UNIVERSITY OF PISA

Performance Evaluation project

Opportunistic cellular network

Martina Criscione, Maria Taibi, Martina Troscia

Specifications

In this project we analyzed how a simple cellular network works. The 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 *Channel Quality Indicator (CQI)* which determines the number of bytes that the antenna can pack into a *Resource Block (RB)*. 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. 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. If the queue can't be emptied, the antenna examines the queue of the following user.

Assumptions

We assumed that the timeslot length is 1ms as it is the usual length in a real LTE network. The timeslot is set by a timer implemented into the antenna; at the beginning of the timeslot, the antenna asks cellulars to send the CQI and waits for all responses before composing the frame. The users simply send the CQI when requested.

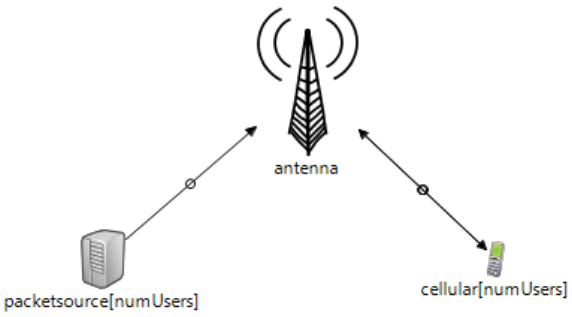
The packets which are on the queues on the antenna have been sent by packet sources, one for each user. The packets are sent asynchronously with respect to the timeslots and their length is uniformly distributed in [1, 75]; the upper bound was chosen so that the largest packet dimension fits a frame at the minimum CQI.

We decided to send to each user only the piece of frame containing its data, instead of sending the whole frame to all. So it is possible that in a timeslot a user doesn't receive anything, if there weren't packets for him or if it wasn't possible to empty his queue.

Model

Modules and network

The submodules we used in the network, defined in .ned files, are



- **cellular**: it is a simple module containing an **id** to identify the user and a variable **distr_cqi** to discriminate the distribution used to generate the CQIs. It has an input and an output gate to interact with the antenna.
- **packetSource**: it is a simple module containing an **id** to relate it to the user, a double **mean** to set the interarrival rate of packets, since arrivals are exponentially distributed, and the variable **max_pkt_len**, to set the maximum packet length. There is only an output gate to send packets to the antenna.
- **antenna**: it is a simple module containing a parameter **num_users** to determine the number of users that are in the system, "inherited" from the network, and a variable **timeslot** to define its

length, set to 1ms as default. It has two vectors of input gates of num_users elements, one for cellular and one for packet sources, and a vector of output gates to interact with every cellular.

So the network has a module antenna, a vector of cellular modules and a vector of packet source modules.

Messages

The messages exchanged among modules, defined in .msg files, are

- **Cqi:** it is used by the cellulars to send its CQI to the antenna. It contains the user **id** and the **cqi**.
- **Data:** it is used by the packet sources to send packets to the antenna. It contains the **id** related to the user and **dim**, the packet length in bytes.
- **Frame:** it is used by the antenna to send the piece of frame for the user, specified by the field **id**. The message also contains **numBytes**, the data length in bytes, **numPkts**, the number of packets for the user, and **tEnq** that is a vector containing the point in time on which each packet was put in the queue.

We also used a *cMessage* message **send_cqi** to let the antenna ask the CQI to the cellular.

Code

In the C++ files, we used the following classes to define the behavior of the modules described above

- **cellular**: it contains the double **p**, needed when we have to generate the CQI with the binomial distribution. The class contains the following functions:
  - initialize(): it initializes the parameter p, if needed.
  - handleMessage(): it handles the arrival of a message: if it is a *send CQI message* it calls createCqi() else if it is a *frame message* it compute statistics on the data arrived.
  - createCqi(): it generates the *CQI message* according to the distribution used and sends it to the antenna.
- **packetSource**: the class contains the following functions:
  - initialize(): it calls createBeep().
  - handleMessage(): when a *beep message* arrives, it generates a *data message* and sends it to the antenna. At the end, it calls createBeep().
  - createBeep(): it creates a *beep message* and schedules it in order to have exponential interarrival times of packets on the antenna.
- **antenna**: the class contains the vector **iusers** of structures of type info. The structure **info** contains a field **id**, the user identifier, a field **cqi** and the queue **q** of structures of type qelem. The structure **qelem** contains a field **dim**, the packet length in bytes, and a field **t**, the time when the packet was enqueued. The class contains the following functions:
  - initialize(): it initializes the vector iusers and starts the first timeslot.
  - handleMessage(): it handles the arrival of a message: if it is a *beep message* it asks the CQI to the users; otherwise, it calls the function related to the type of message received.
  - handleBeep(): it creates a *beep message* and schedules it to establish the start of the timeslot.
  - handleData(): when a *data message* arrives, the antenna inserts the packet into the corresponding user's queue.

- handleCqi(): the antenna updates the field cqi into the users queue. If all users have sent the CQI, it orders the iusers vector using sortCqi(), calls handleFrame() and handleBeep().
- handleFrame(): the antenna creates the frame and sends it to users.
- searchId(): it searches the position into the iusers vector occupied by a certain user.
- swap(): it is used by sortCqi() to swap elements.
- sortCqi(): it orders the iusers vector according to decreasing CQI.

Tuning of factors

The following factors were considered

- **number of users**: we analyzed the system's behavior with a small population of users (10 and 20 users) and with a larger one (50 and 100 users).
- **mean of the exponential distribution** (used for packet interarrival times): we decided to analyze the behavior with a mean equal to 0.02 and 0.04 because real applications such as VOIP and video streaming send packets every 20-40ms. Then we decided to consider means close to this values, slightly smaller (0.005 and 0.01) and slightly larger (0.07 and 0.1).
- **type of distribution used for the CQI generation**: we studied how the system works in the case where users have uniformly distributed CQIs in [1, 15] and where they have binomially distributed CQIs in [1, 15], chosen so that the mean CQI of different users is sensibly different.

Scenarios

We analyzed the following scenarios, where the factors above are varied:

- exponential interarrivals, uniform service demands, uniform CQI;
- exponential interarrivals, uniform service demands, binomial CQI;

Validation

The model has been validated to check if its behavior was as we expected. At first, we checked the behavior of the system with the uniform CQIs and, given a small number of users as input (5 users), we checked if they were served in the correct way, according to their CQI and to the available space on the frame. Then we checked if generated CQIs were coherent with the values of p we set as input, in the case with binomial CQIs.

Performance indices

The two metrics we had to take into account were the **throughput** and the **response time**. The first one was computed as the number of bytes sent by the antenna at each timeslot; the response time was measured by each user and it was computed as the time interval from the packet arrival on the antenna to its reception on the cellular.

We also computed the mean response time of the system, intended as the response time of all packets regardless of the recipient (**global response time**). Since we observed that it corresponds to the mean of users' response time, we decided to study the global response time.

In addition, we analyzed the **used RBs**, computed as the mean number of RB used in a certain configuration over the total frame length (25 RBs); we also studied the mean of the **queues length,** in number of packets, before the antenna composes the frame.

Objectives of the analysis

1. Analyze the trend of throughput, global response time, used RBs and queues length, varying the factors.
2. Find a trade-off among configurations to maximize the throughput and minimize the global response time.
3. Observe if the scenario with uniform CQIs is fairer towards users than the one with the binomial CQIs.

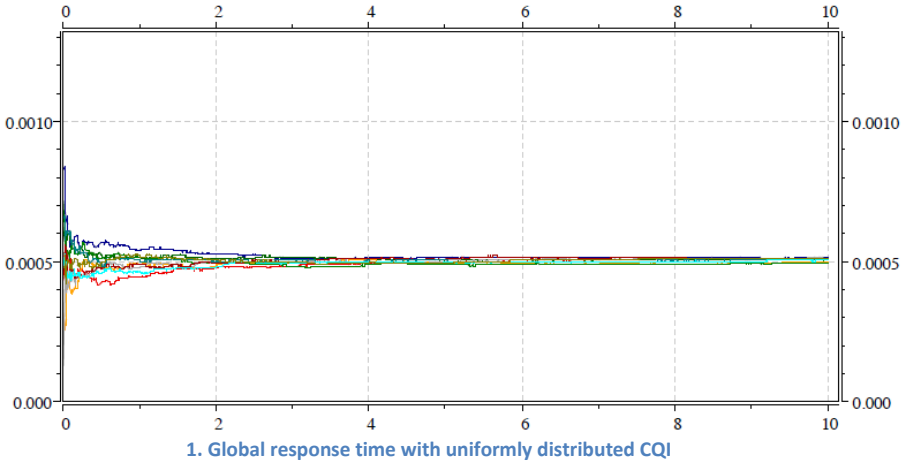
Study of the scenario with uniform CQIs

Estimating the warm-up period and the simulation duration

For each configuration 10 repetitions have been done to look at the average behavior of the system.

At the beginning the system was simulated for 10 seconds to estimate the warm-up period duration. We noticed that the throughput and the queues length converge earlier than the global response time and so we mostly took into account this last one to estimate the warm-up period.

The graph is related to one of the extreme situations from which we inferred the warm-up period, set to 3 seconds.



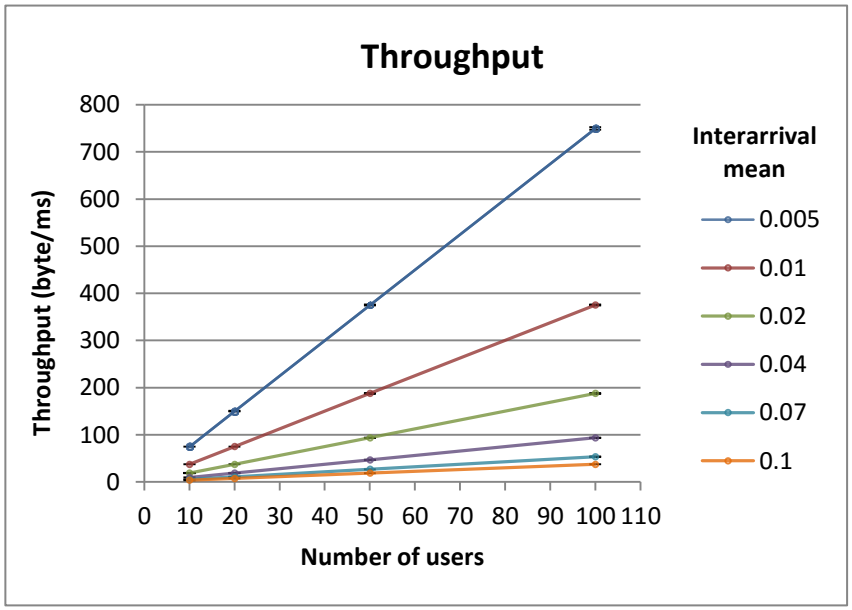
1. Global response time with uniformly distributed CQI

The simulation duration has been set to 8 seconds and it has been chosen so that meaningful data is collected for 5 seconds.

Analysis of data

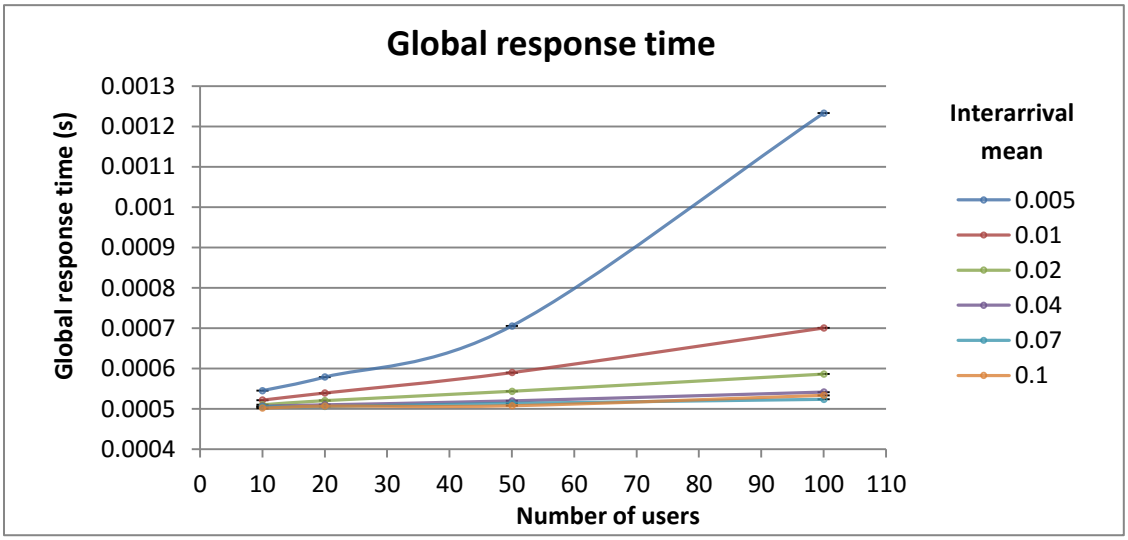
The data for the analysis have been obtained in the following way

- 30 repetitions have been done
- Mean values of the metrics have been extracted from every repetitions
- Mean value of that values have been computed in order to obtain one single value per metric for each configuration
- A 95% CI have been computed on that single value (drawn in the following graphs, but often not visible because too small)



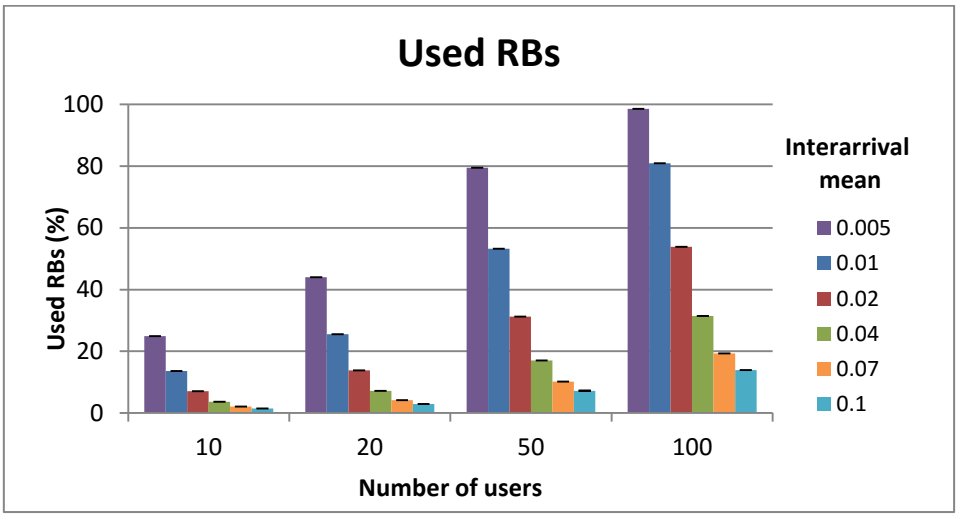
The graph shows how the throughput varies increasing the number of users for every interarrival mean. For few users (10 - 20) the throughput is not influenced by the different interarrival mean. Where there are more users in the system (50 - 100), with high means the throughput doesn't change so much while it is highly influenced by low means.

The mean 0.005 is the one that maximizes the throughput regardless the number of users; this happens because the packet arrives more frequently and there are more packets to send.

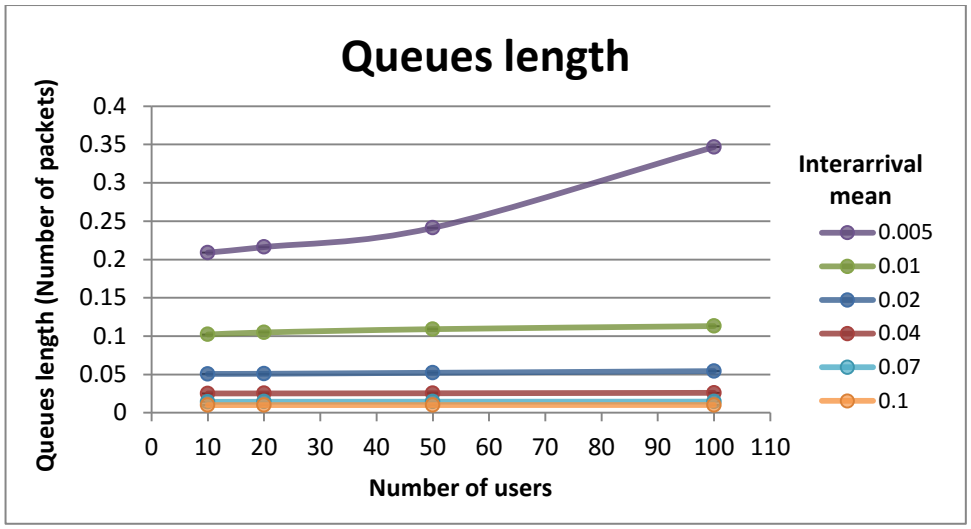


The graph shows how the global response time varies increasing the number of users for every interarrival mean. For few users (10 - 20) the global response time is more or less the same while with a greater number of users it gets worse when the interarrivals are frequent (0.01 and specially with 0.005).

Almost all values of the global response time are below 0.001: this means that on average packets remain on the antenna for less than one timeslot. The only configuration in which more than 1ms is needed is the one with 100 users and interarrival mean 0.005, in fact the percentage of used RBs is almost 100% and the system is near to the saturation.



With the same rate, the utilization is greater increasing the number of users and it increases decreasing the means, being equal the number of users.



On average the queues are almost emptied, there is a slightly greater number of packets for mean 0.005 and more users.

Mean interarrival values that minimize the global response time are the higher ones, 0.04, 0.07 and 0.1 because, as there are not much packets on the antenna, it manages to serve them quickly. However, the throughput is too low with them, so it is better to choose interarrival means like 0.01 and 0.02 so that there is a trade-off between the throughput and the global response time.

Study of the scenario with binomial CQI

Estimating the warm-up period and the simulation duration

We have done 10 repetitions for each configuration also for this scenario. Although we have simulated it for a long period, this time **the system never reaches stability**

- For low and medium interarrival means (from 0.005 to 0.04), the queues of some users with low CQIs grow infinitely.
- For high interarrival means (0.07 and 0.1) the global response time of the system never converges to a value; in this case, all users are able to take part to the system and, as the CQIs are sensibly different from different users, the global response time is too variable in different runs. Moreover, there are some queues that grow infinitely.

However we noticed that after a period of 12 seconds the throughput and the global response time for low and medium interarrival means converge to a value. In fact, in these cases the only active users are the ones with the highest CQIs whose queues are emptied. We studied the behavior of the system, focusing the analysis on them (**Analysis 1**).

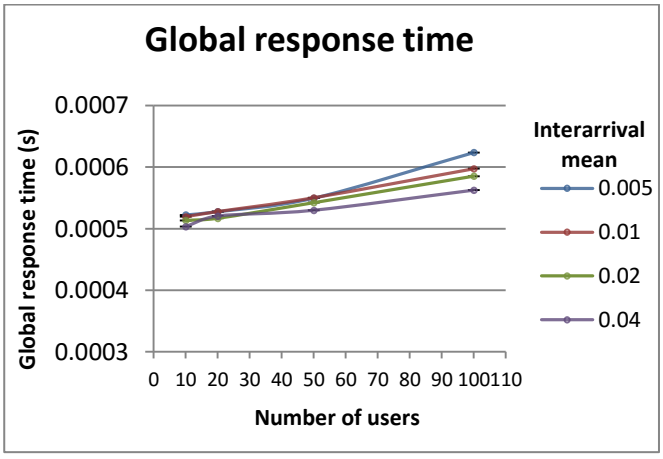
After these considerations, we decided to ignore a constraint and decrease the maximum length of packets generated to 20 bytes. In this way, the system reaches the stability after a warm-up period of 2 seconds, except for cases with low interarrival means (0.005 and 0.01), in which queues doesn't empty for 50 and 100 users. So we analyzed the configurations with interarrival mean of 0.02, 0.04, 0.07 and 0.1. (**Analysis 2**)

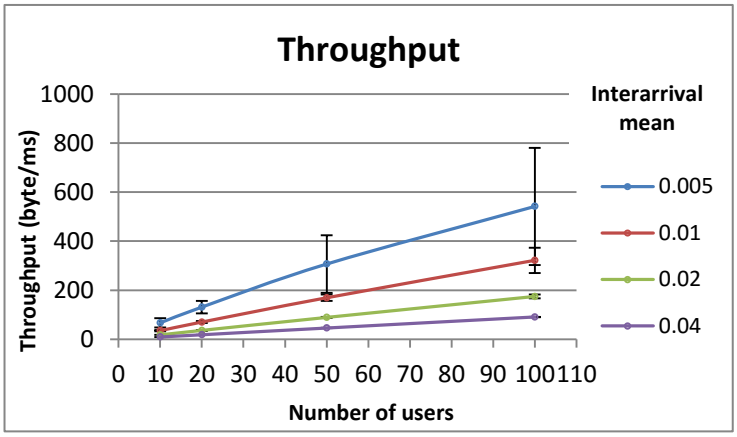
In both the analysis, we added a period of 5 seconds to the warm-up one to set the simulation duration, for the same reasons expressed for the uniform scenario.

The data for the analysis have been obtained with the same procedure adopted for the uniform scenario.

Analysis 1

The throughput and global response time is due to active users, the ones with the highest CQIs, and so the system's behavior is similar to the one in the uniform case.

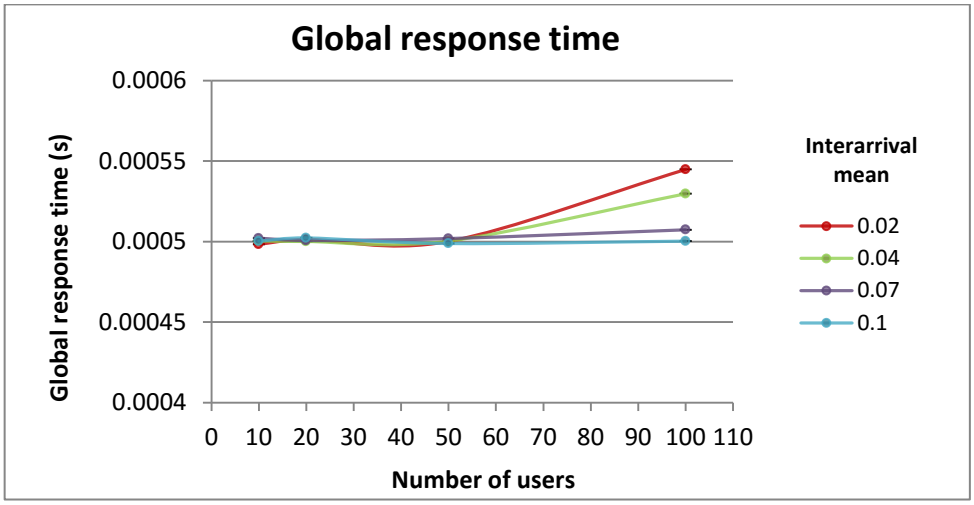




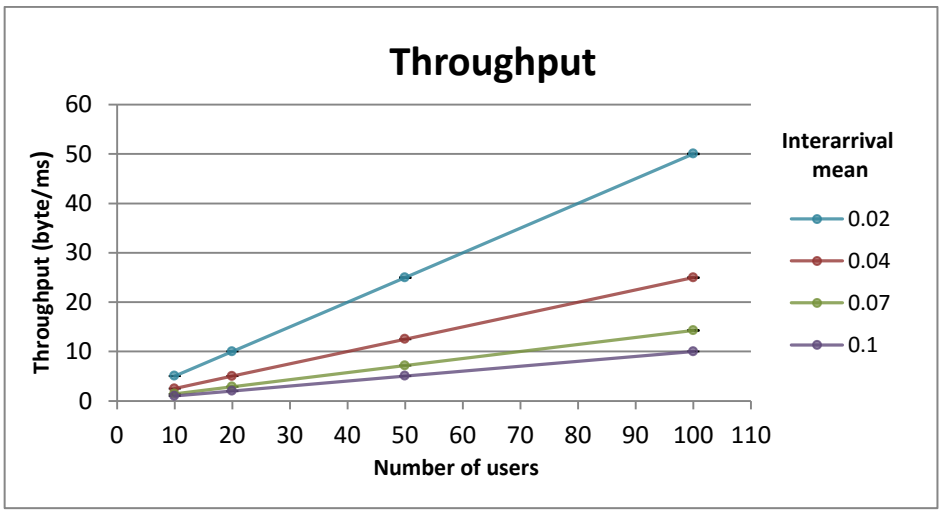
The global response time doesn't vary so much, there is only a slight difference for 100 users; however, the antenna never takes more than one timeslot to serve the users, on average. The differences on the throughput can be noticed on a high workload (a high number of users and the presence of many packets

in the system); though, with mean 0.005 we can't say that the throughput is always better than the case with mean 0.01, as the confidence intervals show.

Analysis 2

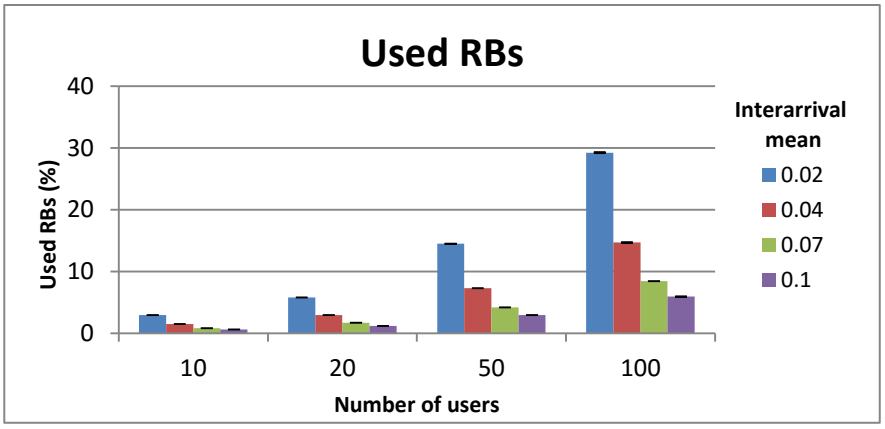


The global response time doesn't change much up to 50 users; for 100 users it is slightly larger decreasing the interarrival mean. However, this variation is minimal around the value 0.0005. This behavior was expected as packets are smaller and are placed more easily in the frame.

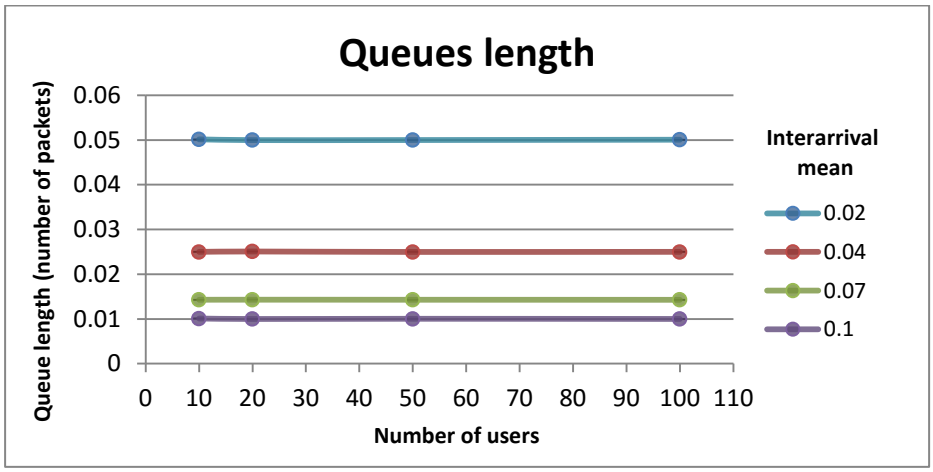


The graph shows how the throughput varies increasing the number of users for every interarrival mean. With 10 and 20 users the throughput doesn't change too much with different means. The best throughput that can be obtained is 50byte/ms with 100 users and an interarrival mean of 0.02; anyway, it is quite low but acceptable, considering the maximum packet size.

It is evident that the system has a low workload, in fact the maximum utilization of the RBs is around 30% for 100 users and low interarrival mean. In all the other configuration, a lot of RBs remain unused every timeslot.



The queues are empty for most of the time and their mean length remains constant increasing the number of users. This happens because the antenna manages to handle all the user data so fast as to leave the queues empty after composing the frame.



The best performance is obtained with interarrival mean of 0.02, as in this way the throughput and the utilization of RBs are maximized and the global response time is more or less constant for different interarrival means.

Conclusions

Our analysis confirmed our intuition about the behavior of the system in the two scenarios: the scenario with uniform CQIs is fair towards users, as the mean value of the CQI is the same for all the users; this not happens with the binomial CQIs, as the mean value for different users is sensibly different.

Although the binomial distribution better reflects the reality, it has worst performances with respect to the uniform distribution under the same conditions. The only way to let the system work is to lighten the workload.

We want to underline that the choice of an opportunistic policy on the antenna emphasize the inequalities that already exist among users when using CQIs distributed in a binomial way. A more “fair” policy might balance the situation.