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:

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

  We can also convert this performance metric in a more standard form, dealing with packets per second:

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

- **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.

- **Mean Number of Packets in the queues**: due to the fact that the this metric is time dependent, will be computed as:

  $$E[N] = \frac{\sum_{i=0}^{N} N_i * (t_{i+1} - t_i)}{t_N - t_0}$$

  Where N_i are the number of packets in the buffer on the interval $[t_i, t_{i+1})$.

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**

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** is reasonable due to the fact that exist some network protocols which work under this assumption, see for instance Slotted Aloha that is the most famous one.
- 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.

- When a packet collide it's reasonable to think that the transmitter will change the transmission channel in order to avoid another collision. Indeed this along with the back-off time are the techniques that should avoid another collision.

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.

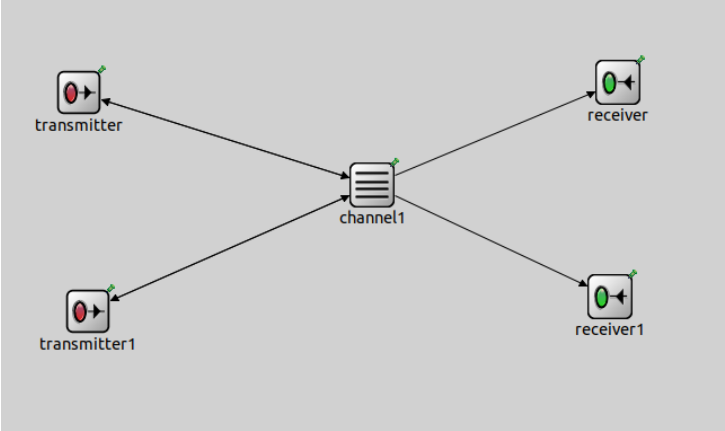


Figure 1: Network example

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 is 0 then the simulation stops immediately because has no sense running a simulation with 0 channels.
- If the time slot size 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 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 perform two tests 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**.

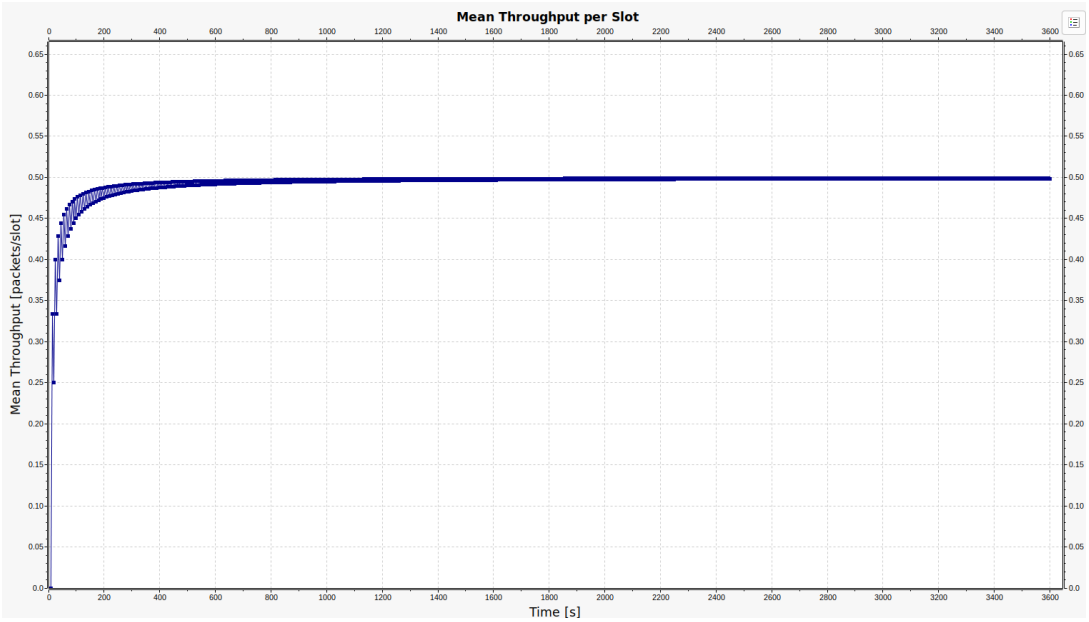


Figure 2: Consistency Test 1

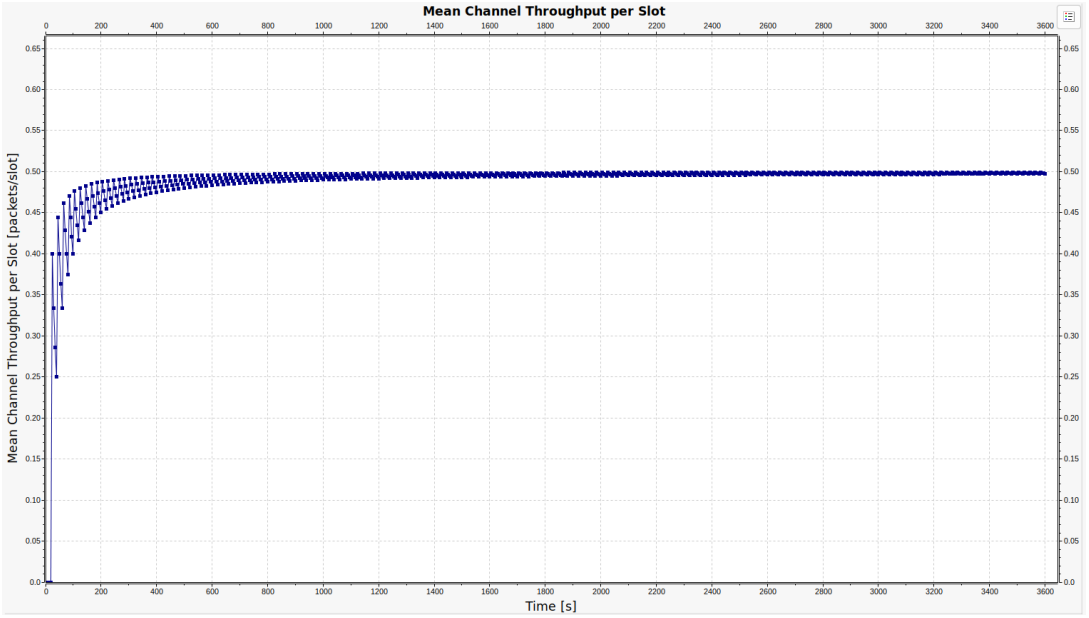


Figure 3: Consistency Test 2

The oscillations at the beginning are due to the fact that the mean inter-arrival time is bigger with respect to the time slot size. In fact we can observe that in the second test there are larger oscillations because the difference between the mean inter-arrival time and the time slot size is bigger than the one of the first test.
Now let analyse the **response time**, we can see its measure in both tests in figure 4 and 5.

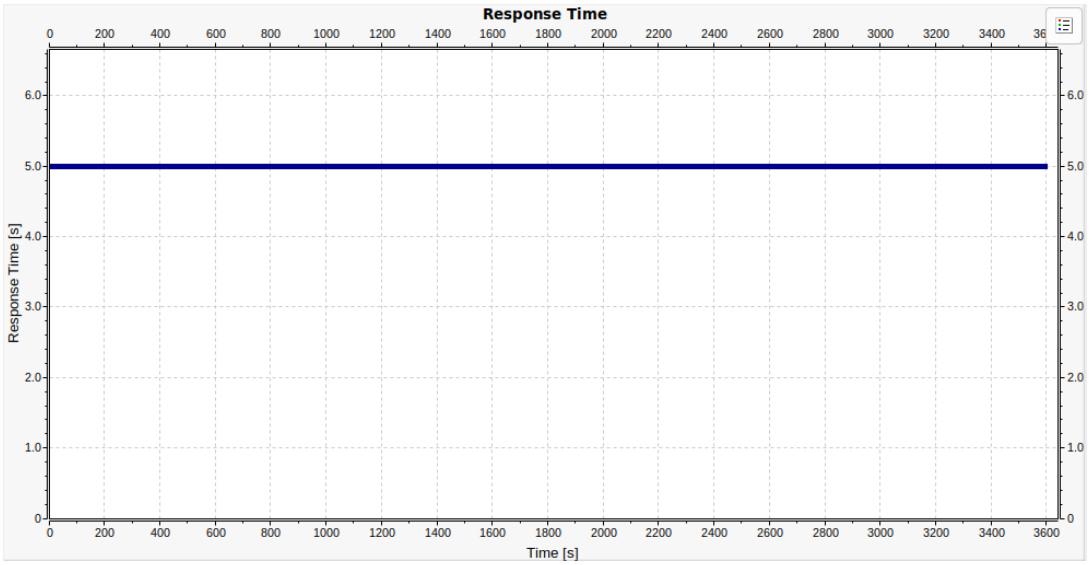


Figure 4: Consistency Test 2 - Response Time

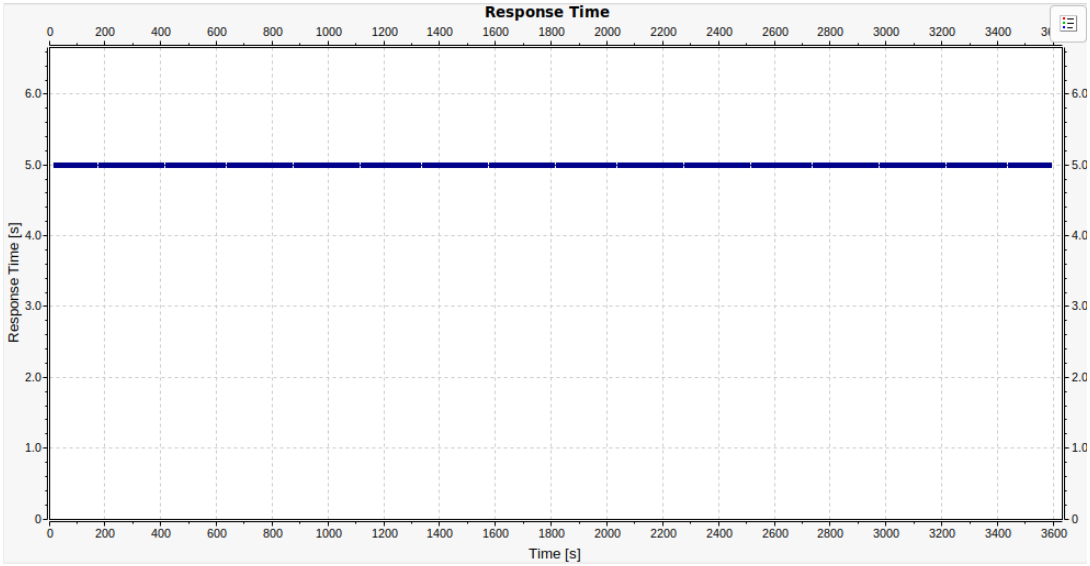


Figure 5: Consistency Test 2 - Response Time

The **mean response time in both cases is 5 seconds**, this is a **lower bound** for the response time because in both tests 5 seconds is the size of the slot time so it's impossible to have a response time lower than the slot size.
We can see that in the second test we have a less continue plot, this is due to the fact that the mean inter-arrival time is bigger in the second case and so the packet are more distant in time. We can conclude that also for what concerns the response time the consistency is ensured because it is the same in the case in which we have a source transmitting each 10 seconds and in the case in which we have two sources transmitting each 20 seconds.

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, 2 simulations have been performed 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.

Test	Number of Transmitters	Number of Channels
1	8	20
2	10	20

Table 1: Continuity test parameters

The remaining parameters are the same of all of four tests:

- Send probability: 1
- Mean Inter-arrival time: 10 sec (deterministic)
- Time slot size: 5 sec
- Threshold: 20 sec

The results of the simulation as shown in the figure 6 and 7 (we measure the mean throughput).

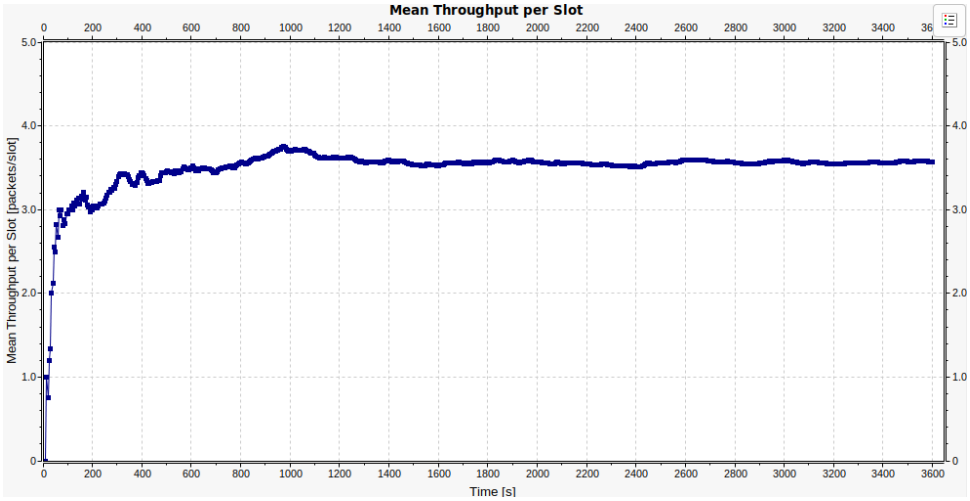


Figure 6: Continuity Test 1 - Mean Throughput per Slot - Collisions detected: 272

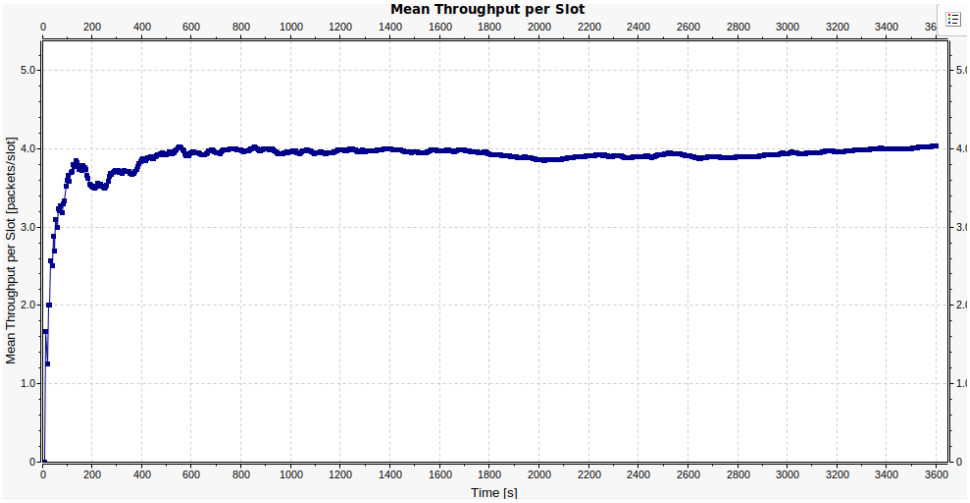


Figure 7: Continuity Test 2 - Mean Throughput per Slot - Collisions detected: 390

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, 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.
At the end of the day we can see that in the first case the mean throughput is settled to a value about 3.6 and in the second case about 4. So changing slightly the input changes slightly the output.
If we analyze the response time we can see that the difference is bigger between the two cases because the response time is sensible to the variation of transmitters and channels, and this is something that must be taken into account. In fact, as we have seen previously, there are a lot of collisions in the second case and this is the main reason for the increasing of response time in the second test w.r.t the first one.

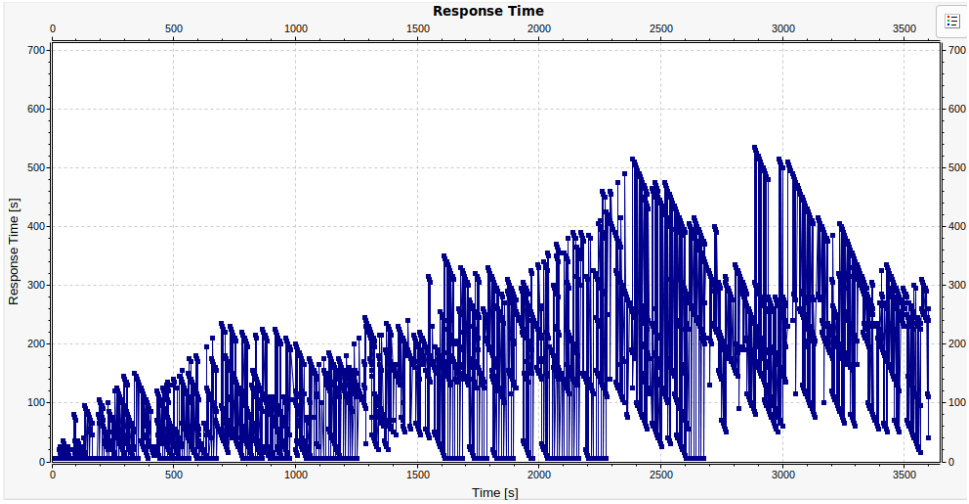


Figure 8: Continuity Test 1 - Response Time - Collisions detected: 272

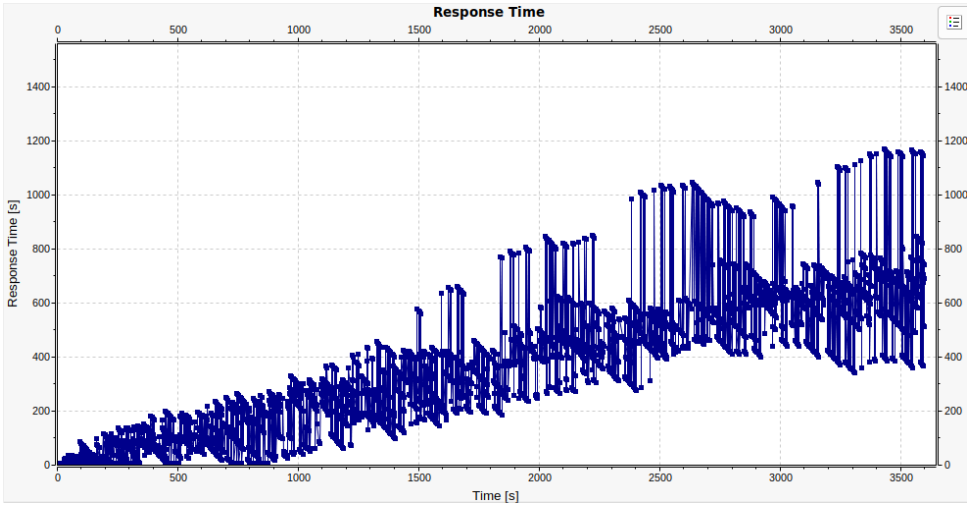


Figure 9: Continuity Test 2 - Response Time - Collisions detected: 390

It's possible to see the response time measured in the two tests in the figure 8 and 9. In any case we can see that the continuity test is correct because the system works as we expect: channels fixed, the more the transmitters, the more the collisions, the more the response time.

In addition to this, some simulation were taken in order to **assess the monotonicity** of some

KPI by changing some factors:

- **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:

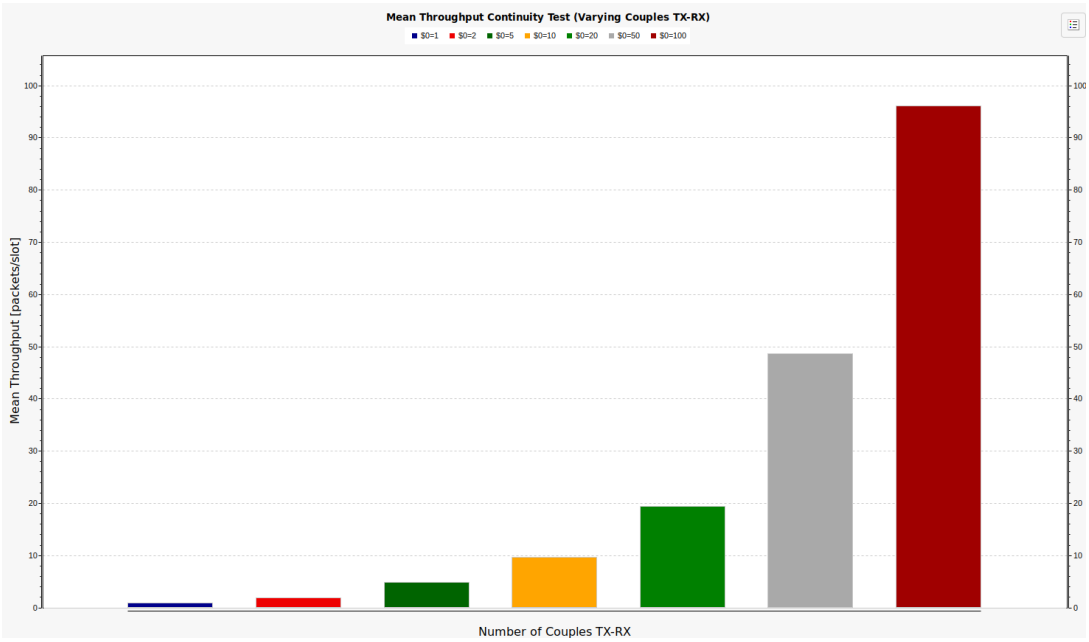


Figure 10: Continuity Test - Increasing Number of TX-RX (Main factors: **N** = 1, 2, 5, 10, 20, 100; **C** = 20000; $\frac{1}{\lambda}$ = 20 ms; T_{slot} = 5ms; $p = 1$)

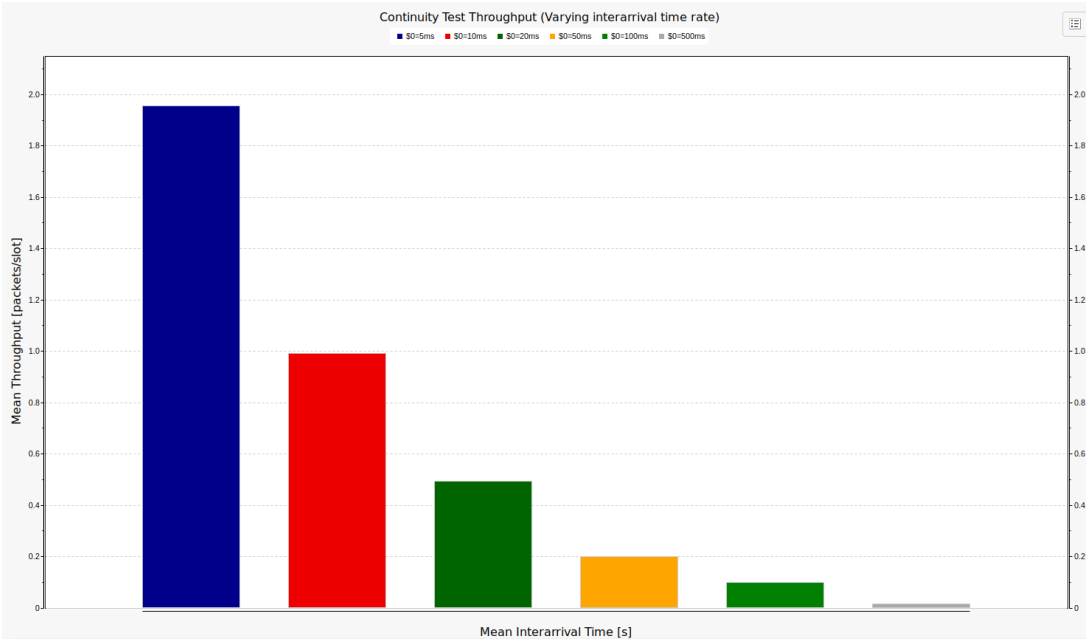


Figure 11: Continuity Test - Increasing Mean Inter-arrival Time (Main factors: **N** = 2; **C** = 20000; $\frac{1}{\lambda}$ = 5ms, 10ms, 20ms, 50ms, 100ms, 500ms; T_{slot} = 5ms; $p = 1$)

- **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:

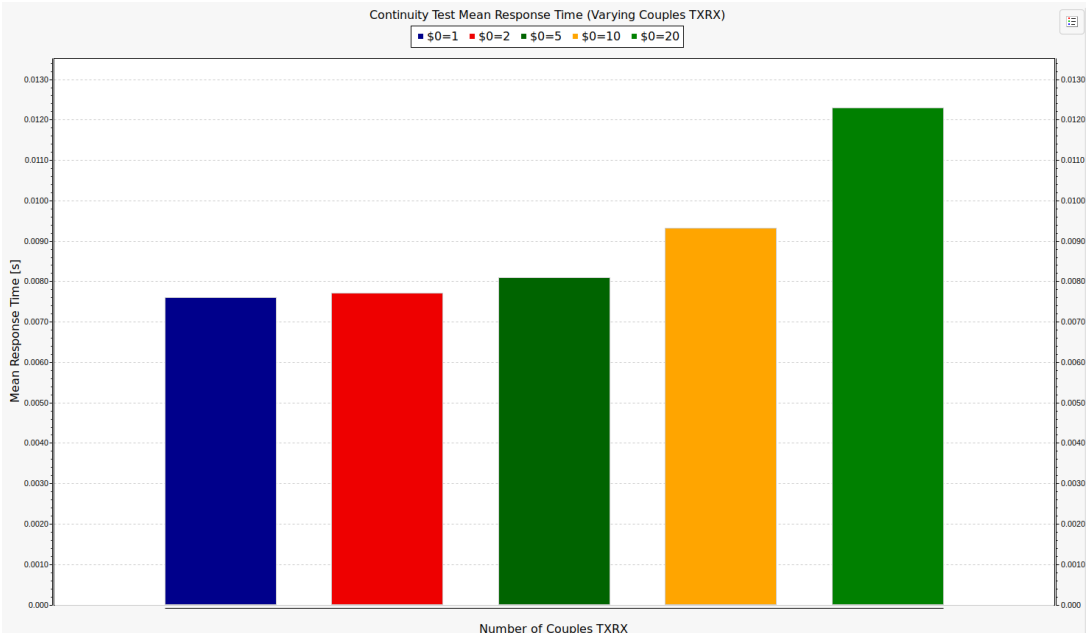


Figure 12: Continuity Test Mean Response Time- Increasing Number of TX-RX

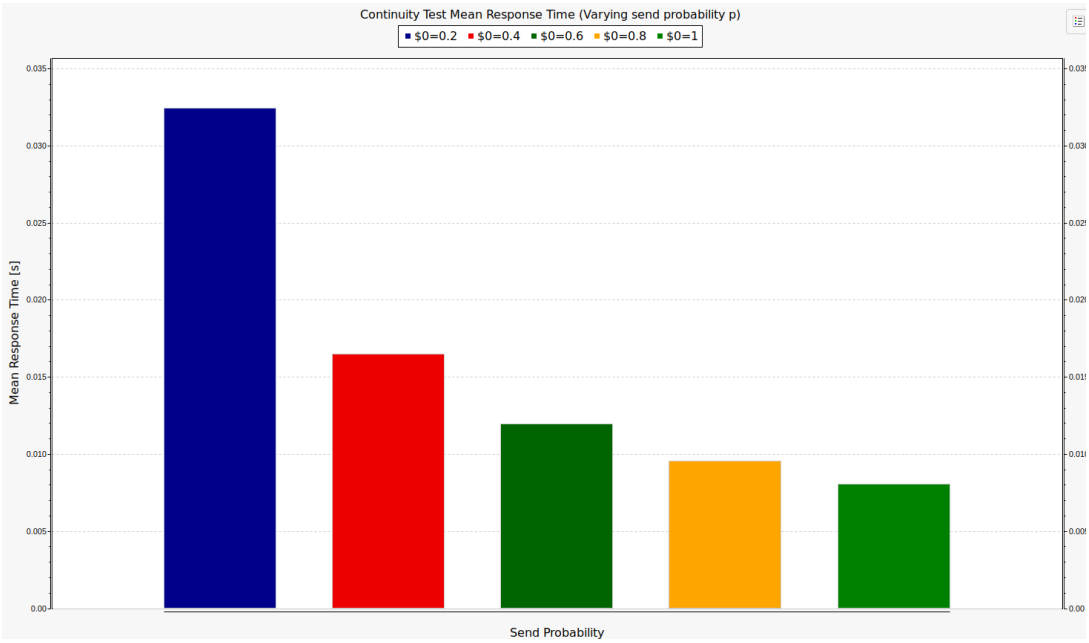
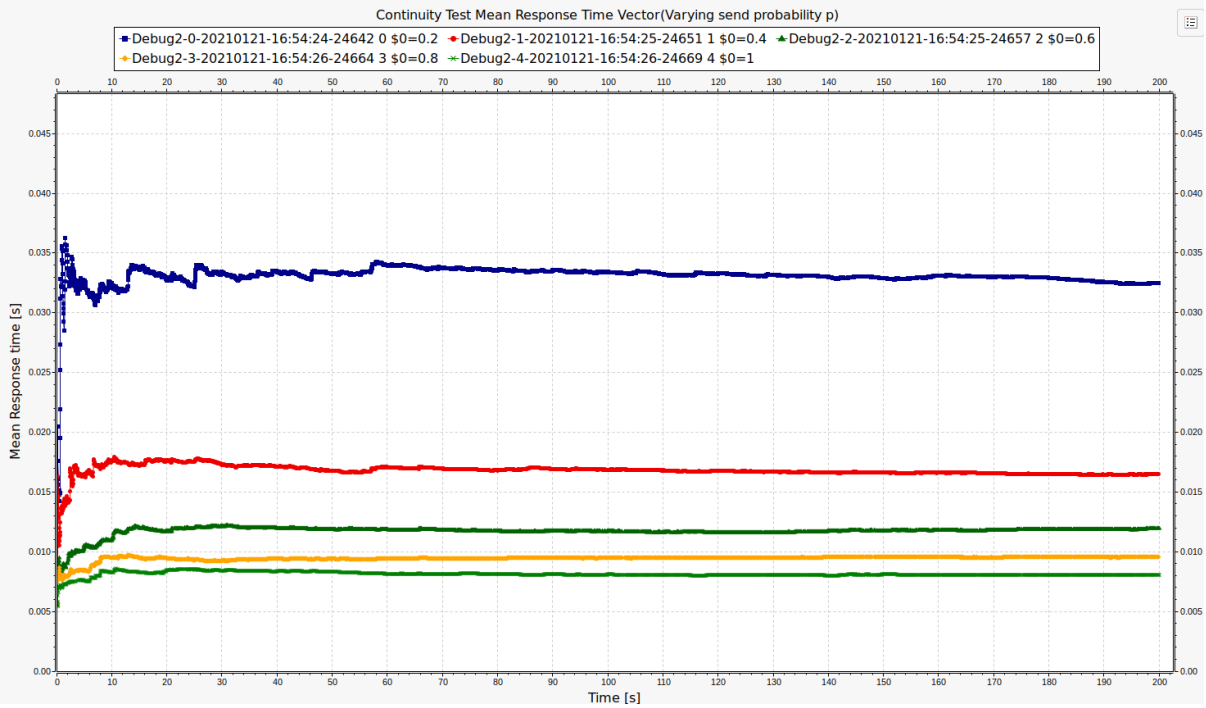


Figure 13: Continuity Test Mean Response Time- Increasing Sending Probability P(Main factors: $\mathbf{N} = 5$; $\mathbf{C} = 4$; $\frac{1}{\lambda} = 200$ms; $T_{slot} = 5$ms; $p = 0.2, 0.4, 0.6, 0.8, 1$)

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:


Continuity Test Mean Response Time Vector(Varying send probability p)
Debug2-0-20210121-16:54:24-24642 0 $0=0.2
Debug2-1-20210121-16:54:25-24651 1 $0=0.4
Debug2-2-20210121-16:54:25-24657 2 $0=0.6
Debug2-3-20210121-16:54:26-24664 3 $0=0.8
Debug2-4-20210121-16:54:26-24669 4 $0=1
Mean Response time [s]
Time [s]

4.5 Test Simulations with Binomial Model (1)

This test simulation has been performed with the following parameters:

Parameters

- Number of couple tx-rx: 1
- Number of channels: 1
- Send probability: ${0.05, 0.1, 0.15, 0.2, 0.4, 0.5, 0.6, 0.8} (p)
- Mean inter-arrival time: 1s (deterministic) ($lambda$)
- Time slot size: 2s (T_{slot})
- simulation-duration: 3600s (T_{sim})
- 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 ($lambda < T_{slot}$). 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:

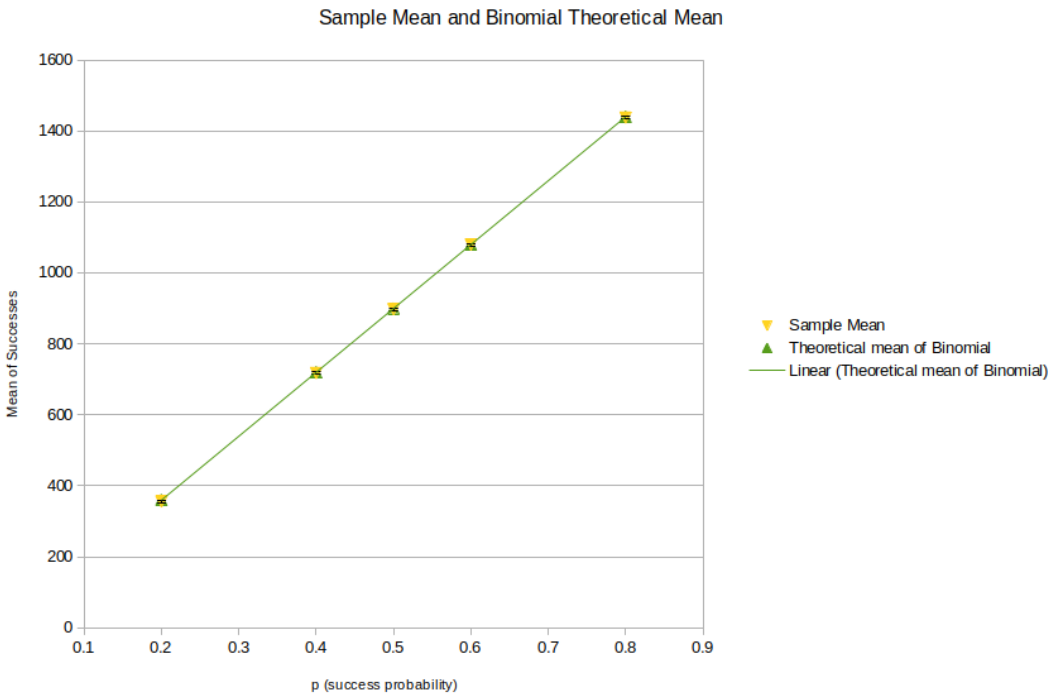


Figure 14: Test Binomial Model

As we can see it is difficult to recognize the theoretical results with the results obtained with the test simulations and the Confidence Interval can be barely seen.

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: ${0.2, 0.4, 0.6, 0.8} ($p$)
- Mean inter-arrival time: 1s (deterministic) ($lambda$)
- Time slot size: 2s (T_{slot})
- simulation-duration: 3600s (T_{sim})
- repeat: 40
- seed-set: ${repetition}
- No Backoff 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.

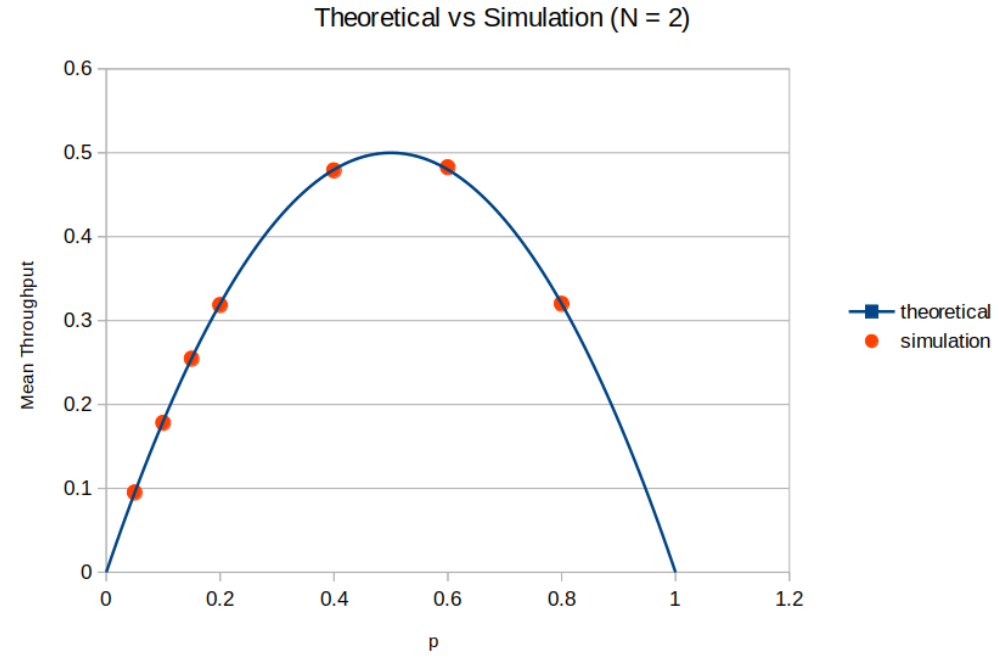
The probability of a successful sent in a particular timeslot, in this case, is the following:

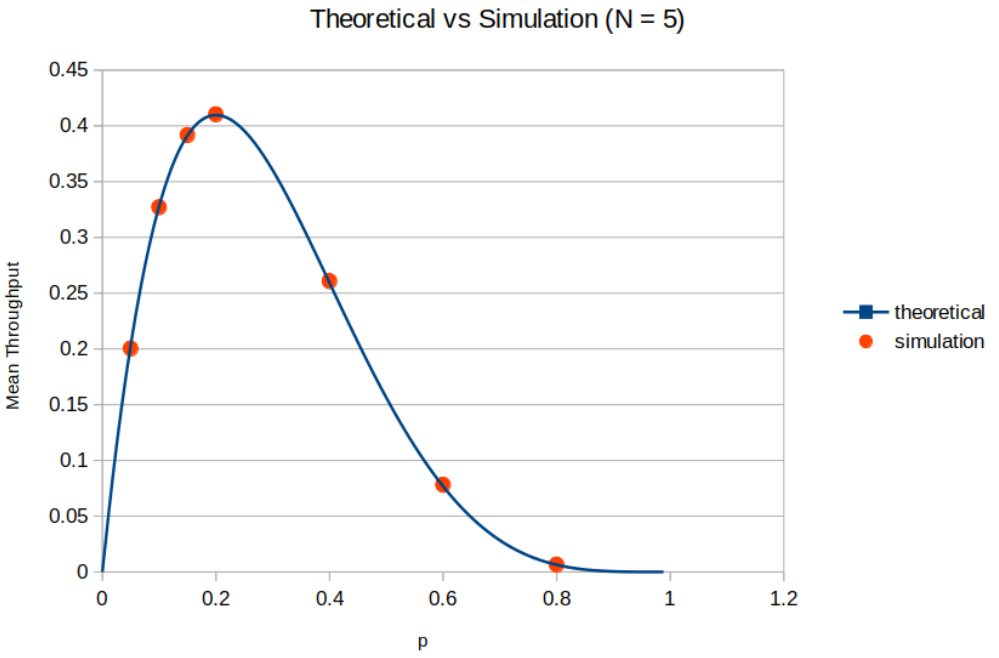
$$P\{"successful\ transmission"\} = P\{"only\ one\ tx\ transmit"\} = N \cdot p \cdot (1-p)^{N-1} \quad (5)$$

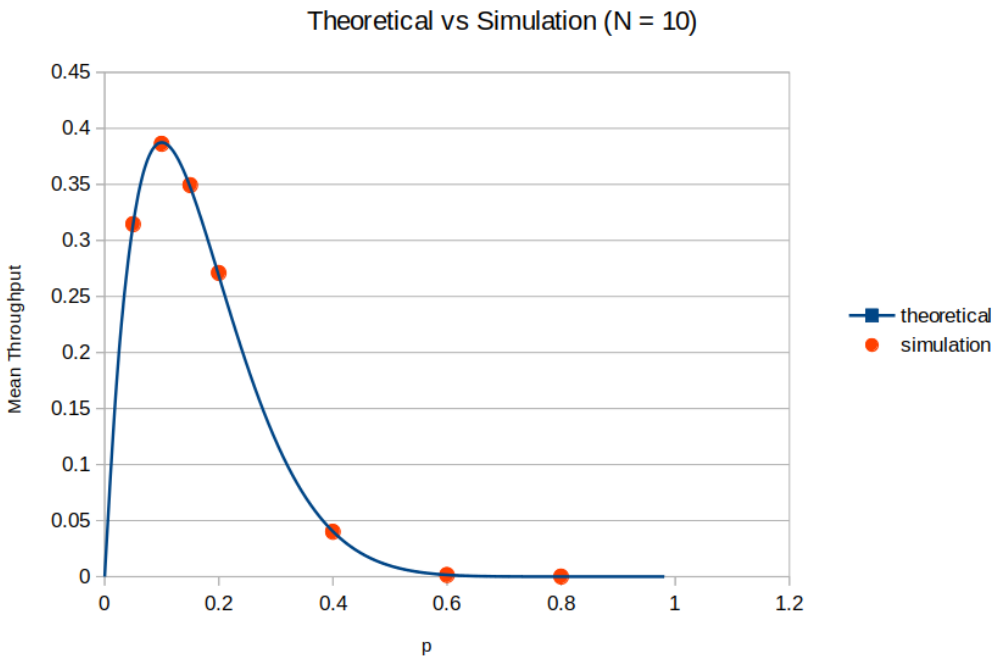
This probability of successful transmission can be seen as the mean throughput of the system in the single channel case. In fact, let N_p the number of packets successfully sent to the corresponding receiver, N_t the number of time-slot considered in the count:

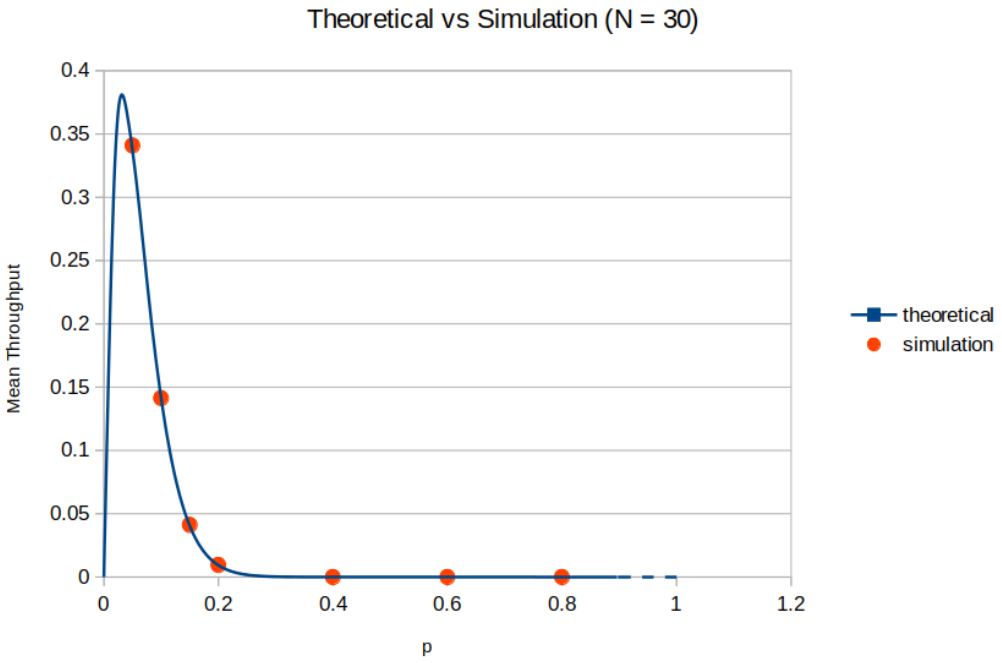
$$Tp\ (slot) = \frac{N_p}{N_t} = \frac{N_t \cdot P\{"successful\ transmission"\}}{N_t} = N \cdot p \cdot (1-p)^{N-1}$$

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. Before doing so, an observation of the result can be carried out at this point: with a good number of couple tx-rx a huge sending probability (i.e. greater than 0.5) is pointless to obtain a high throughput. This result will be considered during the scenario calibration in the next chapter.

5 Simulations Experiments

5.1 Scenario Calibration

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

- **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] (a study was performed in order to choose the mean inter-arrival time that allow us to have meaningful data)
- **Time-slot duration (T_{slot})**: 5 ms
- **Send Probability (p)**: [0.1, 0.5]

5.2 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 is clear 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 15.

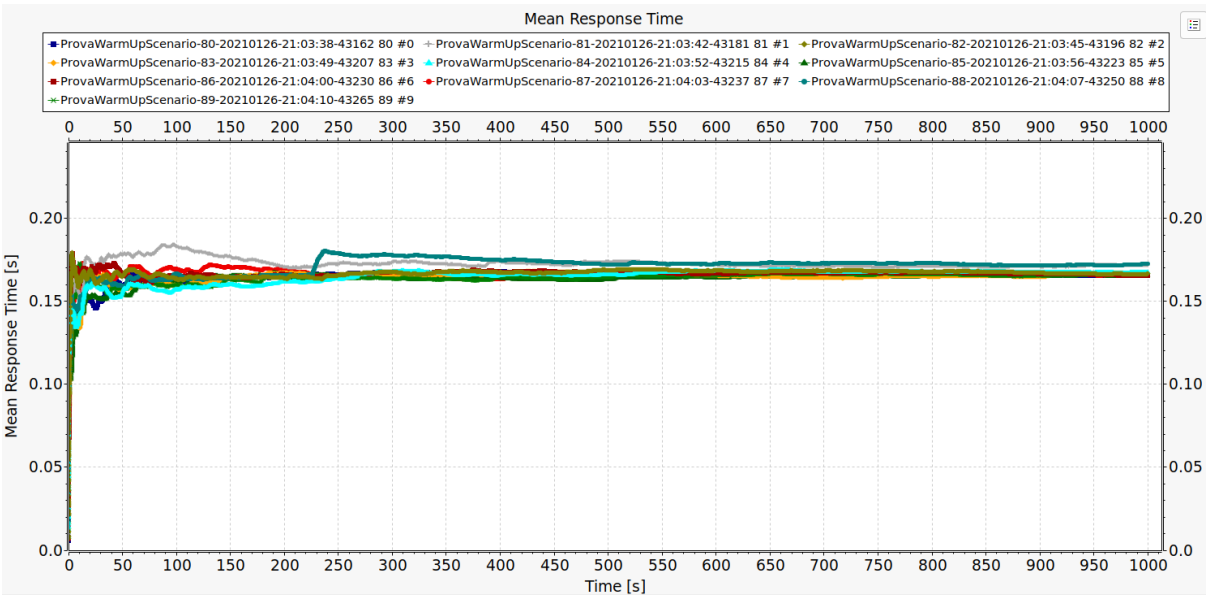


Figure 15: Worst Case Warm-up Response Time

A warm-up period of 250s was chosen.
For what concerns the simulation duration was made a trade-off between the memory consumption for storing data and the length of the simulation itself. 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**.

5.3 Design of Experiments

5.3.1 Factorial Analysis $r2^k$ on Throughput

In order to analyse the contribution of the factors on the throughput performance, we perform a $r2^k$ analysis with $r = 5$ and $k = 4$ (so we perform $5 * 2^4 = 80$ experiments). We take into

account the following factors:

- Number of Couples Tx-Rx: [5, 30] **(A)**
- Number of Channels C : [6, 100] **(B)**
- Send Probability p: [0.1, 0.5] **(C)**
- Mean Inter-arrival Time: [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 16) 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 17) 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.

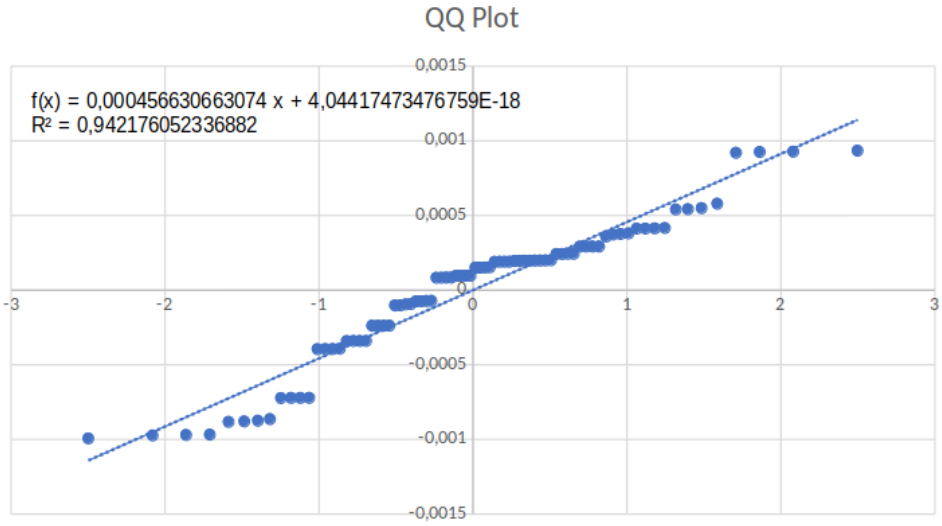


Figure 16: QQ Plot for testing the normal hypothesis

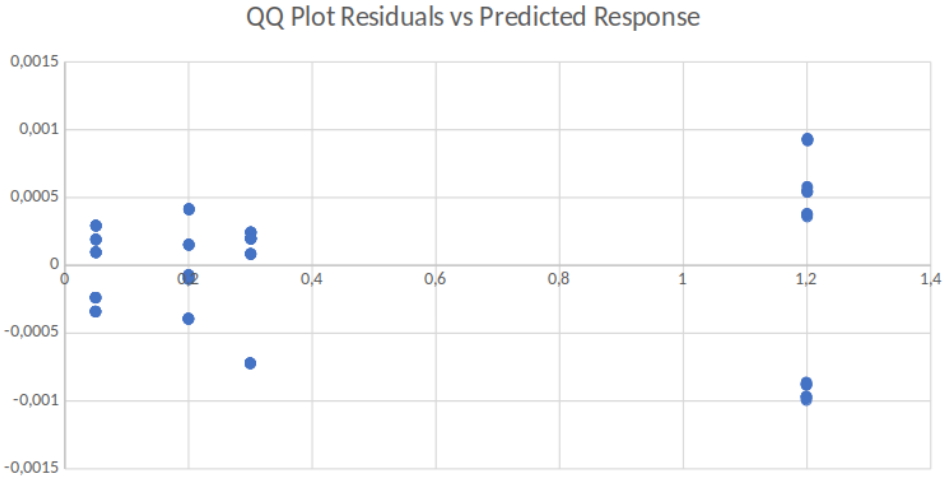


Figure 17: QQ Plot for testing homoskedasticity

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**

  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**

  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.

5.3.2 Factorial Analysis $r2^k$ 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.
We can see the plots to check the hypothesis in figure 18 (for what concerns the normal hp) and in figure 19 (for what concerns the homoskedasticity).

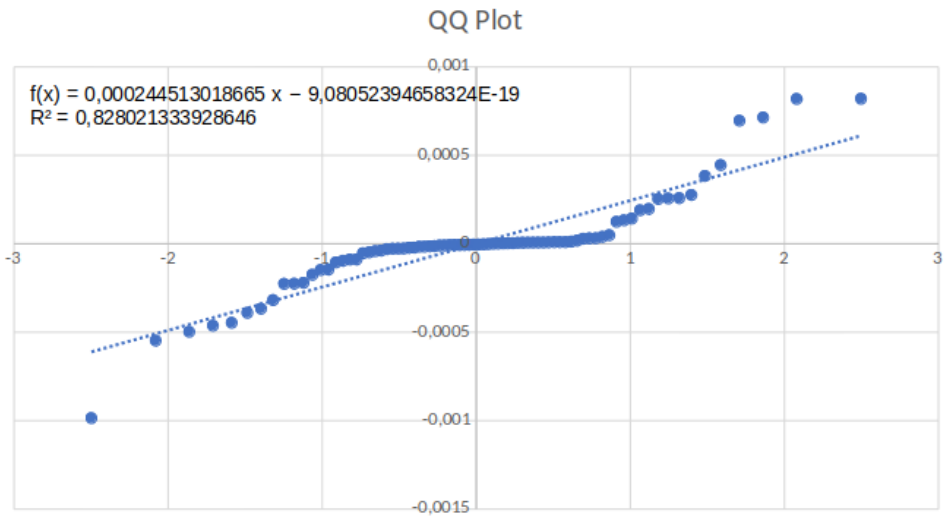


Figure 18: QQ Plot for testing the normal hypothesis

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

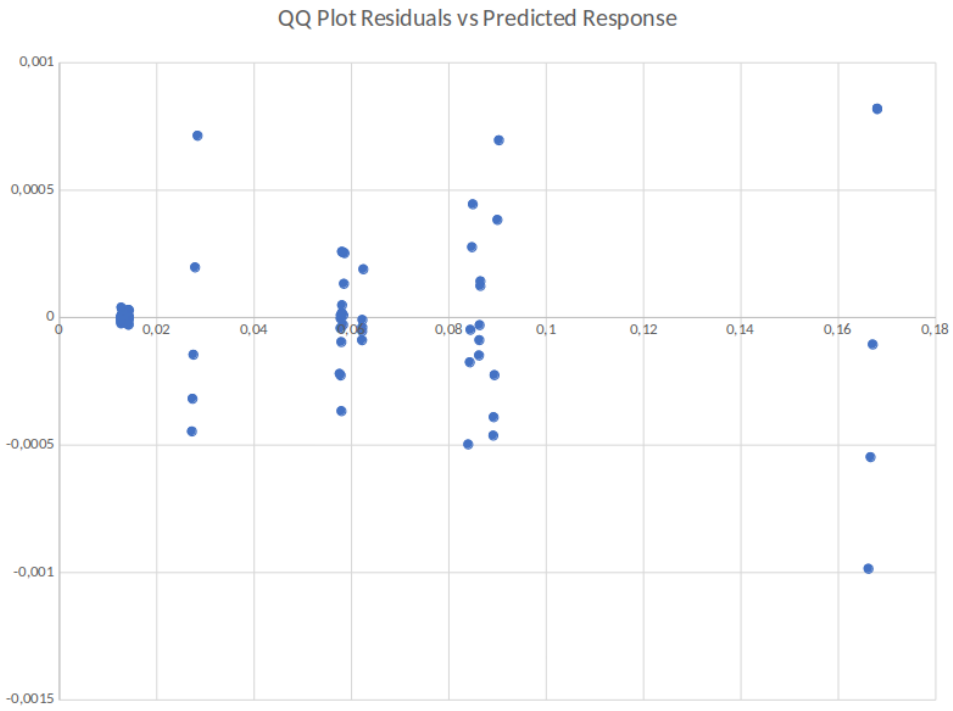


Figure 19: QQ Plot for testing homoskedasticity

The plot about the normal hp is obtained through a logarithmic transformation and it shows an approximating linear trend. Instead for the homoskedasticity we can see that there is a trend but the errors are at least one order of magnitude below the predicted response, and so also this hypothesis is verified.
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**

  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 higher 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.

5.3.3 $r2^k$ Overall Results

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.

5.4 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 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 final conclutions on the network protocol.

5.4.1 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 $r2^k$ 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 20.

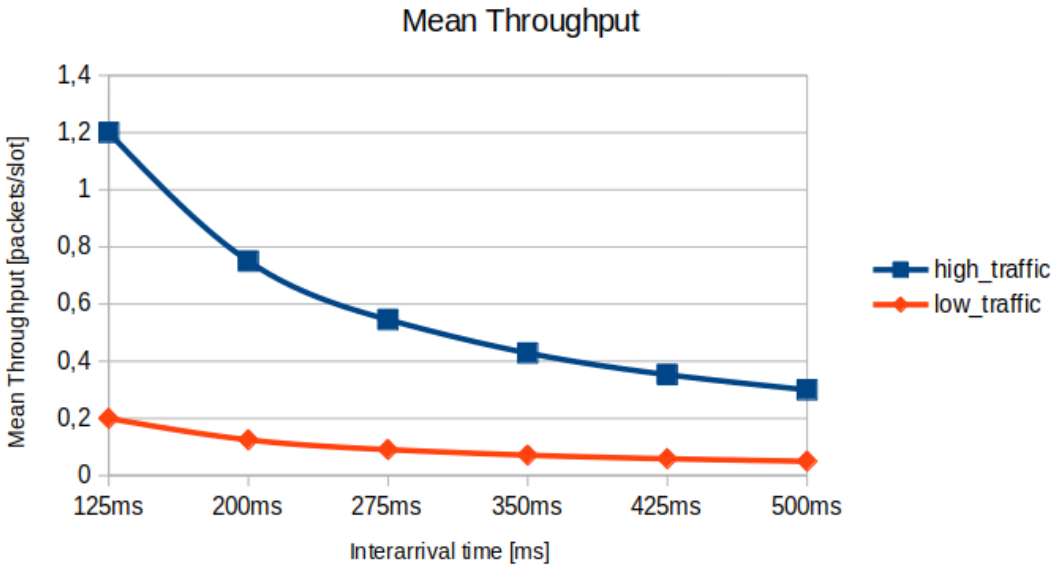


Figure 20: Insight 1

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 $r2^k$ 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.
Now let analyse the fairness of the network protocol for what regards the throughput. It's possible to see the Lorenz Curve for both scenarios in figure 21, we show one figure for both cases because they are indeed equals.

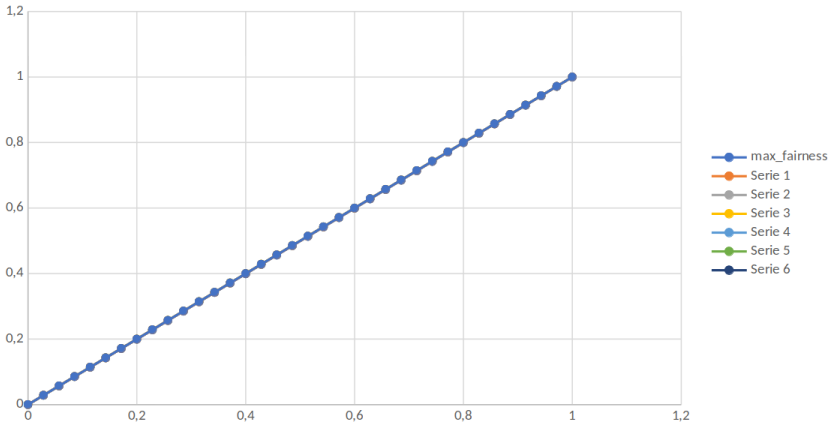


Figure 21: Insight 2

As we can see in the plot in all cases the network protocol is fair for what regards the throughput.

5.4.2 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 $r2^k$ 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 22.

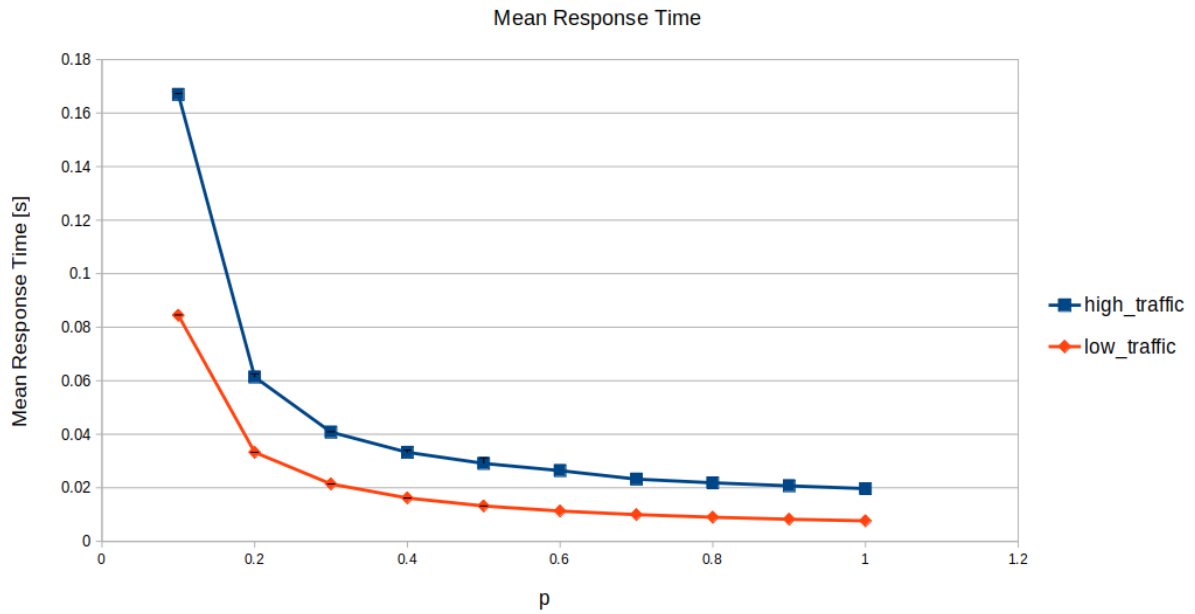


Figure 22: Insight 1

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 mean response time decreases with the increasing of the send probability (as we expected from the $r2^k$ analysis) and that the difference 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.
Now let focus on the fairness of the protocol as regards the response time. We can see Lorenz Curve for the high traffic scenario in figure 23 and Lorenz Curve for the low traffic scenario in figure 24.

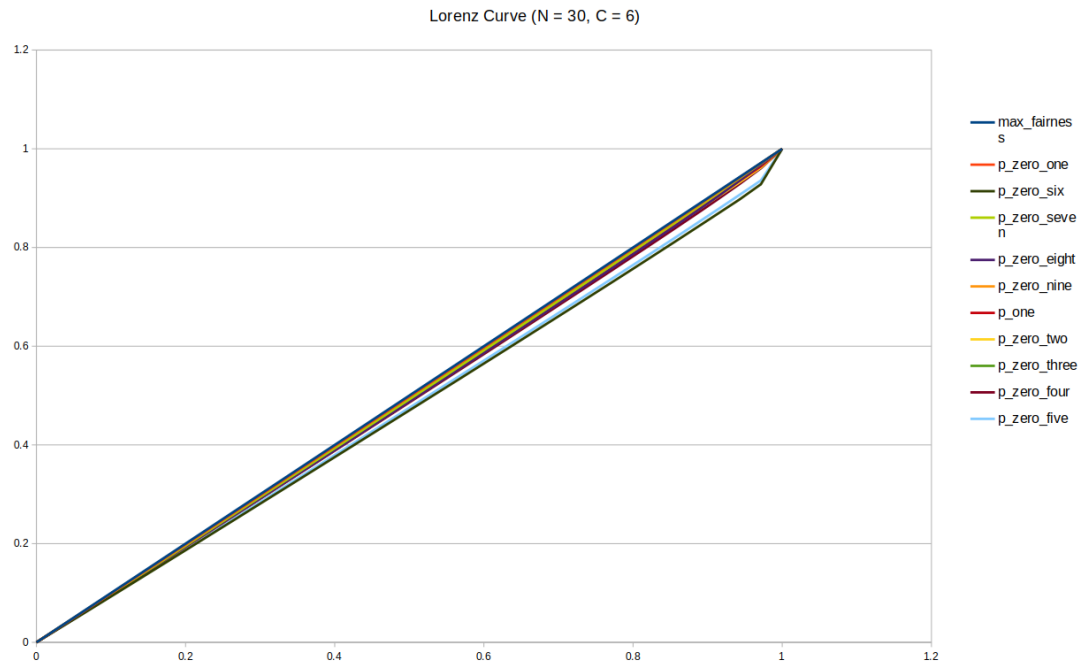


Figure 23: Insight 2

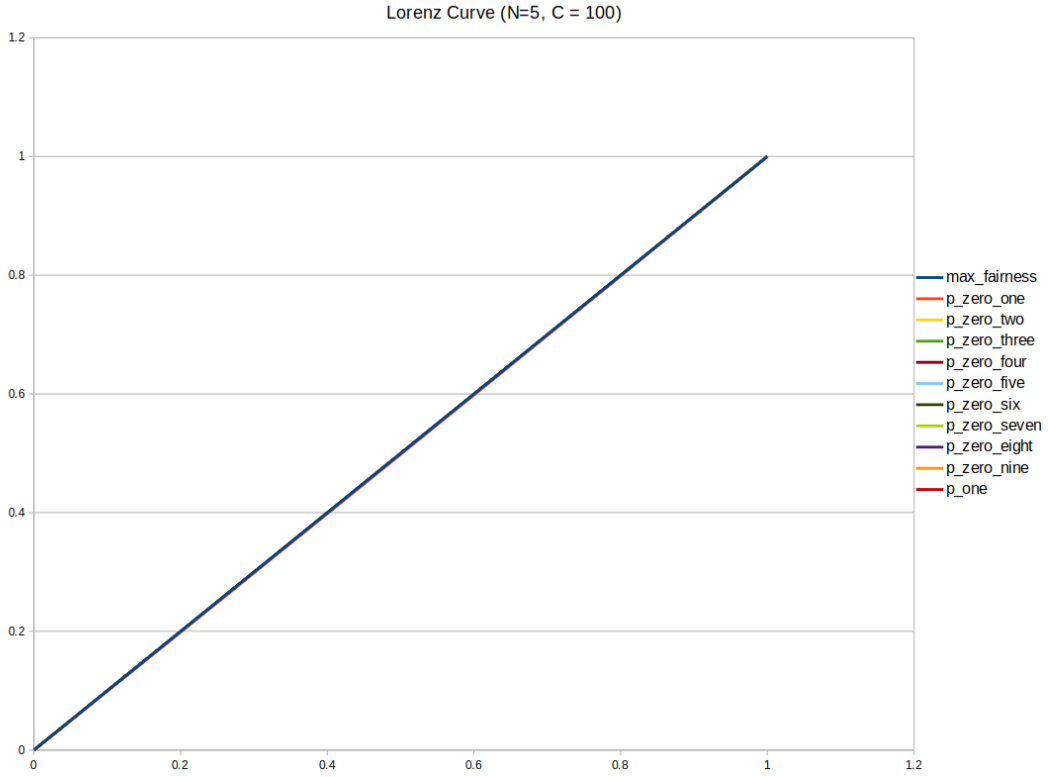


Figure 24: Insight 3

We can see that in both cases the curve related to the simulation scenarios are very close to the line of maximum fairness (in the case of low traffic they are overlapping). So we can say that the network protocol is absolutely fair for what regards the response time, whatsoever be the traffic condition.

6 Conclusions

The final consequences of the report are shown in this section. Considering the join effect of the KPIs we can say that the Slotted Random-Access Wireless Network studied has good performance and the following observations about it can be done.
As first observation we can say that the system is fair. We can say this due to the Lorenz Curve (figure 21, 23, 24).
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.