UNIVERSITÀ DI PISA

Computer Engineering

Performance Evaluation of Computer Systems and Networks

Slotted random-access wireless network

Group Project Report

TEAM MEMBERS:
Tommaso Burlon
Francesco Iemma
Olgerti Xhanej

Academic Year: 2020/2021

Contents

1 — Introduction

1.1 Problem Description

From the group project assignment:

*In a **slotted random-access network**, **N couples transmitter-receiver** share the same communication medium, which consists of **C separate channels**. Multiple attempts to use the same channel in the same slot by different transmissions will lead to collision, hence no receiver listening on that channel will be able to decode the message. Assume that each of the N transmitters generate packets according to an **exponential inter-arrival distribution**, and picks its channel at random on every new transmission. Before sending a packet, it keeps extracting a value from a **Bernoullian RV with success probability p** on every slot, until it achieves success. Then it transmits the packet and starts over. If a collision occurs, then the transmitter backs off for a random number of slots (see later), and then starts over the whole Bernoullian experiment. The number of back-off slots is extracted as $U(1, 2^{x+1})$, where x is the number of collisions experienced by the packet being transmitted.*

1.2 Objectives

The aim of the project report is the *Assessment of the Effectiveness of the Slotted Random-Access Network Protocol* described in the latter paragraph.

1.3 Performance Indexes

In order to define a metric of performance of the objective, the following Performance Indexes are defined:

- **Throughput**: let Tp be the Throughput to be measured, N_p the number of packets successfully sent to the corresponding receiver, N_t the number of time-slot considered in the count of N_p, T_{slot} the period (in seconds) of a time slot, the Throughput(per slot) can be measured as:

  $$Tp(slot) = \frac{N_p}{N_t} \quad [packets/slot]$$

  During all the documentation the latter will be used because more meaningful w.r.t. the scenario that will be defined. We can also convert this performance metric in a more standard form, dealing with packets per second:

  $$Tp = \frac{Tp(slot)}{T_{slot}} \quad [packets/s]$$

- **Response Time**: defined as the time that occurs from the first appearance of one packet at the Transmitter up to the reception of the packet at the Receiver.

2 — Modeling

2.1 Introduction

The system is modeled as **N Transmitter-Receiver couples** which communicate through **C Channels**. A *collision* can occur on a **Channel** if more than one **Transmitter** want to transmit a packet in that **Channel**. The **Transmitter** stores in a queue the packets that it wants to transmit and, then, it sends them; the **Channel** "knows" if a collision occurs and handles it; the **Receiver** only receive packets.

2.2 General Assumptions

The following general assumptions have been made:

- **Slotted**: packets are attempted to be **Transmitter** by the Transmitter only at the **beginning of the time-slot**
- **Constant Packet Size and Transmission Rate**: each packet has a **constant packet size** and each **Transmitter** has a constant and equal transmission rate for which to transmit a packet (without collision) from the **Receiver** to the transmitter will last **one time-slot**.
- **No Propagation Error in the channel**: the only cause of a failed transmission has to be considered as the *packet* collision. Other causes, (i.e. path-loss, shadowing, small-scale fading), are neglected.
- **FIFO Queues of unlimited Capacity at the Transmitter**
- **Transmitters and Receivers always synchronized with the time-slot period**: the **Receiver** knows in which **Channel** the **Transmitter** will try to send his packet in each time-slot and the **Receiver** will be ready to listen in the correct **Channel**.
- **After an eventual collision the transmitter will change his channel choice**: this is the default configuration, on Chapter 5 this configuration will be compared with the Non-Change one.

2.3 Preliminar Validation

Before the implementation a preliminar validation phase is necessary to ensure that the model is correct. Let analyze if the assumptions made in the previous section are reasonable:

- The **slotted assumption** (transmission only at the beginning of the timeslot) is reasonable due to the fact that exist some network protocols which work under this assumption, see for instance Slotted Aloha.
- We consider the **packet size constant** because if the packet length is so large that more than one slots are needed, we can consider, from the viewpoint of the model, that this unique packet send in two different slots is like two packets of fixed-length send each one in a slot. Taken this into account is reasonable to state that the packet size is constant and that one packet is transmitted within a slot (if no collisions occur).
- The critical issue of every slotted network is the one related to the **collisions**: they have an huge impact on the general performance of the network, so it is reasonable to neglect the other propagation errors that are not network-specific or that depend from the environment (as path-loss).

- It's reasonable that the transmitter and the receiver are **synchronized with the time-slot period**, in fact other slotted-based network as slotted ALOHA requires a synchronization of this type.

2.4 Factors

The following factors have been defined which may affect the performance of the system:

- **N**: **Transmitter-Receiver** Couples.
- **C**: numbers of **Channels**.
- **p**: **probability of success** for sending a packet in the current time-slot for a Transmitter.
- $\frac{1}{\lambda}$: exponential distribution **mean inter-arrival time of packets** at the Transmitter.
- T_{slot}: **time-slot duration**.

3 — Implementation

3.1 Modules

The following modules have been defined:

- **Transmitter**: duty of sending a packet to a specific channel.
- **Channel**: duty of checking every channel in each timeslot for any collision.
- **Receiver**: duty of receiving a packet from the channel.

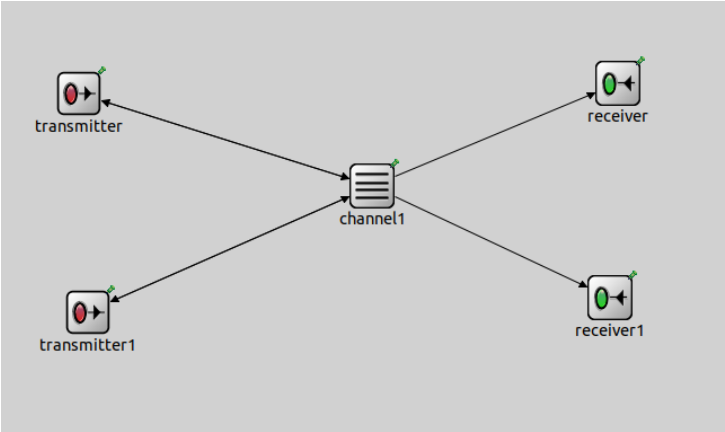


Figure 1: Network example

3.2 Messages

A new format of message has been defined, in order to store all packet information, with the following fields:

- *simtime_t creationTime*: time in which the packet arrives for the first time at the **Transmitter**
- *int idChannel*: **Channel** chosen for the current transmission, may change in case of collision
- *int idTransmitter*: Id of the module **Transmitter** that has sent the message
- *int idGate*: Id of the gate in which the **Transmitter** is linked to the **Channel**

3.3 Modules Behaviour

3.3.1 *Transmitter* Module Behaviour in the implementation

1. Message arrival at the *Transmitter*:
   - **IF** an *ACK* has been received, the packet at the top of the queue can be removed. **GOTO (2)**
   - **ELSE IF** a *NACK* has been received, then the *Transmitter* starts his backoff time and **waits for another message**.
   - **ELSE IF** a *Synchronization message* has been received **AND** the *Transmitter* is not in backoff-time **GOTO (2)**

- **ELSE IF** a *packet* arrives at the *Transmitter*, the *Transmitter* stores the packet in the queue, make a reschedule of the arrival of the next packet and **waits for another message**

2. The *Transmitter* tries to send the packet

   - **IF** success on the Bernoullian Experiment, then the packet will be forwarded to the *Channel*.
   - **ELSE waits for another message**

3.3.2 *Channel* Module Behaviour in the implementation

1. The *Channel* wakes up at the beginning of each time slot and checks his channels status.

   - **IF** two or more packets have arrived in the same channel, the *Channel* will send a NACK to the *Transmitters* that have forwarded the packets in that specific channel.
   - **ELSE IF** one and only one packet has arrived in a channel, the *Channel* will send an ACK to the relative *Transmitter* and will forward the packet to the *Receiver*.
   - **ELSE** the *Transmitters* that did not send a packet will receive from the Channel a *Synchronization Message*

2. The *Channel* will gather of the packets for the current time slot, to be processed in the next one. So this means that if the channel receives packets in time-slot *j*, then the information about collisions are provided to transmitters in time-slot *j+1*. Hence, from the point of view of the transmitter, the information received at *j+1* are referred to events took place in *j*.

3.3.3 *Receiver* Module Behaviour in the implementation:

1. The *Receiver* wakes up when a packet arrives. Due to the fact that the receiver essentially has the only duty of receiving packets. It computes statistics on the number of packets received mainly for debugging purpose.

4 — Verification

In this section we present tests performed in order to verify that our implementation reflects correctly our model.

4.1 Deterministic and Simple Cases

In this scenarios we test the behaviour of the system in **easy and completely deterministic cases** in order to verify the model. In all of these tests the behaviour is the one expected and it has been verified using the graphical environment (Qtenv) and debugging prints.

4.2 Degeneracy Tests

In the **degeneracy tests** we verify the behaviour of our simulator with parameters set to 0 values. In all tests the simulator works properly. In particular the following observations can be inferred:

- If the **number of Channel C** is 0 then the simulation stops immediately because has no sense running a simulation with 0 channels.
- If the **Time-slot size** T_{slot} is 0 then the simulation doesn't stop and goes to infinite because on instant 0 time slots are continuously triggered.
- If the exponential **mean inter-arrival time** $\frac{1}{\lambda}$ is 0 then the simulation doesn't stop and continues to infinite because packets arrive at time 0 continuously.

4.3 Consistency Test

The **consistency test** verifies that the system react consistently with the output. In order to test this we have performed different tests, here an example with the following parameters:

Test 1

- **N** (couples tx-rx): **1**; **C** (channels): **500**; **p** (send probability): **1**; $\frac{1}{\lambda}$ (Mean inter-arrival time): **10s** (deterministic); Time slot size: **5s**;

Test 2: Two couple TX-RX with half packets arrival rate

- **N: 2**; **C: 500**; **p: 1**; $\frac{1}{\lambda}$ = **20s** (deterministic); T_{slot}**: 5s**

We **expect that the result of the two tests are more or less equal** because the behaviour of one source transmitting every 10 seconds must be similar to the behaviour of two sources transmitting every 20 seconds. We set the **number of channels at 500 in order to neglect the effect of collisions** (in any case it is possible that a collision occur, but in the following tests no collisions have been detected).
The graph in figure 2 and in figure 3 show the results of the tests previously explained. We can see that the behaviour is very similar in both cases and so that the systems works as we expect. In fact the Mean Throughput per slot tend in both cases to **0.50 packets per slot**.

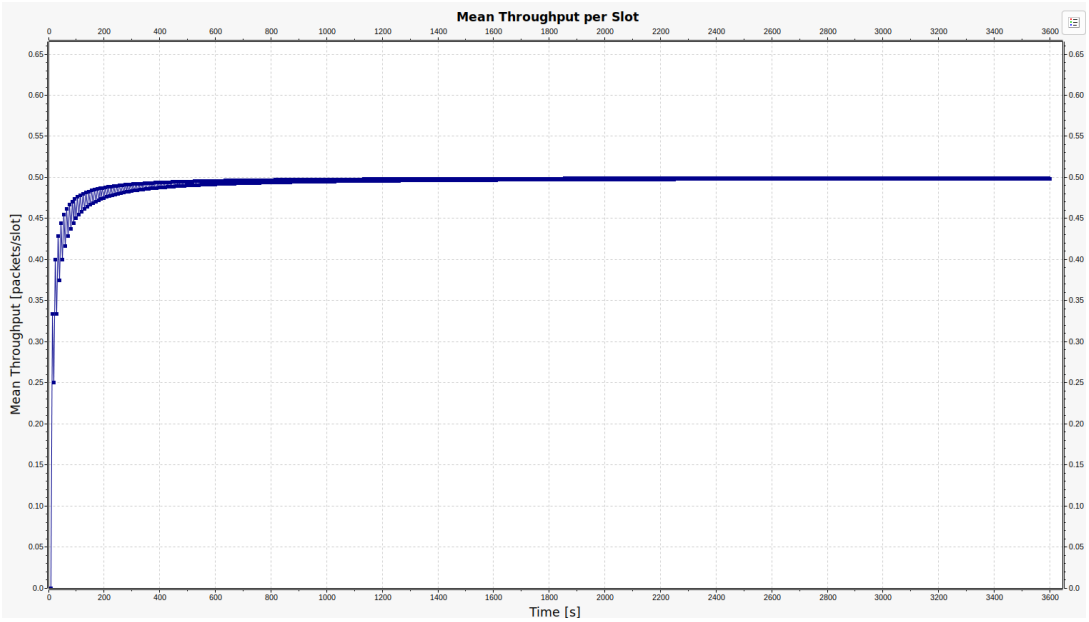


Figure 2: Consistency Test 1

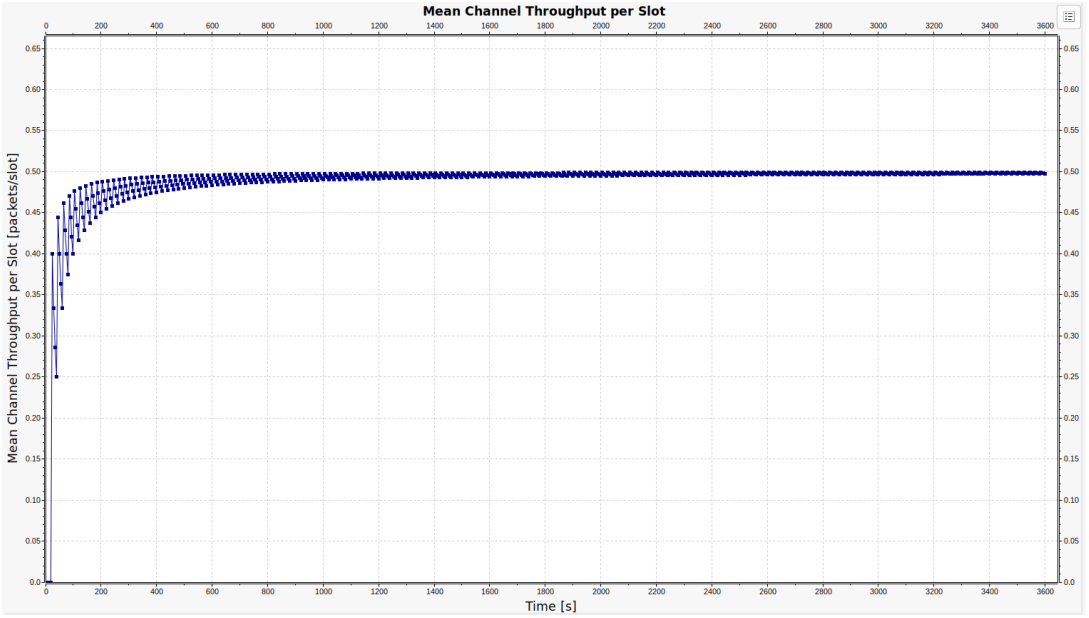


Figure 3: Consistency Test 2

4.4 Continuity Test

In this test the aim is to prove that the output changes slightly if the input changes a bit. In order to do prove this, different test were made. As an example two simulations were carried out with the parameters shown in table 1. With this parameters we **have changed slightly the input**, and so we **expect that the outputs don't show particular differences**. Obviously some differences will be present (in particular due to collisions and the increasing in the number of transmitters) but they shouldn't affect a lot the results.
The remaining parameters are the same of all of four tests:

- Send probability **p**: **1**

Test	N (Couples Tx-Rx)	C (Channels)
1	8	20
2	10	20

Table 1: Continuity test parameters

- Mean Inter-arrival time $\frac{1}{\lambda}$: **10s** (deterministic)
- T_{slot}: **5s**

The results of the simulation as shown in the figure 4 and 5 (we measure the mean throughput).

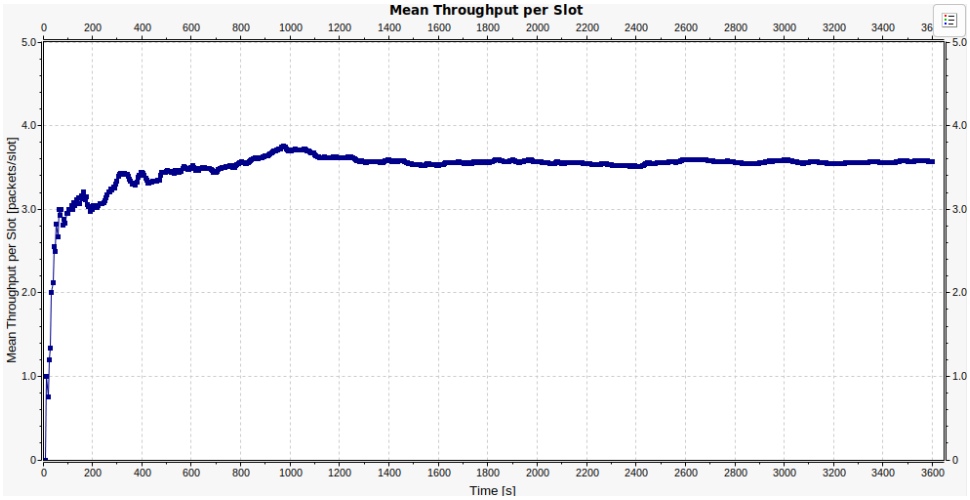


Figure 4: Continuity Test 1 - Mean Throughput per Slot - Collisions detected: 272

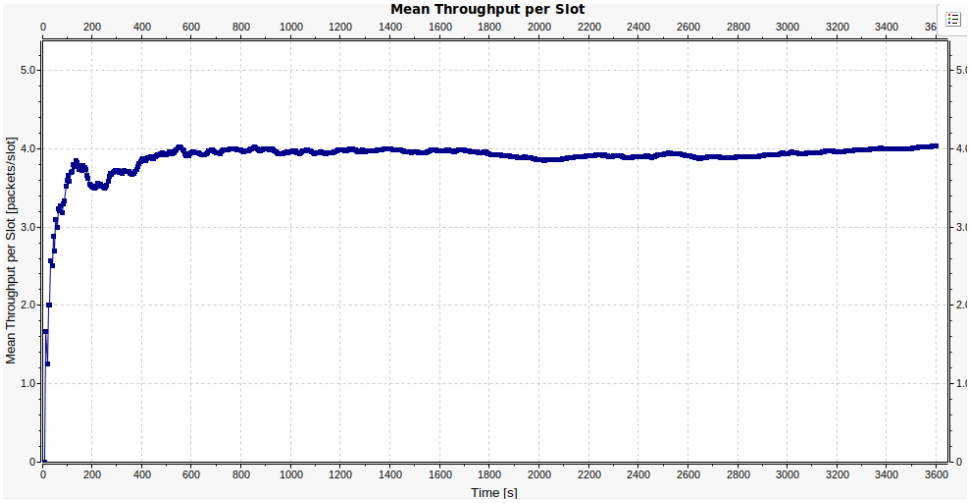


Figure 5: Continuity Test 2 - Mean Throughput per Slot - Collisions detected: 390

We can observe that the output changes slightly between the two cases. In particular it's possible to infer that if the number of transmitter increases then the throughput increases too (this statement will be proved with experiments), but this increasing is reduced by the collisions, in fact if the number of channels is fixed, the more the transmitter the more the collisions.

Monotonicity Assessment: In addition to this, some simulation were taken in order to **assess the monotonicity** of some KPI by changing some factors, here some examples:

- **Mean Throughput**: By **increasing N** (the numbers of couples tx-rx), with a high number of channels to avoid collisions, an **increase on the mean throughput is expected**. On the contrary, by **increasing** $\frac{1}{\lambda}$ (the mean inter-arrival time) an **opposite result is expected**. The following results were obtained:

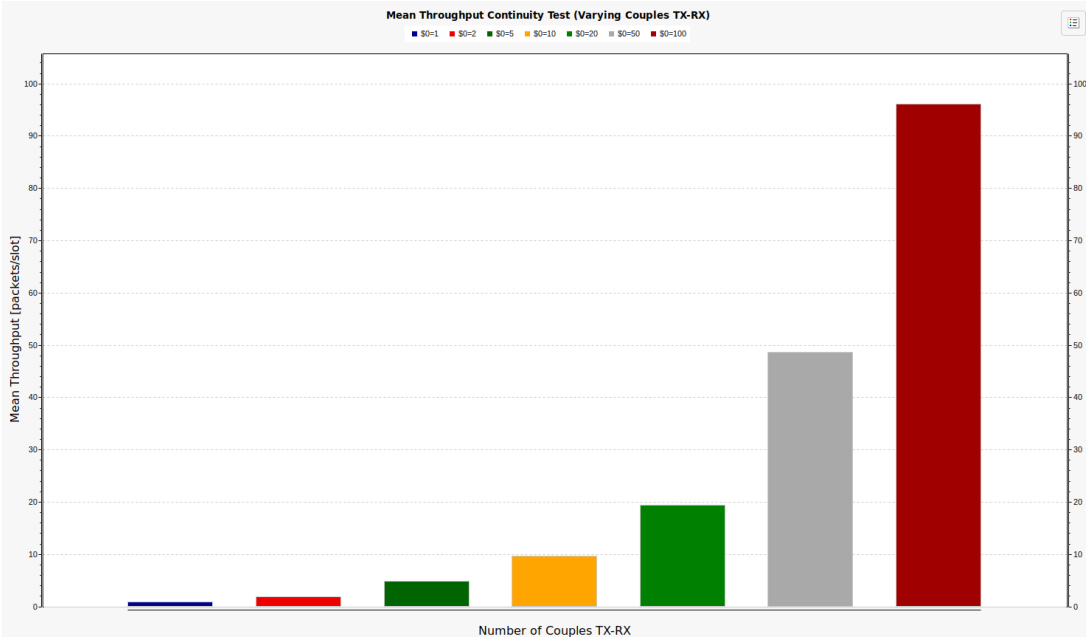


Figure 6: Continuity Test - Increasing Number of TX-RX (Main factors: **N** = 1, 2, 5, 10, 20, 100; **C** = 20000; $\frac{1}{\lambda} = 20$ ms; $T_{slot} = 5$ms; $p = 1$)

- **Mean Response Time**: By **Increasing N** (the number of couples tx-rx), with low numbers of channels, an **increase on the mean response time is expected**. By increasing the transmission probability p (with a low number of couples tx-rx) a decrease on the mean response time is expected due to more transmissions. The following results were obtained:

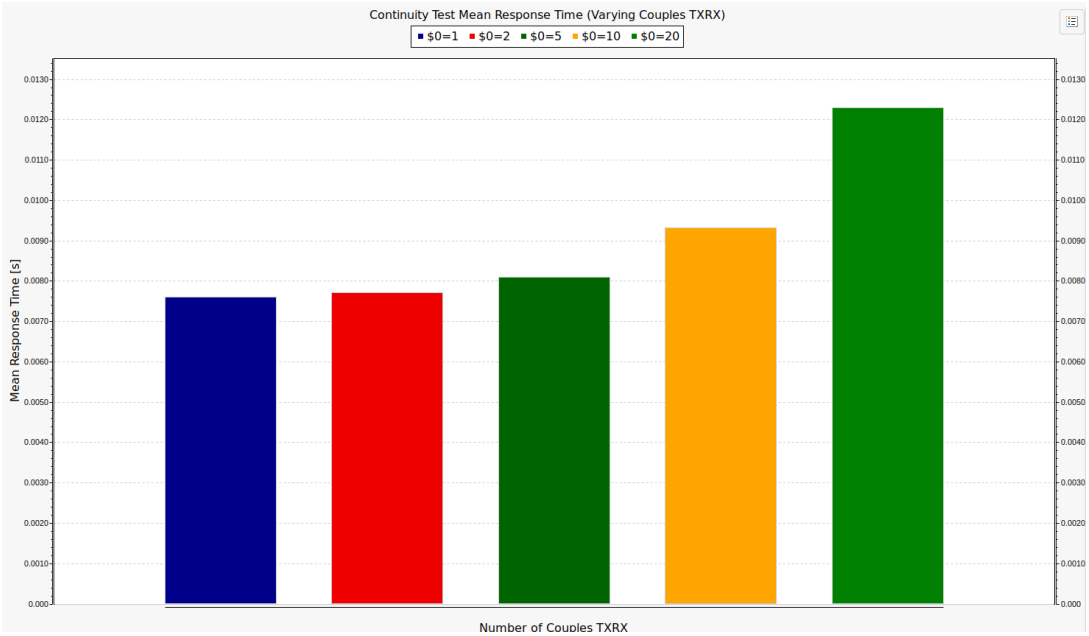
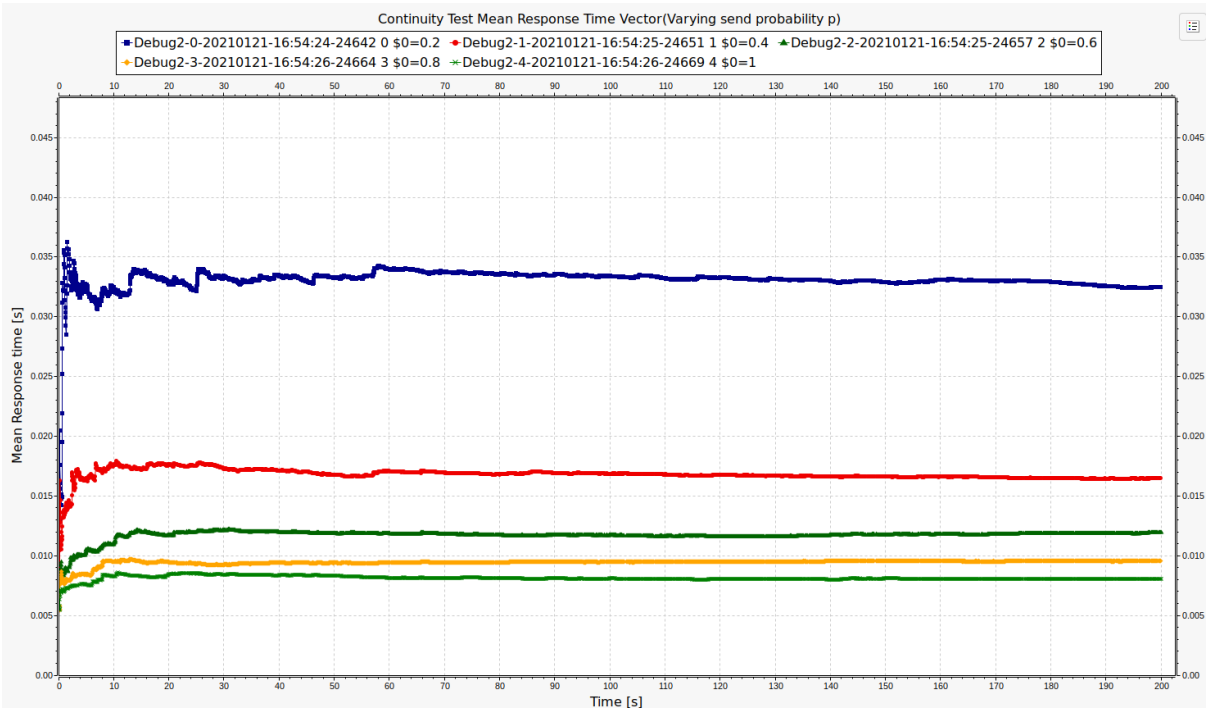


Figure 7: Continuity Test Mean Response Time- Increasing Number of TX-RX

For the Mean Response Time was also checked the steady state reach by just plotting that the relative vector stabilizes at some point (this was done in general to make conclusion with this KPI). Here an example for the varying of the Transmission Probability p:



4.5 Test Simulations with Binomial Model (1)

This test simulation has been performed with the following parameters:

Parameters

- Number of Couples Tx-Rx (**N**): 1
- Number of Channels (**C**): 1
- Send probability (**p**): ${0.05, 0.1, 0.15, 0.2, 0.4, 0.5, 0.6, 0.8}
- Mean inter-arrival time ($\frac{1}{\lambda}$): 1s (deterministic)
- Time-slot size (T_{slot}): 2s
- Simulation duration (T_{sim}): 3600s
- Repeat: 100
- Seed-set: ${repetition}

In this simplified context there are **no collisions** (only one couple) and the transmitter will have, for every slot, at least one packet to sent ($\frac{1}{\lambda} < T_{slot}$). We can model this particular case as a repeated Bernoullian Experiment, in which a success event correspond to a successful packet sent. In this simplified model we can define X as the number of success in n repeated trials (in independent condition), so $X \sim Bin(n, p)$. For this reason the PMF is the following:

$$p(i) = P\{X = i\} = \binom{n}{i} p^i (1-p)^{n-i} \tag{1}$$

Where n represents the number of repeated trials and i the number of successes in those trials. With this distribution the mean and the variance are:

$$E[X] = np \qquad Var(X) = np(1-p)$$

In our context we can state the following:

$$n = \left\lfloor \frac{T_{slot}}{T_{sim}} \right\rfloor = 1800 \tag{2}$$

We would expect in the case of $p = 0.5$:

$$E[X] = np = 900 \tag{3}$$

And the results after the run of 100 test simulation with different seeds, the following results are returned (with 95% CI):

$$\overline{X} \in [893.08, 901.94] \tag{4}$$

Which is in line with our expectations. The latter computations have been repeated for different values of p and the following plot can sum up the results:

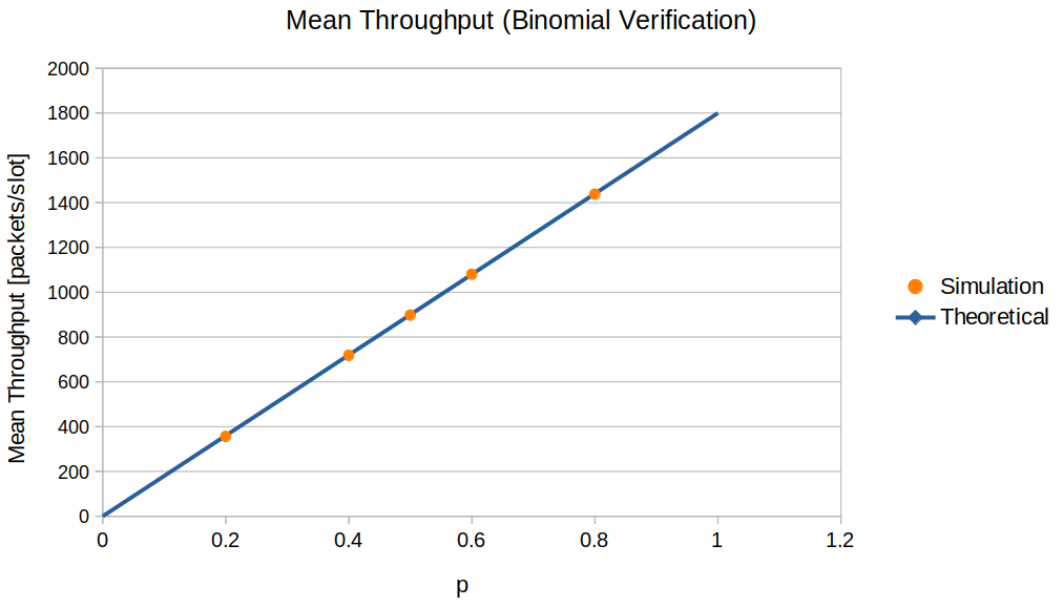


Figure 8: Test Binomial Model

The 95% CI are too small to be seen clearly so were not shown in the latter plot.

4.6 Test Simulations with Collisions (2)

This test simulation has been performed with the following parameters:

Parameters

- Number of couple tx-rx (**N**): ${2, 5, 10, 30}
- Number of channels (**C**): 1
- Send probability (**p**): ${0.2, 0.4, 0.6, 0.8}
- Mean inter-arrival time ($\frac{1}{\lambda}$): 1s (**deterministic**)
- Time-slot size (T_{slot}): 2s
- simulation-duration (T_{sim}): 3600s
- Repeat: 40
- Seed-set: ${repetition}
- **No Back-off** in case of collision

The aim of this verification is to assess if the mean throughput is comparable with some equations that will be found even in the case of presence of collisions. Also in this simulation we are in the hypothesis that each transmitters has a packet to send for every slot ($\frac{1}{\lambda} < T_{slot}$).

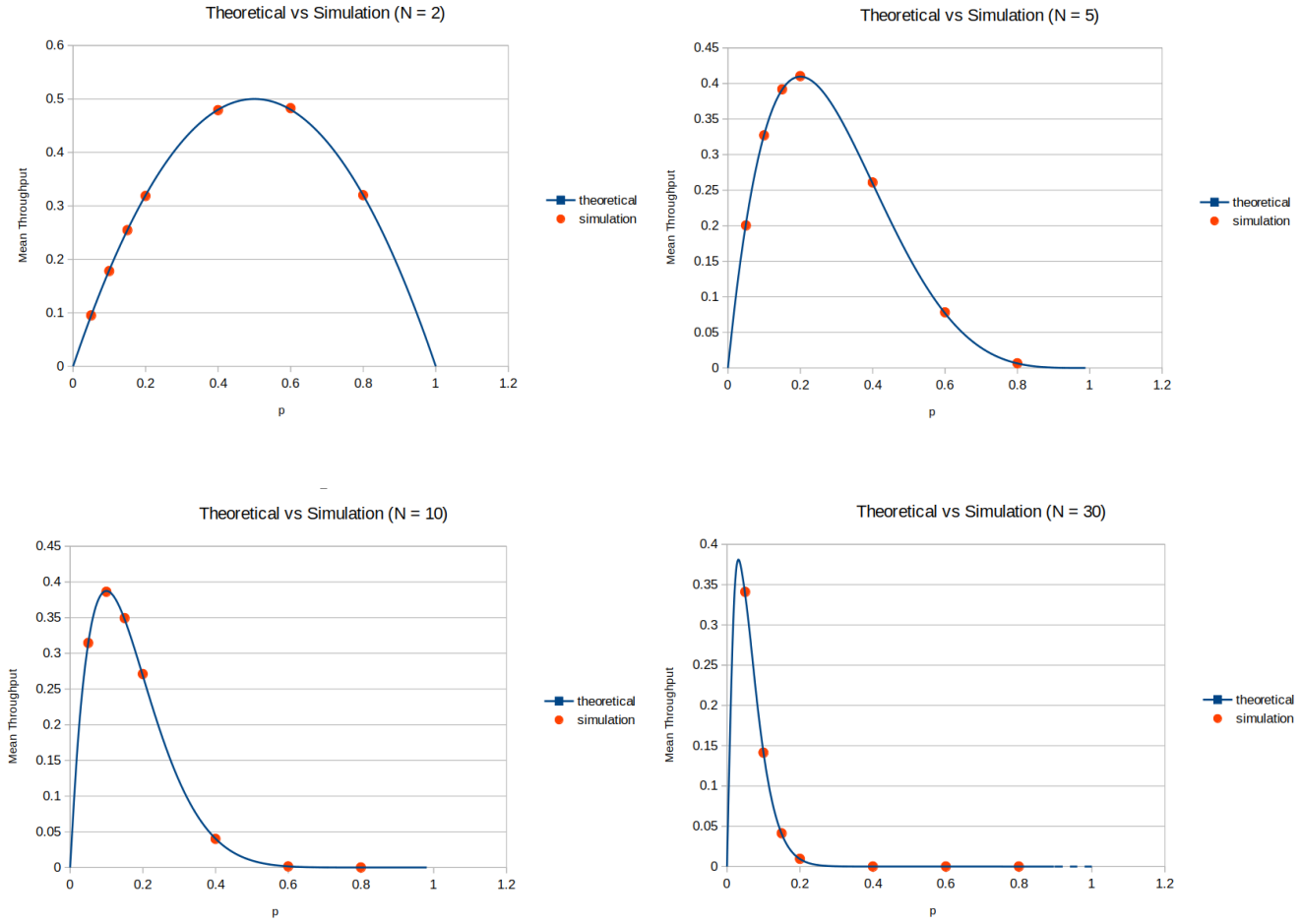
The **probability of a successful transmission in a particular time-slot**, in this case, is the following:

$$P\{"successful\ transmission"\} = P\{"only\ one\ tx\ transmit"\} = N \cdot p \cdot (1-p)^{N-1} \qquad (5)$$

This probability of successful transmission can be seen as the mean throughput of the system in the single channel case. In fact, let N_p the number of packets successfully sent to the corresponding receiver, N_t the number of time-slot considered in the count:

$$Tp\ (slot) = \frac{N_p}{N_t} = \frac{N_t \cdot P\{"successful\ transmission"\}}{N_t} = N \cdot p \cdot (1-p)^{N-1}$$

By comparing the above formula with the results of the simulation the following results are obtained (**95% CI too small to be seen**):



At this point we can state that a proper amount of verification of the implementation of the model has been carried out to make some simulation and gather some insight.

5 — Simulations Experiments

5.1 Study on Response Time Limits

Before choosing the extremes of the **mean inter-arrival time** factor, a study on the mean response time, by changing the latter, has been carried out by comparing limit values of other factors (all the ranges will be shown in the next paragraph).

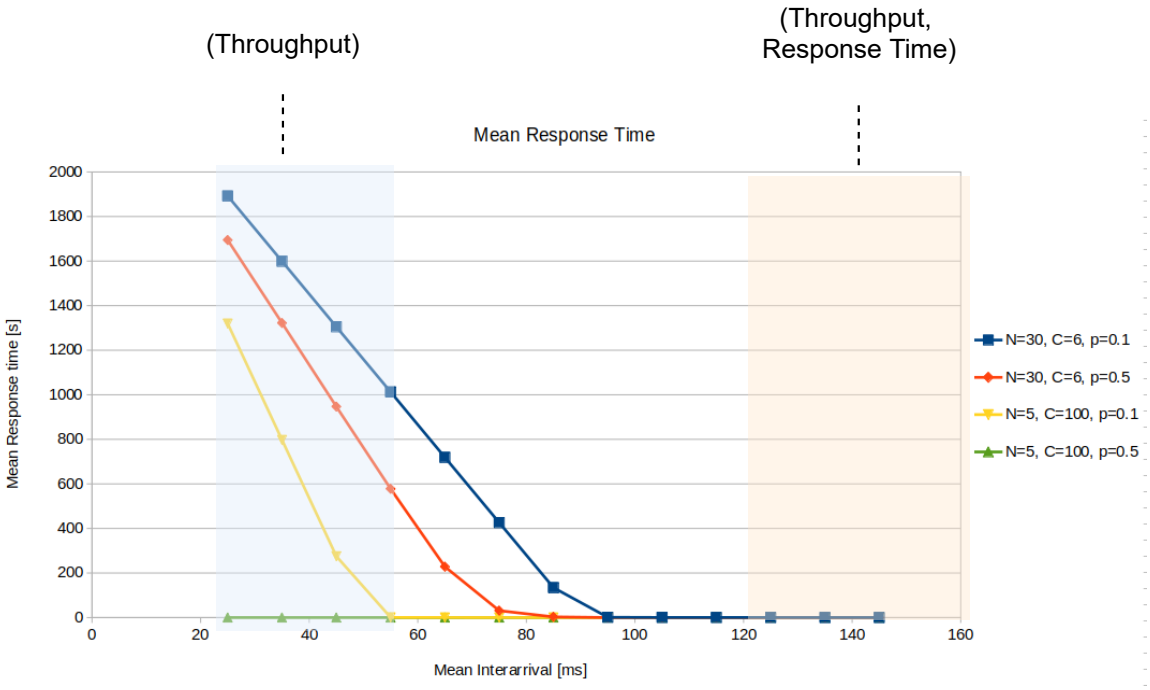


Figure 9: Buffer Explosion

Basically, two different zone of interest of the mean inter-arrival time were found:

- the first one between **[25ms, 55ms]**, for which the mean response time diverges for the majority of the combination of other factors limits. In this area **only information about the throughput** can be carried out, because the **mean response time**, as diverges, is **dependent on the simulation duration**.
- the second one between **[125ms, 500ms]**, for which both information about the **throughput** and the **response time** can be carried out.

For the second interval, the left bound was chosen when the mean-response time began to be comparable with the time-slot duration (which will be 5ms) and the 95% CI scissor is small in comparison with the scale of the response time (basically when the standard deviation becomes very small). Here a "zoom" on the rightmost part of the latter plot:

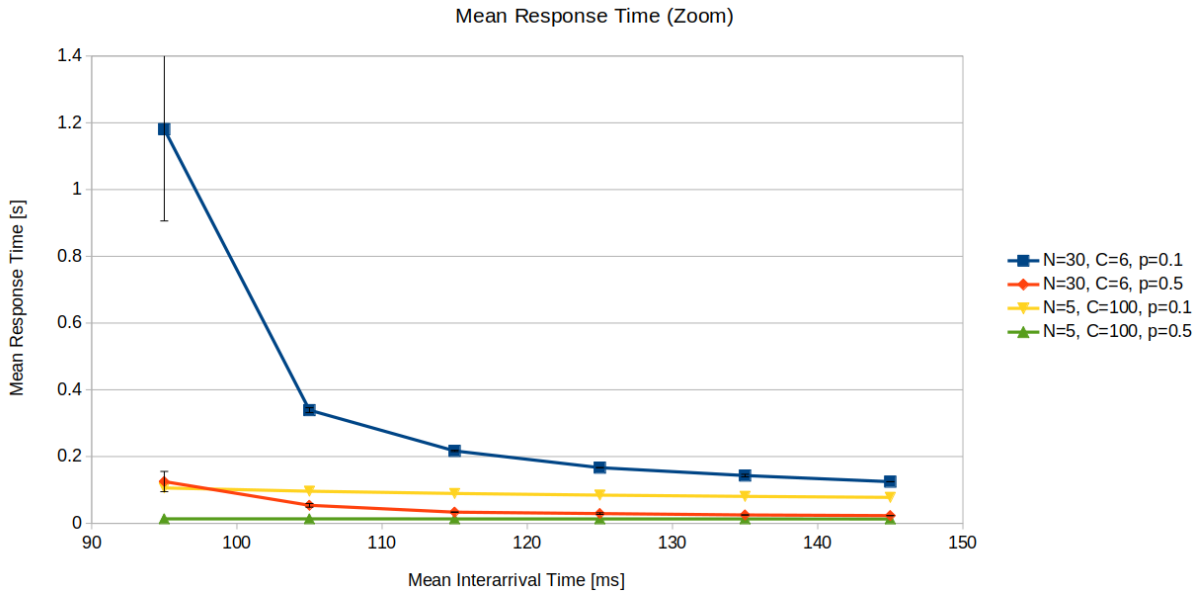


Figure 10: Buffer Explosion between 95ms and 145ms: the 95% CI are too small to be seen on the right part of the plot

The 95% CI were shown only where visible.

5.2 Scenario Calibration

In order to calibrate the simulator parameters, the following range of values were used:

Limited Response Time Scenario

- **Number of Couples Tx-Rx (N)**: [5, 30]
- **Number of Channels (C)** : [6, 100] (Resource Blocks in LTE for different Frequencies)
- **Mean Inter-arrival Time** ($\frac{1}{\lambda}$): [125ms, 500ms]
- **Time-slot duration** (T_{slot}): 5 ms
- **Send Probability (p)**: [0.1, 0.5]

Explosion of Response Time Scenario

- **Number of Couples Tx-Rx (N)**: [5, 30]
- **Number of Channels (C)** : [6, 100]
- **Mean Inter-arrival Time** ($\frac{1}{\lambda}$): [25ms, 55ms]
- **Time-slot duration** (T_{slot}): 5 ms
- **Send Probability (p)**: [0.1, 1]

5.3 Calibration of Warm-Up Period and Simulation duration

For calibrating the warm-up different simulation were made (with the factors range in the latter paragraph). After various test we found that the KPI that impacts heavily in the choose of the warmup time is the *Response Time*. We can see the study about the response time in figure 11. The worst case in terms of convergence time was encountered with $\mathbf{N = 30,\ C = 6,\ \frac{1}{\lambda} = 125ms,\ p = 0.1}$

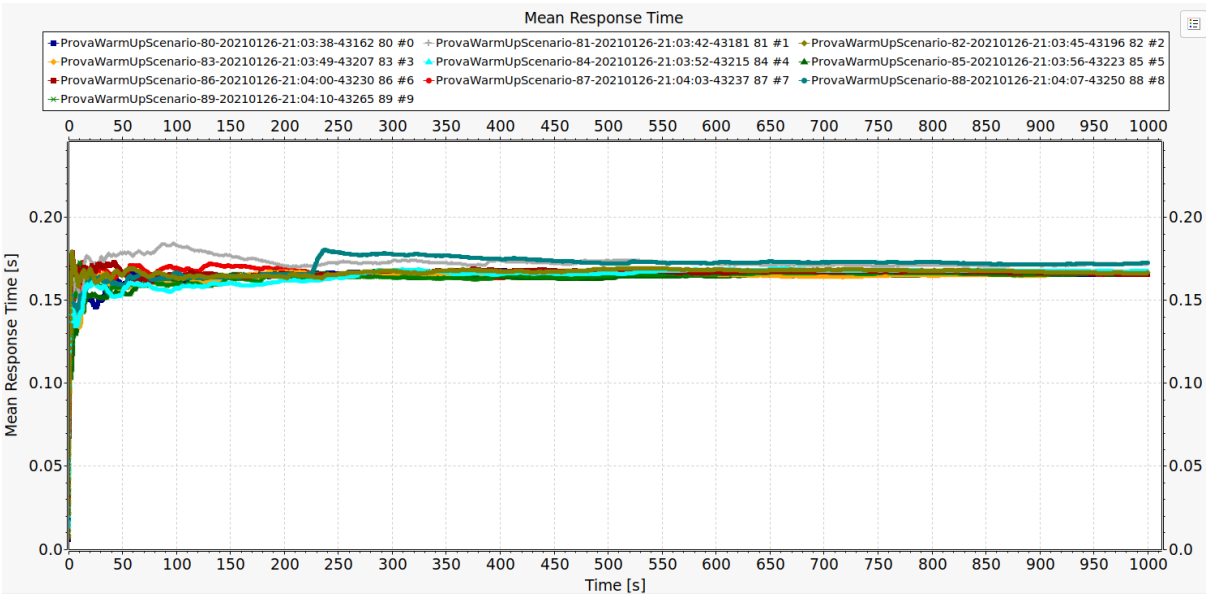


Figure 11: Worst Case Warm-up Response Time

A warm-up period of 250s was chosen.
For what concerns the simulation duration was made a trade-off between the **memory and time consumption** for storing data and performing simulation and the tendency of the KPIs to achieve a **more stable standard deviation**. This was done because there are not stochastic elements in the model (like a particular error probability with a low percentage) that will need a particular amount of time to be shown. Obviously the duration has to be greater than the warm-up duration. All things considered, **a simulation-duration of 5000s was chosen**.

5.4 Limited Response Time Scenario Study

In this chapter we will enter more in the deep to the limited response time scenario, in order to find insights about **throughput** and **response time** (we can due to the fact that in this interval it converges).

5.4.1 Factorial Analysis $2^k r$ on Throughput

In order to analyse the contribution of the factors on the throughput performance, we perform a $2^k r$ analysis with $r = 5$ and $k = 4$ (so we perform $5 \cdot 2^4 = 80$ experiments). We take into account the following factors:

- Number of Couples Tx-Rx (**N**): [5, 30] **(A)**
- Number of Channels (**C**) : [6, 100] **(B)**
- Send Probability (**p**): [0.1, 0.5] **(C)**

- Mean Inter-arrival Time ($\frac{1}{\lambda}$): [125ms, 500ms] **(D)**

The first step is to **check the hypothesis**, in particular we have to control that the **residuals are normal** and that its **standard deviation is constant** (a.k.a. homoskedasticity). For what concerns the normal hypothesis it's possible to see (Figure 12) that the QQ plot of residuals vs normal **show a linear tendency** and so the **hypothesis is verified**.
For the**homoskedasticity**, we have a QQ plot residuals vs predicted response and we can see (Figure 13) that indeed there is a trend, however the **errors** (y axis) **are two order of magnitude below the predicted response** (x axis) and so **we can ignore trends** and state that the **homoskedasticity hypothesis is respected**.

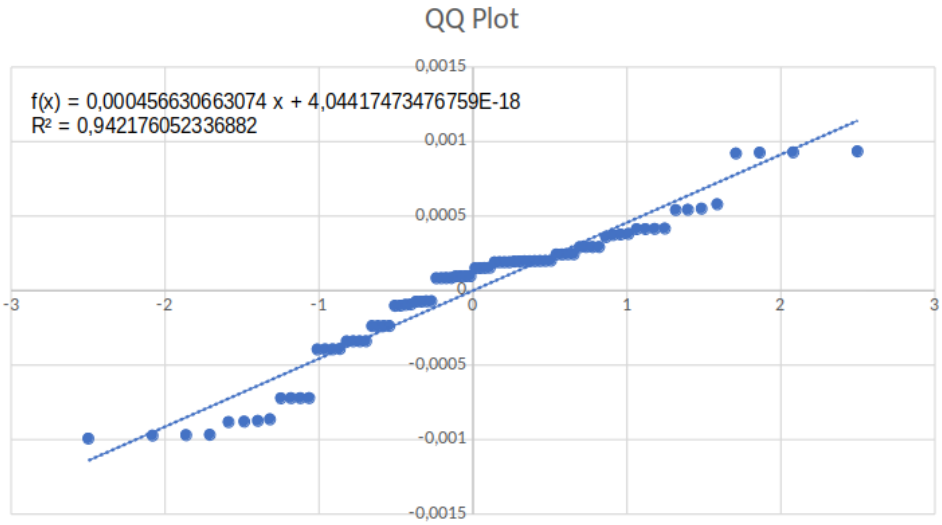


Figure 12: QQ Plot for testing the normal hypothesis

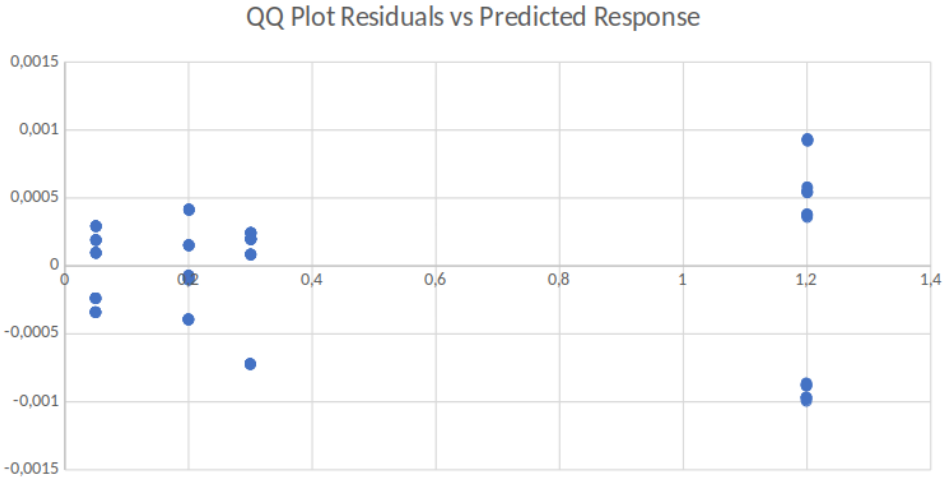


Figure 13: QQ Plot for testing homoskedasticity

Now we can analyse the obtained results. The most relevant ones are the one in the following list, the other factors have an impact on the variability that is very low, ($<<$ 1%) and so they are non relevant:

- **Number of Couples (N)**

It has a **positive impact** on throughput, in particular $qi = [0,312532; 0,312537]$[1] and it accounts for the 48,44% of the variability. This means that the **higher the number of couples the higher the throughput**. In fact with more transmitters we have more packets and so we have an higher throughput.

- **Mean Inter-Arrival Time ($\frac{1}{\lambda}$)**

  It has a **negative impact**: $qi = [-0,262444; -0,262439]$ and it accounts for the 34,15% of the variability. Thus we can say that the **higher the mean inter-arrival time, the lower the throughput**. This happens due to the fact that when we increase the mean inter-arrival time it's more likely that a transmitter have an empty buffer and so it has no packets to transmit, then the throughput decreases.

- **Jointly Effect of Number of Couples and Mean Inter-Arrival Time**

  The **jointly effect of the above factors** accounts for the 17,39% of the variability and it has a **negative impact** ($qi = [-0,187301; -0,187296]$). This because the effect of the mean inter-arrival time is greater with respect to the one of the number of couples, so if both increase then the throughput decreases. Indeed, if we have an higher number of transmitters, but we have the most of them which have an empty buffer (due to the previously explained phenomenon caused by the increasing of the mean inter-arrival time), then the throughput decreases because there are too few packets to transmit.

5.4.2 Factorial Analysis $2^k r$ on Response Time

Now let analyse the contribution of factors on the other KPI, the response time. Also in this case, as the previously, we perform a $r2^k$ analysis with $r = 5$ and $k = 4$ and so 80 experiments in total. The factors are the same of the previously analysis.
The most relevant contributions are the ones explained in the following lists, the others one have an impact in the variability that is negligible w.r.t. the following ones:

- **Send Probability p**

  It has a **negative impact** on the response time, in particular $qi = [-0,033928; -0,033926]$ and it accounts for the **65,67%** of the variability. This means that the h**igher the send probability the lower the response time**. In fact with an high send probability it's more likely that a packet will sent (both in the case of first transmission and in the case of retransmission because of collision) and so it doesn't remain in the transmitter queue increasing its response.

- **Mean Inter-Arrival Time**

  It has a **negative impact**: $qi = [-0,012955; -0,012954]$ and it accounts for the **9,57%** of the variability. Thus we can say that the **higher the mean inter-arrival time, the lower the response time**. This happens due to the fact that when we increase the mean inter-arrival time it's more likely that transmitters send few packets in a slot time and so the probability of having collisions, and so the necessity of retransmitting a packet, is lesser. Moreover even if a collision occurs with an higher mean inter-arrival time there are not several packets in the transmitter queue and so the response time decreases.

[1] This and the following are 95% confidence interval

5.4.3 $2^k r$ Overall Results

As a reference, in the table 2 we can see the results obtained through the $r2^k$ analysis.

	Throughput		***Response Time***	
Factors	qi	Impact on Variability (%)	qi	Impact on Variability (%)
A	0,312534	48,44%	0,006183	2,18%
B	-4,342100E-07	9,35E-11%	-0,006719	2,57%
C	-6,710519E-07	2,23E-10%	-0,033927	65,67%
D	-0,262442	34,15%	-0,012954	9,57%
AB	-1,447366E-07	1,03E-11%	-0,005873	1,96%
AC	-9,078937E-07	4,08E-10%	-0,004269	1,04%
AD	-0,187298	17,39%	-0,005481	1,71%
BC	5,657888E-07	1,58E-10%	0,004619	1,21%
BD	4,342100E-07	9,35E-11%	0,005906	1,99%
CD	1,973682E-07	1,93E-11%	0,010951	6,84%
ABC	4,868415E-07	1,17E-10%	0,004054	0,93%
ABD	1,447366E-07	1,03E-11%	0,005238	1,56%
ACD	8,552622E-07	3,62E-10%	0,003929	0,88%
BCD	-6,710519E-07	2,23E-10%	-0,004240	1,02%
ABCD	-4,868415E-07	1,17E-10%	-0,003761	0,80%

Table 2: Results of $r2^k$ analysis for Throughput and for Response Time. 95% of confidence.

5.5 Limited Response Time Scenario: Result Analysis

Our objective is (as we said at the beginning of the report) the *Assessment of the Effectiveness of the Slotted Random-Access Network Protocol*. In order to do so we choose the mean response time and the mean throughput as Key Performance Indexes. For each KPI we set 2 scenarios: one for **low traffic condition** (in terms of Transmitters and Channels) and the other one for **high traffic condition**. For each factor variation of each scenario of each KPI we perform 35 repetitions in order to obtain meaningful and statistically valid data. Data are computed with a confidence level of 95% (to small to be seen) .Thanks to the analysis performed on this two scenarios and studying the evolution of the KPIs in them, we are able to reach our aim and give some insights on the network protocol.

5.5.1 Throughput

For what concerns the throughput we set these two scenarios:

1. *High Traffic Scenario*

   N = 30, C = 6, Send Probability = 0.1

2. *Low Traffic Scenario*

   N = 5, C = 100, Send Probability = 0.1

In both scenarios we **vary the mean inter-arrival time** from 125 ms to 500 ms with a step of 75 ms. In fact as we seen in the $2^k r$ analysis it is the most relevant factor for the throughput. We can see the results obtained in the outlined scenarios when the mean inter-arrival time grows from 125 ms to 500 ms in figure 14.

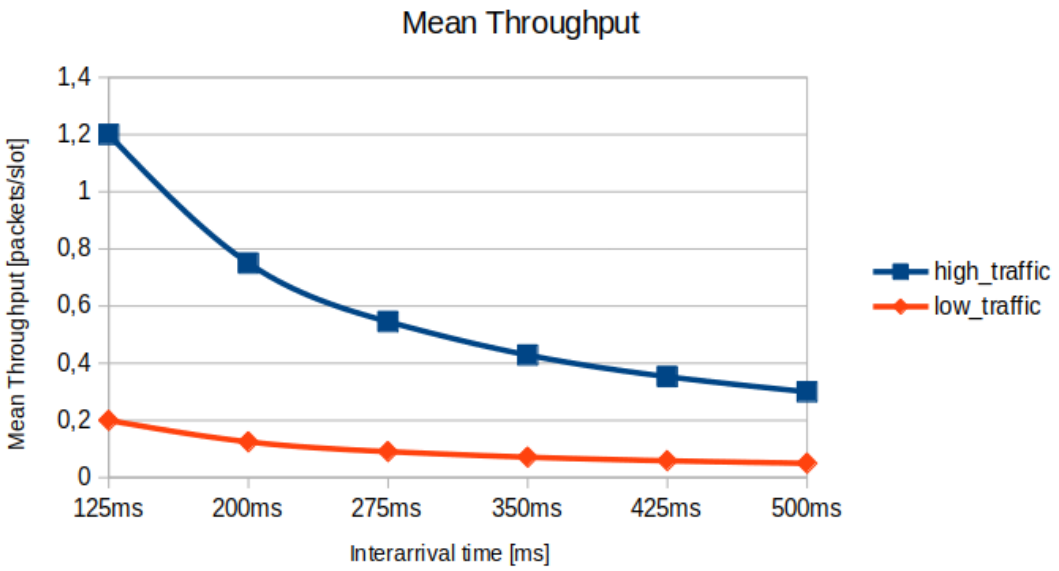


Figure 14: Insight 1

From the plot we can state that the **throughput is higher in the high traffic scenario**, because the more the traffic the more the packets. We can also see that the **best results is obtained with the lower mean inter-arrival time**, this because we have to consider the time slot duration that is 5 ms and it is a lot smaller than the mean inter-arrival time: if the mean inter-arrival time increases, then there are slots where no packets are transmitted (because transmitters have no packets in the queue) and this affect the mean throughput and this is the reason why the throughput decreases with the increase of the mean inter-arrival time. So the latter is another validation of what we obtain from the $2^k r$ analysis. So at the end we can say that the throughput depends mainly from the traffic and from the mean inter-arrival time, this is good because this means that the effect of collisions doesn't not affect so much the overall performance of the network protocol. In fact in the high traffic condition there are more collisions w.r.t. to the scenario with low traffic, but nevertheless the throughput it is not affected by this.

5.5.2 Response Time

For what concerns the response time we set these two scenarios:

1. *High Traffic Scenario*

   N = 30, C = 6, Mean Inter-Arrival Time = 125 ms

2. *Low Traffic Scenario*

   N = 5, C = 100, Mean Inter-Arrival Time = 125 ms

In both scenarios we **vary the send probability** from 0.1 to 1 with a step of 0.1, in fact for the response time the latter is the most relevant factor (as we seen in the $2^k r$ analysis) and so it is the only that if changes causes a big difference.
The results obtained in the previously explained scenario when the send probability (p) varies from 0.1 to 1 is shown in figure 15.

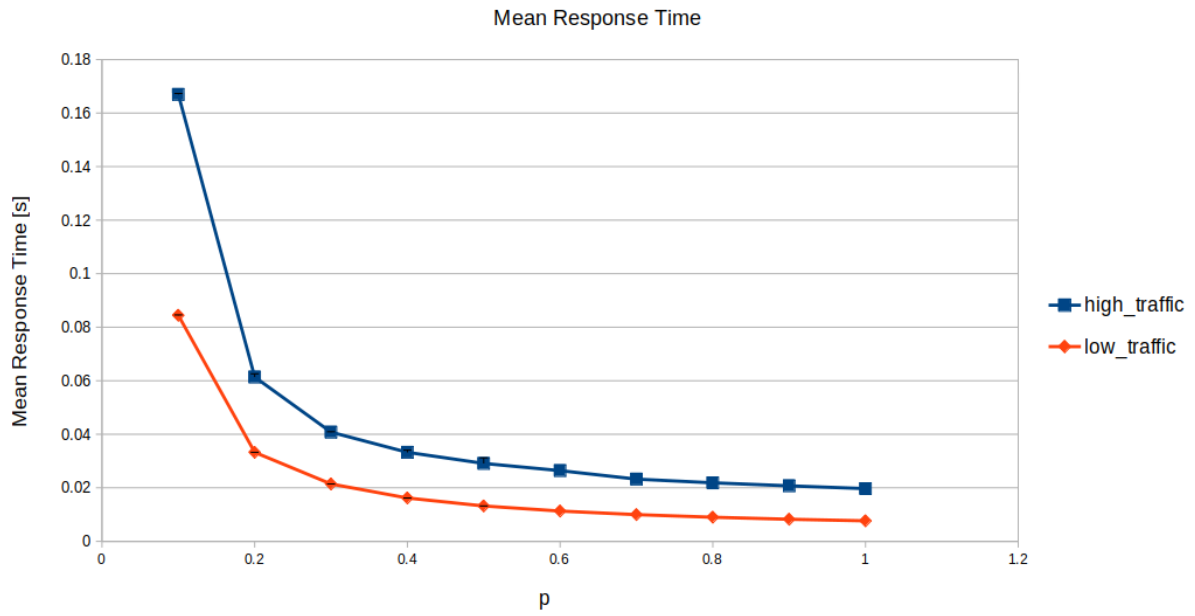


Figure 15: Insight 1

From the figure we can infer that in the **high traffic scenario the response time is higher** than the low traffic scenario, this is something normal. Then we can also say that the m**ean response time decreases with the increasing of the send probability** (as we expected from the $r2^k$ analysis) and that the **differences between the two scenarios are lower when p increases**. The greater value is about 0.17 seconds that are equal to 170 ms which is not a good response time but it is acceptable one and it is a response time which can be experimented also by other network protocols.

5.6 Response Time Explosion Scenario Study

In this chapter we will study more deeply the case in which we have a low mean inter-arrival time which lead to the "explosion" of the response time. Due to the fact that a **huge number of collision is expected**, in order to obtain insights for the throughput we created two other sub-scenarios:

- **Change Of Channel in case of collision (default choice)**: in case of collision of a particular packet, when retrying the retransmission of that packet (after the back-off period) the channel assigned will be picket randomly another time
- **Not Change of Channel in case of collision**

We repeated the $2^k r$ analysis on the throughput for both sub-scenarios and, after verifying the hypothesis (as we shown before) and **checking that the q_i of the most important factors does non include 0 in the relative 95% CI scissor**, we obtained the following result:

- In both sub-scenarios the number of **Couples TX-RX N** is the main factor in terms of impact on the **Throughput** (in both cases an impact greater than 50%)
- In both sub-scenarios the percentage of the Bernoullian experiment success **p** had a low impact (4-5%) on the **Throughput**
- With respect of the limited response time scenario, the mean interarrival time has a low impact on the throughput (around 4%)

5.6.1 $2^k r$ Overall Results

As a reference, in the following table 3 we can see the results obtained through the $r2^k$ analysis.

- Number of Couples Tx-Rx: [5, 30] **(A)**
- Number of Channels C : [6, 100] **(B)**
- Send Probability p: [0.1, 1] **(C)**
- Mean Inter-arrival Time: [25ms, 55ms] **(D)**

	Throughput Change		***Throughput No-Change***	
Factors	qi	Impact on Variability (%)	qi	Impact on Variability (%)
A	1.064	55.93%	1.032	53.00%
B	0.432	9.218%	0.464	10.71%
C	0.308	4.684%	0.290	4.207%
D	-0.285	4.033%	-0.284	4.036%
AB	0.427	9.004%	0.458	10.47%
AC	0.177	1.551%	0.159	1.268%
AD	-0.143	1.0225%	-0.143	1.018%
BC	0.144	1.026%	0.162	1.309%
BD	-0.216	2.314%	-0.216	2.338%
CD	-0.260	3.345%	-0.260	3.372%
ABC	0.149	1.099%	0.167	1.397%
ABD	-0.211	2.210%	-0.211	2.230%
ACD	-0.129	0.830%	-0.129	0.834%
BCD	-0.191	1.817%	-0.192	1.847%
ABCD	-0.196	1.912%	-0.197	1.945%

Table 3: Results of $r2^k$ analysis for Throughput and for Response Time. 95% of confidence.

5.7 Response Time Explosion Scenario: Result Analysis

Due to the fact that the $2^k r$ analysis underlined the importance of the Number of Couples Tx-Rx (**N**), in the following plot we can see more clearly the effects of changing N: the **throughput increase by increasing N**.

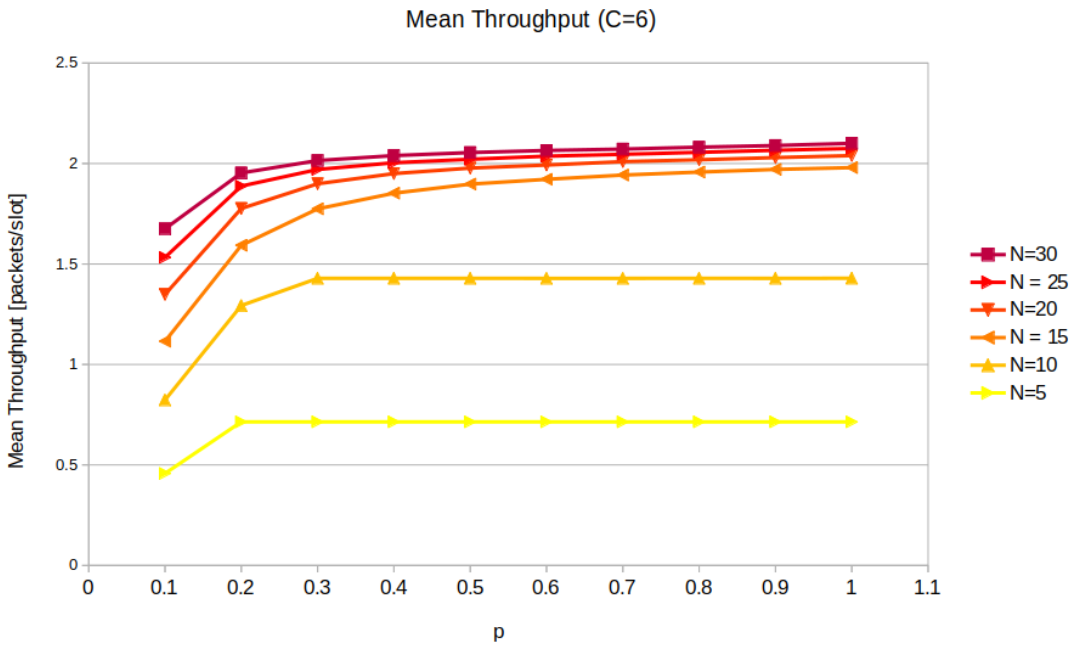


Figure 16: Insight 4

The plot was also done with respect to different values of **p**, even if such factor does not have a really big impact. This was done because, thinking about the real world, the parameter **p** is the simplest thing to change, so, even if his impact is not huge, can be a good idea to show which values of p tunes the **throughput**. All things considered: **increasing p increases slighly the throughput**, however this increase becomes "smaller" with bigger values of p.
For what concerns the **comparison between the Change Channel sub-scenario with the No-Change one**, the following plot can describe well the differences:

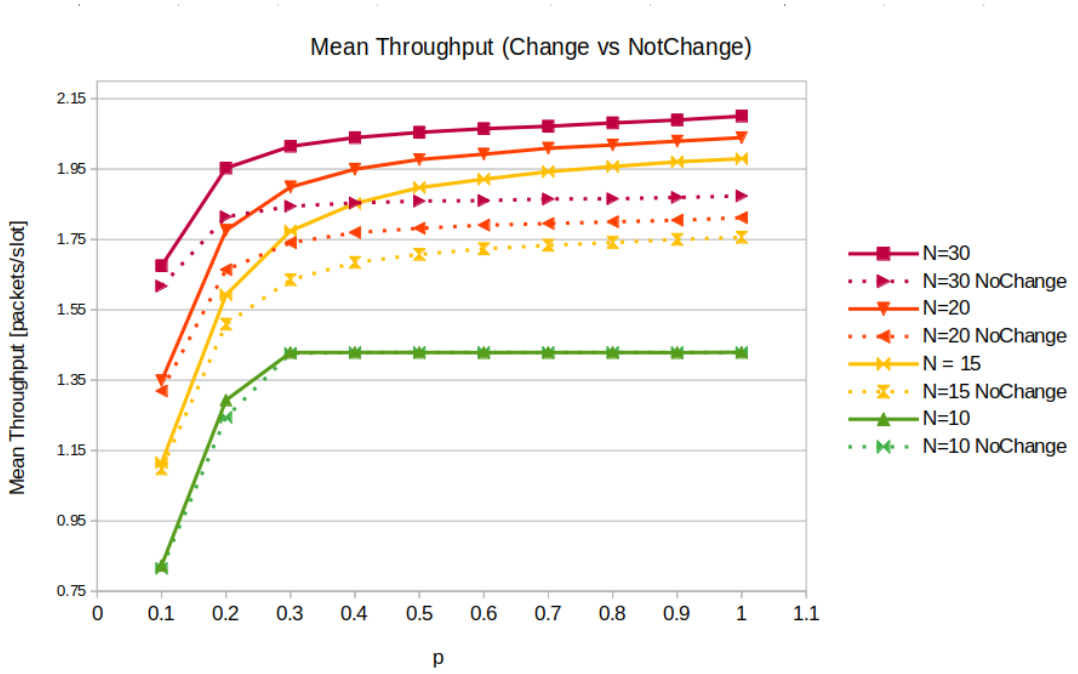


Figure 17: Insight 5

As we can notice, the dotted lines and the continuous lines represent the same simulation in

terms of number N, but with different option of Changing or not the channel in case of collision. So, we can see clearly that for more higher number of N, there is a consistent difference between the two scenarios: **the sub-scenario without change of the channel is worst in terms of throughput**.

6 — Conclusions

The final consequences of the report are shown in this section. The results obtained in the **Response Time Explosion scenario** underlines **three aspects**:

- The **No-Change of channel sub-scenario** (in case of collision), is **worst in terms of throughput** than the Change Channel one, **when the number of trasmitters start to grow** (so more collisions expected mantaining fixed the number of channels that will makes more clear the difference between the two scenarios)
- An **higher Send Probability is better** in this case because **maximizes the throughput** (consideration on Response Time can't be done). This increase start grows faster at the beginning of the increase of p.
- The **Throughput keeps increasing with the number of Transmitters** (with our ranges) and is the **most relevant factor** as underlined in the relative $2^k r$ experiment.

Moreover the overall performance of the system depends heavily from the mean inter-arrival time. In fact if it is lower than 125 ms no observations can be done and no meaningful data can be obtained because the response time diverges. This means that the network protocol studied is good for applications in which the mean inter-arrival time is more or less defined, then if we want to use this type of network for application with mean inter-arrival time lower than 125 ms, this is discouraged. Instead, if for example we want to use the protocol in a network used by a factory to link several machines, and each machine transmits a packet each x ms with $x \geq 125$, then this network protocol can be suggested.
In this context, the results obtained in the conditions previously explained and shown, are good and in particular we can infer the following:

- The throughput maintains a good value for all values and it is greater when the traffic is higher as we expect.
- The response time is acceptable also in the worst case and it is good for the other cases, both in high traffic condition and in low traffic condition. In any case the performance in the worst case is poor because the mean response time is comparable with the mean inter-arrival time and so if we know that the probability of transmission is low (so in a time slot we may have a lot of transmitters that don't transmit) the network doesn't perform well.

At the end it is better to use this network in a not flexible environment in which the packets can arrive at the transmitter with very different inter-arrival time and in which the probability of sending a packet is very low. On the other hand it is well to use it in specific environments in which the good performance of the network and its relativity simplicity (this is something that may require more in-depth studies different from this one) can be a very good choice.

Comparing the two scenarios we found we can also state that, for our ranges of interest, the **presence Bernoullian Experiment tends to decrease the performance** of our system (Throughput in the Response Time Scenario, Response time in the Limited Response time Scenario). So we can think to **remove it from the protocol to tune our system**.