Cloud RAN-based cellular system

Matteo Halilaga
Lorenzo Mancinelli
Rocco Giuseppe Pastore

Introduction

Objectives

Analyze the performances of a Cloud RAN-based cellular system using the following KPIs:

- **Mean end-to-end delay**
- **Queueing time** on the BBU

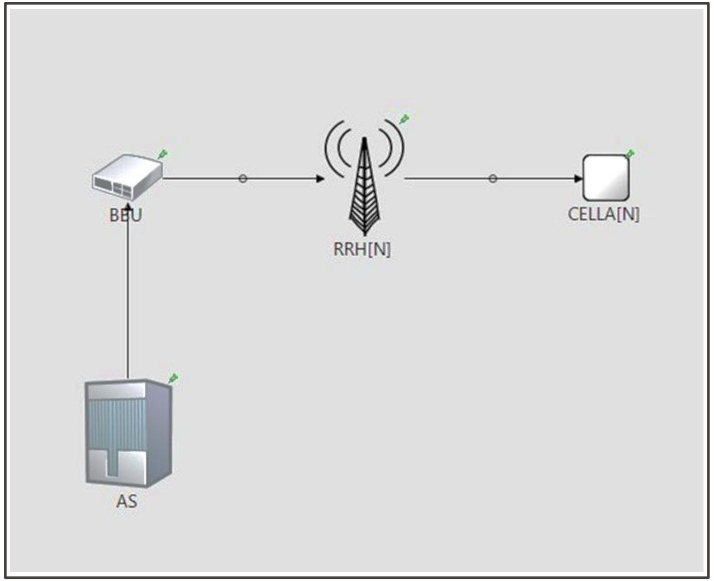
The operating modes of the system are:

1. **No compression** before sending a packet
2. **Compression of X%** before sending a packet

Configuration variables

- Distribution of packet generation rate - $\mathbf{exp}(\lambda_T)$
- Distribution of packet size - $\mathbf{exp}(\lambda_S)$, $\boldsymbol{lognormal}(\mu_S, \sigma_S)$
- Link transmission speed - $\boldsymbol{M}$
- Packet compression - $\boldsymbol{X}$
- Number of cells - $\boldsymbol{N}$

Implementation



- **Application Server**: generates packets of size S every T seconds.
- **BBU**: routes packets to the appropriate RRH, with or without compression.
- **RRH**: forward the packet to the specified cell, maybe after a decompression stage.
- **Cell**: destination of the packet. Here the end-to-end delay is measured.

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

Code verification

Degeneracy tests

Without compression: very high values of link speed, packet size or generation rate lead to **almost zero or infinite end-to-end delay**.

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	∞

With compression

%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

→ Equal to the previous case.

→ Almost zero queueing time.

Code verification

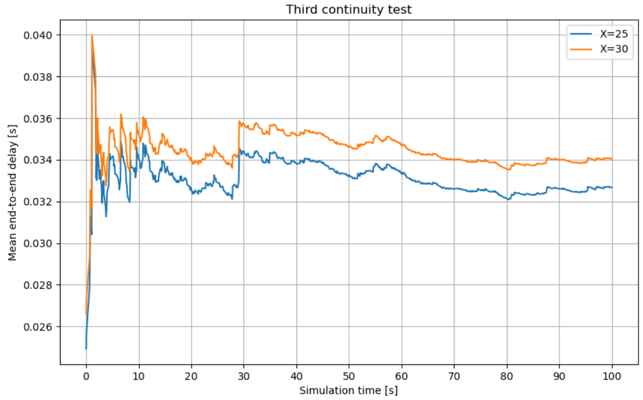
Consistency

Doubling the packet size leads to a **doubling of the end-to-end delay.**

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

Continuity

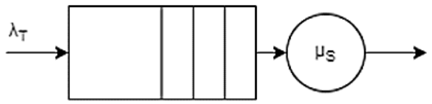
Slight variations in simulation parameters lead to **slight changes** in end-to-end delay.



Theoretical model – exponential case

Without compression

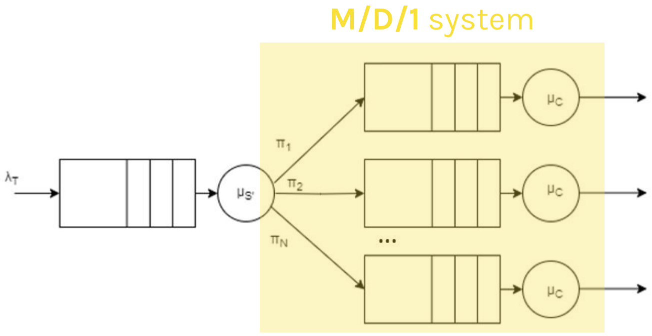
M/M/1 system with $E[N] = \frac{\rho}{1-\rho}, E[R] = \frac{E[N]}{\lambda_T}$



The **stability condition** of the system is

$$\boldsymbol{\rho = \frac{\lambda_T}{\mu_S} < 1}, \qquad \mu_S = \lambda_S \cdot M = \frac{1}{S} \cdot M$$

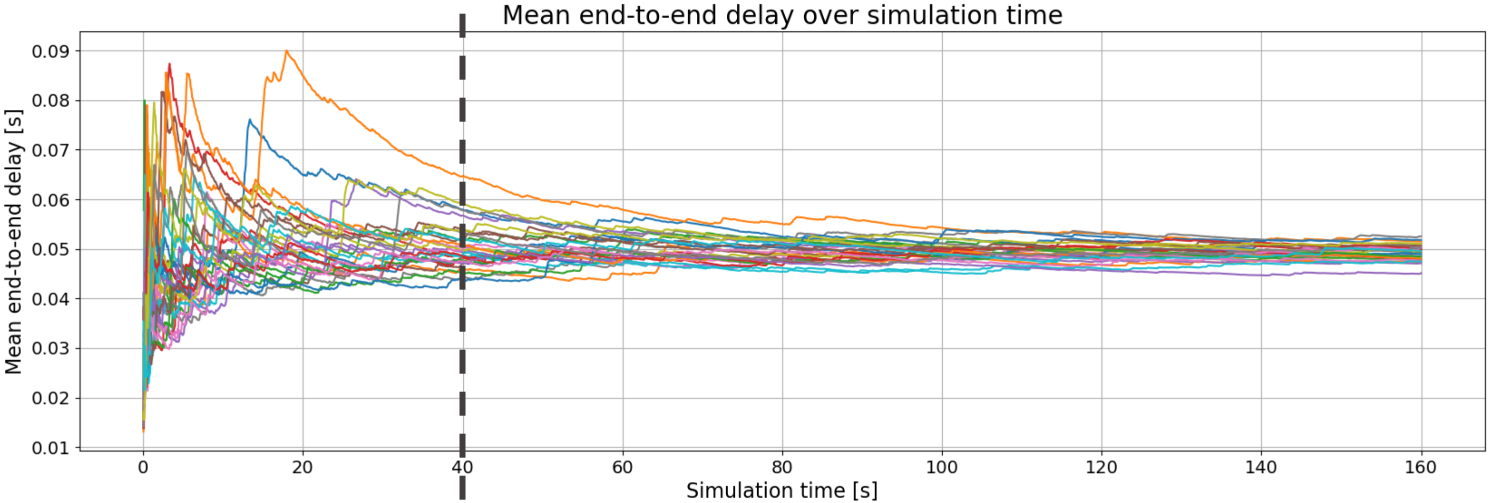
With compression



We have two stability conditions:

$$\begin{cases} \frac{\lambda_T}{\mu_{S'}} < 1, & \mu_{S'} = \lambda_{S'} \cdot M = \frac{1}{S'} \cdot M \\ \frac{\left(\frac{\lambda_T}{N}\right)}{\mu_C} < 1, & \mu_C = 50ms \times X \end{cases}$$

Warmup and simulation time



Warmup period = **40s**

Simulation time = **160s**

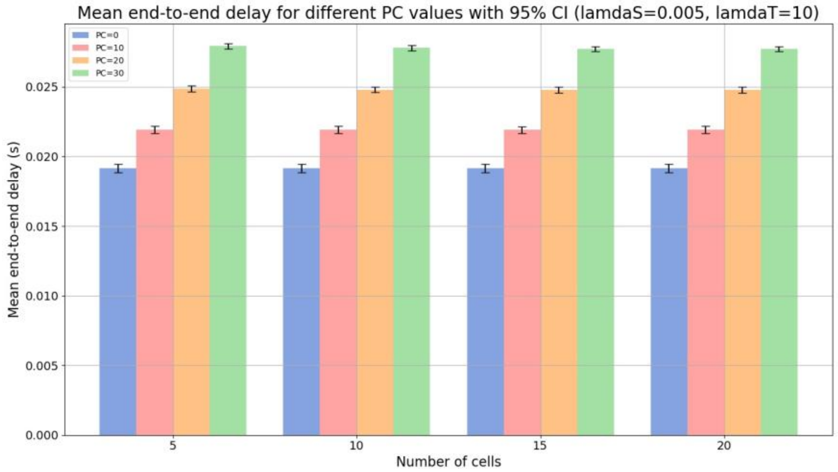
Scenarios

	Load	Link speed
General load – fast link	• Any type of traffic	M = 10Mbps
High load – medium link	• Generic – 500 bytes • Videostream – 1000 bytes	M = 1Mbps
Low load – slow link	• VoIP (100, 200 bytes)	M = 100kbps

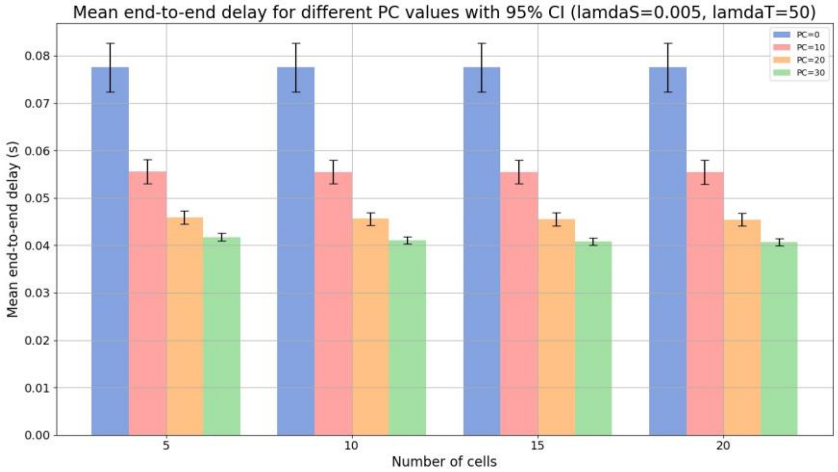
The higher the speed of the M link, the lower the system usage and the less convenient the compression will be.

Exponential distribution – **end-to-end delay**

Low usage ($\rho = 0,16$)



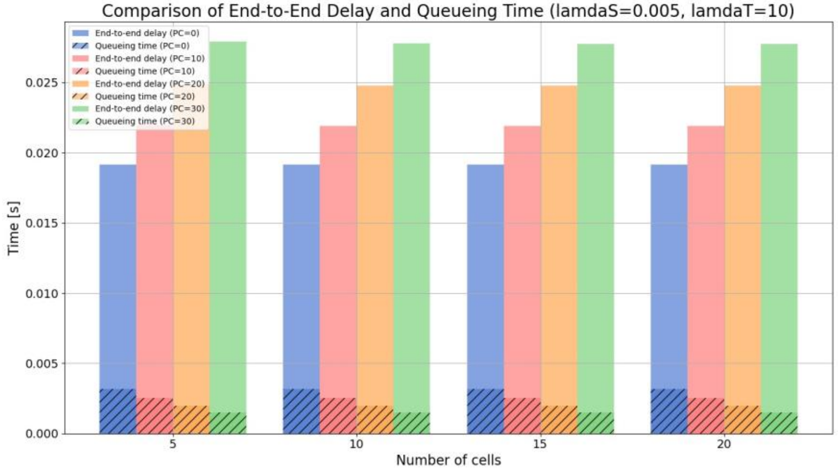
High usage ($\rho = 0,8$)



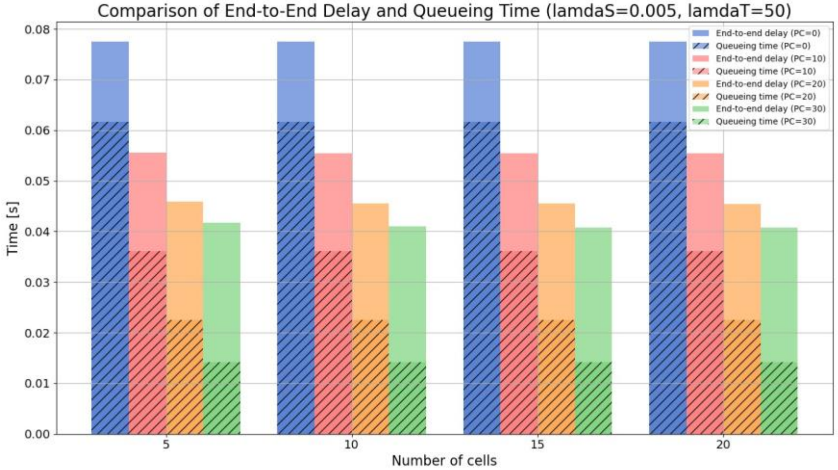
Slow link → $M = 100kbps$

Exponential distribution – **queueing time**

Low usage



High usage

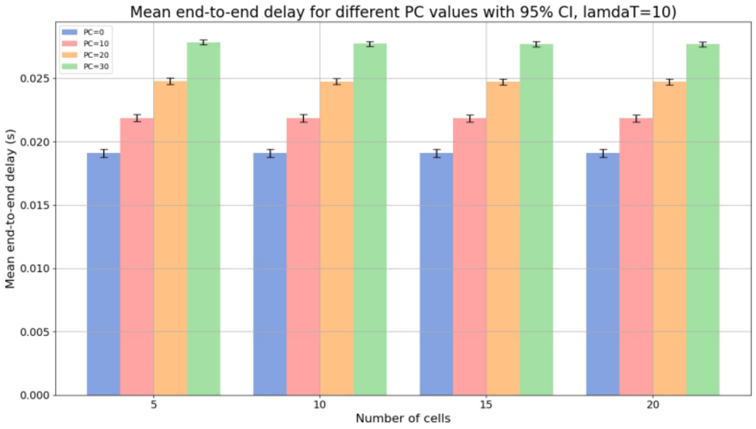


Slow link → $M = 100kbps$

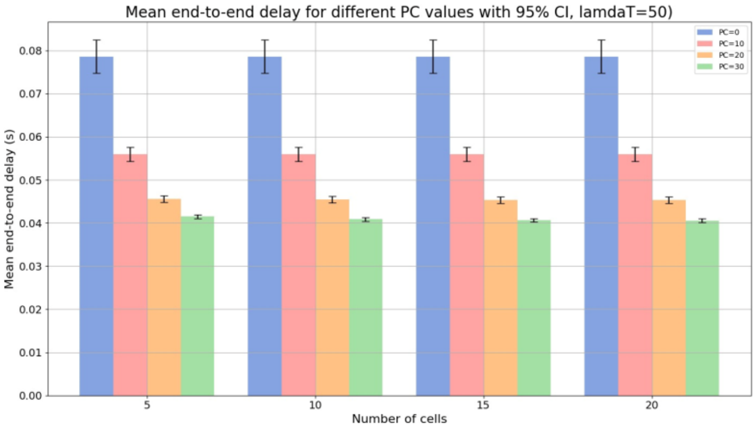
Lognormal distribution

~ same results as the exponential case!

Low usage



High usage



Low load - slow link → $S = 200\ bytes,\ M = 100kbps$

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

Conclusion

- The correct functioning of the system strongly depends on the **sizing of the link** between BBU and RRH.
- With the same packet size and generation rate, the **higher the link**, the **lower the system usage**, given that not much queue will form on the BBU.
- When there is **high utilization**, it’s **convenient performing compression**.
- Increasing the number of cells improves system performance slightly when packet compression is performed.