brought by the monitoring, we decided to carry out some experiments varying the size of the capacity pool. In reality, this could coincide with a rather unstable channel, where the capacity can swing a lot. In particular, we tested a configuration that include a minimum value of 8000 $\frac{byte}{seconds}$ and a maximum of 20000 $\frac{byte}{seconds}$.
At this point we have tried to confirm our initial hypothesis, that is the one according to which, as the monitoring time increases, the benefits are reduced, but as it decreases, the influence of malus becomes too consistent on the response time.

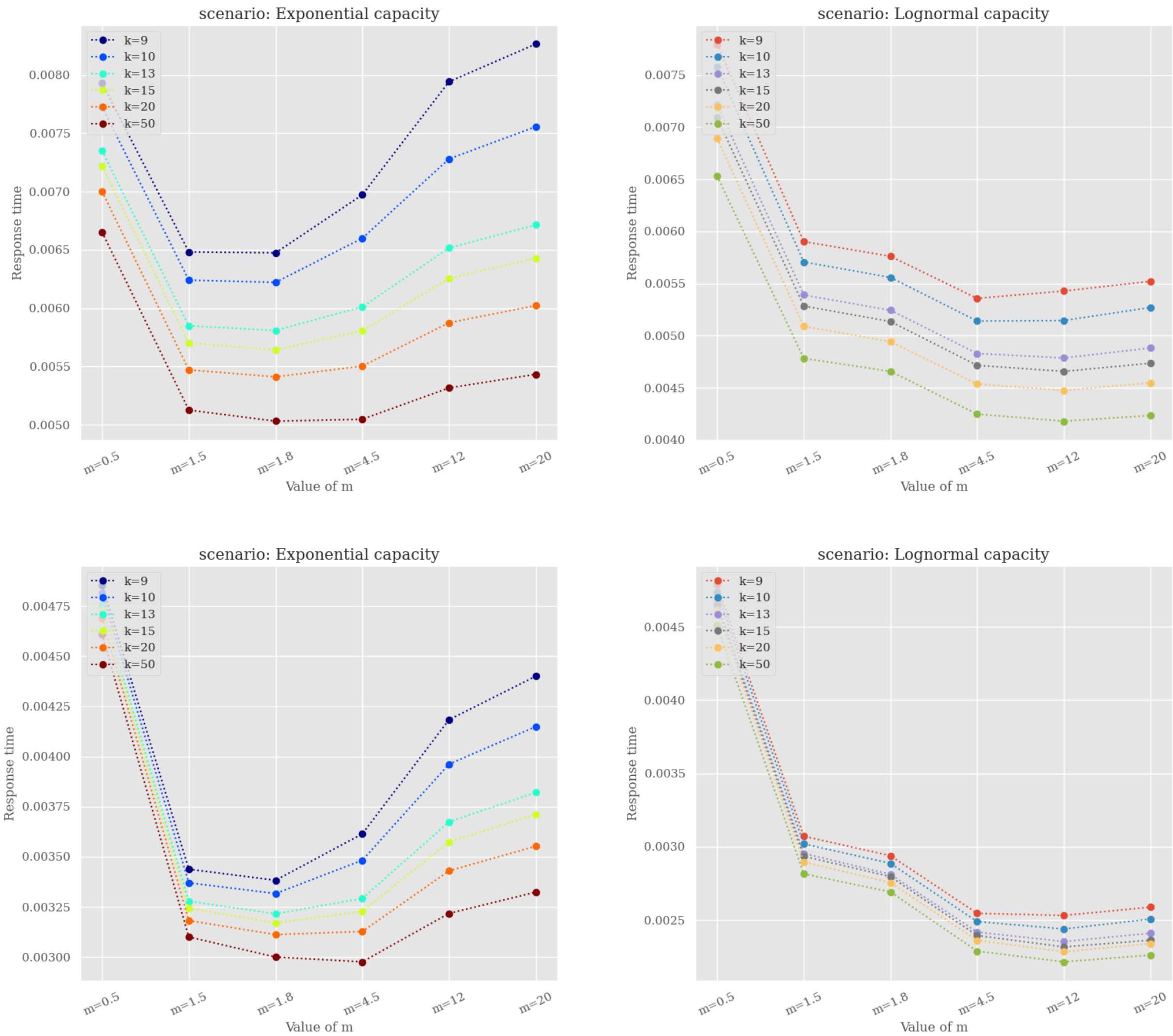

Figure 6.8: Response time, first two pictures with a pool maximum of 20000 last two with a pool maximum of 40000.

As the graph clearly shows, the optimal value of m is different for *exponential* and *lognormal* distribution of t. In particular, for the *exponential* it is between 1.5s and 1.8s, while for the *lognormal* is around 4.5s.

Generally the configuration with *lognormal* distribution of t is more convenient in terms of response time compared to the *exponential* one, at least for the values of m analyzed in this situation.

7 | Factorial Analysis

Results clearly show that penalty due to the malus and monitoring period has a huge impact on our system, so we expect that performing a factorial analysis should yield the same results.
Our study was conducted for both *lognormal* and *exponential* case on the 5 factors: mean capacity setting time t, mean interarrival time k, monitoring period m, penalty time X and number of data link nDL and, using 30 repetitions.
Factors' extremes:

Parameters	**Values**	**Unit**
Mean set-capacity time (t)	${0.5,3}	s
Mean interarrival time (k)	${9,50}	ms
Monitoring period (m)	${0.5,20}	s
Penalty time (X)	${0.01,2}	s
# DL (nDL)	${2,50}	

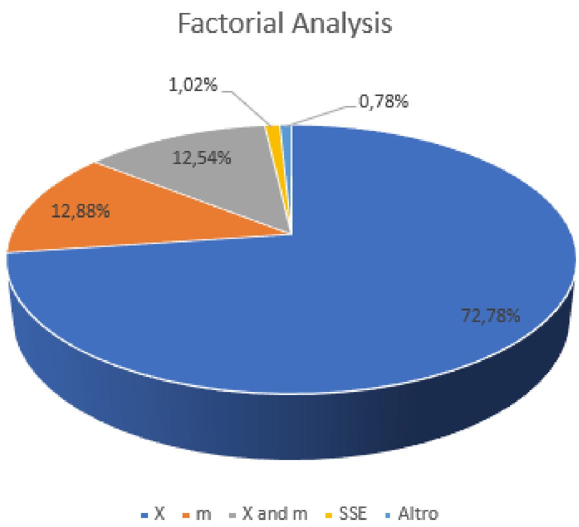

Figure 7.1: Factorial Analysis

8 | Conclusions

Aerocom System is a communication system in which each aircraft communicates with a control tower through a data-link. The system we have studied seems more efficient with high values of the monitoring period and this is because, as we have already said before, our capacity pool is very small. Uder these circumstances, monitoring doesn't bring much benefit because the small improvement in the service time we manage to get choosing the data link with the highest capacity would be wasted because of the malus. However some improvements can be made to modify the system and try to get more benefits from the monitoring mode. In the analysis of our system we are able to fix the time-varying capacity value t and the number of data-link nDL on the values we considered best for the performance. For what concerns the non-monitoring mode we have noticed that the interarrival time k is the most influential factor and for higher value of k better performances are achieved while for the monitoring one it has to be found a trade-off between the monitoring time m and the malus X to achieve better performance. It's important to remember that the malus value X should always be lower than the monitoring time m to assure the correct behaviour of the system.