UNIVERSITÀ DI PISA

COMPUTER ENGINEERING

Performance Evaluation of Computer Systems and Networks

Cloud RAN-based cellular system

Group members:

Matteo Halilaga

Lorenzo Mancinelli

Rocco Giuseppe Pastore

ACADEMIC YEAR 2023/2024

Contents

1 Introduction

1.1 Description of the system

The aim of this project is to analyze a CRAN system by varying some configuration parameters. The system consists of an application that generates messages with a certain frequency and size according to some known distributions, sends them to a BBU which will forward them to a specific RRH following two distinct sending methods: one in which the packet is sent in full and one in which a packet compression is performed. In the first case the packet is directly forwarded to the relevant cell while in the second sending method the RRH will have to carry out a decompression process of the packet and send it to the relevant cell. Given different scenarios, we focus on the study of the system and in particular its end to end delay between the application and the cell with the aim of choosing the optimal compression percentage to use in each situation.

1.2 Assumptions

To study the behavior of the system we focused on a communication system based on a 4G network with different types of traffic. Furthermore, the following design hypotheses were made:

1. queues of infinite capacity;
2. instant compression time;
3. instant communication between AS and BBU;
4. within a simulation all parameters stays constant;
5. during communication the channel does not introduce errors, loss of packets or information within them.

1.3 Simulation parameters

The configuration variables that have been explored are the following:

1. packet generation rate: $\exp(\lambda_T)$;
2. distribution and package size: $\exp(\lambda_S)$, $lognormal(\mu_S, \sigma_S)$;
3. link transmission speed M;
4. packet compression percentages X;
5. number of cells N;

2 OMNeT++ implementation

2.1 Network

The system analized for this project is implemented on OMNeT++, an open-source modular framework designed for discrete-event system modeling and simulation. This choice allows to run the simulation and retrieve all the data in a lightweight and easy way. Each entities described in this chapter is defined by a `.ned` file. The `CloudRANCellularSystem.ned` file shows us how the modules are connected to obtain the network.

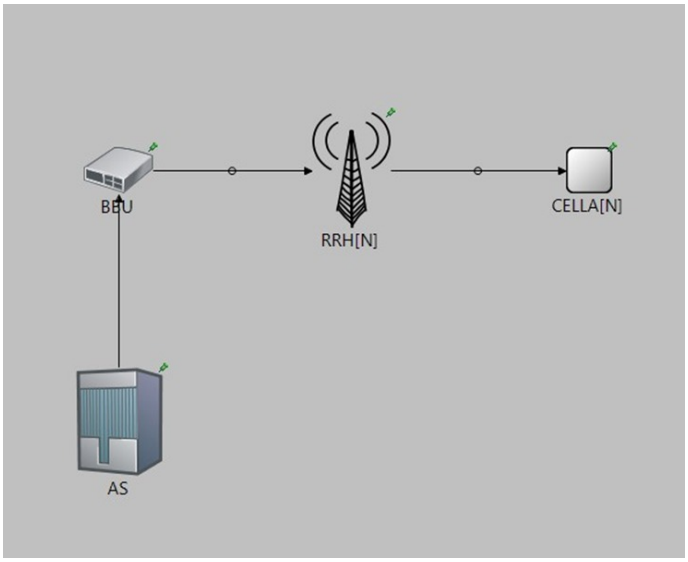

Figure 1: CloudRANCellularSystem network.

2.2 Description of the modules

2.2.1 AS

This module is responsible for generating packets and sending them to the BBU. Each packet is generate at intervals of T seconds, where T is a random variable following an exponential distribution. The size of these packets, S, was examined under two different distributions: the exponential distribution and the lognormal distribution. The format of the generated packet is as follows:

```
message Packet {
    double size;
    int cell;
    simtime_t timestamp;
}
```

2.2.2 BBU

It is the part of the system that is responsible for routing packets to the various RRHs. Since we have the constraint that the BBU can communicate with only one RRH at a time, we see it necessary to create a queue (managed via FIFO policy) in which the packets generated by the AS will queue up waiting to be served. Depending on the sending method, packet compression is performed or not, which is instantaneous and modifies the size of the packet. The BBU is connected to the RRH via a link with a bandwidth of M. Given this limit, the communication between the two modules occurs with a certain delay.

2.2.3 RRH

In the first sending mode (without compression) this module does not perform any function and forwards the packet to the destination cell. In the second it performs a packet decompression phase which takes $50ms \times X$, where X is the percentage of compression used and finally forwards the packet to the cell. Given that the decompression phase requires a non-zero amount of time, it was decided to also equip each RRH with a queue also managed via the FIFO policy.

2.2.4 CELL

It is the last element of the network. Once the packet has been received, it is responsible for measuring the end-to-end delay and storing it in a statistic created within the OMNET++ environment.

3 Code verification

Before proceeding with the simulation of the main scenarios, we carried out some tests to verify the correctness of our code. All system verification tests were carried out considering a single cell N=1 and deterministic values of S and T were used for all degeneracy tests and consistency tests.

3.1 Degeneracy test

These tests have the task of verifying how the system behaves in extreme cases.

3.1.1 Non-compression mode

The conditions under which we considered the system are described in the tables below (Fig. 2). As expected, with a very high speed link compared to the size of the packets, end-to-end delay to the cell is almost zero. Furthermore, by using a very large packet size compared to the capacity of the system link, we notice how the end-to-end delay grows infinitely, recording very high delay values.

LINK M [bytes/s]	PKT SIZE [bytes]	lamdaT [pkts/s]	E[T] (1/lamdaT) [s]	END-TO-END DELAY [s]
10000000	200	10	0.1	0.000020
200	100000	10	0.1	∞
200	200	1000	0.001	∞

Figure 2: Degeneracy test with zero compression.

Finally, by generating packets at a very high rate, an almost infinite end-to-end delay value is obtained.

3.1.2 Compression mode

In the table below (Fig. 3) we show the results of the degeneracy tests referring to the system in the case in which the packets are compressed by the BBU (limit values such as 0% and 99.9% compression were used).

%X	LINK M [bytes/s]	PKT SIZE [bytes]	E[T] (1/lamdaT) [s]	END-TO-END DELAY [s]	QUEUEING TIME [s]
0%	10000000	200	0.1	0.000020	0
99.9%	100000	200	0.1	0.050000	0

Figure 3: Degeneracy test with limit compression percentages.

Note how with a zero compression percentage of the packet size we return to the previous scenario (Fig. 2). For compression levels of the packet size approaching 100%, as expected, the end-to-end delay is solely determined by the decompression process, which amounts to 50ms.

3.2 Consistency test

This test aim to establish whether the system reacts proportionately to input variations. In the case represented by the following table, the size of the incoming packets has been doubled with the same rate and link (valid if there is no queue)

LINK M [bytes/s]	PKT SIZE [bytes]	E[T] (1/lamdaT) [s]	END -TO- END DELAY [s]
10000	500	0.1	0.050
10000	1000	0.1	0.100

Figure 4: Consistency test.

As expected, the end-to-end output delay doubles as the packet size doubles.

3.3 Continuity test

This test aims to verify whether a slight variation in the input data does not cause an abrupt change in the output data from the system. For our system, the generation rate (λ_T), (λ_S) and the compression percentage (X) were varied slightly. Note that these slight variations do not cause an unexpected and important change in the end-to-end delay to the cell but that, on the contrary, the system responds with a slight and sensible variation in the end-to-end delay.

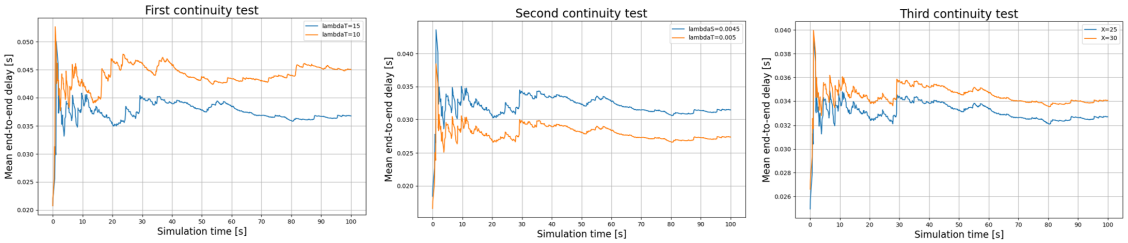

Figure 5: Continuity tests.

4 Stability estimation

4.1 Exponential distribution

The stability estimation model varies depending on the sending mode used, this is due to the fact that the network topology also varies depending on the sending mode, in fact in the case of compression there is also a queue on the RRHs. For this reason we will analyze them separately. In the first case, the one without compression and in which S and T are RV exponentials, the system can be traced back to an M/M/1, in fact in this case the parameter N is irrelevant given that once the BBU is exceeded the packet arrives at the cell without delays. The model is therefore as follows:

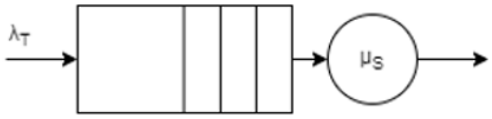

Figure 6: Theoretical model case without compression.

For this model, we know all the mean performance indexes

$$E[N] = \frac{\rho}{1-\rho}, \; E[R] = \frac{E[N]}{\overline{\lambda}} \; (Little's \; Law) \tag{1}$$

and it's stability condition:

$$\rho = \frac{\lambda_T}{\mu_S} < 1, \; \mu_S = \lambda_S \cdot M \tag{2}$$

Since the system is PASTA (Poisson Arrivals See Time Average) we can say that $\overline{\lambda} = \lambda_T$.

4.1.1 Case with compression

In the case of the system that uses the compression mode we are dealing with a queueing network as can be seen in the image. Stability in this case is achieved if both the BBU and RRH systems are stable. So, we have the following conditions:

$$\begin{cases} \frac{\lambda_T}{\mu_{S'}} < 1 \\ \frac{(\lambda_T/N)}{\mu_C} < 1 \end{cases} \tag{3}$$

where $\mu_{S'} = \lambda_{S'} \cdot M = \frac{1}{S'} \cdot M$, $S' = S - S \cdot X$, $\mu_C = 50ms \times X$. In this case the model used to describe the RRH system is an M/D/1 in which the input rate is exponential and equal to $\frac{\lambda_T}{N}$ since each cell have equal probability to be chosen ($\pi_1 = \pi_2 = ... = \pi_N$), while the service time is constant and given by $50ms \times X$.

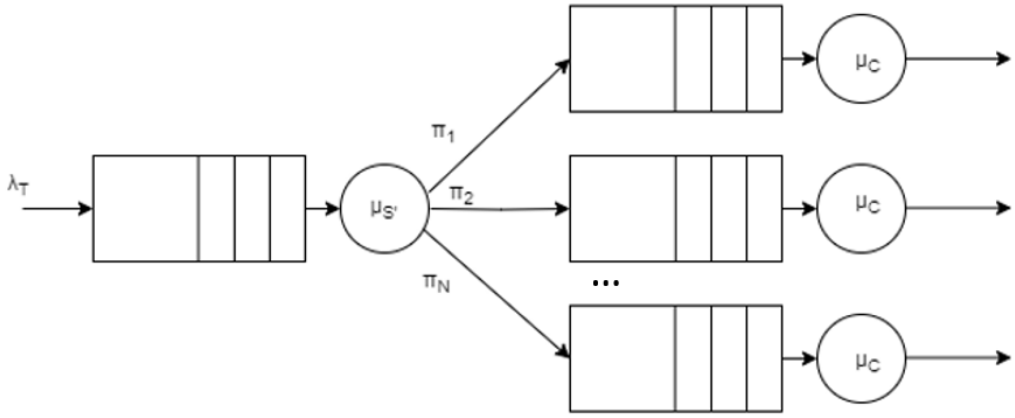

Figure 7: Theoretical model case with compression.

In this case the total end-to-end delay is $E[R] = E[R1] + E[R2]$ where $E[R1]$ is equal to the first case considering S' instead of S and $E[R2]$ is

$$E[R2] = \frac{1}{2\mu_2} \cdot \frac{2 - \rho_2}{1 - \rho_2} \tag{4}$$

Thanks to a correct implementation of the system, the theoretical end-to-end delay are consistent with those obtained from the simulator for both methods.

4.2 Lognormal distribution

In the case in which the S is lognormal we have that the system becomes a M/G/1 in which the stability is still $\rho < 1$, while the average number of jobs in the system

is:

$$E[N] = \rho + \frac{\rho^2 + \lambda^2 \cdot Var(t_S)}{2 \cdot (1 - \rho)} = \rho + \frac{\rho^2 \cdot [1 + CoV(t_S)^2]}{2 \cdot (1 - \rho)} \tag{5}$$

For the case involving compression, the system becomes too complex to be analyzed using standard theoretical models.

5 Warm-up time and simulation time

Before analyzing system performance, we determined the warm-up time needed for system stabilization and set a maximum simulation time for data collection. We focused on lognormal packet size distribution, with exponential packet arrival times, as the worst-case scenario due to its higher probability of generating large packets.

5.1 Warm-up time

As regards the warm-up period, the reference configuration was set with

$$N = 1,\ M = 12500,\ \lambda_T = 50,\ \mu_S = 4.9517437762680645,$$

$$\sigma_S^2 = 0.6931471805599453,\ X = 20$$

This configuration allows us to obtain a compromise between high packet size, since the average size is 200 bytes, and a high generation rate of these, i.e. 50 packets per second, maintaining the stability of the system even in the presence of compression, in this case equal to $X = 20\%$.

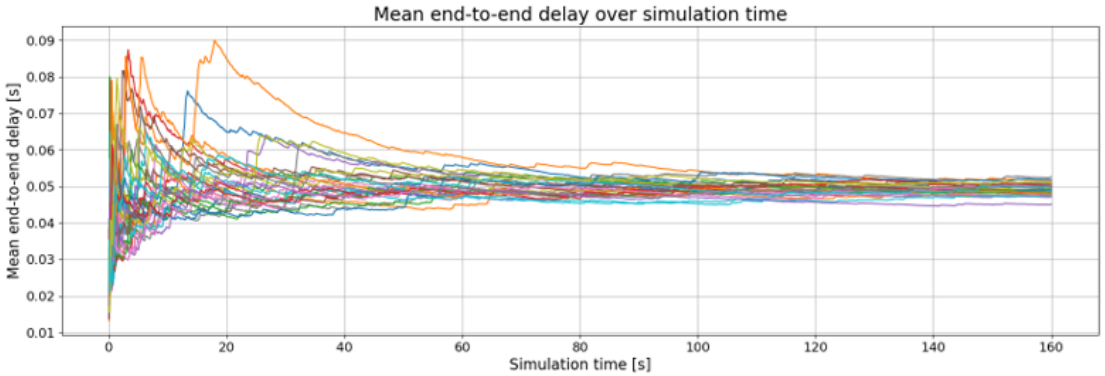

Figure 8: Estimation of the warm-up time with exponential distribution of T and lognormal distribution of S, for compression ratio equal to 20.

As can be observed from the graph of Figure 8, the end-to-end delay surpasses the initial transient phase and begins to stabilize in the time interval $[35s, 40s]$ of simulation time. So, to be completely sure, we can set the warm-up period to $40s$.

5.2 Simulation time

Regarding the simulation time, we analyzed the standard deviation of the average end-to-end delay for each repetition. The Table 1 presents the average end-to-end delay at different time intervals and the corresponding standard deviation, excluding the warm-up period identified in the previous section. As observed, the average end-to-end delay does not significantly vary, because we surpassed the initial transient phase, while the standard deviation decreases. However, at a certain point this decrease becomes less relevant. This suggests that, for longer simulation times, the standard deviation is unlikely to have important changes. Consequently, we have chosen a maximum observation period of 160 seconds for our simulations to avoid excessively prolonged durations.

Simulation time [s]	**Mean end-to-end delay [ms]**	**stdev [ms]**
60	49.007	6.736
80	48.241	4.181
100	48.789	3.274
120	48.674	2.580
140	48.728	2.259
160	48.907	2.162

Table 1: Mean end-to-end delay for different simulation times and their standard deviations.

6 Simulations

Before starting our simulations, it is essential to note that the samples within a single repetition are not independent of each other. Therefore, for each simulation, we conducted 30 independent repetitions to ensure that the means of the end-to-end delays at the cell, which themselves are random variables, could follow a Normal distribution. A comprehensive analysis is presented in Appendix A.1.

6.1 Exponential distribution of S

In this section we analyze the trend of the average end-to-end delay as the number of cells varies and for different compression values, setting an exponential distribution for both S and T.

6.1.1 Selected configurations

Our simulations were based on the following configurations:

- VoIP traffic (100, 200 bytes), generic traffic (500 bytes), and videostream traffic (1000 bytes);

- 4G network with transmission speeds between BBU and RRH of 100kbps, 1Mbps, and 10Mbps;
- number of cells equal to 5, 10, 15, and 20;
- compression rates of 0%, 10%, 20%, and 30%, as these are the most common in such applications;
- λ_T equal to 10, 50, 100, and 150 to analyze various traffic intensities.

6.1.2 Scenarios

To optimally analyze our system, we decided to consider three different scenarios:

- general load - fast link (any traffic, M = 10Mbps);
- high load - medium link (generic and videostream traffic, M = 1Mbps);
- low load - slow link (VoIP traffic, M = 100kbps).

In the tables shown below, each cell indicates the system usage, while the cell color represents the stability condition of the system (green = stable, red = unstable). It's important to take a look to this parameters because it's the main indicator that suggest if it's convenient to make the compression.

High load - fast link Under these conditions, the system becomes less interesting to study due to low system utilization resulting from the high transmission link speed. This leads to performance degradation as the packet compression rate increases. Therefore, in a system with these characteristics, the best choice is to never compress the packets sent to the destination cells.

S [byte] \ lambdaT	10	50	100	150
100	0.0008	0.004	0.008	0.012
200	0.0016	0.008	0.016	0.024
500	0.004	0.02	0.04	0.06
1000	0.008	0.04	0.08	0.12

Figure 9: High load - fast link (10Mbps) system usage.

For example, with a configuration of $\lambda_T = 150$ and a packet size of 1000 bytes, the average delay increases from 0.00091s without compression to 0.01285s with 20% compression (considering N = 5).

High load - medium link In the table in Figure 10, we have highlighted in bold the most interesting configurations to analyze, and their graphs are shown in Figure 11. As we can see, in only one case is it better to compress the packets before sending them to the RRH, that is the case with S=1000 bytes and $\lambda_T = 100$. Unfortunately, due to the partial overlap of the confidence intervals, we cannot say with certainty that the 20% of compression is the best for all the different number of cells. The convenience in compressing is given first and foremost by the high percentage usage of the system in the BBU. Compression generates a double effect:

- In the BBU, we have the advantage of decreasing the transmission time towards the RRHs, as the packet size decreases, going from S to S'.
- In RRH, as the compression percentage increases we have an increase in decompression time which can cause potential delays on the RRH queue.

S [byte] \ lambdaT	10	50	100	150
100	0,008	0,04	0,08	0,12
200	0,016	0,08	0,16	0,24
500	0,04	**0,2**	**0,4**	0,6
1000	0,08	**0,4**	**0,8**	1,2

Figure 10: High load - medium link (1Mbps) system usage.

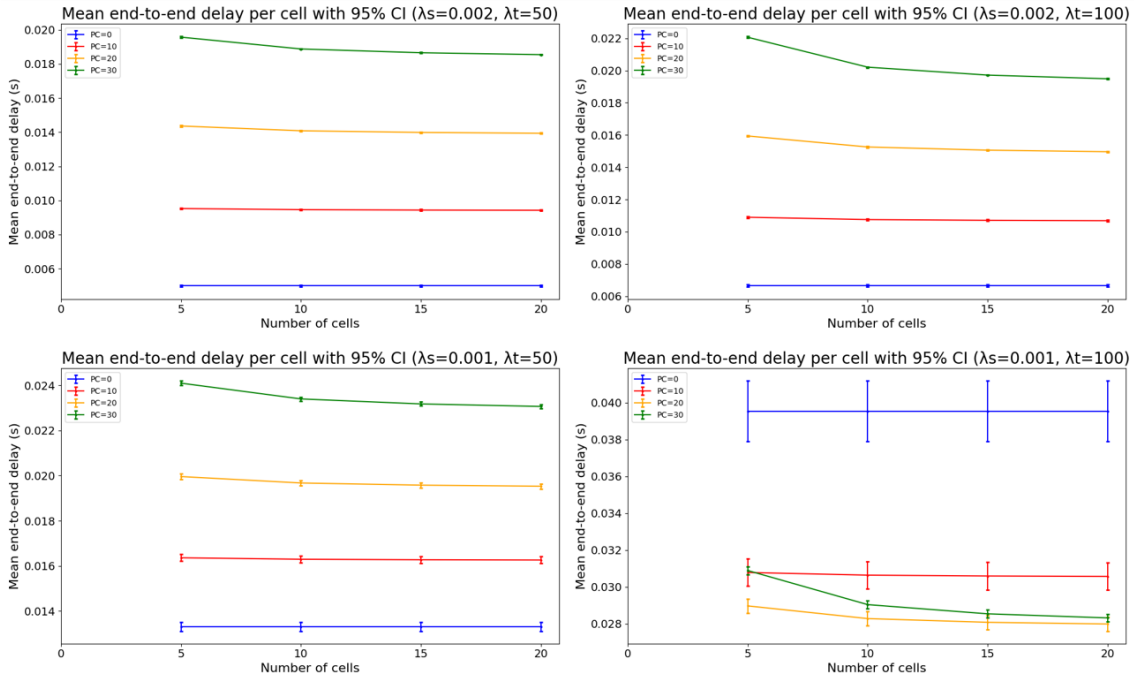

Figure 11: Mean end-to-end delay medium link (1Mbps).

Furthermore, we see how as the number of cells increases the average end-to-end delay decreases slightly, as the output rate from the BBU is spread over a greater number of RRHs.

Low load - slow link As before, in the table we have highlighted in bold the most interesting configurations to analyze, the graphs of which are shown in Figure 13 and 14.

S [byte] \ lambdaT	10	50	100	150
100	0,08	0,4	0,8	1,2
200	0,16	0,8	1,6	2,4
500	0,4	2	4	6
1000	0,8	4	8	12

Figure 12: Low load - slow link (100Kbps) system usage.

The graphs confirm the observations made in the previous paragraph.

- For systems with low utilization, compression offers no benefit. This is because small packet sizes, relative to the high BBU-to-RRH communication speed, result in transmission times shorter than decompression times. Furthermore, low packet generation rates lead to minimal queuing at the BBU.
- High system utilization benefits from compression. For example, with packet generation rate $\lambda_S = 0.01$ and arrival rate $\lambda_T = 100$, compressing up to 20% improves performance. Beyond this threshold, average packet delay rises, but remains significantly lower than without compression, even considering overlapping confidence intervals for five cells. While it's inconclusive whether 30% compression surpasses 20%, both outperform 10% compression in high-cell scenarios.

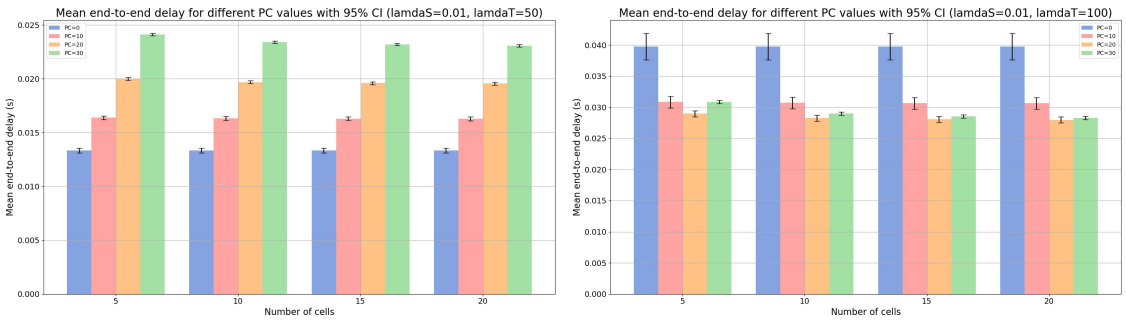

Figure 13: Mean end-to-end delays for a slow link (S = 100 bytes).

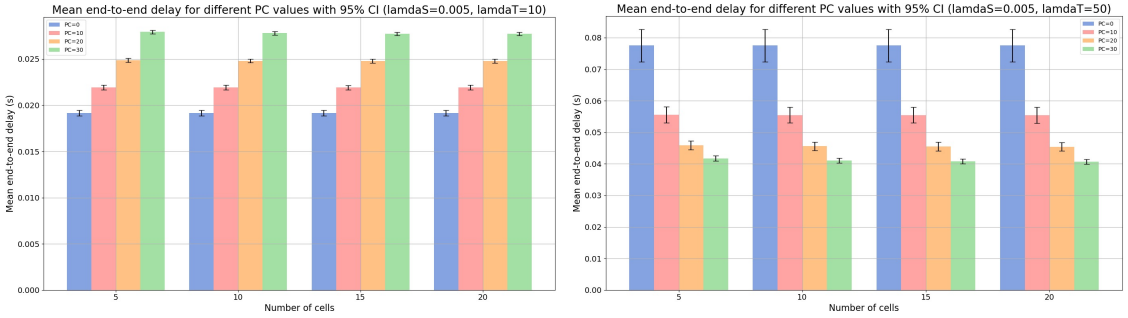

Figure 14: Mean end-to-end delays for a slow link (S = 200 bytes).

6.1.3 Impact of BBU queueing time on end-to-end delay

Referring to the last scenario (low load - slow link), we have added a statistic within our code to evaluate the impact of BBU queuing time on the end-to-end delay. This statistic allowed us to confirm our results. As shown in the graphs below, we can observe that the queuing time decreases as the compression percentage increases, independent of the number of cells.

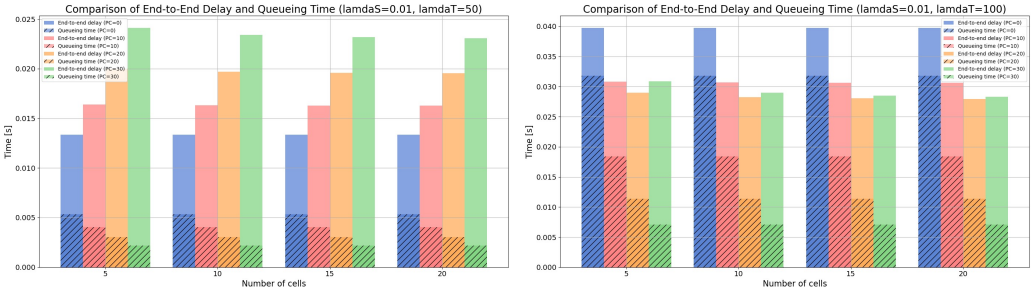

Figure 15: Comparison between queueing time and end-to-end delay in slow link (S = 100 bytes).

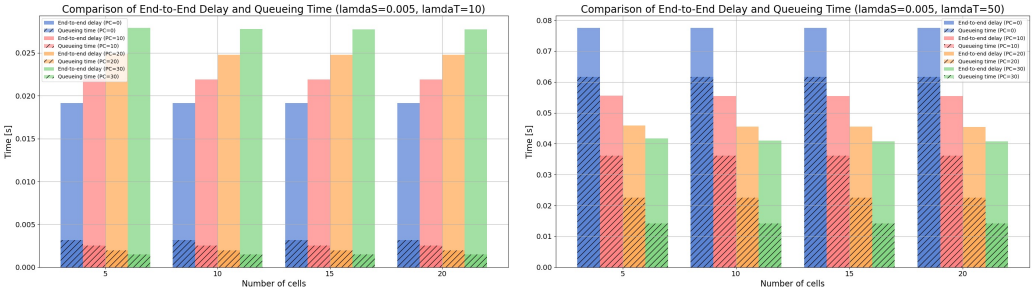

Figure 16: Comparison between queueing time and end-to-end delay in slow link (S = 200 bytes).

6.2 Lognormal distribution of S

By analyzing the last 4 configurations with a lognormal distribution of S (see Appendix A.2 for more details), we can observe that the results obtained are very similar to the previous ones.

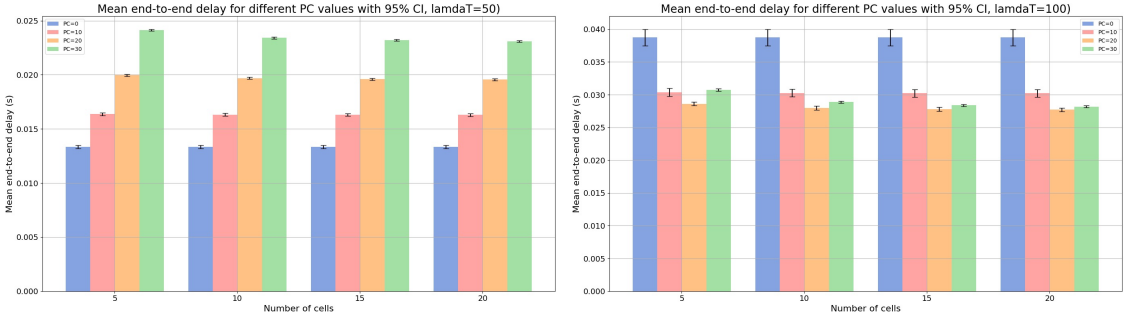

Figure 17: Mean end-to-end delays for a slow link (S = 100 bytes).

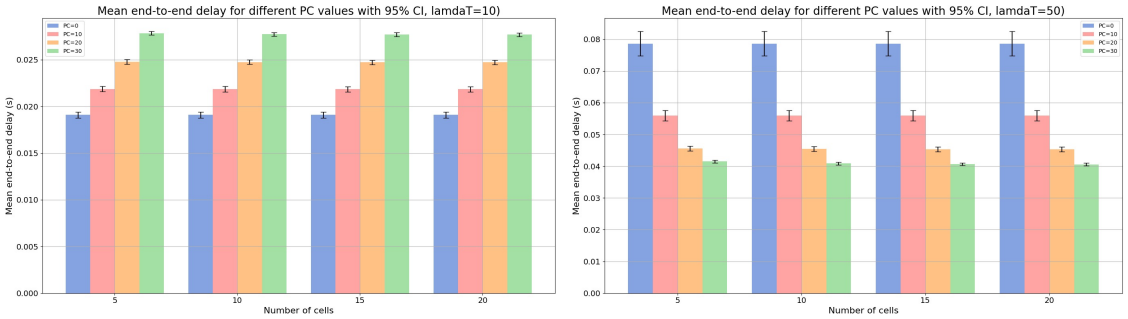

Figure 18: Mean end-to-end delays for a slow link (S = 200 bytes).

This is due to the fact that although extreme values for the lognormal are more likely, this difference is not that significant and since we also use low rates it is even less visible. The main difference between the two distributions is for small values far from the average, in fact we can see from the table that values between 0 and 100 are 10 times more likely with an exponential distribution.

Distribution	Parameters		P(0<X<20)	P(200<X<1000)	P(1000<X<2000)
Exponential	λ = 0.005		0.0952	0.3611	0.0067
Lognormal	μ=4.9517437762680645	σ = 0.832554611157697	0.0094	0.3292	0.0087

Figure 19: Comparison packet probability.

Despite this difference, our system is not very sensitive to small values of packet size, a small packet will simply be disposed of first and will have no impact on subsequent ones. For these reasons we obtain very similar results. For completeness we also show the graph with the queuing time but it remains almost identical for the same reasons.

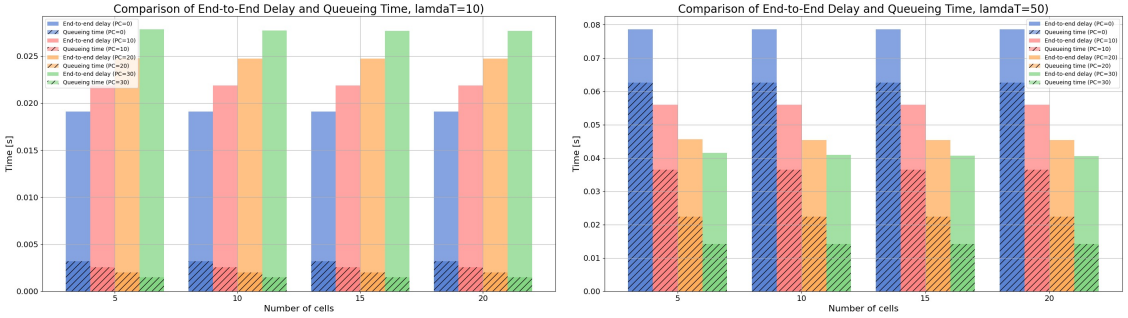

Figure 20: Comparison between queueing time and end-to-end delay in slow link with lognormal distribution (S = 200 bytes).

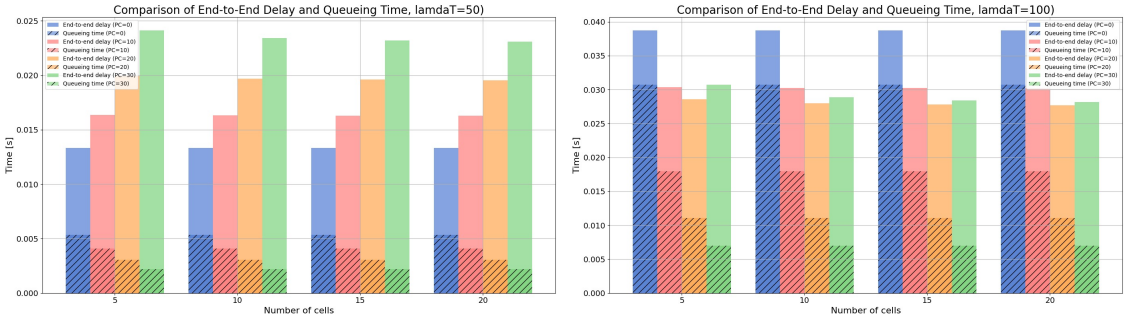

Figure 21: Comparison between queueing time and end-to-end delay in slow link with lognormal distribution (S = 100 bytes).

7 Conclusions

We observed that the system operates similarly whether the packet size S follows an exponential or lognormal distribution. Therefore, the system's performance heavily relies on appropriately sizing the link between the BBU and the RRH. Moreover, if the transmission link M is too fast, the packet transmission via compression becomes unnecessary. Without queues forming at the BBU, compression would only introduce additional delays at the RRHs due to the decompression process. However, compression becomes beneficial when system usage is high and queues form at the BBU. This is evident in specific scenarios such as [M=12500, $\lambda_S = 0.01$, $\lambda_T = 100$ (Fig. 13)], where increasing the compression percentage results in a greater reduction in packet waiting time than the increase in decompression time. Finally, the number of cells becomes a bit more significant when the system employs packet compression, as the uniformly chosen target cells experience slightly reduced waiting time at the RRHs. However, this differences are little because of the low utilization of the RRH.

A Appendix

A.1 Independence assumption of mean end-to-end delay.

Chapter 6 focuses on the investigation of the average end-to-end delay observed across various cells, calculated for each repetition of the experiment under a specific configuration and for different values of the compression ratio. Since the end-to-end delay is dependent on the number of packets queued at either the BBU or the RRH, the samples observed in a single repetition cannot be entirely independent of each other. In fact, if an incoming packet encounters a very long queue at the BBU, for example, its end-to-end delay will be observed to increase, and the same principle applies to the subsequent packet. This behavior can be easily seen in Figure 22. Therefore, to enable the application of the central limit theorem and to establish that the average end-to-end delay is a Normal RV, simulations with 30 repetitions each were conducted. For each repetition, we calculated the average end-to-end delay recorded for each cell. Since each average delay is also a (RV), and since the repetitions were set to be independent of each other (Figure 22), we can assert that the average delay experienced by each cell has a normal distribution, that is

$$\overline{R}(i,x) = \frac{1}{30} \cdot \sum_{j=0}^{29} E[R](i,j,x) \sim Normal \tag{6}$$

where $E[R](i,j,x)$ is the average end-to-end delay recorded by cell i in the $j-th$ repetition of the experiment with a compression percentage of x. This statement is proven also by the experiments, as shown in Figure 23. Under these conditions, we can compute the confidence intervals for each $E[R](i,j,x)$ as follows:

$$CI_{95\%} = \left[\overline{R}(i,x) - z_{0.025} \cdot \frac{S}{\sqrt{30}}, \overline{R}(i,x) + z_{0.025} \cdot \frac{S}{\sqrt{30}}\right] \tag{7}$$

where S is the standard deviation computed from $E[R](i,j,x)$.

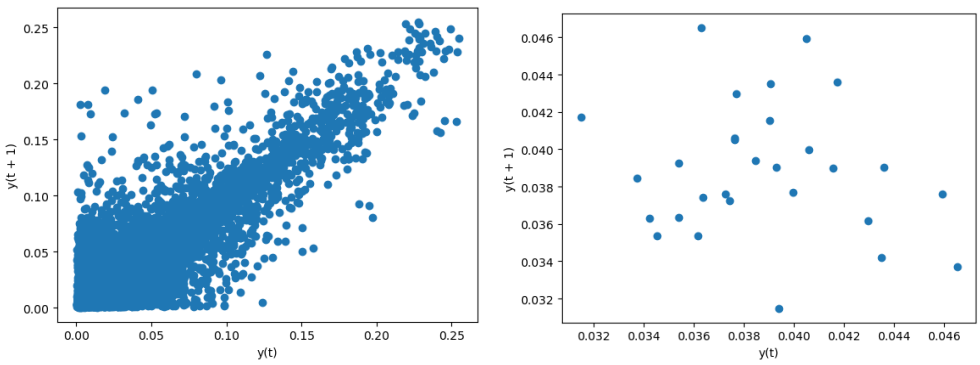

Figure 22: Lag plots of the samples within a single repetition (left) and of the mean values of the 30 repetitions of a single simulation (right).

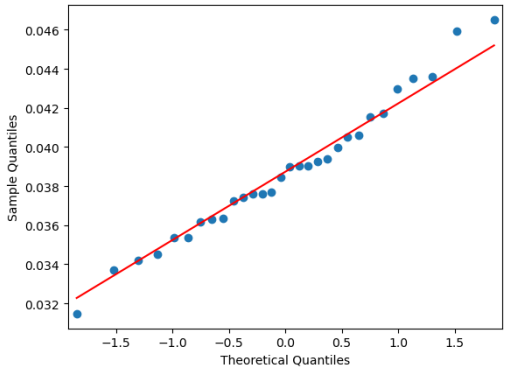

Figure 23: QQ plot of the mean values of the 30 repetitions of a single simulation.

A.2 Conversion parameter from exponential to lognormal

In order to make fair experiment keeping the same value of mean and variance for the two distribution we have used the following formula to calculate the parameters of a lognormal distribution

$$\mu = \ln\left(\frac{E[S]^2}{\sqrt{Var(S) + E[S]^2}}\right) \qquad \sigma = \sqrt{\ln\left(1 + \frac{Var(S)}{E[S]^2}\right)} \tag{8}$$