Università di Pisa

Computer Engineering

Performance Evaluation of Computer Systems and Networks

Opportunistic Cellular Network

Group Project Report

TEAM MEMBERS:
Federico Casu
Daniel Deiana
Erica Raffa

Academic Year: 2022/2023

Contents

Chapter 1

Introduction

1.1 Opportunistic cellular network

A cellular network transmits its traffic to n users. Each user has its own FIFO queue on the transmitting antenna. On each timeslot, users report to the antenna a **C**hannel **Q**uality **I**ndicator (**CQI**), i.e. a number from 1 to 15, which determines the number of bytes that the antenna can pack into a **R**esource **B**lock (**RB**) according to the table below. Then the antenna composes a frame of 25 RBs by scheduling traffic from the users, and sends the frame to the users. A packet that cannot be transmitted entirely will not be scheduled. An RB can only carry traffic for one user. However, two or more packets for the same user can share the same RB. (e.g., packet 1 is 1.5 RBs and packet 2 is 1.3 RBs, hence ceiling(1.5 + 1.3) = 3 RBs are required to transmit them).

CQI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bytes	3	3	6	11	15	20	25	36	39	50	63	72	80	93	93

The antenna serves its users using an opportunistic policy: backlogged users are served by decreasing CQI. When a user is considered for service, its queue is emptied, if the number of unallocated RBs is large enough.

1.2 Project Objectives

Consider the following workload for all users:

a) Packet interarrival times are IID RVs (to be described later).

b) Service demands (in bytes) are IID RV (to be described later).

Study the ***throughput*** and ***response time*** of the above system with a varying workload at least in the following scenarios:

1) Exponential interarrivals, uniform service demands (the largest packet dimension is such that it fits a frame at the minimum CQI), uniform CQIs.

2) Same as above, with binomial CQIs, chosen so that the mean CQI of different users are sensibly different.

1.3 Model description

Since the objective of this project was to study a simplified version of a cellular network protocol, we have introduced 3 entities to model the system:

- **Antenna**: an Antenna is an entity that models the transmitting antenna within a cellular network. The Antenna is in charge of transmitting the traffic to the users. User traffic is represented by packets. When a packet arrives to the Antenna, it queues the packet in the buffer[1] (FIFO queue) associated to the destination user. On each timeslot, the Antenna asks to its users a CQI and then transmits a Frame. A Frame is composed of 25 Resource Blocks (RB). User traffic is allocated in RBs according to a Opportunistic policy: backlogged users are served by decreasing CQI.

- **Source**: a Source is an entity that models an external or internal user that forwards packets to a specific queue in the antenna. We have a Source entity for every backlogged user. A single Source will generate packets to be forwarded **only** to its associated user. Packets interarrival times are ***exponential*** RVs and service demands are ***uniform*** RVs $\sim U(1, 75)$.

- **Cellular**: a Cellular is an entity that models an internal user of the network served by the Antenna. The Cellular sends its CQI to the Antenna, following the latter's request, on each timeslot. Afterward, the Cellular will receive the packets addressed to it from a Source through the Antenna. CQIs are integer RVs generated according the following scenarios:
  - ***Uniform*** - $CQI \sim U(1, 15)$.
  - ***Binomial*** - $CQI \sim Bin(n, p_i)$, where n is the number of user and p_i depends on the user i.

[1]Each user has its own buffer on the Antenna.

Chapter 2

Implementation

2.1 Network

To build our model and to run simulations we used the framework ***OMNeT++***. Each entities described in the previous chapter is defined by a `.ned` file.
The `OpportunisticCellularNetwork.ned` file shows us how the modules are connected to obtain the network.

`OpportunisticCellularNetwork.ned`

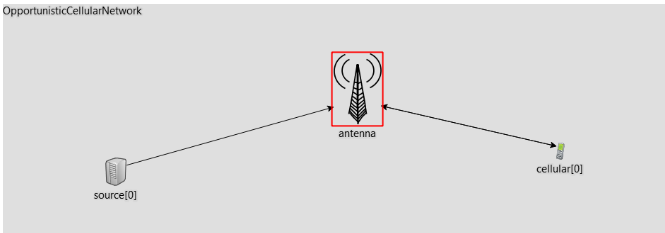


Figure 2.1: A simple instance of the network (1 user).

2.2 Antenna

The Antenna was modeled through a `SimpleModule` called '`Antenna`'. The behaviour of the simple module is the following:

- On each timeslot, the Antenna asks to its users a CQI. To handle this event the Antenna sends to itself a message every `timeslot` milliseconds and then sends to each user a CQI REQUEST.
- When a new CQI arrives, the Antenna analyzes the CQI RESPONSE to obtain the user who transmits the CQI and the current CQI. The Antenna stores the couple ¡user, CQI¿ into a `std::vector<>`. The latter will be used to implement the service policy.
- When a new packet arrives the Antenna analyzes the received packet to obtain the destination user and the size of the packet: these informations are stored in the message that triggered the event. Once these informations have been retrieved, the Antenna place the packet into the queue associated to destination user.
- To transmit the queued packets, the Antenna allocates a Frame at each time slot. A Frame will be packed only when each user has transmitted his current CQI. When the last user sends his CQI RESPONSE, the Antenna handles the event as described before and then it performs the following actions:

- It sorts the `CQIs` vector in decreasing order according to the user's current CQI. In this way the Antenna allocates RBs to the user whose current CQI is the highest (opportunistic policy).
- It serves the "first" backlogged user.
- If the user under service can be no longer served[1], the Antenna will serve the user with the second highest CQI and so on.

2.3 Cellular

The entity Cellular is defined by a `SimpleModule` of equal name. The Cellular's behaviour is the following:

- In order to simulate the quality of the network, the Cellular generates a new CQI every `timeslot`. According to the parameters of the simulation, the distribution will be uniform or binomial.
- Since the mean values of the CQIs has to be sensibly different for every Cellular, the p values are calculated so that they are equally parted from one another, based on the `population` number. The p value is calculated separately for every Cellular during the initialization.
- A Cellular can receive two types of messages: CQI REQUEST and RBs. In case of a CQI REQUEST, the Cellular sends the new generated CQI back to the Antenna. In the other case, the Cellular receives n RBs addressed to itself. In the last scenario, the Cellular compute the *response time* and the number of received bytes is tracked to analyze the throughput.

2.4 Source

The **Source** entity is defined as a module in "Source.ned" file. The Source models traffic incoming to antenna queues. Each source sends to the antenna a `Packet` message. The inter packet arrival is a RV taken from an exponential distribuition with a rate λ. Packet size is a RV taken from an integer uniform distribuition going from 1 to 75. 75 is chosen as maximum value because the Antenna that is using minimum RB size can fit the packet into the frame.

[1] Either the remaining RBs are not enough to pack the remaining queued packets or the queue has been emptied

2.5 Messages

We defined tree new format of messages to properly represent a packet, a CQI RESPONSE and a RB.

`Packet.msg`

```
message Packet {
    int size;
    int index;
    simtime_t timestamp;
}
```

`CQI.msg`

```
message CQI {
    int id;
    int CQI;
}
```

`RBsPacket.msg`

```
message RBsPacket {
    int destinationUser;
    int usedRBs;
    int usedBytes;
    simtime_t arrivalTimes[];
}
```

2.6 Validation Tests

Before going to simulate the main scenarios we performed tests to validate our model, more details can be found in A. We put here a brief description of the tests we made:

- **Degeneracy Test** - This test aims to verify the behaviour of the cellular network in degenerate conditions, i.e. when the system becomes unstable.
- **Consistency Test** - This test aims to verify the behaviour of the cellular network in two different workloads where we expect that the KPIs are the same.
- **Continuity Test** - This test aims to verify the behaviour of the cellular network in different workloads that are slightly different. We expect that the KPIs are different from each others but the differences are small.
- **Scheduling Policy Test** - This test aims to verify the behaviour of the scheduling policy adopted by the cellular network. In particular, our objective was to verify if the performance of cellulars with high CQIs are better than cellulars with low CQIs.

Chapter 3

Warm-up time and simulation period estimation

After initial consideration we proceeded to estimate values for warm-up time (and simulation period) using the Antenna slotted throughput as a reference and seeing its evolution over simulation time. This statistic was chosen because, unlike the response time, its value stabilizes in all scenarios[1]. Moreover, the Antenna slotted throughput summarize the state of the entire network since it is just the sum of the cellular throughputs.
For the **warm-up time** estimation we observed the trajectory of the mean throughput to see when the transitory has passed. The process was:

- Computing a vector whose element *i-th* is the mean value calculated from samples $\in [0, i]$ (for each repetition). The formula that has been used is the following:

$$v_n(i) = \frac{\sum_{k=1}^{i} THR_n(k)}{i}$$

  where $v_n(i)$ is the *i-th* element of the vector associated to *n-th* repetition.

- Computing the time $\bar{t}$ at which the mean throughput approached a good value. The **warm-up time** was chosen such that $\bar{t} = \frac{i}{1000}$, where $i = max(j) \ : \ M(j) - E[THR] \geq 10$, where $M(j)$ is the mean throughput at timeslot j calculated starting from the n repetitions

$$M(j) = \frac{\sum_{i=0}^{n-1} v_i(j)}{n}$$

  and $E[THR]$ is the mean throughput calculated by aggregating the mean throughput of the n repetitions:

$$E[THR] = \frac{\sum_{i=0}^{n-1} E[THR]_i}{n}$$

For the **simulation period** estimation we observed not only the trajectory of the mean value, to see when the transitory has passed, but we observed also the sample standard deviation among different repetitions[2]. The process was:

- Computing a vector whose element *i-th* is the mean value calculated from samples $\in [0, i]$ (for each repetition). The formula used to compute $v_n(i)$ is the same seen for warm-up time estimation.

- Computing the time $\bar{t}$ at which the sample standard deviation approached a good value. The sample standard deviation at timeslot i was computed among the mean values extracted at each repetition. More in detail, the sample standard deviation was computed

[1] In the case of saturation the value of response time increases with the simulation duration.
[2] We run 10 repetitions for each scenario that has been studied.

as follows:

$$S(i) = \sqrt{\frac{\sum_{j=0}^{n-1}(v_j(i) - \bar{v})^2}{n-1}}$$

where n represents the number of repetitions for each scenario. The **simulation period** was chosen such that $\bar{t} = \frac{i}{1000}$, where $i = max(j) \quad : \quad S(j) - S(j-1) \geq 0.01$.

Considering the worst case scenario (Uniform CQIs, $\lambda = 5$), we decided to chose the interval [0, 2] s as the warm-up period and 10 s as the simulation duration. Detailed test results can be found in Appendix B.

Chapter 4

Simulations

4.1 Introduction

We decided to analyze our KPIs[1] in the scenario described in Table 4.1.

Scenario		
Parameter	**Value**	**Measure unit**
Timeslot	1	ms
Population	10	
Transmission delay	0	s
#RB	25	
Max packet size	75	bytes
Warm-up time	1.75	s
Simulation duration	10	s

Table 4.1: Simulation scenario.

To obtain true independent random arrivals, 10 RNGs are required, each of which feeds the exponential distribution used by the `Sources`. Continuing the `Sources`, we need RNGs to generate packets whose size is a RV $\sim U(1, 75)$: others 10 RNGs are needed. We also need a RNG for each `Cellular` to generate random and independent CQIs at each timeslot. Furthermore, to model the random order in which the `Antenna` receives CQIs from the `Cellulars`, we have decided to shuffle the vector in which we stored the CQIs received at each timeslot. To do that we need to instantiate a RGN to feed a uniform distribution used to randomly shuffle the CQI's vector.

Overall the model requires 31 RNGs each of them initialized with a different seed in order to have independent pseudo-RVs. The seed-set is changed at every repetition to have different independent experiments. At the end of all repetitions the results are aggregated by computing the mean and 95% confidence intervals.

4.2 Uniform CQIs

In this section we will consider `cellulars` which, at each timeslot, generate CQIs with an integer uniform distribution. This approximation simulates the scenario where users randomly move within the cell. If the user experiences a strong signal then its `cellular` will send high CQIs. Otherwise, if the user experiences a weak signal, it will send low CQIs.

In Chapter A we proved that network is fair[2] so we expect that all users will have similar values for the mean throughput and the mean response time.

[1] Mean slotted throughput and mean response time.

[2] If it is stable. The network is considered stable if the mean number of queued packets at the Antenna will reach a constant value.

4.2.1 Throughput

For each repetition we gathered the mean slotted throughput seen by the cellulars connected to the network. We consider as workload, for each `source`, the mean arrival rate of packets λ on each timeslot, so that we can study the variation of the *mean throughput* as λ increases.

The simulation runs 30 repetitions for each value of λ. We expect that the throughput increases as λ increases, because there are more and more packets, traveling across the network, to be delivered.

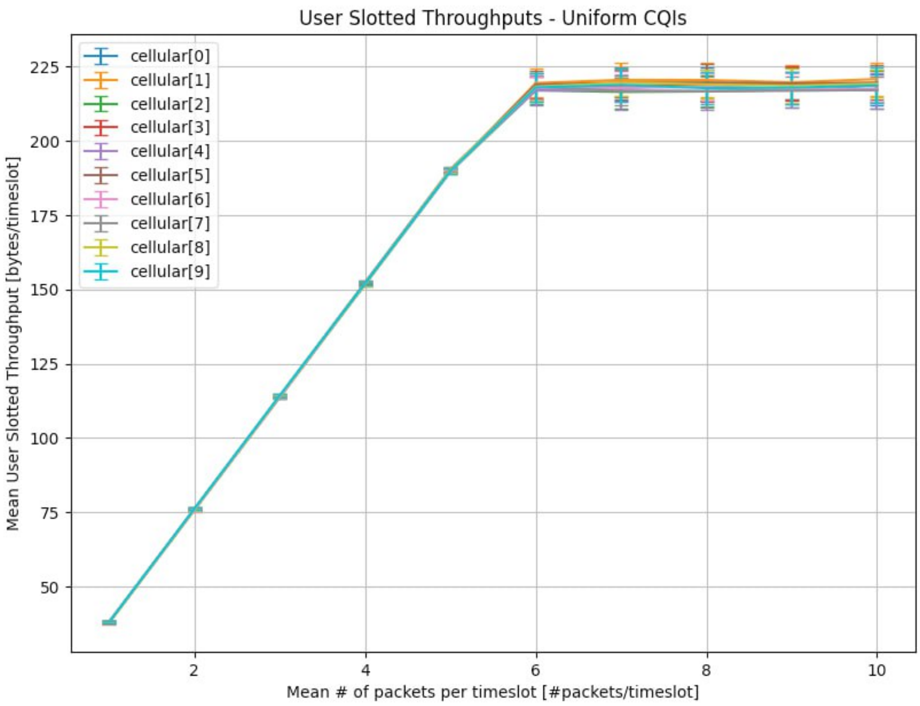


Figure 4.1: Mean user slotted throughput as a function of packets arrival rate - $\lambda \in [0.125, 10]$ (Exponential interarrivals, Uniform Service Demands, Uniform CQIs).

According to Figure 4.1, when the number of packets grows significantly, the mean throughput **E[Th]** reaches the maximum value at λ_{SAT} (which depends on configuration factors) before going in **saturation** as the Antenna gets busier. The saturation occurs at the same λ for each cellular because the opportunistic policy doesn't come in play: each cellular has the same probability to be served due to the uniform CQIs. Furthermore, again due to the fact that each user has the same probability of being served, the mean throughput remains similar for all users even after the system has reached saturation. Also, it's worth to notice that the system becomes unstable once it reaches its saturation point.

4.2.2 Response time

Before starting with the analysis of the results obtained, it must be said that the collected response times were not independent (the complete analysis can be found at Appendix C). To deal with this situation we decided to run simulations consisting of 30 repetitions. From each repetition we collected the mean response time experienced by each cellular connect to the Antenna. Each mean response time is a RV itself and, given the fact that the repetitions are set up to be independent from each others, we computed the mean response time each cellular experienced simply aggregating the mean response times obtained from each repetitions.

The mean response time should be analyzed if and only if the system is stable. As shown by Figure 4.2, if $\lambda \geq 5$, the mean user response time starts to grow abnormally. We suspect that the system cannot reach a steady state if the packet arrival rate $\lambda \geq 5$ packets per timeslot.

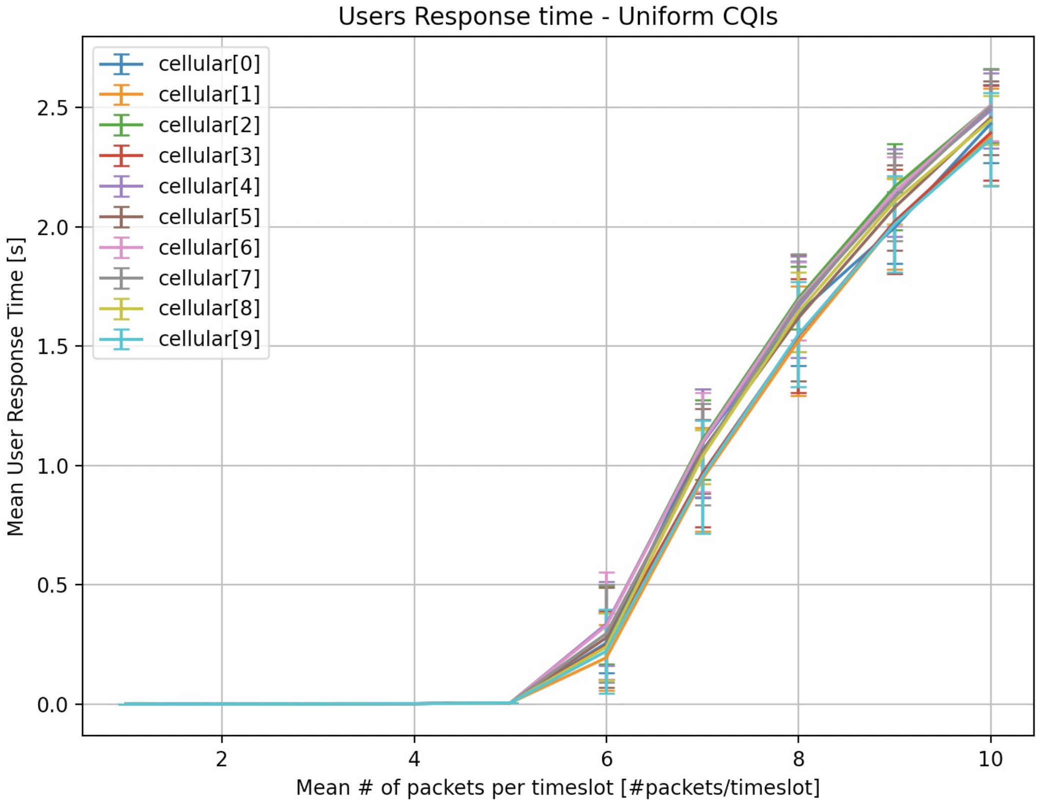
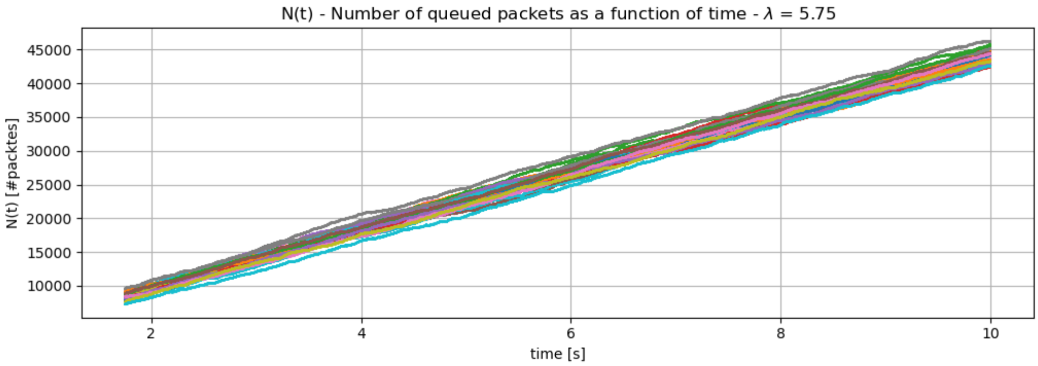


Figure 4.2: Mean user response time as a function of packets arrival rate - $\lambda \in [0.125,\ 10]$ (Exponential interarrivals, Uniform Service Demands, Uniform CQIs). Mean response times starting from $\lambda \geq 5$ shouldn't be considered because the system is no longer stable.

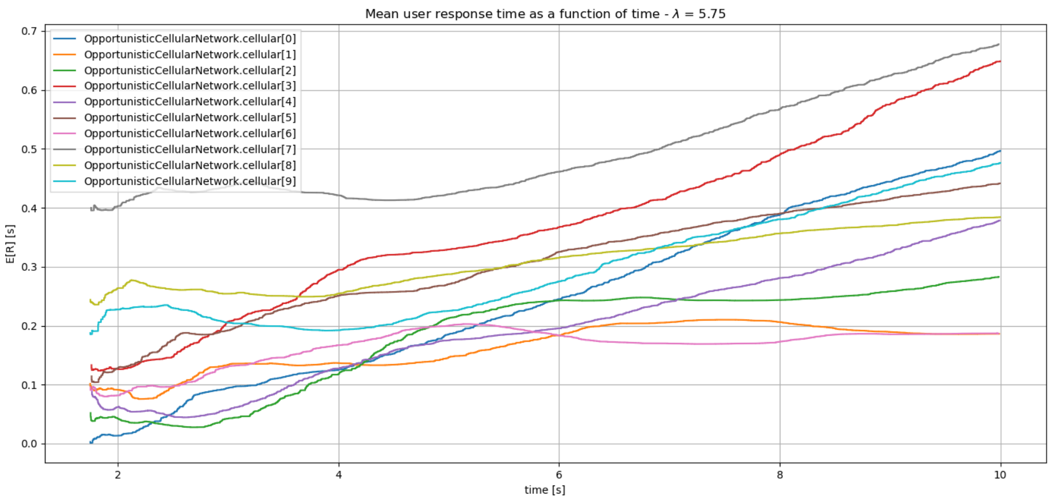
Our intuition is confirmed by Figure 4.3. When the arrival rate $\lambda \geq 5.75$ packets start to queue up because the Antenna can't handle the workload. For this reason mean response times should be considered only for $\lambda \in (0, 5.5]$.

As shown in Figure 4.4, the Antenna fairly serves its users. For each arrival rate that has been considered, the mean response times experienced by users are very similar. This behaviour can be explained noticing that the effect of opportunistic policy doesn't come in play because each user has the same probability to experience a strong/weak signal (high/low CQIs). If the CQIs are uniformly distributed, each user has the same probability to be served so the response times are very similar.

The cellular network reaches a saturation point around $\lambda \in [5.5,\ 5.75]$.



(a) Number of queued packets as a function of time.



(b) Mean user response time as a function of time.

Figure 4.3: Unstable scenario: λ = 5.75 #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

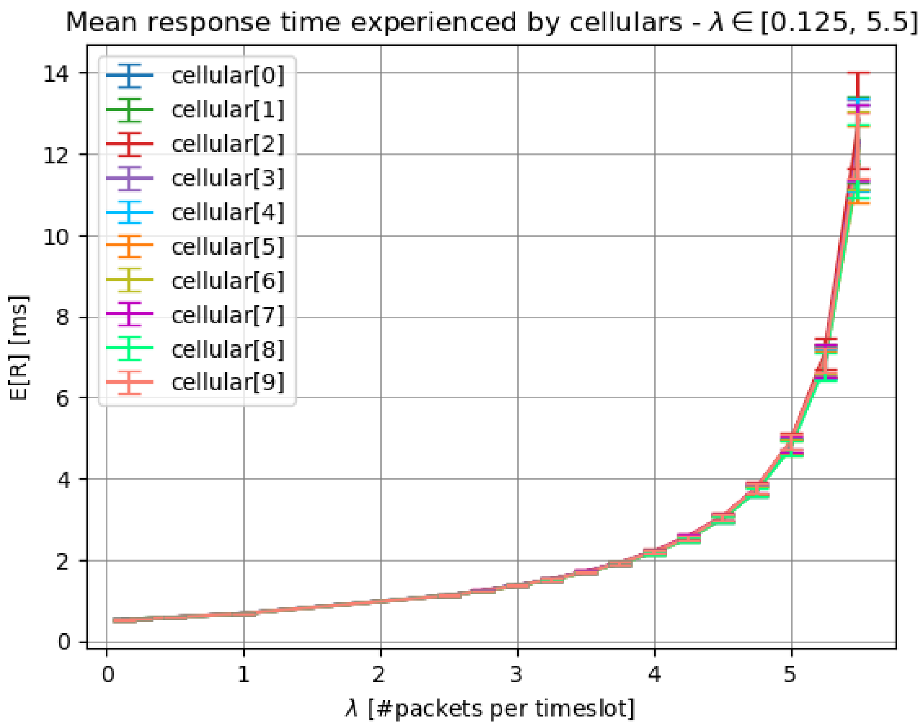


Figure 4.4: Mean user response time as a function of packets arrival rate - $\lambda \in [0.125, 5.5]$ (Exponential interarrivals, Uniform Service Demands, Uniform CQIs).

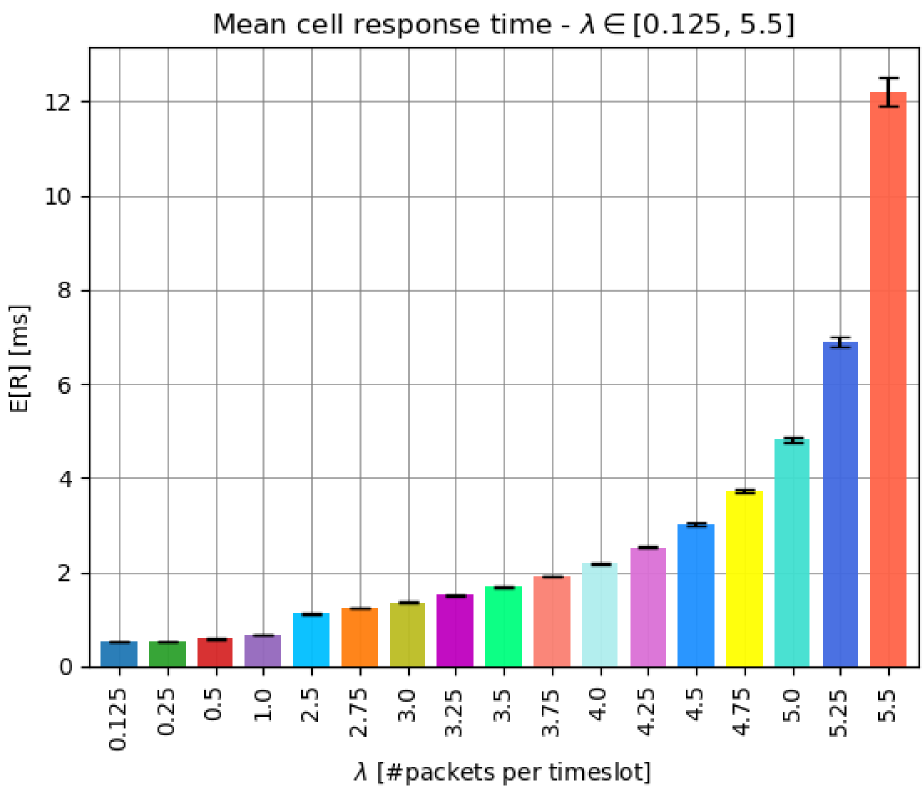


Figure 4.5: Mean cell response time as a function of packets arrival rate - $\lambda \in [0.125, 5.5]$ (Exponential interarrivals, Uniform Service Demands, Uniform CQIs).

4.3 Binomial CQIs

In this section we consider `cellulars` which CQI number for each timeslot will be taken from a Binomial distribution. Apart from the parameters regarding CQIs, the value used are the same seen in the case of uniform CQIs as seen in Table 4.1.
The specification tells us to have sensibly different mean CQIs for each user so the RV CQI at each timeslot for user i was computed as

$$CQI = 1 + Bin(p_i, n), \qquad n = 14,\ i = 0, 1, ..9,\ p_i \in [0, 1]$$

Those values were chosen to get CQIs $\in [1, 15]$. The values for p_i were chosen as follows:

User	p_i
0	0.1
1	0.2
2	0.3
3	0.4
4	0.5
5	0.6
6	0.7
7	0.8
8	0.9
9	0.95

Table 4.2: Binomial CQI parameters

This models a network were each user senses a different Channel Quality from the others. We choose only to have 10 users because it was easier to make this difference noticeable in the plots, and we can always increase the overall workload of the `Antenna` by increasing λ.

4.3.1 Throughput

We first study the **Mean Slotted Throughput**. The workload values in this case were chosen to bring up the effect of the scheduling policy. For this statistic we expect that with increasing workloads the users with lower mean CQIs will experience lesser and lesser throughputs than the ones with highest means.

Our intuition on the behavior of the system is confirmed by Figure 4.6. For every user i, apart from the ones with highest mean CQI, we see that $E[Th]_i$ reaches its maximum around λ_i^{SAT} [3] then it decreases more and more eventually tending to 0. This behaviour is due to the effect of opportunistic policy: the Antenna allocates an increasing number of RBs to cellular with high CQIs, causing the number of allocated RBs for cellular with low CQIs to decrease to 0 (and so their throughputs tend to 0), as the packet arrival rate increases. It's straightforward to notice that as the workload increases the Antenna tends to exclude cellulars with the lowest mean CQIs, penalizing their performances.

Lorenz Curves

After observing how the throughputs tend to decrease, it was interesting to plot a Lorenz Curve of the **Slotted Throughput** (Figure 4.7) too see better how the system in this scenario increases its unfairness as values of λ increase. It is also interesting to compare the Lorenz Curve in the case of Uniform CQIs (Figure 4.8). We can clearly see that in this case, the fairness of the network its not influenced by the workload because each user has the same probability of experiencing high/low CQI. In fact, for every λ, the curve almost fits perfectly the maximum fairness line.

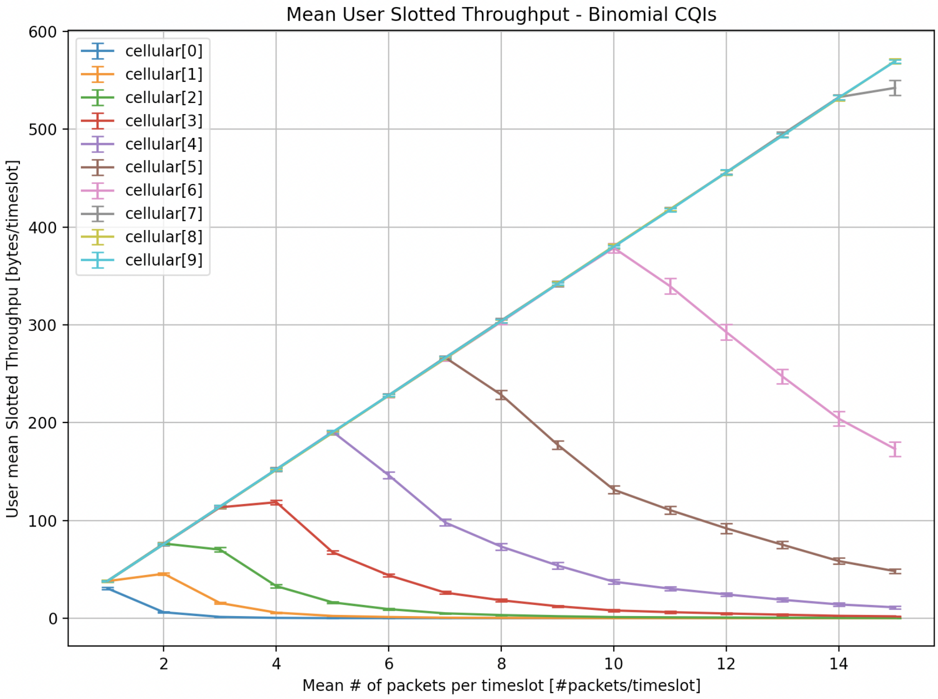


Figure 4.6: Mean slotted Throughput (Exponential interarrivals, Uniform Service Demands, Binomial CQIs)

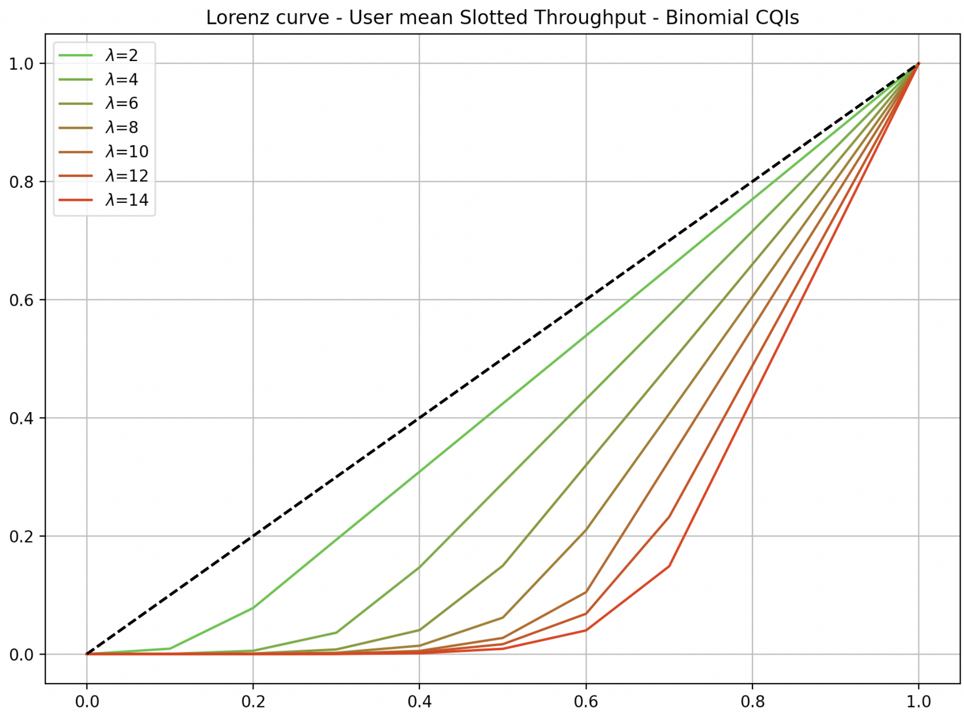


Figure 4.7: Lorenz Curve - Mean slotted Throughput (Exponential interarrivals, Uniform Service Demands, Binomial CQIs)

[3]This value depends on configuration factors. Such as **population** and p_i

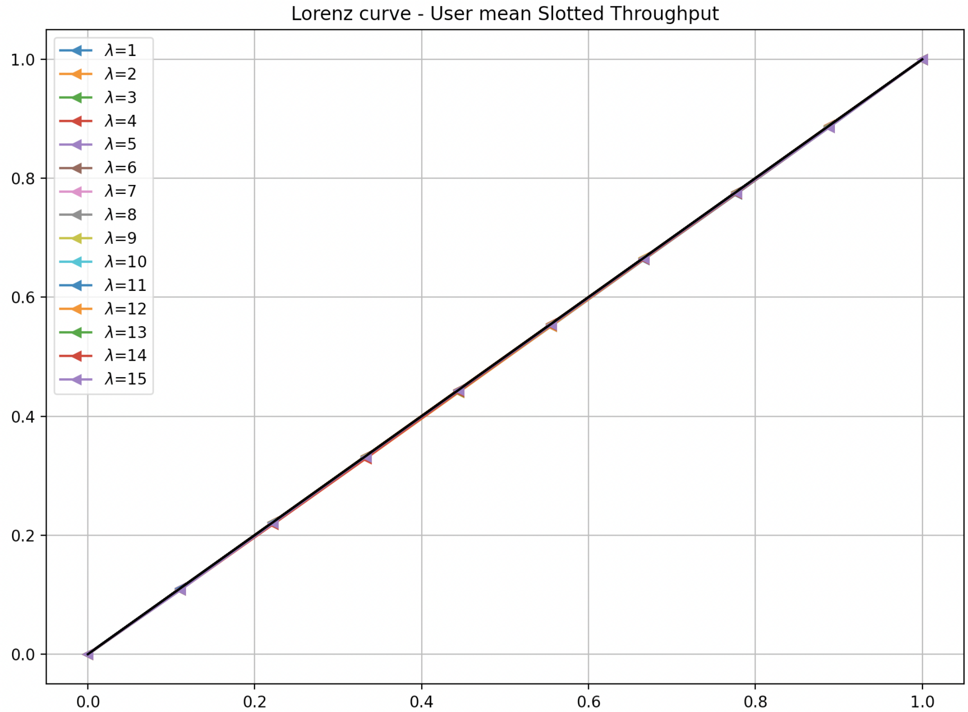


Figure 4.8: Lorenz Curve - Mean slotted Throughput (Exponential interarrivals, Uniform Service Demands, Uniform CQIs)

4.3.2 Response Time

As explained in Section 4.2.2, we run a simulation of 30 repetitions in order to obtain independent RVs for the mean response time of each cellular.

Due to the unfairness of the network, cellulars with low CQIs are disadvantaged. When the packet arrival rate reaches the saturation rate of one of the cellulars with low CQI, it is served less and less due to the opportunistic policy. In addition to the decrease in throughput, the cellphone in question experiences longer response times (as expected). Unfortunately, the response time increases with the increase of the simulation time as, from the point of view of the cellphone, the system is unstable.

No matter how long the simulation is, the number of packets received by the low CQIs is always less than that of the high CQIs, since the sources are constantly sending new messages so the network will never unload. This leads to the fact that the mean response time for the cellular with lower CQIs will be computed on fewer samples and also the magnitude of the samples tends to increase. In other words the sample width and the sample variance will not be constant. The results[4] would not be consistent.

Fixed number of packets N

To study the phenomenon we decided to gather the results obtained until the reception of N packets for every cellular without changing the load of the network. When even the last cellular has reached the fixed number the current run is stopped. The network is at his full potential, but we study the behaviour in function of N packets received by each cellular, so it does not depends on the duration of the simulation. For the experiment we consider N = 2000 packets to receive.

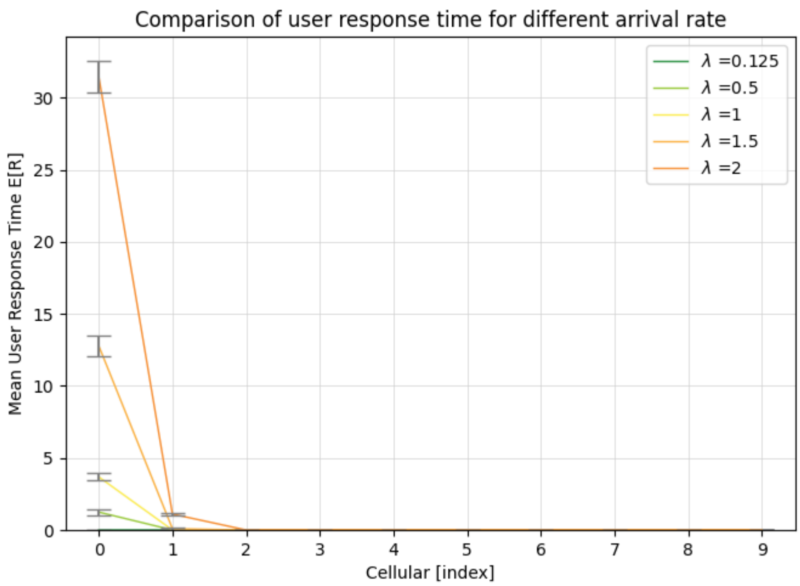


Figure 4.9: Mean user response time as a function of CQI - $\lambda \in [0.125,\ 2]$.

Figure 4.9 shows how the increases of λ results in the evident rise of the mean response time E[R], especially for the low range of CQIs, while the cellulars in the high range experience a more stable condition. Figure 4.10 shows a zoomed-in view to point out the differences of the decreasing speed of the lines.

[4]Figure D.1 shows us how unreliable it is the mean response if computed without considering a minimum number of packets received for each cellular.

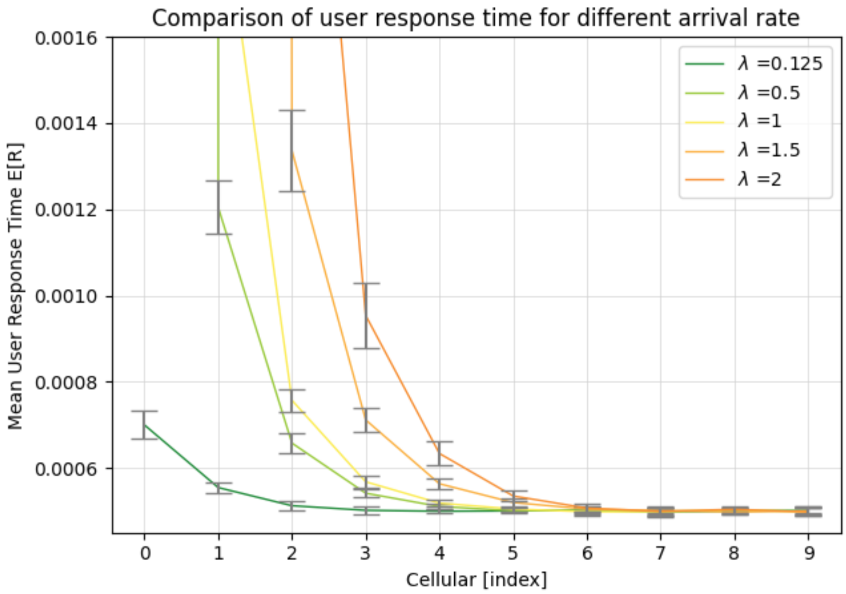


Figure 4.10: Mean user response time as a function of CQI (zoom) - $\lambda \in [0.125,\ 2]$.

Given the previous considerations on the difficulties of analyzing response times in this case we decided to analyze another scenario where instead of having different p_i for every user i we had 2 groups of users, divided in classes. We called those 2 "Low class" and "High class", having respectively the Binomial success probabilities $P_{low} = 0.4$ and $P_{high} = 0.8$. [5]

We plotted the Mean Response Time (Figure 4.11) for the two classes computing the mean across every user in that class. In this case we also had a population of 20 users instead of 10 (10 for each class) to have better confidence intervals.

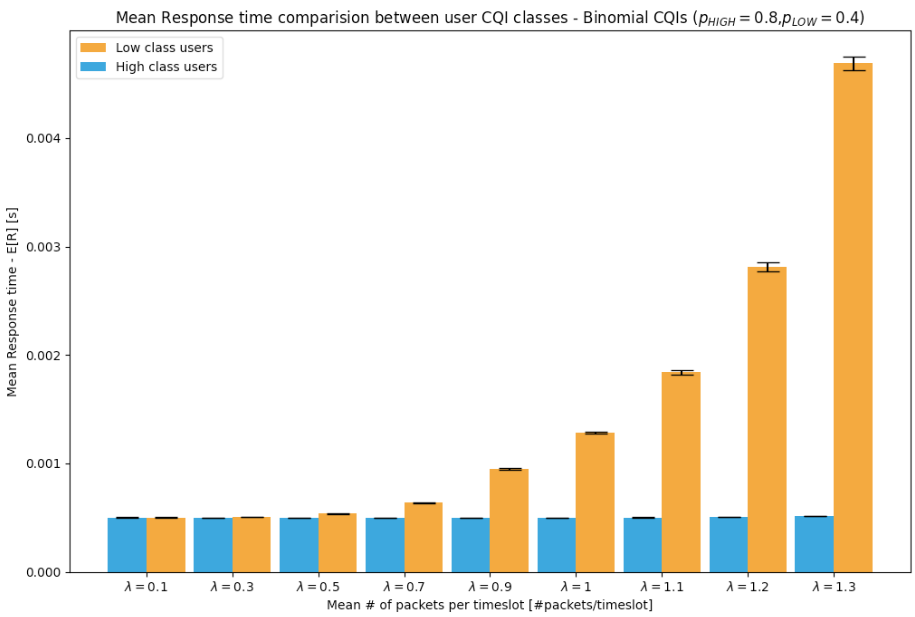


Figure 4.11: Mean User Response Time (per class) - $\lambda \in [0.1,\ 1.3]$.

[5] This choice was made because it was easier to calibrate a workload where we didn't have the previous problem on collecting a sufficient number of samples but still conveys the unfairness of Binomial Scenario.

It's clear to see that after a certain workload the low class user starts to experience higher response times as the Antenna prioritizes the higher class ones, that instead remain stably around the same quality of service. Furthermore, it is worth to note that although the mean response time of the two classes is comparable at least for $\lambda \in [0.1, 0.5]$, low CQI class experiences worse response times respect to high CQI class starting from $\lambda \geq 0.3$. This can be inferred from the empirical CDFs (Figure 4.12, Figure 4.13 and Figure 4.14).

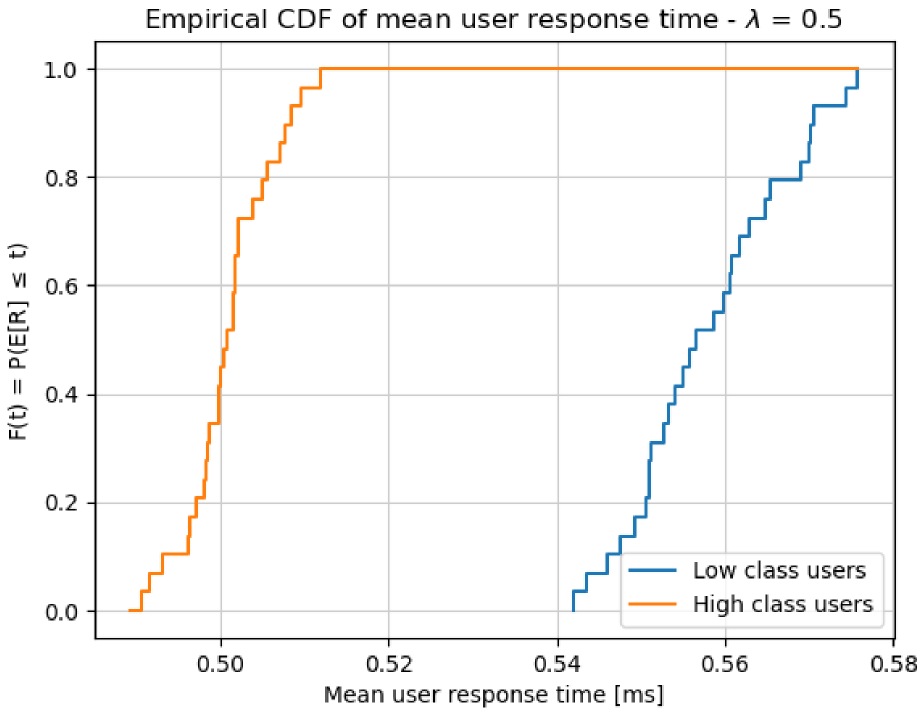


Figure 4.12: Empirical CDF of mean response time (per class) - $\lambda = 0.5$.

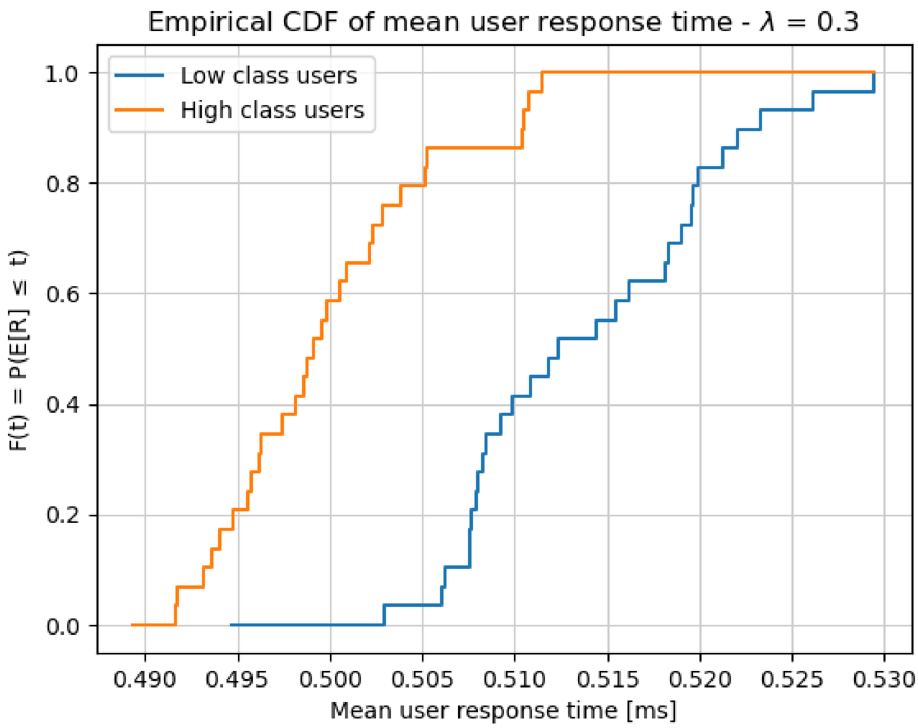


Figure 4.13: Empirical CDF of mean response time (per class) - $\lambda = 0.3$.

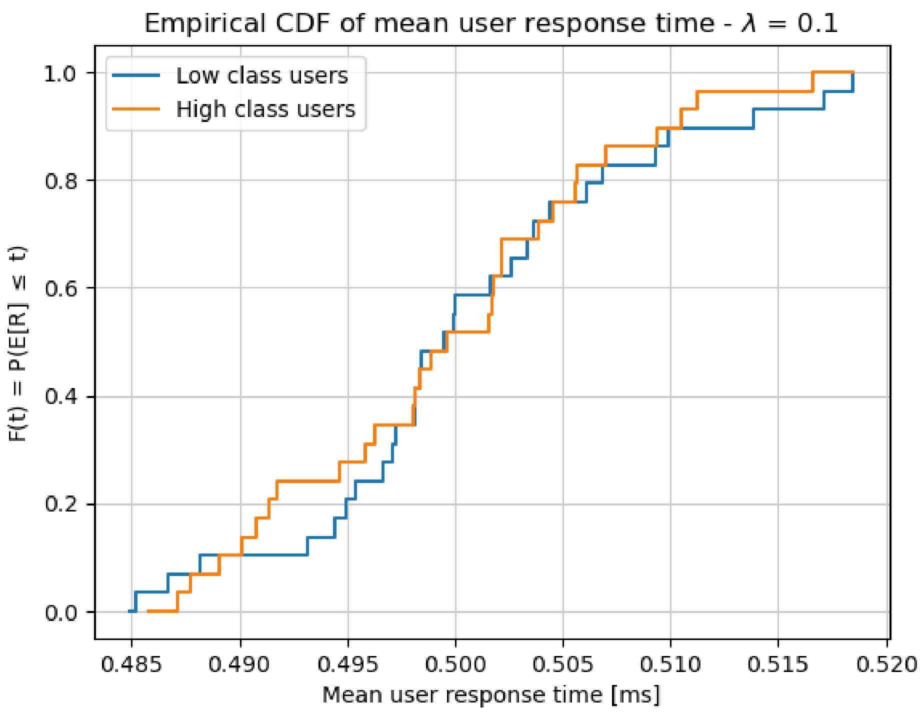


Figure 4.14: Empirical CDF of mean response time (per class) - $\lambda = 0.1$.

4.3.3 Resource Block Allocation

We used the same class scenario to give some insights on the Frame utilization with users that sense different qualities of the channel. We plotted the Mean number of Resource block utilized per user across users of a certain group.

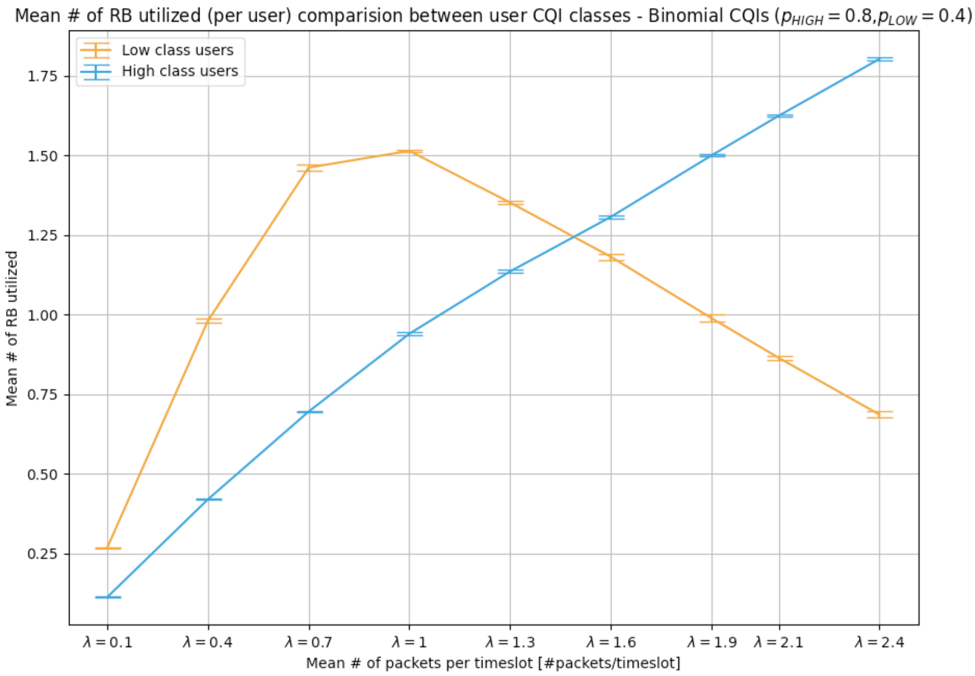


Figure 4.15: Mean Number of RB utilized per user comparison between classes- $\lambda \in [0.1, 2.4]$.

This statistic gives us informations about how much of the frame is utilized by each group. We can say that, at low workloads, the majority of the frame is used by low class users because the Antenna allocates a high number of RBs to handle all the packets in low class' queues since the mean RB size is much lower than the high ones. Then around the same λ for which the response times for the low class group starts to increase, the high ones start to gain priority over them, for even higher workloads the entire frame gets allocated only to high class users leaving the firsts to experience starvation.

Chapter 5

Conclusions

The opportunistic scheduling policy, in general, prioritize users which experience a better quality of network. The scheduling policy aims to maximize throughput and, as a result, the cellulars experiencing high quality of network also experience high throughput.

What is the purpose of having an opportunistic policy? We recommend this type of scheduling policy for the purpose of building a network which advantages the user depending on its perceived network quality.

On the other hand, if we want to ensure a minimum quality of service for all users, this type of policy could lead to unsatisfactory results if the perceived network quality among all users is heterogeneous. Note that the performance experienced by users depends only on the quality of the network and never on the load received previously.

A possible solution to obtain fairer scheduling policy, even in the case of heterogeneous network quality, is to consider the load of the packets previously sent as a parameter in resource allocation policy.

Appendix A

Validation and verification

A.1 Fixed CQI, fixed interarrival times, fixed packet size, 1 user

In this test we have just one `Cellular` connected to the antenna wich always generates the same CQI at each timeslot. `Source` sends packet with a fixed rate; the packet size is fixed as well. This is a very simple system with deterministic arrivals and deterministic service demand.

Since we know exactly the number of bytes that the Antenna receives/sends at each timeslot (we are in a scenario with deterministic arrivals and fixed service demand[1]) the throughput can be computed as described below:

$$THR = \frac{\#bytes\_sent \times 8}{timeslot} \quad [bps]$$

The mean response time can be computed only considering a single timeslot since the behaviour of the system repeats itself periodically (period = timeslot). The periodic behaviour can be explained noticing that:

- In our model the service time is zero. Therefore, the departure times of the packets are the same because each served packet leaves the Antenna at the instant in which a new timeslot begins.
- With an arrival period equal to 0.20 ms, 5 packet will arrive to the Antenna at each timeslot.
- Considering the last two statements, a packet whose arrival time $\in [t_i, t_{i+1}]$[2] will leave the queue at t_{i+1}. No packets whose arrival time $\in [t_i, t_{i+1}]$ will leave the queue at t_{i+2} (no overflow) because the the Frame's size is large enough to handle the workload.

Considering the first timeslot, $[0, 1]\, ms$, the formula is the following:

$$E[R] = \frac{1}{n} \sum_{n}^{i=1} (1 - Arrival\_time\_Queue(i)) \quad [ms]$$

where n is the number of arrivals at each timeslot. With these parameters we expect the following results:

- The system is at its *steady state* because the total number of bytes that the Antenna receives at each timeslot ($50 \times (\frac{1}{0.20}) = 250$) are less than the maximum number of bytes which the Frame can carry out ($93 \times 25 = 2325$).

[1]Packet dimension.
[2]$[t_i, t_{i+1}]$ represents a timeslot.

- Since the system is at its steady state, the number of bytes that the Antenna sees incoming at each timeslot must be equal to the number of bytes that the Antenna sends at each timeslot. The following statement can be proved checking if the Antenna's throughput (output throughput) is equal to Cellular's throughput (input throughput).

Parameters				
	Description	**Formula**	**Value**	**Measure unit**
P	Arrival period		0.20	ms
T	Timeslot		1	ms
n	# arrivals at each timeslot	n = T / P	5	
D	Packet dimension		50	bytes
At(i)	Arrival time packet i^{th}		[0.2, 0.4, 0.6, 0.8, 1]	ms
Expected Results				
	Description	**Formula**	**Value**	**Measure unit**
$\#BS_A$	# bytes sent Antenna	$\#BS_A = n \times D$	250	bytes
THR_A	Antenna's throughput	$\frac{\#BS_A \times 8}{timeslot}$	2	Mbps
$\#BS_C$	# bytes received Cellular	$\#BS_C = n \times D$	250	bytes
THR_C	Cellular's throughput	$\frac{\#BS_C \times 8}{timeslot}$	2	Mbps
E[R]	Mean Response Time	$\frac{1}{n}\sum_{n}^{i=1}(1 - At(i))$	0.4	ms

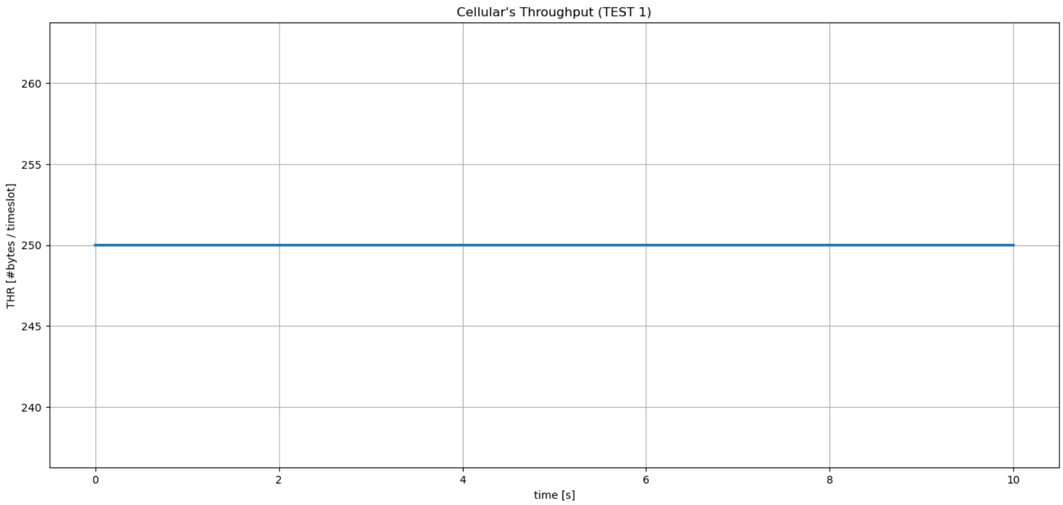


Figure A.1: 1^{st} Test - Cellular Throughput.

As shown in Figure A.1 and Figure A.2, both `Cellular`'s throughput and `Antenna`'s throughput are equal to `250 bytes/ms (2 Mbps)` and both are constant through the entire simulation.
Also, we computed $E[R]$ as a function of time. The formula is the following:

$$E[R](t) = \frac{\sum_{t_i \le t} R(t_i)}{\sum_{t_i \le t} 1_{R(t_i)}} \quad [ms]$$

where $R(t_i)$ is the response time of the packet whose departure time is t_i.
As shown in Figure A.3, the plot reports a peak around $t = 0$. The behaviour shown by $E[R](t)$ around $t = 0$ is due to the fact that the sum by which it is computed has too few values.

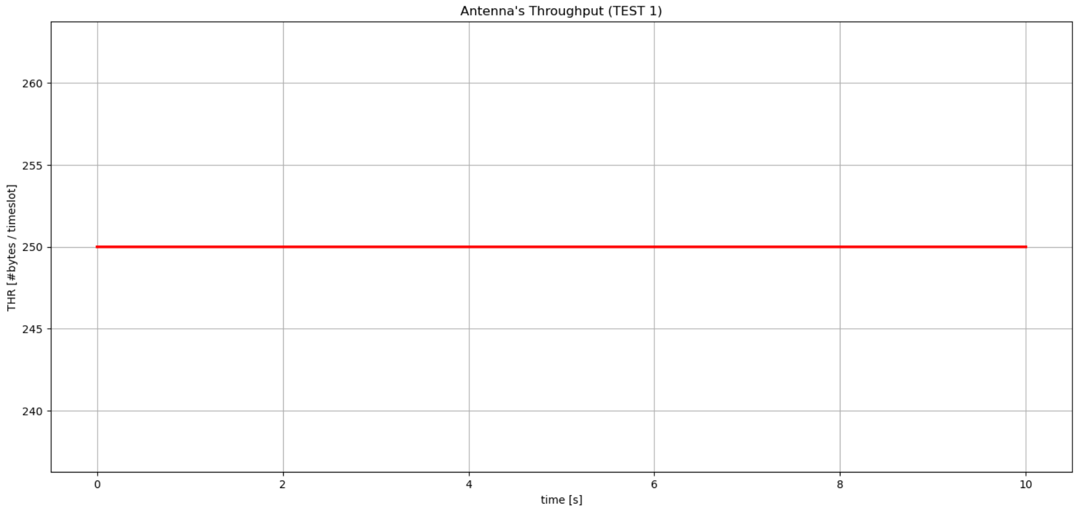


Figure A.2: 1^{st} Test - Antenna Throughput.

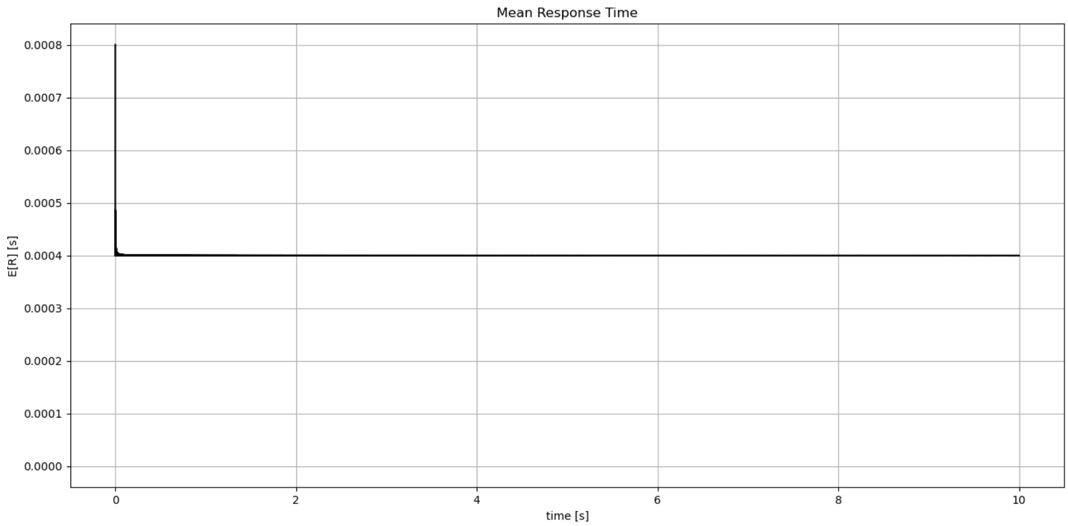


Figure A.3: 1^{st} Test - Mean Response Time.

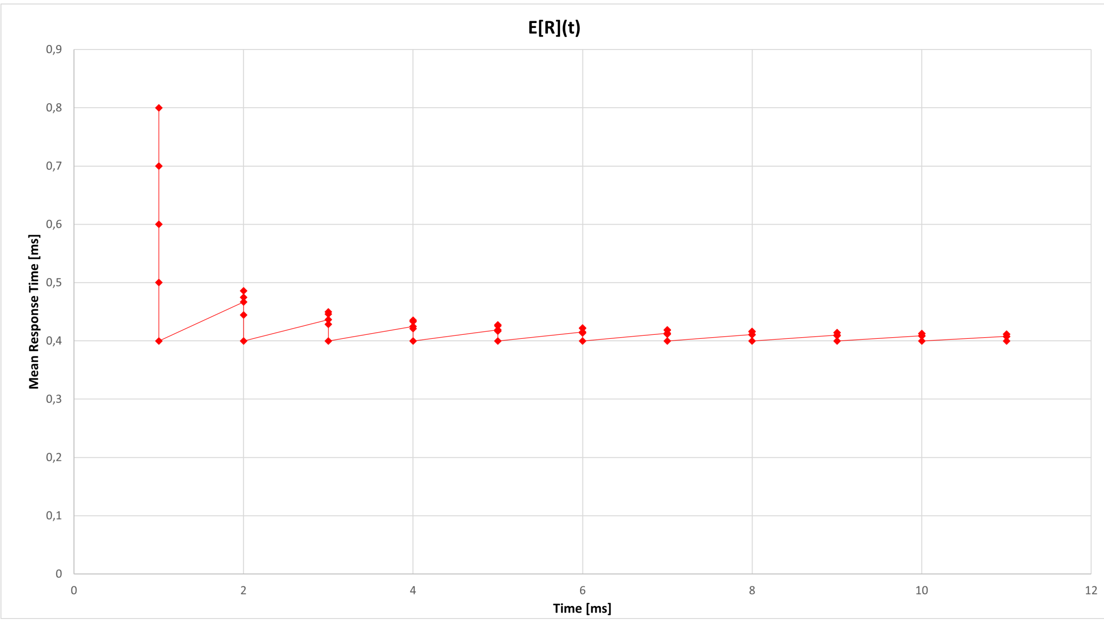


Figure A.4: 1^{st} Test - Mean Response Time - Zoom around t = 0s.

A.2 Degeneracy Test

With this test we want to study the state of the system as a function of the packet arrival rate. Let's consider the following scenario:

- One `Cellular` connected to the `Antenna`.
- Arrivals are deterministic and periodic: a `Source` sends packet with a fixed rate.
- Service times are deterministic (the system has been modeled with service time = 0).
- Both packet size and CQI are fixed and set to the maximum value allowed by the system.

If we consider the state of the system as the *number of backlogged packets* we expect that, if the system is at its steady state, the mean number of backlogged packets will reach a constant value. On the other hand, when the system cannot reach a steady state, we expect that the number of backlogged packets diverge. The previous statement can be explained reasoning in the following terms:

- Since arrivals are periodic, at each timeslot the `Antenna` receives N packets.
- Also, both packet size and CQI are set to the maximum value allowed by the system. In this scenario, at each timeslot, the `Antenna` allocates a Frame made of 25 RBs. Each RB carries 1 backlogged packet.
- At each timeslot, the `Antenna` sends exactly 25 packets. If $N > 25$ then at timeslot $i \geq 1$, $U_p(i) = N - 25 + U_p(i-1)$ packets will not be sent. Notice that $U_p(i)$ represents exactly the number of backlogged packets at timeslot i.
- Since the queue has infinite capacity, no packet will be rejected upon its arrival.
- Due to the previous considerations, the number of backlogged packets will diverge as time increases.

As shown in Table A.1, it is quite easily to compute the number of unsent packets at timeslot i. For instance, if we instantiate the arrival period $P = 0.02$ ms, we can clearly see that after few timeslots the number of backlogged packets is quite big.

$U_p(i)$ - Number of unsent packets at timeslot i				
Parameters				
	Description	**Formula**	**Value**	**Unit of measure**
P	Arrival period		0.02	ms
T	Timeslot		1	ms
N	# arrivals at each timeslot	N = T / P	50	
$U_p(0)$	Initial condition		0	
Results				
timeslot i		$U_p(i) = N + U_p(i-1) - 25$, $i \geq 1$		
1		25		
2		50		
5		125		
10		250		
25		625		
50		1250		
100		2500		

Table A.1: Number of unsent packets at timeslot i.

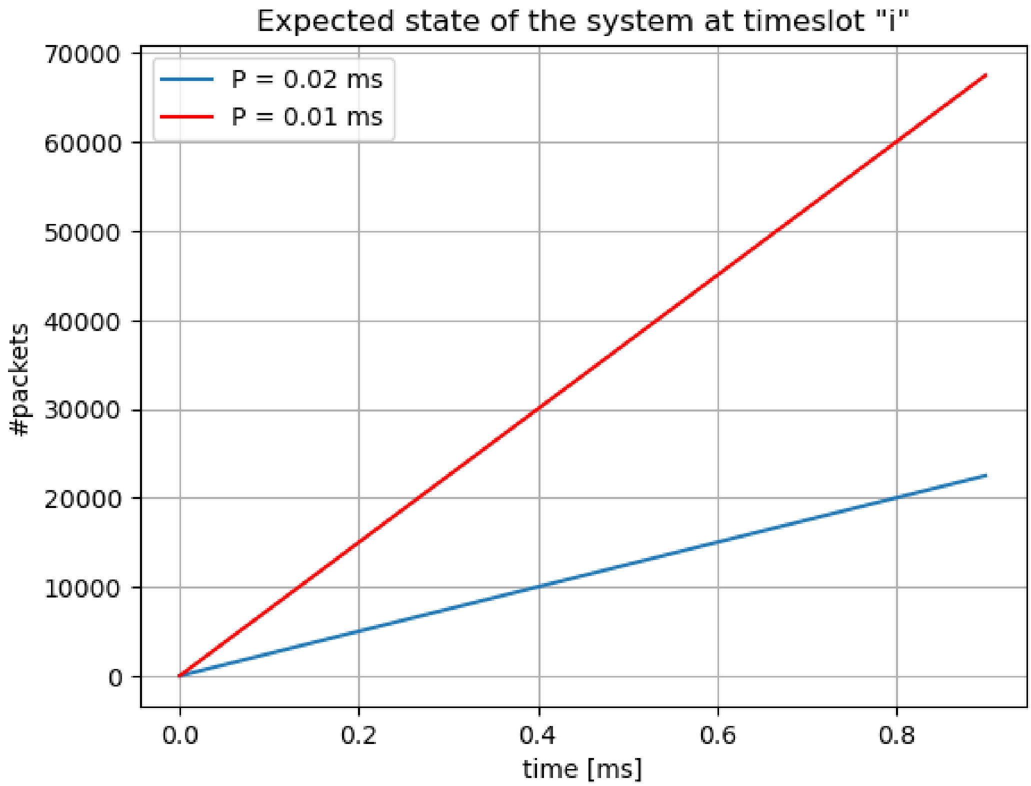


Figure A.5: Degeneracy test - Expected state of the system as a function of time.

To prof that the system will not reach a steady state (when the arrival rate > departure rate), we can approximate the state of the system with the number of unsent packets $U_p(i)$. The latter is a good approximation because at any time t the state of the system is given by $A_p(t_i, t) + U_p(i)$:

- $A_p(t_i, t)$ - Number of packets that have arrived between $[t_i, t]$, where t_i is the previous expired timeslot.
- $U_p(i)$ - Number of unsent packets at previous expired timeslot t_i.

As time increases, $A_p(t_i, t)$ becomes negligible in comparison with $U_p(i)$.

From a theoretical standpoint, we expect that $N(t)$, i.e. the number of backlogged packets, behaves similarly to the number of unsent packets, $U_p(t)$ (Figure A.5).

The simulation has been conducted with different arrivals period: $0.20, 0.10, 0.05, 0.02, 0.01$ ms. The reason why we conducted several simulations is due to the fact that we wanted to validate the behavior of the system both in its steady state and in unstable scenarios. In a deterministic scenario such as the one that we are validating, we can compute the *threshold* arrival rate:

λ_{TH} - **Threshold arrival rate**				
Parameters				
	Description	**Formula**	**Value**	**Unit of measure**
T	Timeslot		1	ms
P	Arrival period			ms
λ	Arrival rate	T / P		#packets per T
μ	Service rate		25	#packets per T
ρ	Utilization	λ/μ		
Result				
λ_{TH}	Threshold arrival rate	$\rho \geq 1 \rightarrow \lambda_{TH} \geq \mu$	25	#packets per T
P_{TH}	Threshold arrival period	T/λ_{TH}	0.04	ms

The simulations that have been conducted with arrival period above P_{TH} (Figure A.6) confirm our expectations: the plot clearly shows us that the state of the system diverges as time increases. The simulations that have been conducted with arrival period below P_{TH} (Figure A.7) confirm our expectations as well. Since the system is at its steady state, we studied the mean number of backlogged packets.

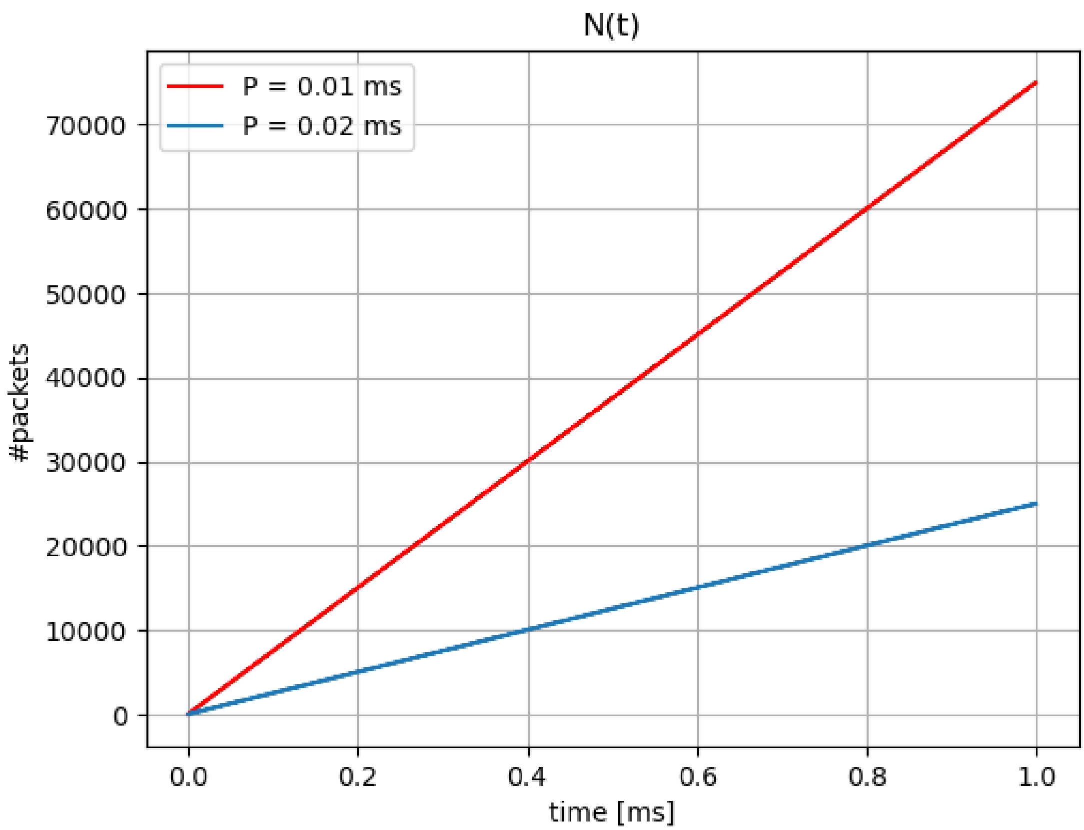


Figure A.6: Degeneracy test - Actual state of the system as a function of time.

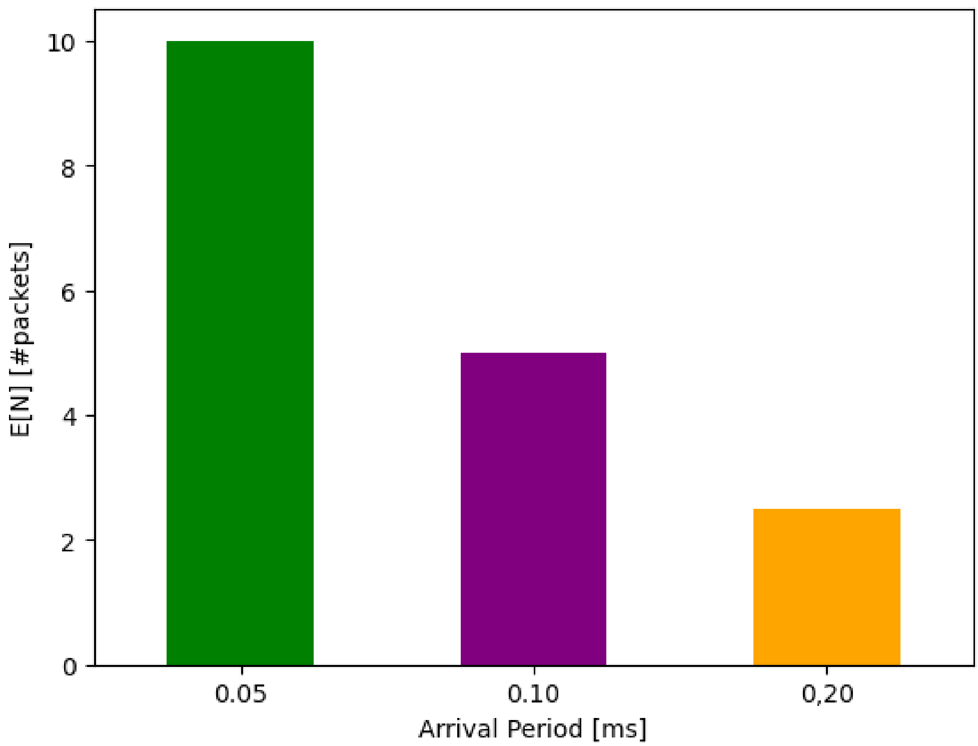


Figure A.7: Degeneracy test - Actual state of the system as a function of time (steady state scenario).

A.3 Consistency Test

With this test we want to evaluate the system in two different configurations so that the loads are the same. This can be done simply by noticing that, with different parameters for the two (arrival rate and population) the system will experience the same load. This test doesn't involve random parameters so that we can compute easily by hand the results obtained. For this instance the comparison is made by observing the Antenna `Throughput`.

The simulation has been set up with the following parameters:

```
# omnetpp.ini
# Consistency Test (first configuration): 1 user, fixed CQI = 10,
#                   deterministic arrivals, queue of infite dimension,
#                   fixed packet dimension = 25
network = OpportunisticCellularNetwork
[consistencyTest0]
**.population           = 1
**.cellular[*].TEST     = true
**.cellular[*].TEST_CQI = 10
**.source[*].TEST       = true
**.source[*].TEST_RATE  = 4
*.source[*].TEST_SIZE   = 25
```

We want to compare the following configuration with a similar scenario when we have 2 users, with halved arrival period in respect to the first config.

```
# omnetpp.ini
# Consistency Test (second configuration): 2 user, fixed CQI = 10,
#                   deterministic arrivals, queue of infite dimension,
#                   fixed packet dimension = 25
network = OpportunisticCellularNetwork
[consistencyTest1]
**.population           = 2
**.cellular[*].TEST     = true
**.cellular[*].TEST_CQI = 10
**.source[*].TEST       = true
**.source[*].TEST_RATE  = 2
*.source[*].TEST_SIZE   = 25
```

We expect to have same Antenna throughput in both cases. That can be computed as the sum of the single throughput of users. The formula is the same as in the first test. The first case with 1 user is trivial. By sending fixed dimension packets each 25 bytes long 4 times per timeslot we have a `Slotted Throughput` of 100 bytes.

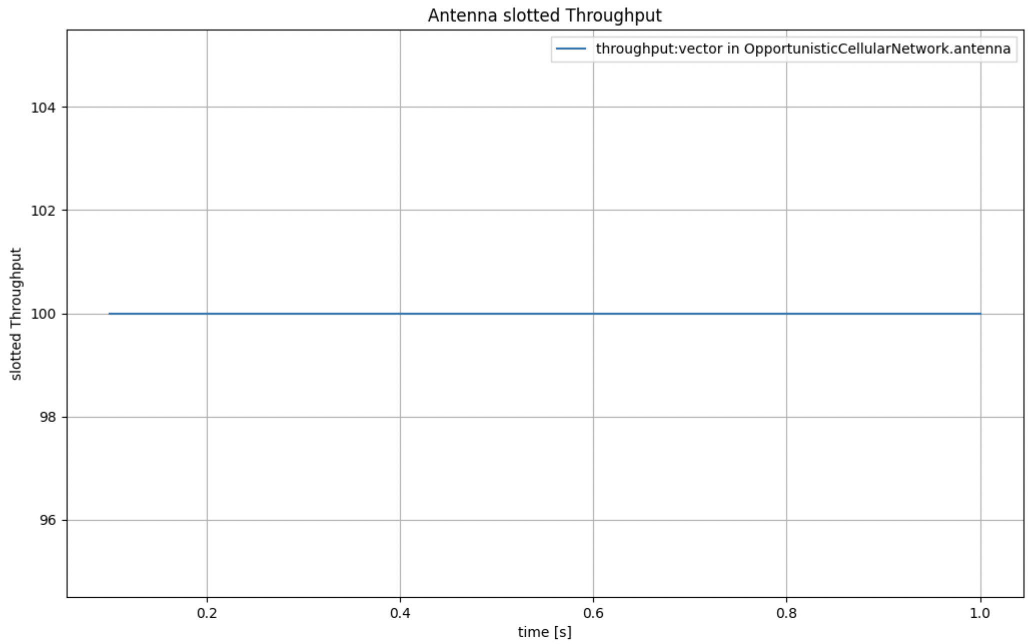


Figure A.8: Antenna Throughput (Consistency test - first configuration).

The second configuration gives us what we expected. By doubling users and halving the rate we have the same result as for configuration 1.

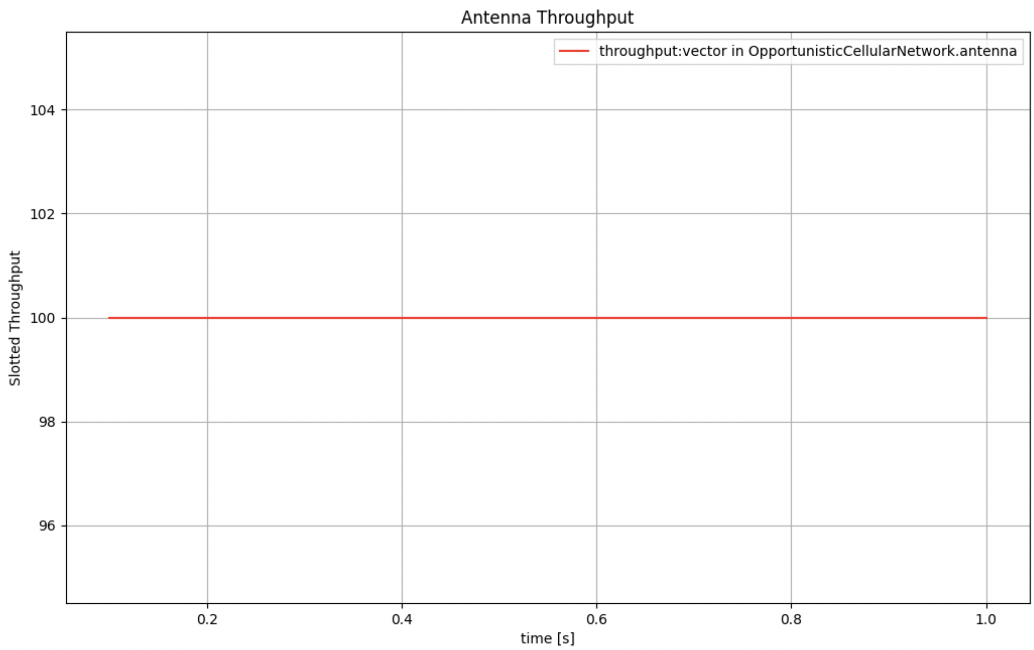


Figure A.9: Antenna Throughput (Consistency test - second configuration).

A.4 Continuity Test

The purpose of this test is to verify how the system behaves to small variations.

The mean of the User response time shows how much a cellular waits until it's served by the Antenna. The antenna, every time, will choose the mobile phone with higher CQI to be served.

If the arrival period of new packets increases, it is likely that the time taken to serve the phones with the lower CQIs decreases more and more.

For the experiment the CQIs of the phones are fixed to the same value (15 in this case), with the purpose to study the continuity of the response time to small variations. The graph below is obtained slightly varying the arrival period with values 0.27 ms, 0.29 ms and 0.31 ms and running multiple repetition.

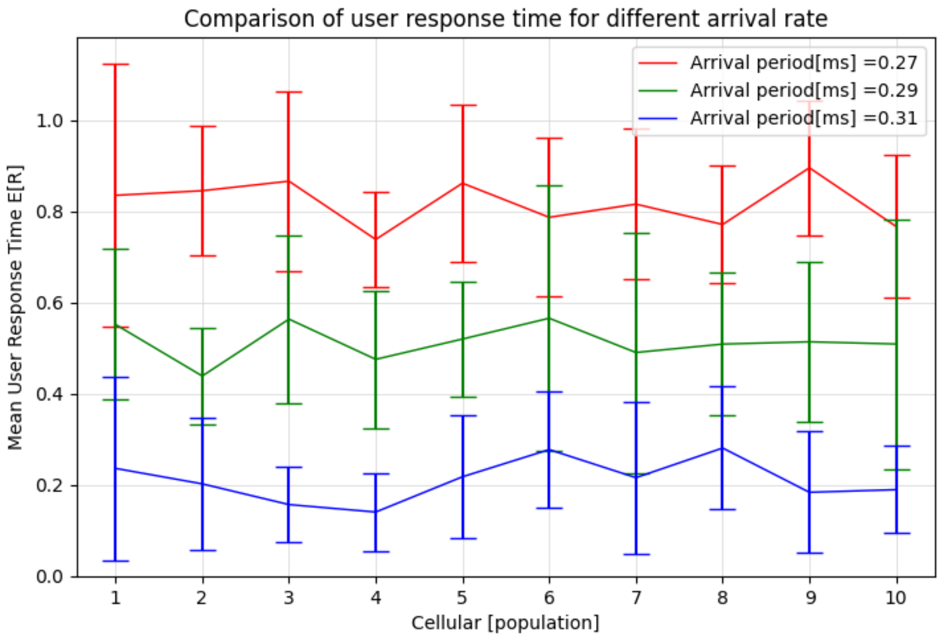


Figure A.10

According to the earlier considerations, the figure A.10 shows the expected behaviour. The small variation of the arrival periods results in the little variation of the response time too. This confirms that the system response is continuous and it can be seen also by the ranges of the CIs. The variation between arrival times is, in fact, so small that the response times experienced may overlap a little.

A.5 Validating the scheduling policy

Before showing results concerning the two different CQI distributions, we need to verify the behavior of the simulator core, the **scheduling policy** used by the Antenna. We recall that the Resource Block allocation policy of the network we study is an *opportunistic* one. So the users with higher **C**hannel **Q**uality **I**ndicator (i.e. CQI) for a particular timeslot will get priority in RB allocation over the others. Since the maximum frame dimension is fixed to 25 RBs, users with lower CQIs will tend to experience lower throughput in situations where the loads start to get higher.

The scenario for this verification will be the following:

- A population of 3 users.
- Different fixed CQIs for each user.
- Incremental workload (λ).

We want to study the system by incrementing the arrival rate of packet (λ) and noticing how it affects user's slotted throughput.

We expect that:

- With low values for λ the system is in a steady state, hence at each timeslot the Antenna can empty all users' queues and satisfy all users demands. In this scenario we expect a **fair behavior with regards to user throughput**.
- With higher values of λ we expect the opportunistic behavior of the scheduling policy used by Antenna to come in play. With so, we expect the users with lower CQIs to experience lower throughput[3] than the user with the highest. With the worst case approaching a scenario where the user with the highest CQI will get all RBs, leaving other 2 users to "starve" and experience very low throughputs.

Here's a chart of the mean slotted throughput for each user at varying workload values.

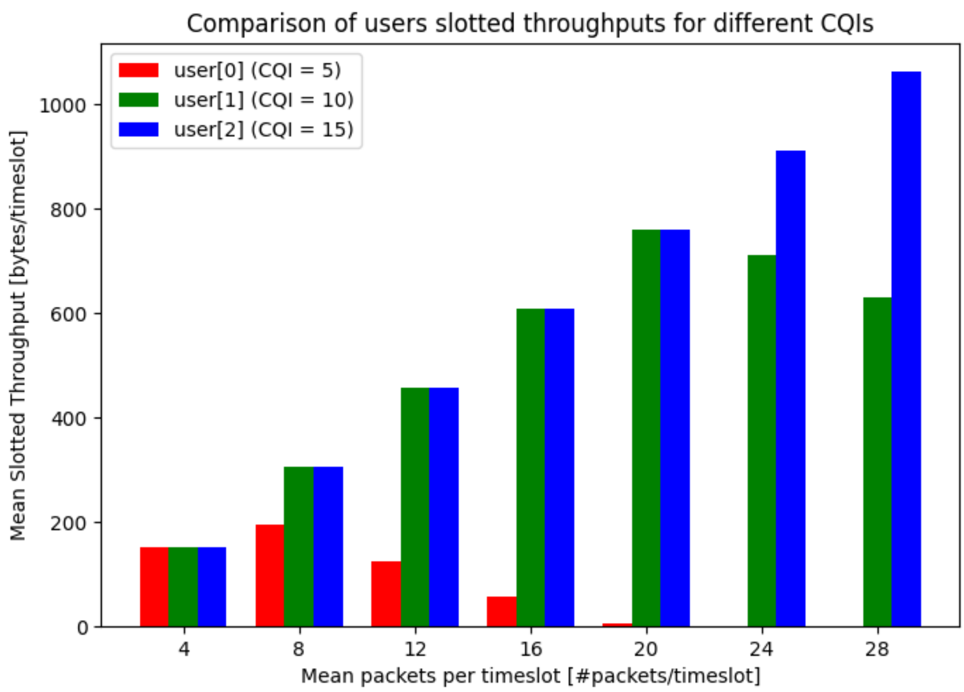


Figure A.11: Comparison of users slotted throughputs for different CQI. Confidence intervals are not shown because they are so small that it is impossible to see them.

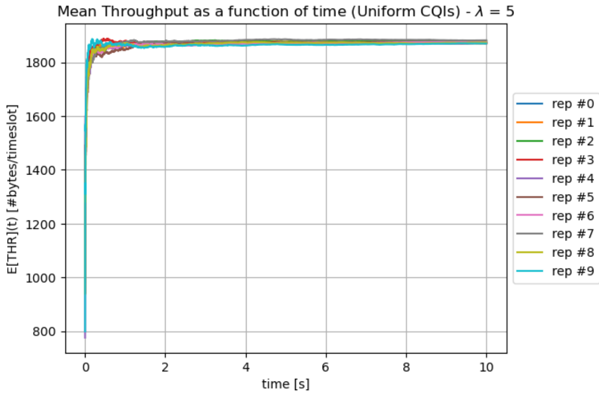
This gives us what we expected. We can see that for very low workloads ($\lambda = 4$) all 3 users have very similar throughputs because the system can allocate the right number of RBs unregarding of CQI number (which represent RB size). For increasing λ we can see that user[0] stars to get lower throughputs than the others and also we can see that if we keep increasing the workload the throughput actually tends to 0. Same reasoning can be done for user[1], for $\lambda \leq 20$ the latter will get a fair treatment compared to user[2]. But we see also that for $\lambda \geq 24$ its throughput starts to decrease and in the limit case of very high workloads the total throughput of the Antenna will be destined only to user[2].

[3] And by this also increasing values in packets response time

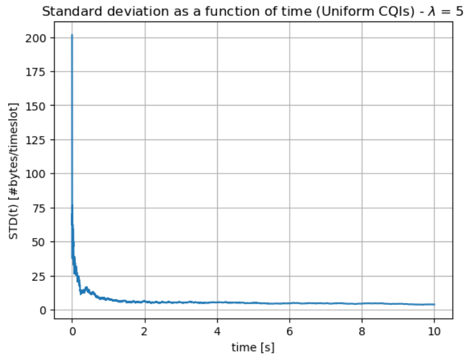
Appendix B

Plots regarding warm-up time and simulation period estimation

Warm-up time and **simulation period** estimation				
Parameters				
	Description	**Formula**	**Value**	**Measure unit**
T	Timeslot		1	ms
S(i)	STD at timeslot i	$\sqrt{\frac{\sum_{j=0}^{n}(v_j(i)-\bar{v})^2}{k-1}}$		#bytes/T
th_W	Warm-up time threshold		0.25	#bytes/T
th_S	Sim. duration threshold		0.01	#bytes/T
i		$i = max(k) : S(k) - S(k-1) \geq th_W$		
j		$j = max(k) : S(k) - S(k-1) \geq th_S$		
Results - Uniform CQIs, λ = 5 #packets per timeslot				
$\bar{T}_W$	Warm-up time	$i * 0.001$	1.824	s
$\bar{T}_S$	Simulation duration	$j * 0.001$	9.995	s
Results - Uniform CQIs, λ = 2.5 #packets per timeslot				
$\bar{T}_W$	Warm-up time	$i * 0.001$	0.261	s
$\bar{T}_S$	Simulation duration	$j * 0.001$	9.994	s
Results - Uniform CQIs, λ = 0.5 #packets per timeslot				
$\bar{T}_W$	Warm-up time	$i * 0.001$	0.054	s
$\bar{T}_S$	Simulation duration	$j * 0.001$	9.926	s
Results - Uniform CQIs, λ = 0.25 #packets per timeslot				
$\bar{T}_W$	Warm-up time	$i * 0.001$	0.055	s
$\bar{T}_S$	Simulation duration	$j * 0.001$	7.278	s
Results - Binomial CQIs, λ = 2.5 #packets per timeslot				
$\bar{T}_W$	Warm-up time	$i * 0.001$	0.242	s
$\bar{T}_S$	Simulation duration	$j * 0.001$	9.973	s
Results - Binomial CQIs, λ = 0.5 #packets per timeslot				
$\bar{T}_W$	Warm-up time	$i * 0.001$	0.07	s
$\bar{T}_S$	Simulation duration	$j * 0.001$	9.991	s
Results - Binomial CQIs, λ = 0.25 #packets per timeslot				
$\bar{T}_W$	Warm-up time	$i * 0.001$	0.039	s
$\bar{T}_S$	Simulation duration	$j * 0.001$	6.465	s

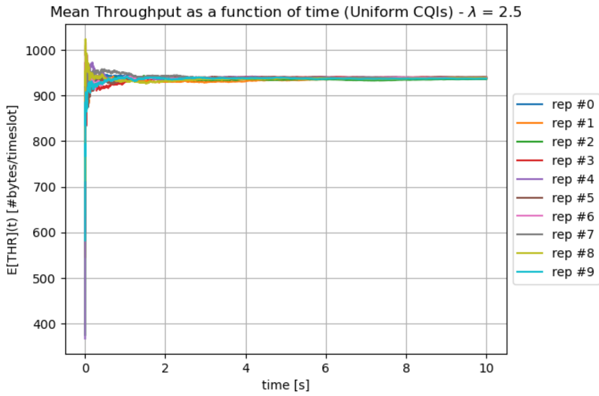


(a) Mean cell throughput for 10 repetitions.

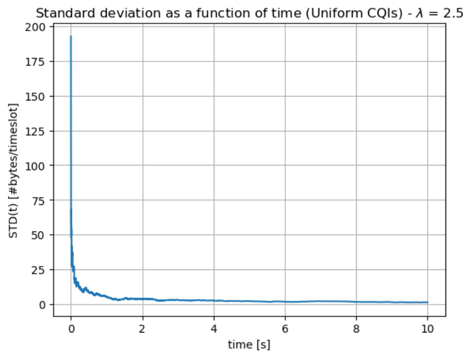


(b) Mean cell throughput's STD among 10 repetitions.

Figure B.1: Scenario: $\lambda = 5$ #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

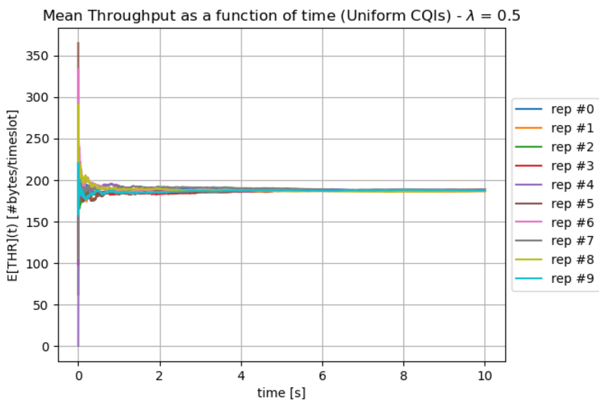


(a) Mean cell throughput for 10 repetitions.

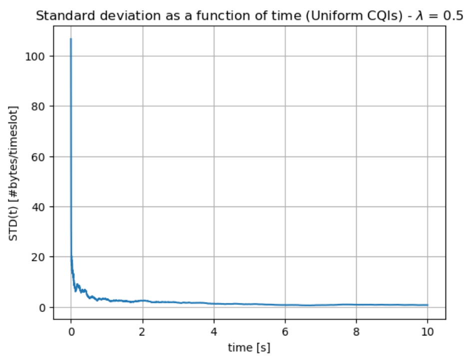


(b) Mean cell throughput's STD among 10 repetitions.

Figure B.2: Scenario: $\lambda = 2.5$ #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

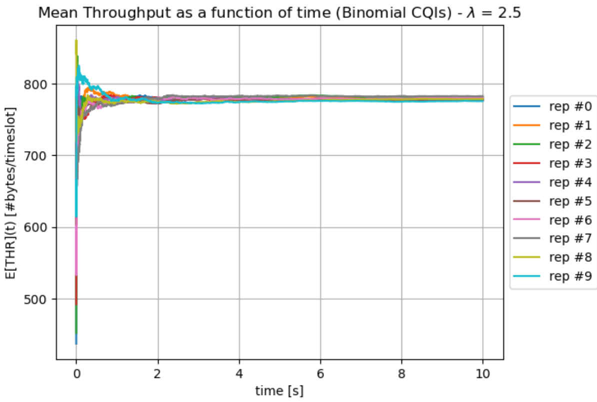


(a) Mean cell throughput for 10 repetitions.

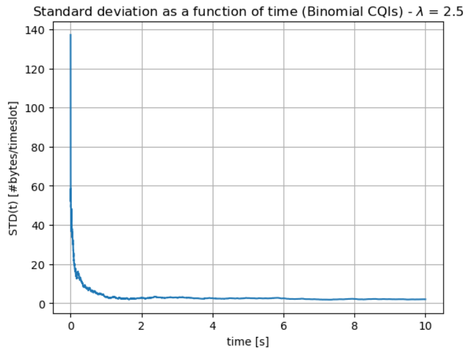


(b) Mean cell throughput's STD among 10 repetitions.

Figure B.3: Scenario: $\lambda = 0.5$ #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

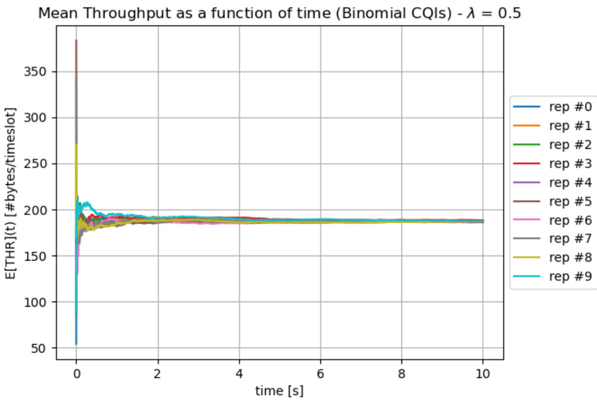


(a) Mean cell throughput for 10 repetitions.

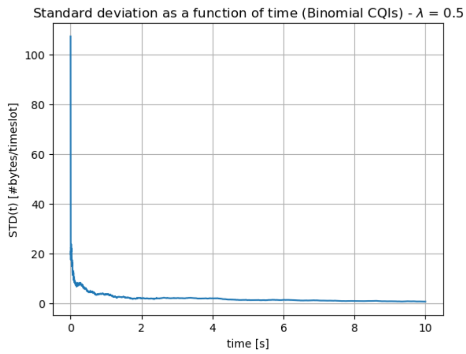


(b) Mean cell throughput's STD among 10 repetitions.

Figure B.4: Scenario: $\lambda = 2.5$ #packets per timeslot, population = 10, Binomial CQIs, Uniform Service Demands.

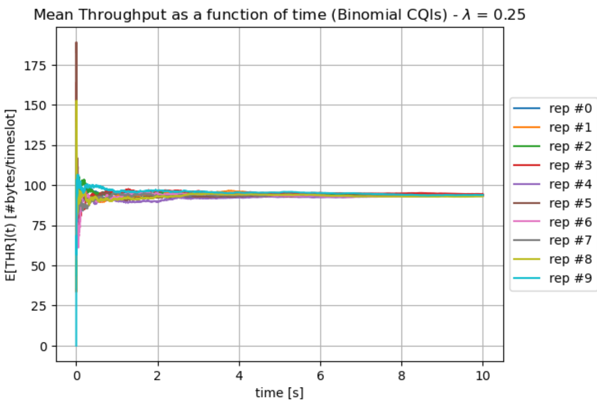


(a) Mean cell throughput for 10 repetitions.

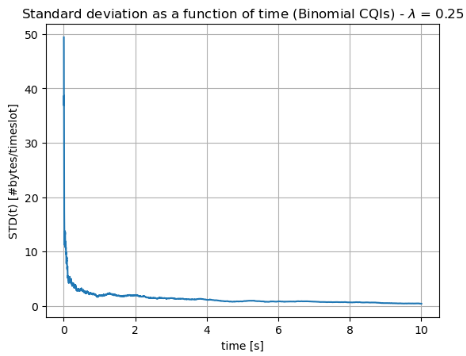


(b) Mean cell throughput's STD among 10 repetitions.

Figure B.5: Scenario: $\lambda = 0.5$ #packets per timeslot, population = 10, Binomial CQIs, Uniform Service Demands.



(a) Mean cell throughput for 10 repetitions.



(b) Mean cell throughput's STD among 10 repetitions.

Figure B.6: Scenario: $\lambda = 0.25$ #packets per timeslot, population = 10, Binomial CQIs, Uniform Service Demands.

Appendix C

Testing the independence assumption of response time samples

Before starting with the analysis of the results obtained, it is mandatory to explain how data were collected and aggregated. For each repetition we collected the mean response time experienced by users within the cellular network. Given the nature of the response time, even if for each repetition we collected a huge number of samples, we can't exploit the CLT and say that the mean response time is a Normal RV because we can't ensure independence among samples. For instance, if a packet at its arrival sees a huge number of queued packets it will experience a considerable response time. The next packet that arrives at the Antenna has a non negligible probability to see a huge number of queued packets as well as the previous one. Given that, it's very likely that the response times of the two packets are not truly independent. To prove the previous statement we decided to plot a correlogram of the mean user response time for each workload under study. We noticed that our intuition is true for $\lambda \geq 1$. After the results of the independence analysis we decided to reject the independence assumption for each studied workload.

To obtain independent RVs we decided to run simulations composed of 30 repetitions. For each repetition we collected the mean response time experienced by each `cellular`. Each mean response time is a RV itself and, given the fact that the repetitions are set up to be independent from each others, we can exploit CLT and say the following:

$$\bar{R}(i,\lambda) = \frac{\sum_{j=0}^{29} E[R](i,j,\lambda)}{30} \sim Normal$$

where $E[R](i,j,\lambda)$ is the mean response time experienced by `cellular`[i] at the j-th repetition of the simulation whose arrival rate was λ.

The confidence intervals were computed as follows:

$$CI_{95\%} = [\bar{R}(i,\lambda) - z_{0.025}\frac{S}{\sqrt{30}},\ \bar{R}(i,\lambda) + z_{0.025}\frac{S}{\sqrt{30}}]$$

where S is the sample standard deviation computed from $E[R](i,j,\lambda),\ j = 0,...,29$.

Also, we computed the mean cell response time as a function of λ. In particular, the formula that has been used is the following:

$$\bar{R}(\lambda) = \frac{\sum_{i=0}^{9} \bar{R}(i,\lambda)}{10}$$

In that particular case, the samples are normally distributed so we computed confidence intervals using the Student-T distribution.

$$CI_{95\%} = [\bar{R}(\lambda) - t_{0.025,9}\frac{S}{\sqrt{10}},\ \bar{R}(\lambda) + t_{0.025,9}\frac{S}{\sqrt{10}}]$$

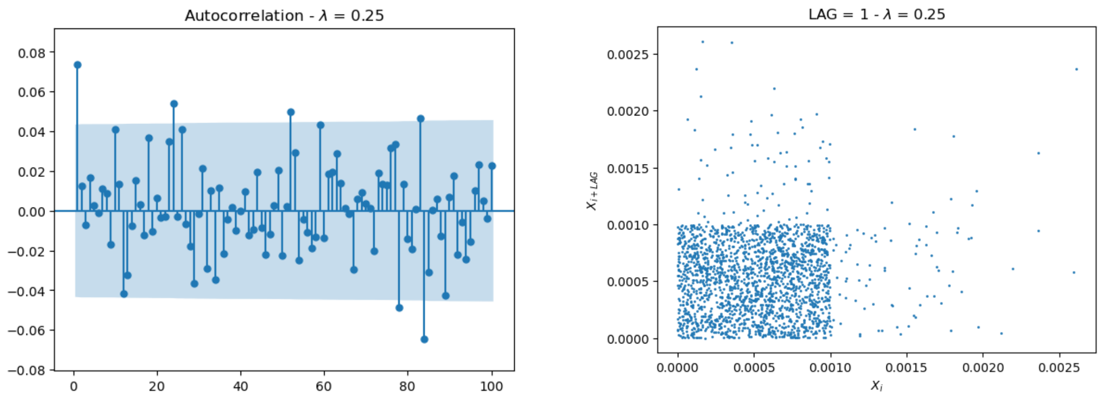


Figure C.1: Scenario: $\lambda = 0.25$ #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

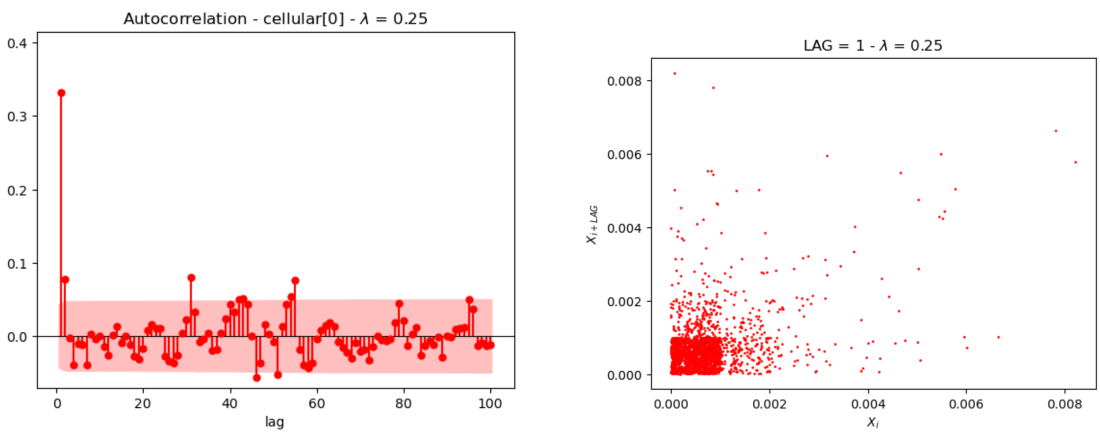


Figure C.2: Scenario: $\lambda = 0.25$ #packets per timeslot, population = 10, Binomial CQIs (cellular[0] $\sim Bin(14,\, 0.1)$), Uniform Service Demands.

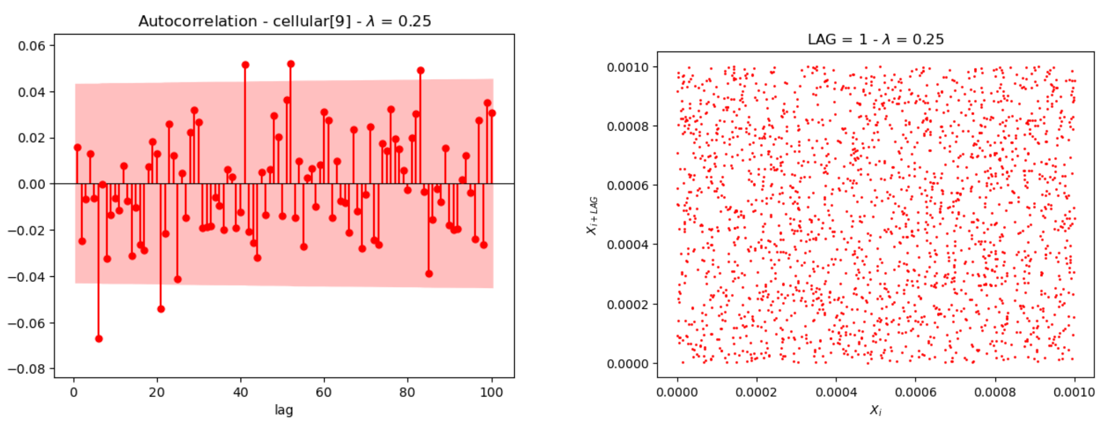


Figure C.3: Scenario: $\lambda = 0.25$ #packets per timeslot, population = 10, Binomial CQIs (cellular[9] $\sim Bin(14,\, 0.95)$), Uniform Service Demands.

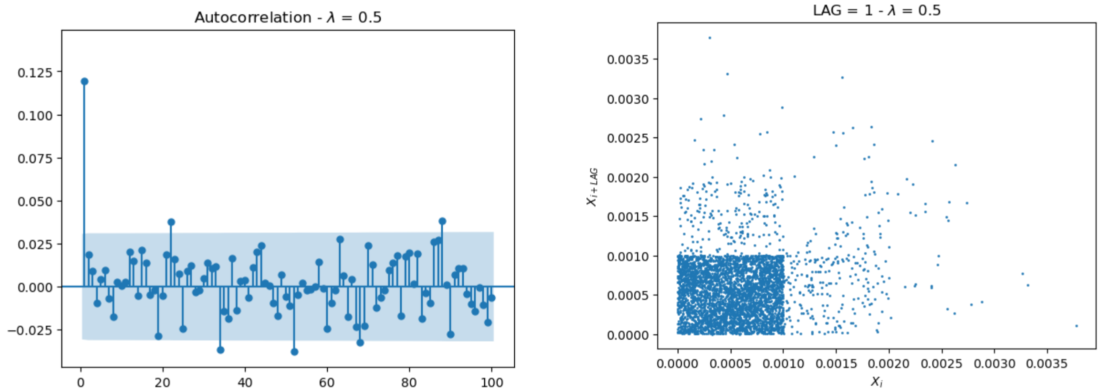


Figure C.4: Scenario: $\lambda = 0.5$ #packets per timeslot, population $= 10$, Uniform CQIs, Uniform Service Demands.

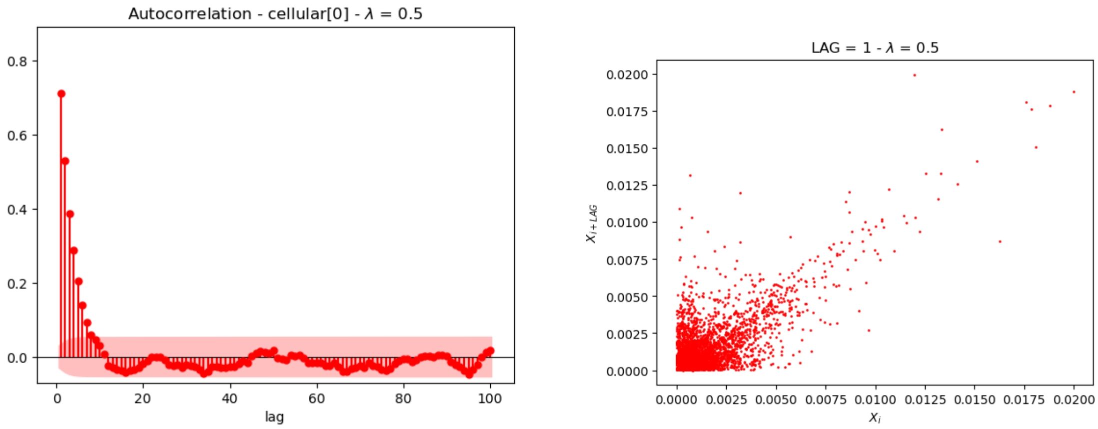


Figure C.5: Scenario: $\lambda = 0.5$ #packets per timeslot, population $= 10$, Binomial CQIs (cellular[0] $\sim Bin(14,\ 0.1)$), Uniform Service Demands.

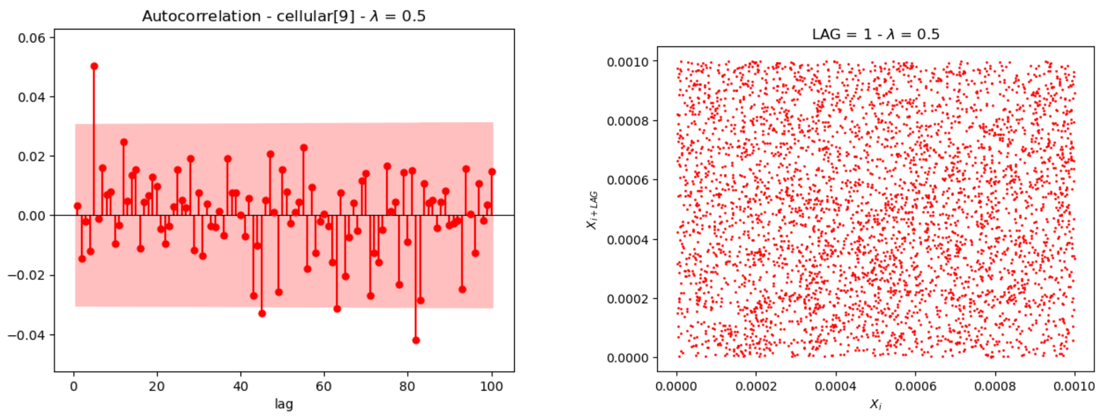


Figure C.6: Scenario: $\lambda = 0.5$ #packets per timeslot, population $= 10$, Binomial CQIs (cellular[9] $\sim Bin(14,\ 0.95)$), Uniform Service Demands.

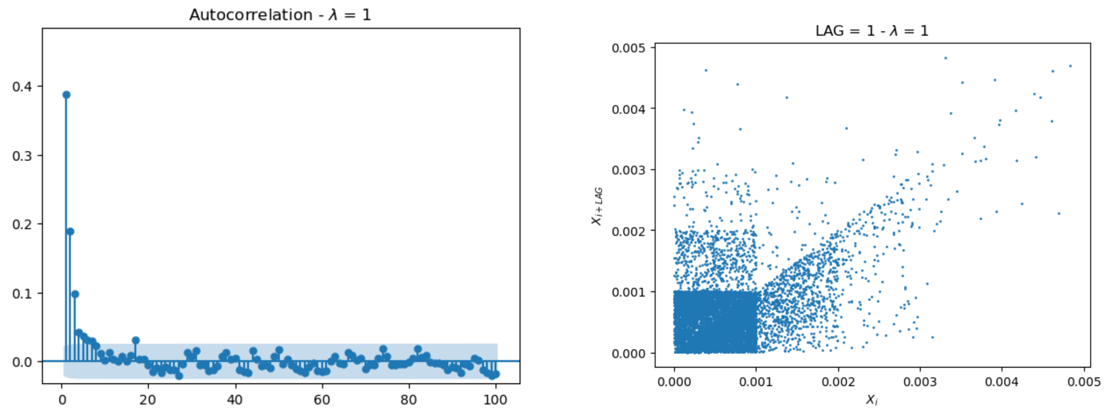


Figure C.7: Scenario: $\lambda = 1$ #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

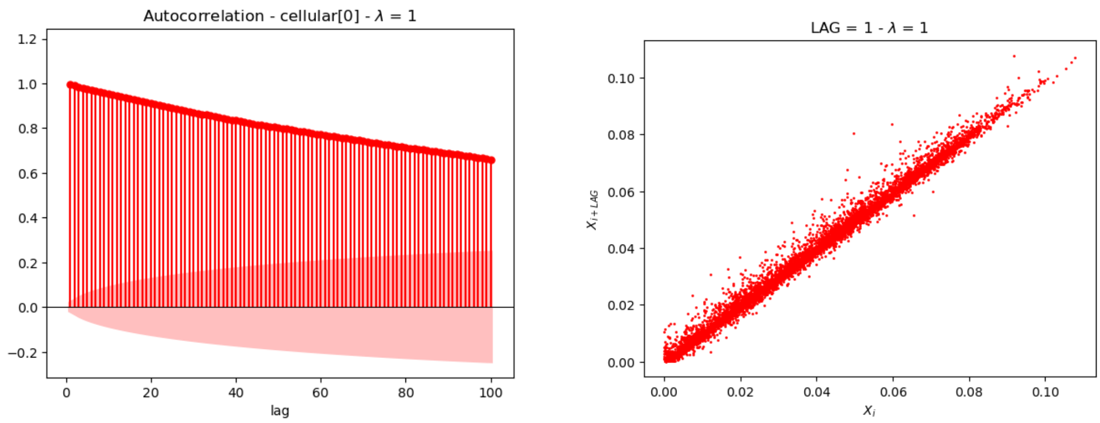


Figure C.8: Scenario: $\lambda = 1$ #packets per timeslot, population = 10, Binomial CQIs (cellular[0] $\sim Bin(14,\ 0.1)$), Uniform Service Demands.

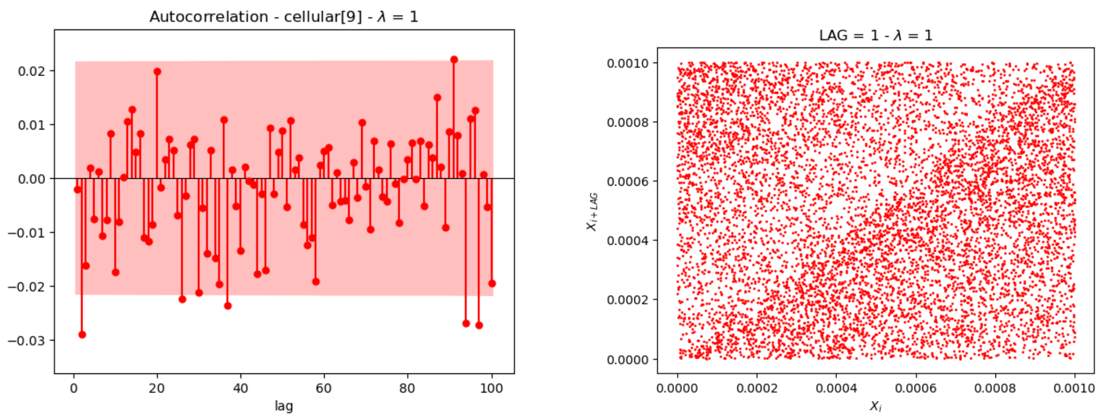


Figure C.9: Scenario: $\lambda = 1$ #packets per timeslot, population = 10, Binomial CQIs (cellular[9] $\sim Bin(14,\ 0.95)$), Uniform Service Demands.

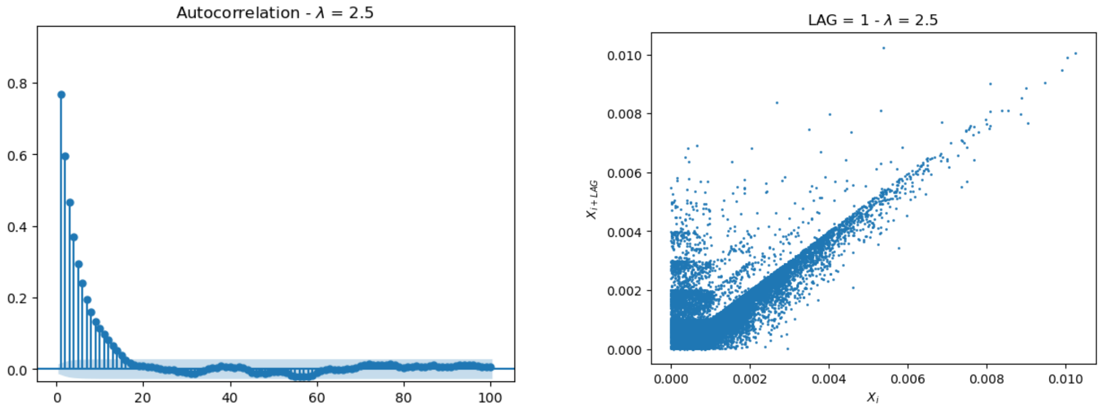


Figure C.10: Scenario: λ = 2.5 #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

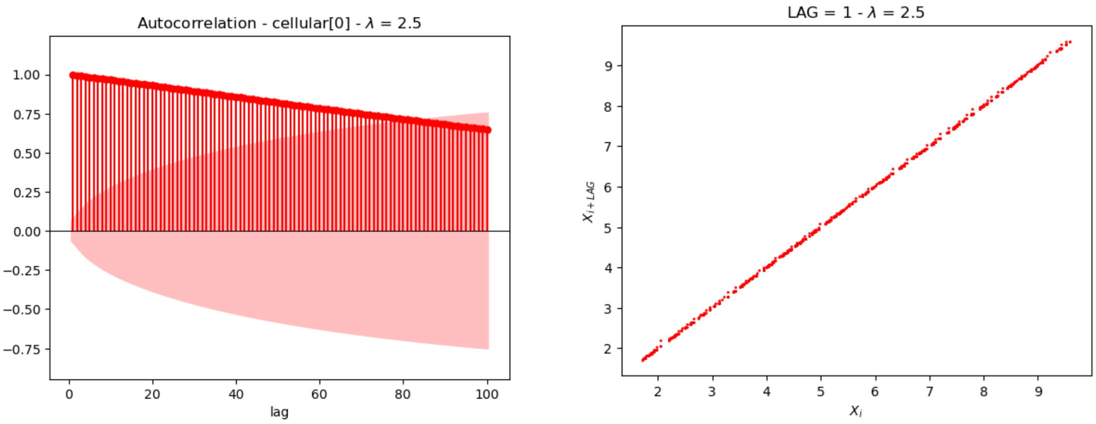


Figure C.11: Scenario: λ = 2.5 #packets per timeslot, population = 10, Binomial CQIs (cellular[0] $\sim Bin(14,\ 0.1)$), Uniform Service Demands.

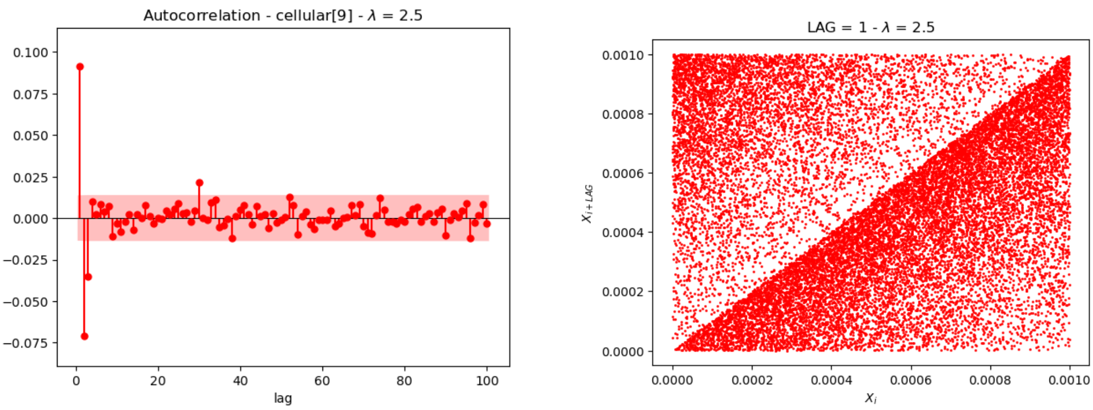


Figure C.12: Scenario: λ = 2.5 #packets per timeslot, population = 10, Binomial CQIs (cellular[9] $\sim Bin(14,\ 0.95)$), Uniform Service Demands.

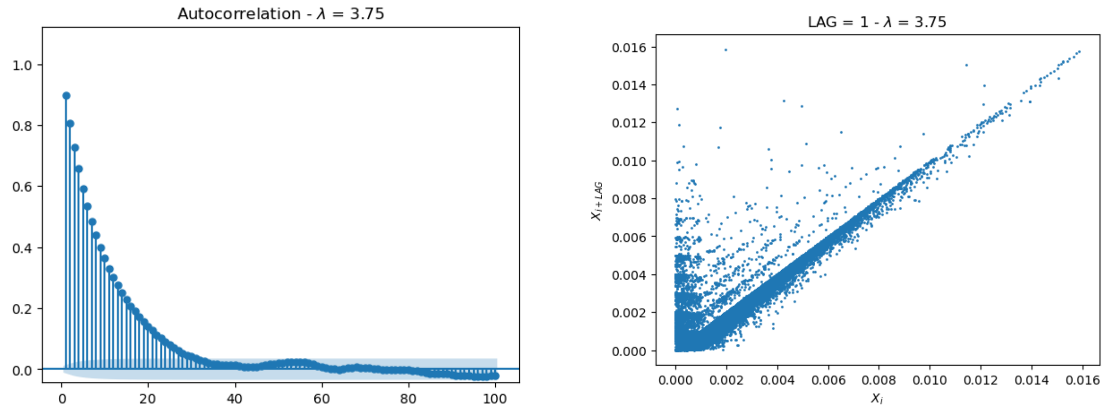


Figure C.13: Scenario: $\lambda = 3.75$ #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

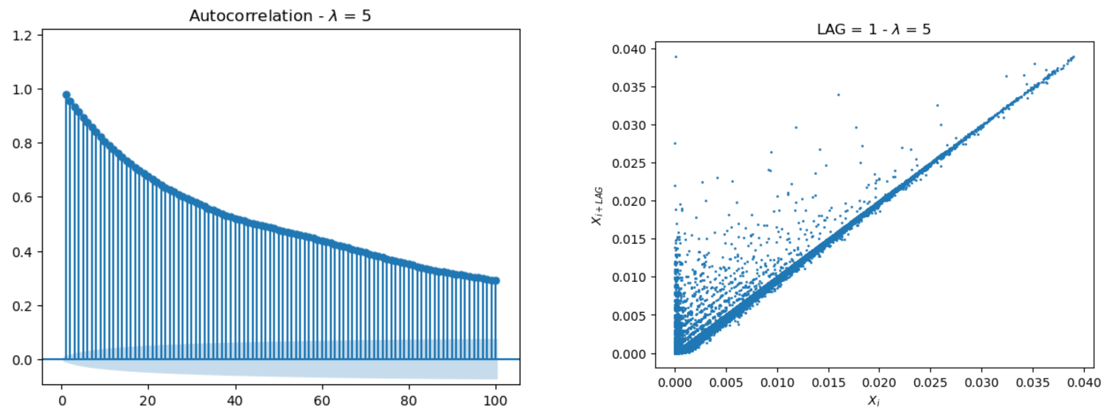


Figure C.14: Scenario: $\lambda = 5$ #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

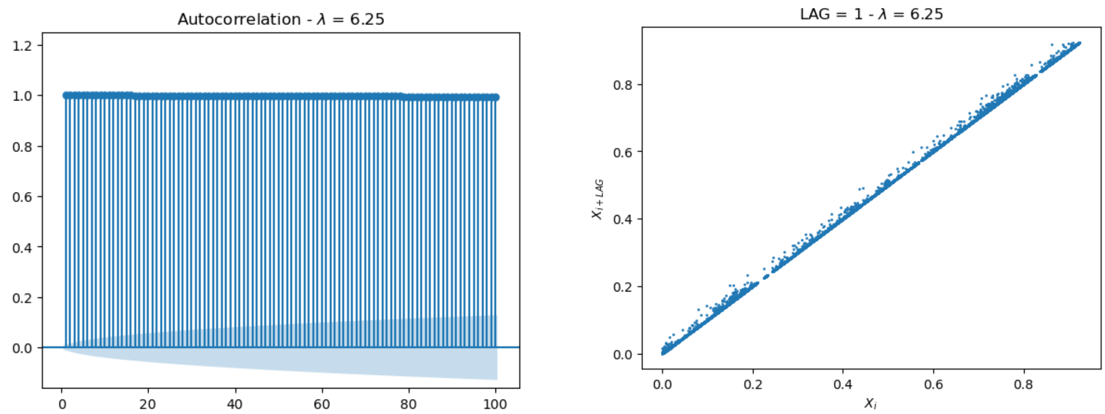


Figure C.15: Scenario: $\lambda = 6.25$ #packets per timeslot, population = 10, Uniform CQIs, Uniform Service Demands.

Appendix D

Failed analysis of mean user response time in a Binomial scenario

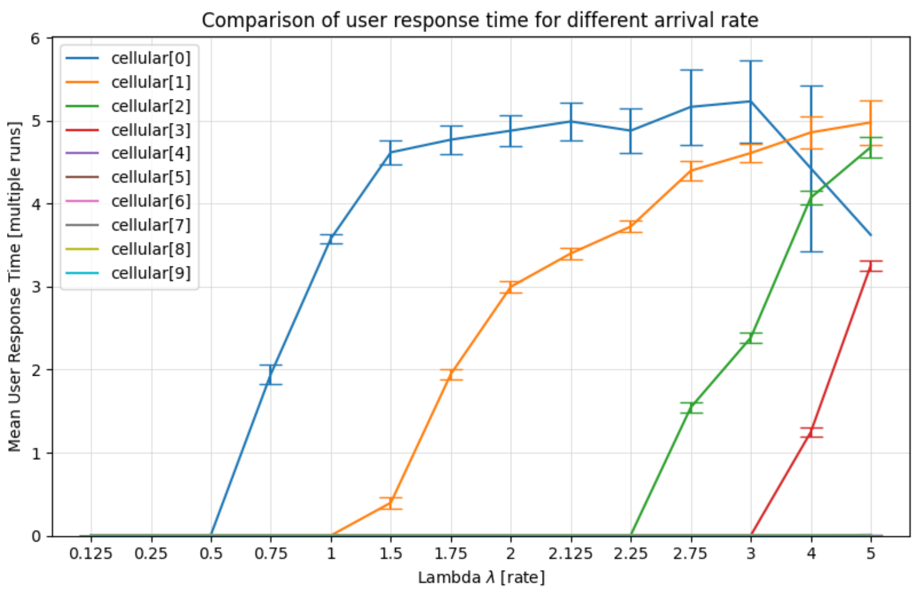


Figure D.1: Mean user response time as a function of packets arrival rate - $\lambda \in [0.125, 5]$. This plot shouldn't be considered because the sampling process yields sets of samples with different width.