



Norwegian University of
Science and Technology

Distributed interference and data buffer management for cognitive radio

Alberto Bayo Moya

Master of Science in Electronics

Submission date: June 2010

Supervisor: Torbjørn Ekman, IET

Co-supervisor: Kimmo Kansanen, IET

Problem Description

Resurs- og støykontroll innen kognitive nettverk basert på netverket sensing. Studenten skal evaluere algoritmer for system optimering med MATLAB.

Assignment given: 01. February 2010
Supervisor: Torbjørn Ekman, IET

Abstract

Future wireless communications are focused on develop new paradigms of spectrum reuse, i.e. cognitive radio, where radio frequencies can be taken into used if no other use is detected. In this way, these systems need mechanisms for interference control.

In this work we study the use of a distributed system based on cognitive radio. Scheduling transmitters and data rate optimization algorithms have been implemented in order to balance the secondary user transmissions and maximize the data rate transmission. Radio sensors are used to monitor the interference levels generated by the different transmitters. The system model has been implemented in MATLAB and analyzes its behaviour through several simulations.

Contents

1	Context and environment	3
2	Objectives of work	5
3	Distributed system description	6
3.1	Elements and basic parameters	6
3.1.1	The whole system	6
3.1.1.1	Plane Earth loss	7
3.1.1.2	Shadowing	8
3.1.1.3	Multipath fading	8
3.1.1.4	Overall system description	9
3.1.2	Primary system	11
3.1.3	Secondary system	12
3.1.4	Sensors	13
3.1.5	Fusion center	14
3.1.6	Silent & Busy periods	15
3.1.7	Interference queue	16
3.2	Algorithms	17
3.2.1	Admission protocol	17
3.2.2	Scheduling secondary user transmitters	18
3.2.3	Datarate optimization	20

<i>CONTENTS</i>	2
4 Results	22
4.1 Secondary users close to secondary receiver	22
4.1.1 Experiment 1	24
4.1.2 Experiment 2	29
4.1.3 Experiment 3	34
4.2 Secondary users further from secondary receiver.	39
4.2.1 Experiment 4	41
4.2.2 Experiment 5	46
4.2.3 Experiment 6	51
4.3 Best value constraint for intercell interference	56
4.3.1 Experiment 7	57
4.3.2 Experiment 8	58
5 Conclusions and discussion	60
A Matlab code	61
A.1 Cell structure function	62
A.2 Fixed losses function	62
A.3 Fast fading function	63
A.4 Arrival process function	63
A.5 Scheduling second transmitter function	64
A.6 Data rate optimization function	64
A.7 Data rate test function	65
A.8 Main program	65

Chapter 1

Context and environment

Today, it is more and more important to be in touch with anyone and in anywhere. For this reason, mobile communications and specially wireless communications have become much more essential during the last two decades.

In order to supply the increasing demand of this type of technology, more and more frequencies have taken up. Mobile communication companies have invested amounts of money in obtaining licenses. Most of them are focused on mobile communications and high speed wireless internet.

During the last years, smart phones and laptops have become quite popular in many countries. Furthermore, most of these dispositives have a wireless connection or at least, are potential consumers of these type of products. On the other hand, the radiofrequency spectrum is limited. Besides, mobile operators want to optimize their licenses in order to maximize their investments in frequency licenses.

For all these reasons, it is necessary to research and analyze new technologies in the wireless communication field. One of them is called *Cognitive radio*. This is the base of our project. *Cognitive radio* is a new concept that Joseph Mitola III introduced in 1998 for wireless communication, in which either a network or a wireless node changes its transmission or reception parameters to communicate efficiently, avoiding interference with licensed or unlicensed users [1].

The basic idea is to reuse frequency bands whose communication capacity is lower than mobile bands. Thereby, it is possible to implement systems that can transmit over these same frequency, in order to increase the data rate of different mobile systems based on mobile wireless, etc.

The main disadvantage of *Cognitive radio* is that these frequency bands are licensed ones. In other words, the secondary user do not know if any primary user in its same transmission range is listening or not. In that moment, the secondary one would interfere in the communication between the two primary users.

A valid solution is a sensor grid that protects the primary user from exceeds certain interference constraints. In this way, the sensors can monitor and limit the power level of the secondary users. However, a feedback between sensors and secondary transmitter is necessary. Basically, the sensor grid will establish if the secondary transmitter is able to transmit or not, because an On/Off power allocation is implemented.

Before a secondary user transmits, it is necessary to decide which user will do. To achieve this goal, a scheduling algorithm is implemented. This one is based on the channel state between transmitter and receiver from secondary system. Besides, interference queue and primary interference are also parameters.

Afterwards, a data rate optimization algorithm is executed in order to maximize the data rate between the users of secondary system. This procedure is implemented because channel state changes during a busy period in a cell. The secondary interference from the neighbour cell is unknown. So, it is necessary to develop an algorithm that can estimate that unknown interference and in addition, calculated an estimated data rate.

The basic aim of the work is to maximize the data rate between users in the secondary system and keep an acceptable interference level that lets the primary system works without problems.

The project is focused on four different things: the data queue at the secondary transmitter, and how its behaviour is when its size changes, the optimization data rate and its accuracy, the interference neighbour cell, this parameter attempts to balance the power level between cells, and finally, how the interference queue monitors the whole system.

All of these issues are explained in depth in the following chapters.

Chapter 2

Objectives of work

In this section I try to describe the main goals of this project. In the following lines it will be listed all of them.

- Maximize the data rate between the transmitter and the receiver of the secondary system
- The user data rates are maximized while fulfilling the interference constraints at the sensors.
- Discuss if the data rate estimation is close to the real data rate, calculated after the transmission.
- Test that all interference queues fulfil the constraints.
- Test average interference from neighbour cell, in order to know the specific weight of this parameter in the interference queue and its most suitable value.
- Analyze the relationship between the position of different secondary transmitters and its estimated data rate.

Chapter 3

Distributed system description

3.1 Elements and basic parameters

This part tries to describe the different units and elements that make up the whole system.

3.1.1 The whole system

The whole system is composed of four different types of elements: primary system, secondary system, fusion center and sensors, whose characteristics and main behaviours will be explain in depth in the following chapters.

As it was explained before, the basic aim of this project is that the whole system can work simultaneously. On one hand, there is a primary system which has a license for transmitting and obviously, their communications must be successful. In other words, the interference that comes from the secondary system must be lower than the interference constraint that the primary receivers have.

On the other hand, improving the efficiency in these frequency bands need to develope different strategies and researches that enhance the use of these frequencies. Due to most of the elements that belong to the whole system, this part is focused on describing the theoretical issues related to channel mobile propagation model.

As it was described in the previous project, the mobile channel characterization it is defined by three parameters: path loss, shadowing and multipath fading [2]. There are many models that attempt to estimate properly which is the path loss between two elements. In this project, plane earth loss model has been implemented.

3.1.1.1 Plane Earth loss

This model is an improvement compared to Free Space model, because it takes into account the height of the different users. I mean, the height of the base station is usually about 30 m. However, the one of the mobile user is about 1.5 m or 2 m, according to the human being's height.

The propagation take place via both a direct path between the antennas and a reflection from the ground. These two paths sum at the receiver with a phase difference related to the difference in length between the two paths. A simple way to analyse the situation is to make use of image theory.

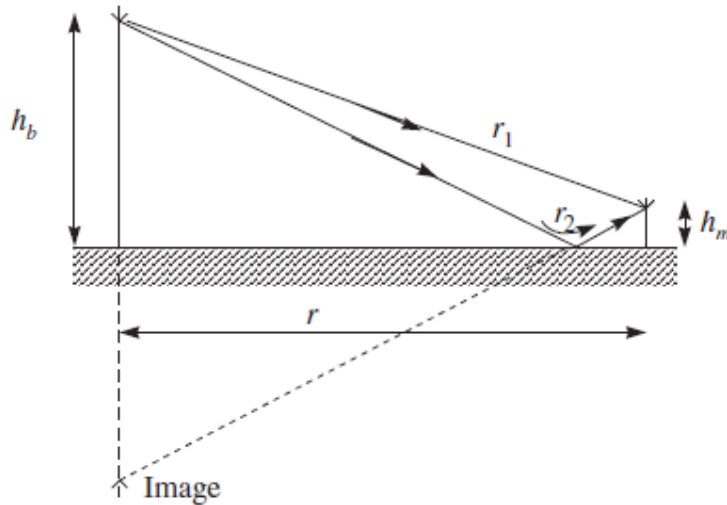


Figure 3.1: Physical situation for plane earth loss

The equation that it is used in the program is:

$$L = 40 \cdot \log_{10}(r) - 20 \cdot \log_{10}(h_m) - 20 \cdot \log_{10}(h_b) \quad (3.1)$$

where

- r is defined as the distance between transmitter and receiver in meters.
- h_m is the mobile height, typically between 1.5 m and 2 m.
- h_b is the base station height, typically between 20 m and 30 m.

The plane earth loss model is not the most accurate model of real-world propagation when taken in isolation. It only hold for long distances and for cases where the amplitude and phased of the reflected wave is very close th the idealised -1.

Besides, more information about different path loss models is available in [3, Ch:5.6].

3.1.1.2 Shadowing

This phenomenon is the second one which defines the characterization of a mobile channel. Large obstacles, like buildings, cars, different geographical features are the main causes of shadowing, because they cause variations of hundreds of wavelength. It must be denoted that shadowing is focused on characterizing big obstacles, and its value does not change if transmitter and receiver are static, in the same position.

The shadowing variations modify the original value of path loss that has been calculated. Different propagation studies conclude that long term fading follows a log-normal distribution. Besides, the distribution has an average of 0 dB and a standard deviation from 6 dB to 8 dB. [3, Ch:9]

Log-normal distribution random variables is generated following this equation, in linear scale:

$$S = e^{(x \cdot \sigma + \mu)} \quad (3.2)$$

where

- $X \sim N(\mu, \sigma^2)$

As it was written in the previous paragraph, further information about shadowing theory can be found in [3, Ch:9] and [4, Ch:2].

3.1.1.3 Multipath fading

This parameter can be defined like what occurs when the coherence time of the channel is small compared to the delay of the channel [3, Ch:10].

In other words, this phenomenon attempt to take into account all those variations that the main signal suffers when is divided in lots of different contributions, scattered and reflected from many obstacles.

This fading appears especially in urban centers, inside buildings, etc, where the original signal, because of the propagation, is separated in different signals. Line of sight is often blocked by obstacles, and a collection of several waves is all what a mobile antenna receives.

According to the various scenarios that can exist, several models of multipath fading have been developed. In this project, Rayleigh distribution has been implemented.

This model assumes that there is not line-of-sight amplitude. In this way, scattered and reflected waves are the principal contribution.

Furthermore, Rayleigh distribution is quite useful in heavily built-up city centers, where is usually the lack of line-of-sight.

Rayleigh distribution is used in this project, in order to generate random variables that follow this type of distribution. For this reason, it has been implemented a small function which generates variables based on [2, Ch:2.2.3.2]:

$$R = \sqrt{X^2 + Y^2} \quad (3.3)$$

where

- $X \sim N(\mu, \sigma^2)$
- $Y \sim N(\mu, \sigma^2)$

3.1.1.4 Overall system description

Once the theoretical background about the mobile channel propagation losses has been explained, it is important to illustrate how the whole system works. In this way, the following diagram will show a possible framework for a future implementation.

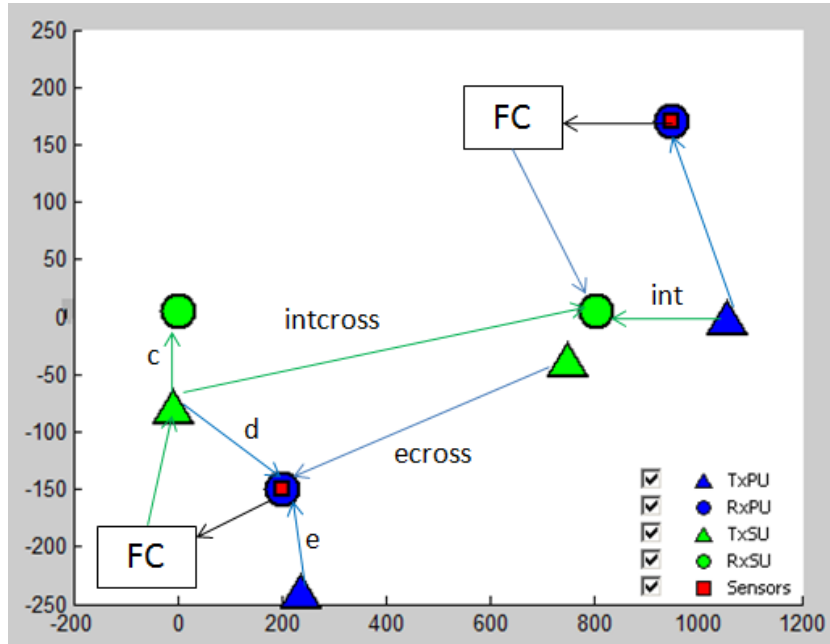


Figure 3.2: Whole system diagram

As it can be seen in the previous picture, the project is based on two separate cells. Each one has its own fusion center, in order to follow a distributed implementation.

The different arrows try to explain the different channel states implemented. It is necessary to denote that some channel states have been omitted in order not to duplicate and get confused.

Up to now, a complete description about the setting of the different elements and the channel states between them will be given.

Firstly, the channel states will be described. One of them is the channel $c(t)$, this channel is the path loss between any secondary transmitter and the secondary receiver of the same cell. The next channel state is $d(t)$, this is focused on the losses that exist in the distance between a secondary receiver and the associated sensor to primary receiver. The following channel is $e(t)$, this one shows the path loss among a primary transmitter and the sensor close to the primary receiver. The last channel state that just operates in one cell is $int(t)$, this channel is the path loss between a primary transmitter and the secondary receiver.

Besides, there are two more channel states that must be taken into account. These ones are $ecross(t)$ and $intcross(t)$. The first one shows the losses between any transmitter from one cell to the sensor close to the primary receiver of the other cell. The second one is focused on the losses that exist between any transmitters from one cell to the secondary receiver of the other cell.

Once the several channel states have been described, the next step is to describe how it works the main program.

The program calculates the distance between the secondary receivers of the two cells, from a file called *fixed_parameters.txt*.

Thereby, the program estimates the coverage radio of each cell. It is assumed that the position of the receivers, not only secondary but also primary, it is known. This is a natural assumption because it is supposed that the receivers are base stations.

Afterwards, the sensor structure is calculated. It is significant to denote that one of the sensors must be set close the primary receiver, in order to have a good estimation of the interference that this element receives.

Later, the rest of the users are set in the scenario according to the coverage radio calculated previously and a random process that is implemented by the function *create_cell.m*. This function establishes an array structure based on the two cells. Most of the parameters are implemented inside this type of format. The basic idea is that an overall parameter, called *cell*, is composed by different fields that represents the several elements and parameters that exist in a real cell.

The following diagram illustrates the main configuration of most parameters.

Field	Value
RxSU	[1,5]
TxPU	[197.39 -47.241;-181.34 353]
RxPU	[50;70]
TxSU	[37.804 69.102 -72.52;-52.704 30.854 -53.793]
Sensors	<2x6 double>
c	[0 0 0]
d	<3x6 double>
e	[0 0]
irt	[0 0]
ecross	[0 0 0 0]
irtcross	[0 0 0 0]
c0	[9.3512e-017;1.2848e-016;3.045e-018]
d0	<3x7 double>
e0	<2x6 double>
irt0	[4.8778e-019;5.4413e-019]
e0cross	[0 0 0 0]
irt0cross	[0 0 0 0]
Q	[0 0 0]
datarate	0
X	[5.0119e-018 5.0119e-018 5.0119e-018 5.0119e-018 5.0119e-018 5.0119e-018 5.0119e-018 5.0119e-020]
ropt	1
whoTx	0
whoTxOtherCell	0
ack	0

Figure 3.3: Cell structure

As the cell structure figure illustrates, this way of programming it is much easier because all the variables that depend on a cell are also implemented in a cell structure.

Not all the variables are included in this array structure. Nevertheless, all of them will be explained during the report, as soon as the project description requires its explanation.

On the other hand it is important to explain one variable, at least. This one is the noise power. Due to work with power level, the noise power is a constant. Its value is -120 dBm over 200 kHz. However, working with density power causes that the final value is -173 dBm/Hz.

3.1.2 Primary system

This is usually referred to the GSM system, but the principles of the primary system are: timeslots and carriers for separate users.

The reason why it is called “*primary system*” is because this one has an absolute priority on the transmission and working process over the “*secondary system*”.

In this section, it will be also described the different assumptions about transmitters and receivers that have been taken into account. In this way, the reader can have an overall view about how this system affects the secondary one.

As it was described previously, the primary system is the original one. In other words, this system is usually a 2G mobile system, like GSM or D-AMPS. The mobile communication companies have invested amounts of money in obtaining one of the few licenses that the different governments put out to tender.

For this reason, it is quite important that the primary system can work without problem, I mean; the interference that can introduce the secondary system must be lower than the one that the primary system can support. In this project, some assumptions have been established in order to simplify the extension and difficulty of the report:

- It is assumed that there will be just one receiver per cell: it is supposed that the receiver is the base station. So, it is a valid assumption to consider just a receiver per cell.
- The primary receiver position is known: for the same reason that in the previous assumption, it is natural to suppose that the position of the primary receiver is fixed and known a priori.
- The transmitted power is constant, not only its power level, but also during the whole experiment period: this assumption is quite important. First of all, the use of a constant power level and also the same through the time, makes easier most of the calculations and equations related to almost all of the algorithms implemented.
- The primary transmitters are set randomly: in this case, there is no constraint about the number of primary transmitters. In addition to this, their positions in the scenario follow a random distribution. It is assumed that they are users, so at first, it is impossible to know where they can be located.

3.1.3 Secondary system

The secondary system is the most important one. This system can be managed and controlled by the fusion center. For this reason, it is quite helpful to describe it as accurate as we are able to.

On the other hand, there are a lot of similarities between the primary and the secondary system. This is a usual situation and some assumptions come from the primary system to make easier the second one.

The main purpose of this system is to work at the same time with the primary one. In order to achieve that purpose a power control must be fixed over the different secondary transmitters.

One of the main goals of this project is to check if this control works and if it is enough. It exists a compromise between the data rate maximization and the interference level over the primary system.

This is the base of the project. The main objective is that this system can work with the primary one together. The system is defined, based on several assumptions, as follows:

- The receiver is set in the middle of the cell: the coverage radio is based on the distance between the different secondary receivers. Besides, the function that sets the rest of the elements in the scenario uses this position as a point reference.
- The receiver position is known: as it was described in the previous chapter, it is assumed that the receiver is a base station. Then, its position is known and fixed.
- The transmitters are set randomly: in the same way that the primary transmitters are set in the scenario, the secondary ones are supposed to be users. So a random distribution over the scenario is the most suitable one.
- An On/Off power allocation is defined: thereby, a secondary user transmits with the maximum power level when the fusion center let it and when the interference level is not exceeded. If one of the previous conditions is not achieve, no power is transmitted.
- A TDMA-FDD communication system is defined between transmitters and receivers: this assumption avoids much interference between different elements. Besides, it is a general and extended implementation, like GSM used by primary system.
- Data buffers are implemented in the different transmitters: most of the transmitters will have to wait until they are allowed to transmit. Besides, it will be analyzed the behaviour of the secondary system when the buffer size changes.

3.1.4 Sensors

These elements are focused on measuring the interference level that generates the secondary transmitters. The different sensors are deployed in a grid.

One of the main problems that exist in this project is the way of monitoring and knowing the interference level that the secondary transmitters produce in the whole system. For this reason, a grid of sensors has been implemented.

The different sensors are set in the scenario following a ring, whose centre is the secondary receiver and whose radio is usually a half of cell radio. Within this one, the sensors are distributed throughout the ring, and the distance between two consecutives sensors is always the same.

However, it is important to take into account these assumptions and notes about the sensors:

- One sensor is set close enough to the primary receiver: In this way, it is assumed that what the sensor receives is the same that what primary receiver receives. The sensor grid measures power levels and afterwards, sends those levels to the fusion center.
- Sensors measure power levels: the different elements just measure power, so they just send the measurement to the fusion center after getting it. In other words, no signal process is made by the sensors.

Furthermore, it is important to denote that the sensors are not able to distinguish between the different powers transmitted at the same time. In other words, if a primary and a secondary user are transmitting at the same time, the sensor cannot tell the difference between them.

For this reason, a busy and silent period has been implemented. This issue has a specific chapter where all these questions will be clarified.

3.1.5 Fusion center

The fusion center can be described as the core of the system. This element is the place where all signals come from the different sensors and are processed. Besides this, it allocates the secondary user that is going to transmit. Mainly, these reasons are the ones that define the fusion center as “core” of the whole system. Busy and silent period (BP and SP), are also fixed by this element.

But the detailed treatment is outside the scope of this work. So, we will just establish some general rules and protocols about how the element could work with the rest of the units of the system.

As it was described in the brief description, the intelligence of the overall system lies in the fusion center. As the project is developed in a distributed way, it is important to note that there is one fusion center per cell, instead of just one for the whole system.

In the previous chapter, it was explained that the measurements of different sensors were sent to the fusion center. This one saves them in order to process and run different algorithms.

The basic aim of this project consists of analyzing the suitable behaviour of the secondary system, estimating the most suitable secondary transmitter and its optimal data rate, avoiding high interference level over primary system. For this reason, most of the communications between fusion center and the rest of the elements are not the purpose of this project.

Nevertheless, on order to show that there is an overall idea about how the whole system works, a general description of how the fusion center could work and communicate with the rest of the units will be explained.

Otherwise, before listing the several assumptions or guidelines, it is necessary to know the information that the fusion center receives and obtain and on the other

hand, the information that fusion center needs to run the different algorithms that have been implemented.

As it will be explained in further chapters, there are three significant algorithms implemented. Two of them, scheduling secondary transmitters and data rate optimization, should be executed by the fusion center.

The first one relies on: data queue from the suitable transmitter, channel state $c(t)$, estimated channel state $\hat{d}(t)$, interference queue and interference power from primary users and neighbour cell. The three last parameters are known by the fusion center, but the other two remaining ones are unknown, so they should be sent by the relevant secondary transmitter.

The second algorithm, data rate optimization, depends on: interference power from primary transmitters, average losses between the secondary receiver of each cell f and the channel state $c(t)$. In this case, the last two parameters are unknown, and just one parameter is known by the fusion center. Maybe, the best option is sent the primary transmitter interference to the suitable transmitter, and it calculates the optimal data rate.

Besides this, the signal quantification and modulation should be studied in further researches.

3.1.6 Silent & Busy periods

In this part of the report we will try to explain how the silent and busy periods work, and also the small relative shift that exists between the periods of different cells. Measuring the interference isolated from the signal, is one of the main reasons why there is a small gap among the period of the cells. Different graphics and diagrams will make easier for the reader to understand the running of this part of the system. Also a brief generalization about the silent periods in a larger system will be given.

As it was explained in the sensor's chapter, these elements cannot distinguish between power levels. Basically, the sensors just measure a power level, even if that power level is a sum of various contributions coming from many transmitters.

Because of this situation, it is not possible to determinate the interference level that exists in the environment. It must be note that secondary and primary users are transmitting at the same time and from different cells.

This new technique alternates busy and silent period of secondary transmitters. Not only the alternation between silent and busy period in a cell is important, but also the existing relative movement in time between the two cells.

For this reason, the following figure shows a time diagram of the two cell periods.

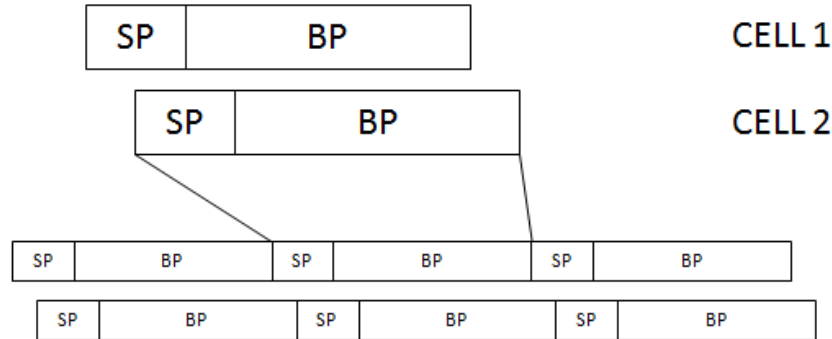


Figure 3.4: Cell periods

This type of configuration is the most suitable one. In this way, it is possible to measure two types of power level during a silent period. This characteristic is quite important because the main program implements algorithms that need these two kinds of power level. For instance, scheduling secondary users and interference queue need a power level when the neighbour cell is also in a silent period. On the other hand, the data rate optimization algorithm needs to know the power level when the neighbour cell is in a busy period.

Nevertheless, this configuration is not the perfect one. The main disadvantage is suffered by *cell 1*. This one needs to know the power level in the busy period of *cell 2*, in order to estimate the data rate. However, this power level is related to the previous busy period. So, the estimated data rate will be based on a busy period that does not exist when the cell 1 will transmit.

3.1.7 Interference queue

The interference queue has the main purpose of measuring the level of interference that exists in the environment. A net of sensors is deployed over the two cells in order to measure the interference that produces the secondary system. Another weight has been included in the interference queue: the neighbour cell interference. This weight will be also explained deeply in this section, but the basic idea is to balance the interference among the cells. The significant influence that these parameters have over the scheduling algorithm will be taken into account as well.

As it was described previously, the interference queue attempts to control the interference that the secondary system introduces in the environment.

One of the main purposes is to stabilize the interference queue. This one is based on the following equation from [6] and [5]:

$$X(t+1) = [X(t) - X_{av}]^+ + P(t) \cdot d(t) \quad (3.4)$$

The following interference queue value is updated every iteration by each sensor. As it can be seen in the equation, the queue depends on the transmit power from secondary transmitter $P(t)$, channel state $d(t)$, that represents the path loss between secondary user and the sensor allocated to primary receiver, and average interference X_{av} , that shows the amount of interference that the primary receiver is able to tolerate.

The average interference is one of the key parameters. The larger its value is, the higher interference is allowed in the whole system. Consequently, a higher data rate is obtained and the system works better.

Furthermore, X_{min} parameter attempts to limit the interference level. If the interference queue is 0, the values that determine the most suitable secondary transmitter would be 0. That would mean no one of the users can transmit, when the real situation is the opposite one.

The last weight in the interference queue vector is the neighbour interference or intercell interference. This value attempts to control the interference that one cell receives from the other one.

The equation of intercell queue is:

$$X_{NC}(t+1) = [X(t) - X_{av}^{nc}]^+ + P(t) \cdot ecross(t) \quad (3.5)$$

The equation has a lot of similarities to (3.4). The only difference is the channel state. In this case, $ecross(t)$ represents the path loss between a secondary user from one cell to the secondary receiver of the other cell.

3.2 Algorithms

The basic aim is that the reader has a detailed view about the several algorithms implemented and the different inputs used.

3.2.1 Admission protocol

This section will show and describe the algorithm which has to control the arrival of packets to the different buffers of each secondary transmitter. This algorithm also has to drop packets if the buffer length is overloaded.

The arrival of different transmission packets to the secondary transmitter must be monitored. For that reason, it is necessary to implement an algorithm that fixes the buffer size and drives the different packets to the transmitters. In this

way, the function is based on several inputs that are going to be explained in the following lines. El first variable is *cell*: this parameter is a structure array one, and one of its various fields is the amount of data that each secondary transmitter has. So, it is one of the needs for the right running of the function.

The second one is *prob*: this parameter denotes the probability that a packet of bits arrives to anyone of the secondary transmitters. The arrival of packet has been defined as a Bernoulli process.

$$Ap(t) = \begin{cases} n & k < prob \\ 0 & k > prob \end{cases} \quad (3.6)$$

where

- n is the number of bits.
- $k \in [0, 1]$ is a random number with uniform distribution.

The third one is *bits*: this parameter just shows the number of bits that is going to be added to the relevant transmitter's queue.

The last input is *controlqueue*: this parameter fixes the buffer size. The upper constraint is calculated in bits, so a multiplication of controlqueue and bits is made in order to know the buffer size that is allowed during a certain experiment.

Once all of the inputs have been described in depth, the next step is to show the data queue behavior as it is described in [5] and [6].

$$Q(t+1) = [Q(t) - r_s(t)]^+ + Ap(t) \quad (3.7)$$

where

- $Q(t)$ is the data queue length in the previous instant.
- $r_s(t)$ is the amount of succesfully transmitted data during time t .

Similar to the interference queue, the data queue is updated every instant. It depends on the estimated data rate and the arrival process.

3.2.2 Scheduling secondary user transmitters

This algorithm is one of the most important ones. It has a basic purpose: decide which secondary transmitter is the most suitable one to send information to the secondary receiver. To achieve this purpose, the algorithm implements some equations in order to obtain a metric. The transmitter which has the best metric is the one that transmits. During this section it will be also explained

the most relevant variables and parameters, as well as how these ones may affect the whole metric.

Once the several data queues have received the packets, it is important to decide which secondary transmitter is going to send the information. Up to now, this second algorithm will establish the most suitable one. This decision is based on objective parameters like: channel state $c(t)$ between secondary transmitter and secondary receiver, the interference queue, channel state $d(t)$ between secondary transmitter and the sensor close to primary receiver, and interference power coming from primary users and secondary user from neighbour cell.

The equations that decide which is the most suitable secondary transmitter have been obtained from [6]. This optimization power is quite close to the model that it has been implemented in this project. The formula is:

$$p = \arg \max_P \{q^T r - \bar{y}^T p\} \quad (3.8)$$

As it can be seen in the previous formula. The secondary power is higher if the data queue q or the data rate $r(t)$ is higher. On the other hand, the parameter \bar{y} can be denoted as $\bar{y} = \bar{D}^T x$, where $d(t)$ is the channel state between secondary transmitter and the sensor close to the primary user and x is the interference queue. The maximum power will decrease if \bar{y} or p , allocated power in the previous instant, increases.

If the secondary users utilize orthogonal channel access between themselves, the data rate function of each one can be estimated with the Shannon capacity, defined by the next equation.

$$r_k(p_k) = \log_2 \left(1 + \frac{p_k c_k}{N_0 + I_k} \right) \quad (3.9)$$

where

- c_k is the channel state between the secondary transmitter and receiver.
- N_0 is the power level of AWGN.
- I_k is interference from other transmitter, like primary or secondary ones.

Furhtermore, an On/Off power allocation has been implemented, so the maximum power is always tansmitted. In order to obtain the best metric equation, p_k can be substituted by:

$$p_k = 1 \quad (3.10)$$

In this way, the most suitable secondary transmitter equation will be the one that obtains the best metric from (3.8). So, if (3.9) and (3.10) are replaced in (3.8).

$$J_k = q_k \cdot \log_2\left(1 + \frac{c_k}{N_0 + I_k}\right) - \bar{D}^T \cdot x \quad (3.11)$$

In the previous equation, the data rate parameter has been replaced by Shannon capacity equation.

3.2.3 Datarate optimization

This is the last of the implemented algorithms. Its mathematical complexity makes it much more important and that is why it should be quite interesting to describe the whole process.

The implementation tries to estimate which is the most suitable data rate in a radio link. The reason why it is necessary to estimate the data rate is because the secondary interference that comes from the neighbour cell is unknown.

This situation causes that the secondary interference must be defined as an exponential distribution. All the development and assumptions that have been taken into account in order to work the data rate optimization will be also explained.

The next step in the secondary process transmission is to know the data rate that the secondary user is going to transmit with. This data rate depends on several parameters that will be described in the following lines.

First of all, it is important to realize that the data rate calculated is an estimate. This situation happens because the channel state changes during the busy period related to the secondary interference that comes from the neighbour cell. As it was established in previous chapters, the primary power is constant in time and in its power level.

In the same way, it is assumed that the channel state between all the primary users and the secondary receiver is known in order to make easier the computation and simplify the problem. Otherwise, the secondary interference coming from the neighbour cell is assumed to be unknown. This point of view is closer to the real world, and similar to what happens in the system. For this reason, it is also assumed that the secondary interference which comes from the neighbour cell follows an exponential distribution.

$$P_{I_{SV}}(z) = \frac{1}{f} \cdot e^{-\frac{z}{f}} \quad (3.12)$$

where

- $P_{I_{SV}}(z)$ is defined as the probability that a level of secondary interference, defined by z , was achieved.

- \bar{f} is defined as the average losses, path loss and shadowing, that exists between the secondary receivers from different cells.

It must also be said that the outage probability P_{out} , is defined as the probability that the secondary transmission fails. In other words, the instantaneous *SINR* is lower than the threshold *SINR*.

$$P_{out} = P(\gamma(z) < \gamma_{th}) \quad (3.13)$$

where

- $\gamma(z)$ is the instantaneous SINR with a secondary interference denoted by z .
- $\gamma_{th} = 2^r - 1$ the threshold SINR is calculated based on the estimated data rate.

Furthermore, γ_{th} can be described as.

$$\gamma_{th} = \frac{p_k C_k}{z_{th} + N_0 + I_{pu}} \quad (3.14)$$

If the outage probability is expressed in interference terms and z_{th} is replaced.

$$P_{out} = P\left(z > \frac{p_k C_k}{2^r - 1} - N_0 - I_{pu}\right) \quad (3.15)$$

Finally, the outage probability can be shown in the following equation.

$$P_{out} = 1 - P\left(z < \frac{p_k C_k}{2^r - 1} - N_0 - I_{pu}\right) = e^{-\frac{\left(\frac{p_k C_k}{2^r - 1} - N_0 - I_{pu}\right)}{\bar{f}}} \quad (3.16)$$

Hence, the throughput is defined as.

$$Th = r \cdot (1 - P_{out}) = r \cdot \left(1 - e^{-\frac{\left(\frac{p_k C_k}{2^r - 1} - N_0 - I_{pu}\right)}{\bar{f}}}\right) \quad (3.17)$$

This is the equation that it must be optimized.

Chapter 4

Results

This section is focused on illustrating the different experiments that we have performed during this master degree thesis. This experiments will be explained objectively the results that it has been obtained. Graphics and diagrams will also be added to make easier the basic behavior of the system.

The following experiments are divided in three large groups, according to the way of analyzing the whole system.

4.1 Secondary users close to secondary receiver

This part is focused on how the system works when the secondary users are near the secondary receiver.

This implementation has been chosen because one of the most significant constraints is the interference that comes from the primary system.

Thereby, in order to let the secondary system works, it is necessary to establish a medium distance between the primary users and the secondary receiver.

The following picture illustrates the first scenario that has been considered.

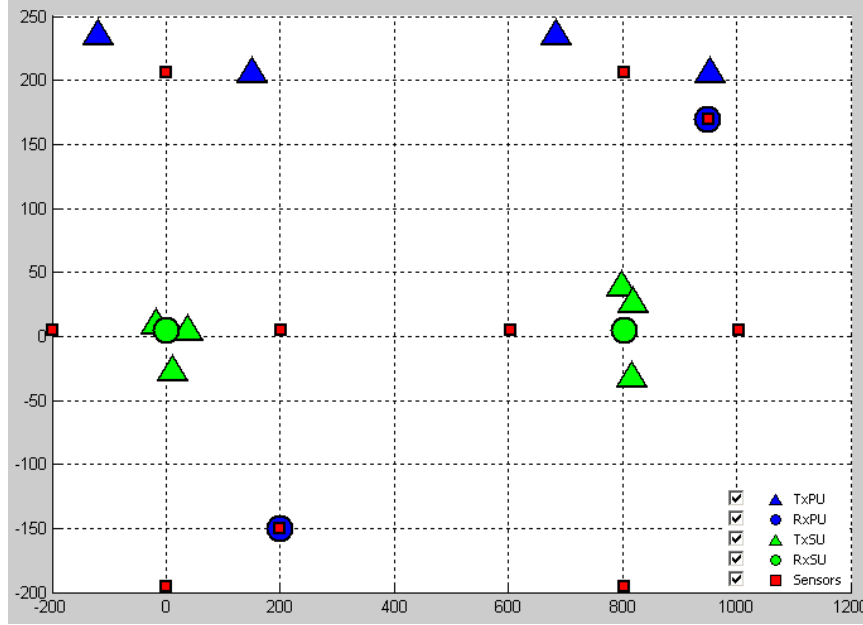


Figure 4.1: First scenario implemented

As it can be seen in the figure, the secondary users are deployed close to the secondary user. Actually, the coverage radio cell used as a parameter is $r/10$, in the function that sets the positions. Where r is defined as the normal coverage radio cell.

Furthermore, secondary system is absolutely fixed. Transmitters and receivers are set far away from the secondary receiver in order to avoid a higher primary interference over the secondary system.

During the following experiments all these parameters will be the same:

- Three secondary transmitters per cell, whose positions are fixed.
- Two primary transmitters per cell, whose positions are fixed.
- Five sensors per cell, whose positions are also fixed.
- Receiver Height, $h_b=30$ m.
- Transmitter height, $h_m=1.5$ m.
- Number of bits per packet, $bpp=2$ bits.
- Transmission density power, $P_{tx}=1e-5$ W/Hz.
- Average interference in the own cell, $X_{av}=5.011e-18$ W/Hz (equal to -80 dBm over a bandwidth of 200 kHz)

- Neighbour average interference, $X_{av}^{nc}=5.011e-21$ W/Hz (equal to -110 dBm over a bandwidth of 200 kHz)
- Number of iterations per experiment: 10000.

On the other hand, there are still two parameters that have not been defined and they will change in order to know how the system works.

These parameters are: packet arrival probability and size of data buffers.

4.1.1 Experiment 1

In this experiment the rest of parameters have the following values:

- Packet arrival probability, $p=0.1$.
- Data buffer size, $DBS=20$ bits.

Further, a set of diagrams and graphics will explain how the secondary system works and if it fulfills the different constraints. The first graphic shows the value, for both cells, of SINR threshold, calculated by the data rate optimization algorithm, and the real SINR.

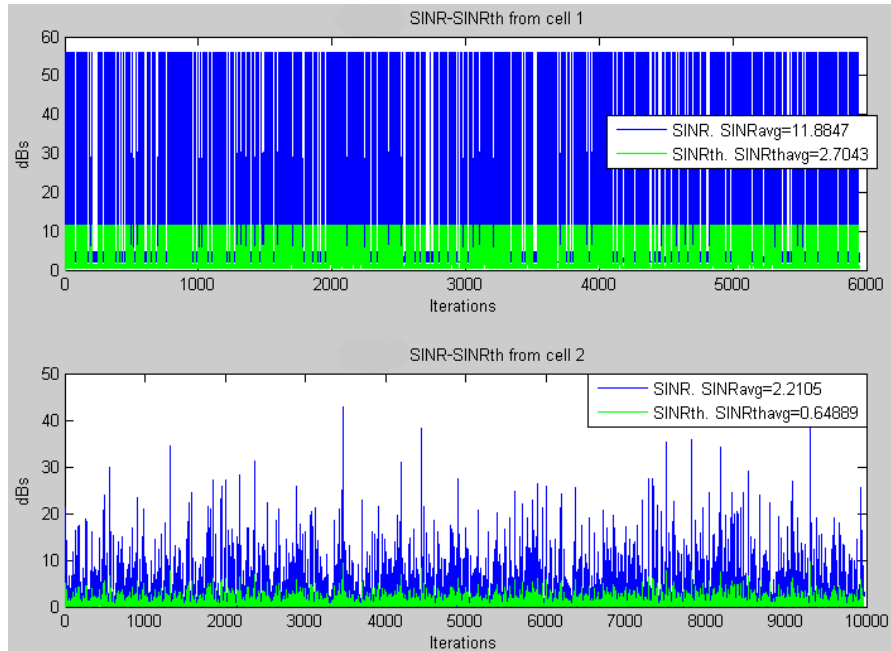


Figure 4.2: SINR and SINRth from cell 1 and cell 2

This picture shows the instantaneous SINR and SINR_{th} during 10000 iterations in the two cells.

As it can be seen in the upper part of the figure, the SINR_{th} related to cell 1 is higher than the one related to cell 2. It means that the secondary transmitters from cell 1 are able to transmit more data than the ones from cell 2.

On the other hand, it can be noticed that the data rate optimization algorithm is close to the real value in the second cell.

Although the SINR values seem to be quite high, in the range of 50 dBm for the first graphic and 30 dB for the second one, the average value of them is quite lower. In the first case the average SINR is 11.88 dB and in the second case is 2.21 dB.

As it was described before, the relation between the SINR and SINR_{th} in each cell is different. For cell 1 the relation is 4.39, and for cell 2 is 3.41. These relations give an idea of the accuracy of the data rate optimization algorithm.

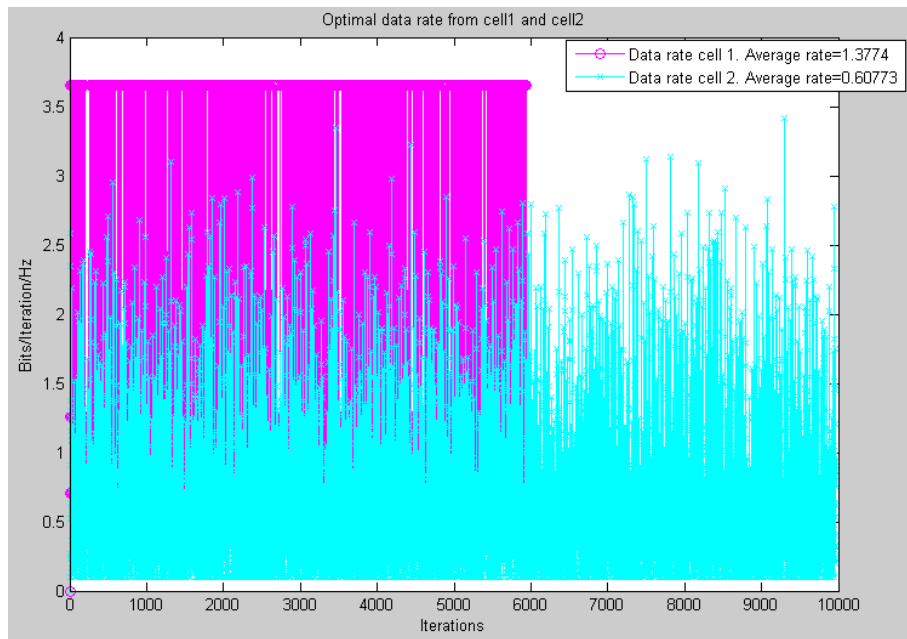


Figure 4.3: Optimal data rate from cell 1 and cell 2

This picture describes the different data rates that the secondary systems from each cell achieve.

As it was explained in the figure (4.3), the first cell has, at least one transmitter whose channel state is better than the rest ones. For this reason, the average data rate is higher than in the second cell.

Otherwise, the second cell is able to transmit data during much more iterations than the first one. So, the second cell has a higher efficiency than the first one.

If we compare the average data rate of each cell, with the maximum data rate that can be obtained from figure (4.3), some conclusions can be clarified.

For the first cell, the average data rate is 1.37 bits/iteration/Hz. The maximum data rate, taken from the SINR=11.88, is 3.68 bits/iteration/Hz. So, a difference of 2.31 bits/iteration/Hz exists between the estimated data rate and the real one.

For the second cell, the value of the average data rate is 0.607 bits/iteration/Hz, and the average value of SINR from figure (4.2) is 2.21 dB, so the average data rate is 1.68 bits/iteration/Hz. Thereby, the gap that exists between values is 1.08 bits/iteration/Hz.

After these two figures, it is possible to say that the first cell is able to transmit more data because one of the channel states between transmitters and receiver is better. On the other hand, the second cell has a higher efficiency in the data rate optimization algorithm. However, it transmits lower than the first one.

The following picture shows the interference level of the two cells.

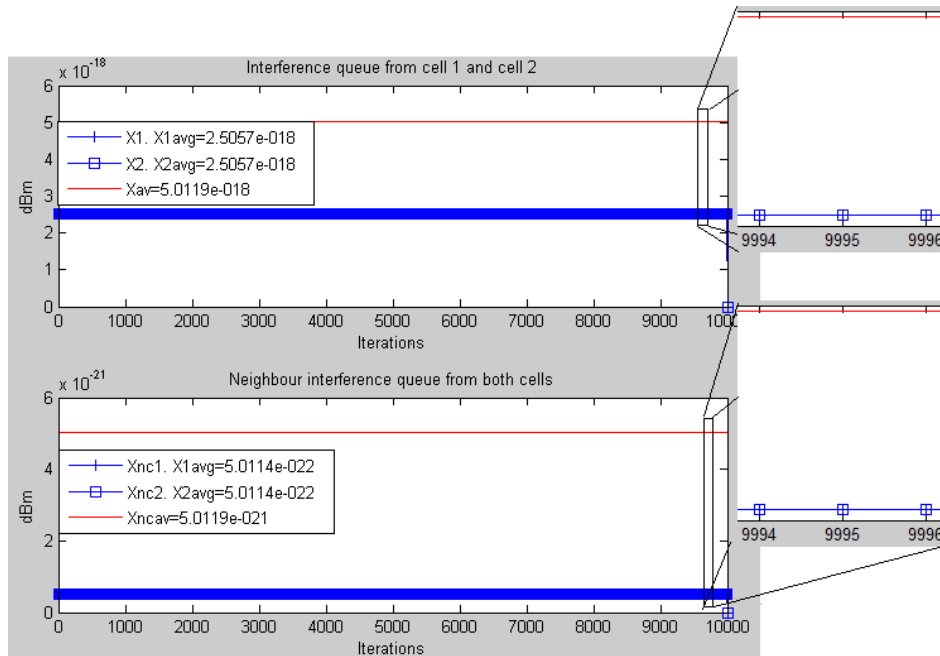


Figure 4.4: Interference queues from cell 1 and cell 2

The previous picture describes the interference level for both cells. The graphic is divided in two smaller ones.

The upper one shows the average interference level, in red, that each primary receiver is able to support. This value is -80 dBm over a bandwidth of 200 kHz, so it is considered a suitable value.

Besides, two blue lines determinate the instantaneous interference level that has the interference queue. As it can be seen, the interference level is always below the average one. It means that the primary system works properly. In other words, the secondary system does not affect the normal running of the primary one.

Actually, interference values are so low that the X_{\min} constraint is the value that appears in this graphic. It must be considered that the interference value is a sum of all the interference values that come from each sensor. According to this, the interference value is $5 * X_{\min}$.

The bottom figure illustrates the neighbour interference queue. Its behaviour is quite close to the interference queue. In this case the average neighbour interference is -110dBm over 200 kHz. So the value is quite conservative, and the interference level is close AWG noise level.

As it can be seen, the interference queue always has a lower value than the average one, so it does not affect the primary system.

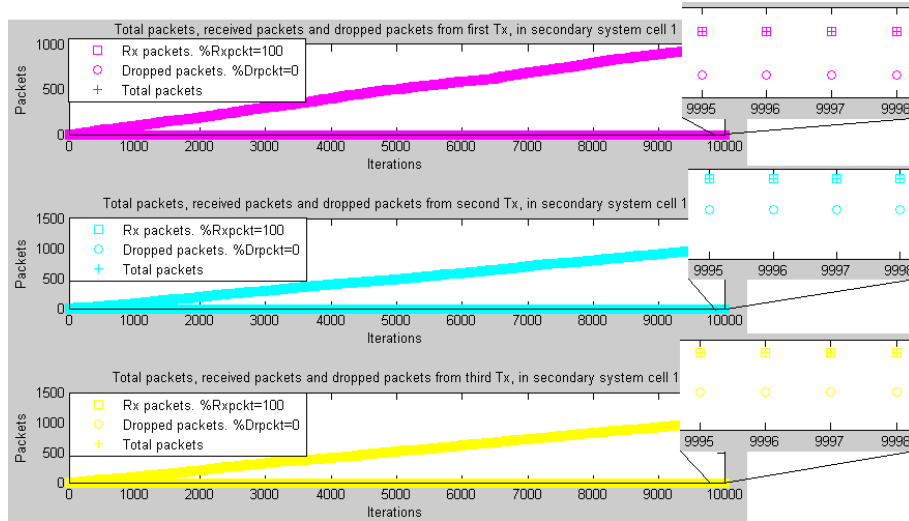


Figure 4.5: Received and dropped packets from cell 1

This picture shows the percentage of packets that has been received or dropped at each data queue of each secondary transmitter in cell 1.

As it can be seen in the previous picture, this configuration is most suitable one. All the secondary transmitters from cell 1 are able to receive all the packets and no one of them are dropped.

The overall average data rate for the cell is much higher than the average packet arrival, that is 0.3 bits/iteration/Hz. Noone of the packets are dropped.

This configuration is quite conservative because the packet arrival probability can be increased.

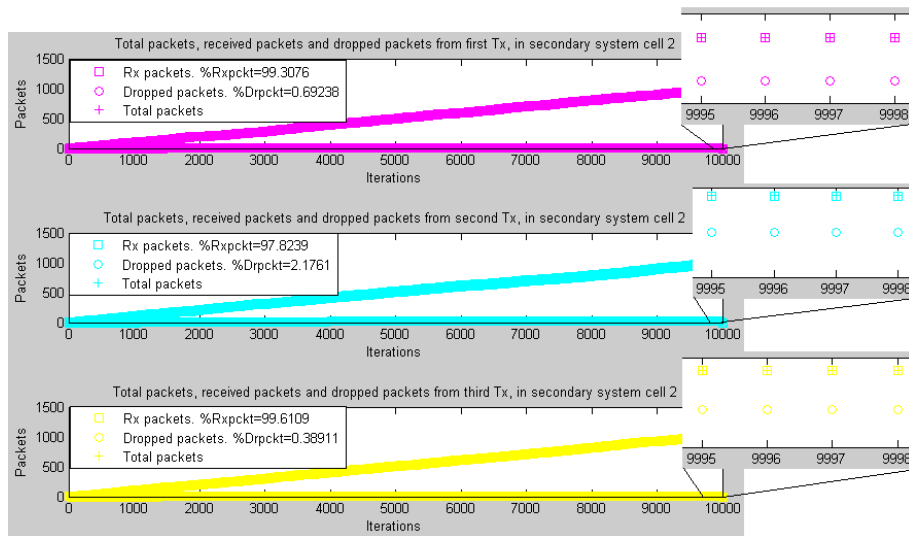


Figure 4.6: Received and dropped packets from cell 2

This picture has the same meaning like the previous one, but in this case, is focused on the second cell.

The main behaviour and data is almost the same, and as it was explained before the higher overall data rate makes almost impossible that there are dropped packets.

In this case, there is a small amount of packets that have been dropped. This amount is quite small and in the first and in the third secondary transmitter is less than 1%. In the case of the second transmitter, the percentage is a bit more than 2%.

The main cause of this situation is the average data rate, which is lower in the second cell than in the first cell.

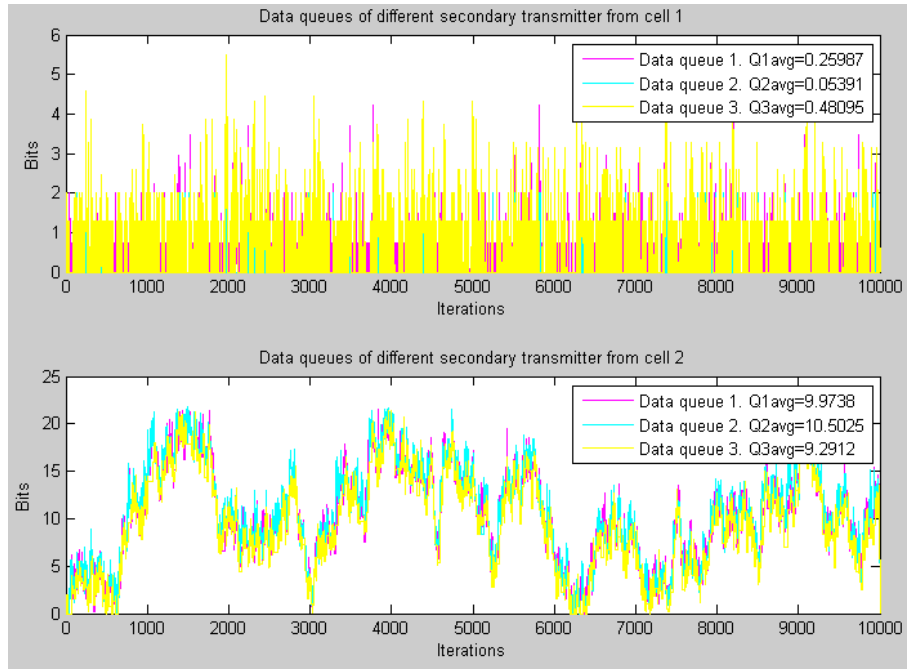


Figure 4.7: Data queues from cell 1 and cell 2

This figure describes the behaviour of the data queues. The different colours are set according to its respective secondary transmitter, like in figures (4.5) and (4.6).

As it was described before, the data queue from the cell 1 is empty many times. This is the reason why in figure (4.3), the data rate is transmitted less times than in cell 2.

In this way, it can be seen that the second transmitter from cell 1 has a quite good channel state between secondary transmitter, because its average queue is much smaller than the other transmitters.

On the other hand, the second cell has worse levels of average data queue. Because of its data rate, the queue is larger and it means worse channel states. However, the different secondary transmitter has almost the same channel states, because their average data queue is in the same range. This is a good thing, because all of the users can transmit frequently.

4.1.2 Experiment 2

According to what it has happened in the previous experiment, the next step is to increase the arrival probability, in order to make more efficient the secondary system.

In this way, the basic parameters of this new experiment are:

- Packet arrival probability, $p=0.5$.
- Data buffer size, $DBS=20$ bits.

In this experiment, the interference queue graphic will not be shown because the values are the same that in figure (4.4). In order to be as less repetitive as it is possible I have considered not showing it.

As it was made in the previous experiment, a set of pictures will explain how the system works with this new parameter value.

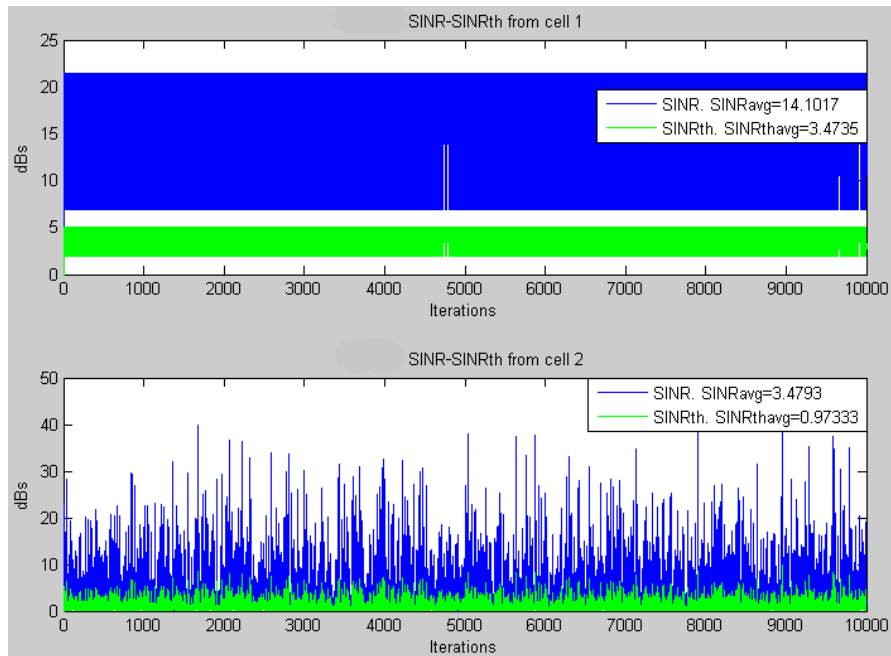


Figure 4.8: SINR and SINRth from cell 1 and cell 2

As it was described before, in this experiment the packet arrival probability has increased from 0.1 to 0.5. It means that the input average data rate is 1 bit/iteration/Hz, compared to 0.2 bit/Hz of the previous experiment, the step is quite large.

In this way, it can be seen in the upper picture that SINRth have increased. The main reason is because the data queues are larger, so the data rate optimization algorithm calculates larger rates.

In this case, the $SINR_{th}$ from cell 1 is 3.4735 dB. It is about 0.7 dB higher than the same value in the previous experiment. It is not a huge increase, but it indicates that the average data rate will probably increase.

For the second cell, the SINR_{th} is 0.97 dB. The gap is around 0.3 dB, but if it is seen in relative terms, the increase is almost 50%.

As it was explained in Experiment 1, it seems to be a relationship between the accuracy of data rate optimization algorithm and the SINR level. In other words, the larger SINR_{th} is, the larger difference exists between SINR_{th} and SINR.

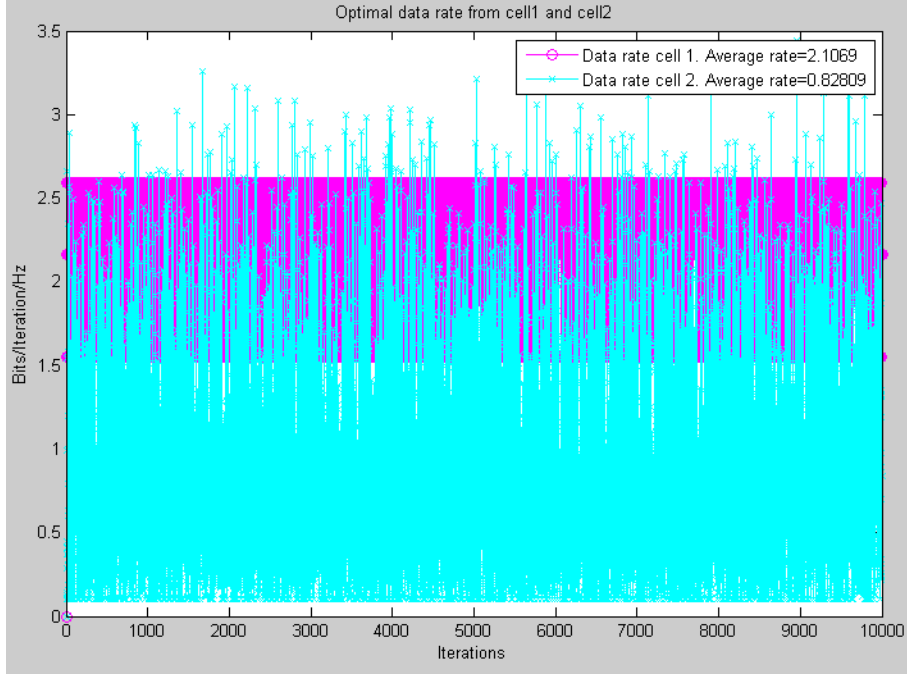


Figure 4.9: Optimal data rate from cell 1 and cell 2

In this case, due to the higher arrival probability, the transmissions in cell 1 are more frequent than the previous experiment.

Furthermore, the average value of data rate has increased. For cell 1, the value is 2.1 bits/iteration/Hz. It means an increase of 0.73 bits/iteration/Hz if it is compared to the data rate from Experiment 1. In the second cell case, the gap is 0.22 bits//iteration/Hz. It is lower than in cell 1 and it seems to be close to the limit, because the input average data rate is 1 bit/iteration/Hz. It is quite probably that the second cell has more losses than the first one.

If we compared the real and estimated data rates, we obtained the following values: the real data rate from cell 1 is 3.91 bit/iteration/Hz and from cell 2 is 2.16 bits/iteration/Hz. In this case, the differences between the estimated ones are 1.81 bits/iteration/Hz and 0.99 bits/iteration/Hz. These differences are lower than the ones obtained in the previous experiment.

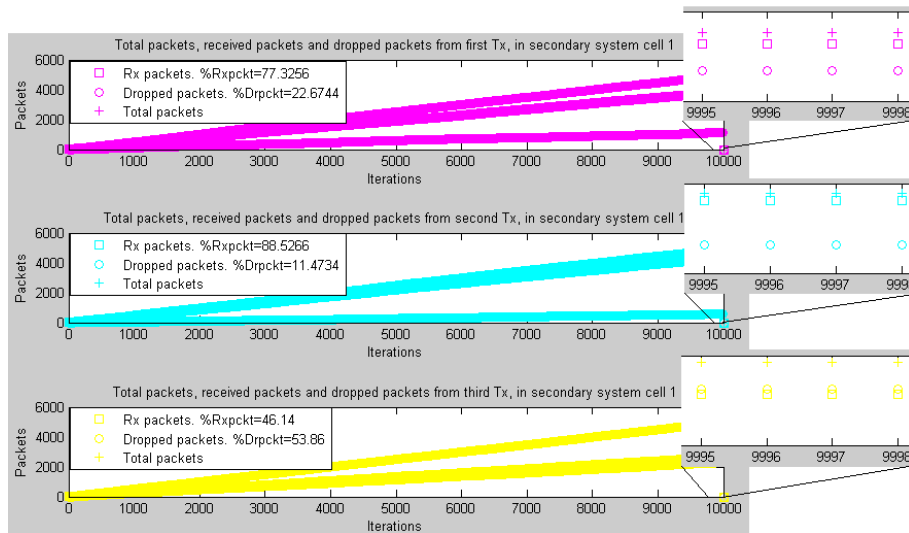


Figure 4.10: Received and dropped packets from cell 1

In this case, the system does not work as well as in the first time. The level of dropped packets is more significant and, for instance, in the third transmitter is 53.86%, which is larger than the received ones.

The best secondary transmitter is also the second one. Its percentage of dropped packets is high, 11.46%, but not too much. On the other hand, primary and even more third transmitter, have worse levels of dropped packets.

Nevertheless, the average packet arrival is lower than the general average data rate, so theoretically, the cell should be able to received all the packets that arrive.

This percentage of dropped packets is too high, so it is necessary to reduce the packet arrival probability and increase the buffer size.

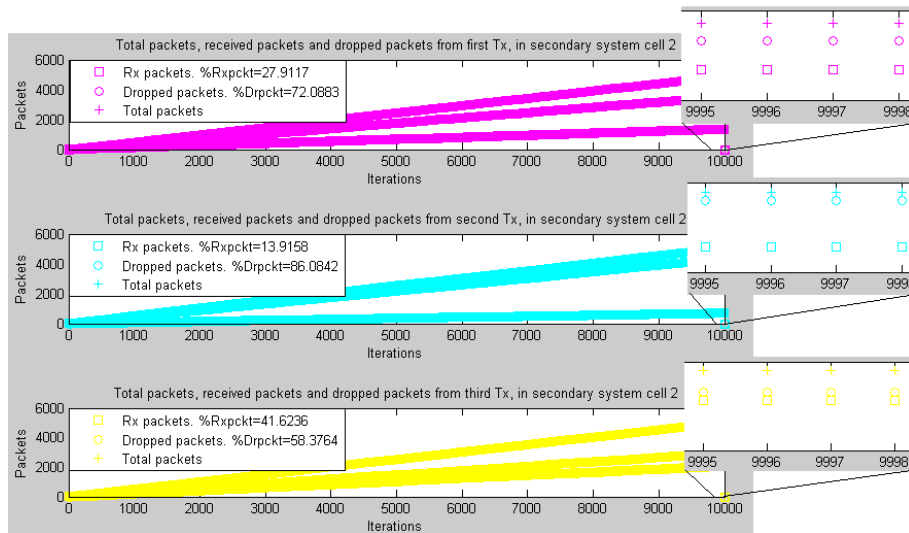


Figure 4.11: Received and dropped packets from cell 2

As it was described previously, the received and dropped packets figure from cell 2 is much worse than the one from cell 1.

Its average data rate is lower than the input average rate, so it is natural to suppose that the amount of dropped packets will be higher than the amount of received ones.

As it can be seen, the first transmitter has a percentage of dropped packets equal to 72.08%. For the second transmitter the percentage is 86.08% and for the third one the relative value of dropped packets is 58.37%

None of the secondary transmitters have more received packets than dropped packets. This is a bad situation because the amount of received packets must be much larger than the amount of dropped ones.

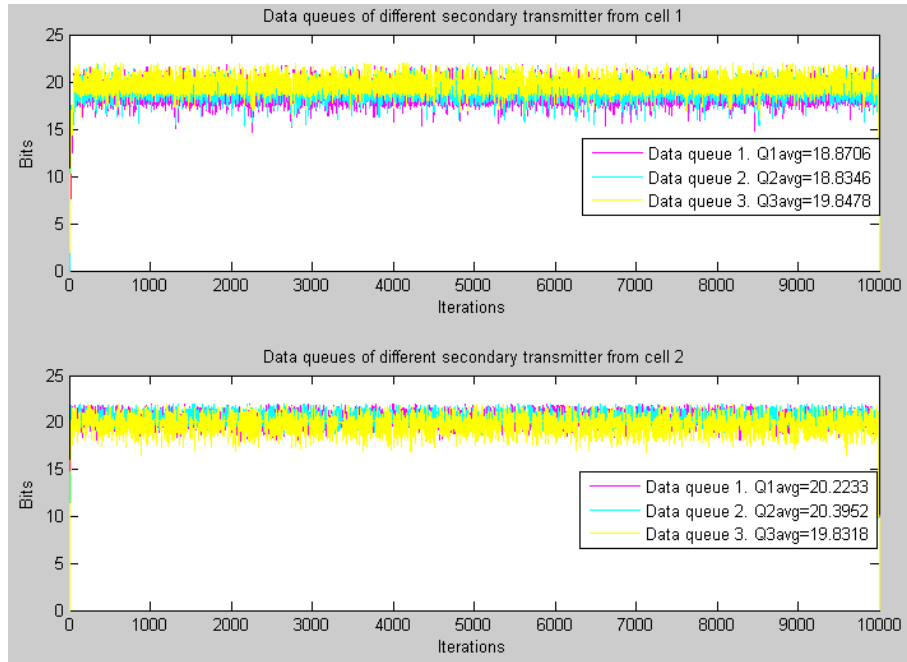


Figure 4.12: Data queues from cell 1 and cell 2

The data queues figure also shows what the previous figures describe. The arrival of packets is higher than the data rate of the system, so the data queues are full too soon and the level of dropped packets is significant.

On the other hand, the data queue from cell 1 is higher. However, as it is possible to see in figure (4.10), the percentage of received packets is quite high related to the average data queue length.

It means that an average data queue value close to buffer size does not mean a bad level of received packets. For instance, the second transmitter has an average data queue of 18.83 bits and its percentage of arrival packets is close to 90%.

4.1.3 Experiment 3

This experiment is an average of the two previous ones.

From the first one we conclude that a higher packet arrival probability should be possible. From the second one we conclude that the probability was too high and that the buffer size should be higher.

For these reasons, the basic parameters of this experiment are:

- Packet arrival probability, $p=0.5$.
- Data buffer size, $DBS=60$ bits.

In this way, the experiment will try to figure out if a larger data queue causes an improvement of the average data rate and the percentage of received packets in the different secondary transmitters.

In this experiment, the interference queue graphic will not be shown because the values are the same that in figure (4.4). In order to be as less repetitive as it is possible I have considered not showing it.

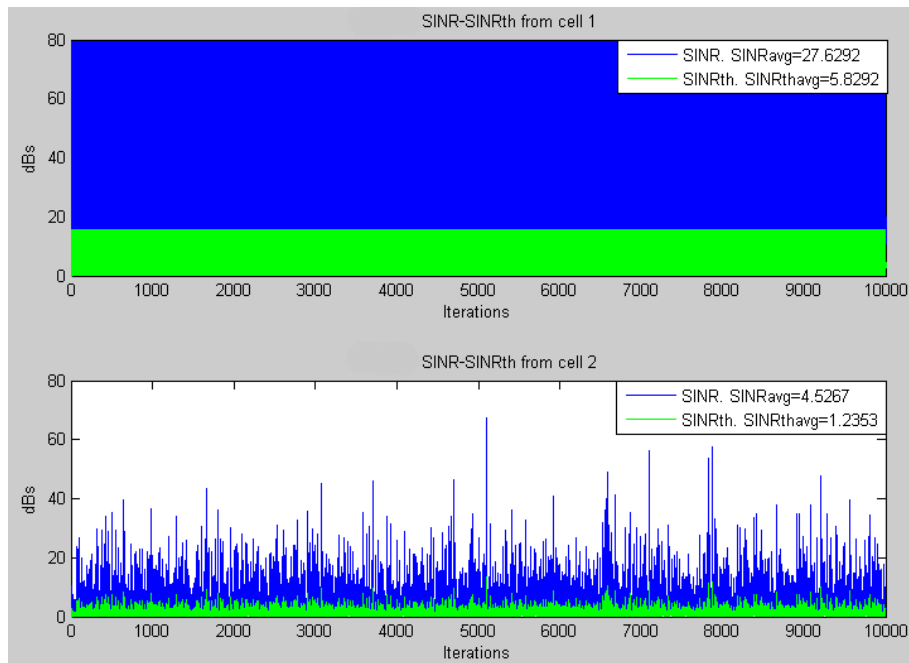


Figure 4.13: SINR and SINRth from cell 1 and cell 2

The previous picture represents the best $SINR_{th}$ of the three experiments. Their values are the highest ones and that means a higher data rate.

For example, if it is compared the $SINR_{th}$ value from cell 1 to the $SINR_{th}$ values from previous experiments, there is an improvement of 2.4 dB and 3.1 dB, from experiment 2 and experiment 1, respectively. Even if the packet arrival probability is lower.

There is still a better relation between SINR and $SINR_{th}$ when their values are lower. For example, from the cell 1, the quotient is 4.74 and for the second cell is 3.67.

The same situation occurs in cell 2. In this case, the improvement is not so large, but it exists. The difference between SINR_{th} is: 0.23 dB and 0.6 dB. It is not significant but it is still an improvement.

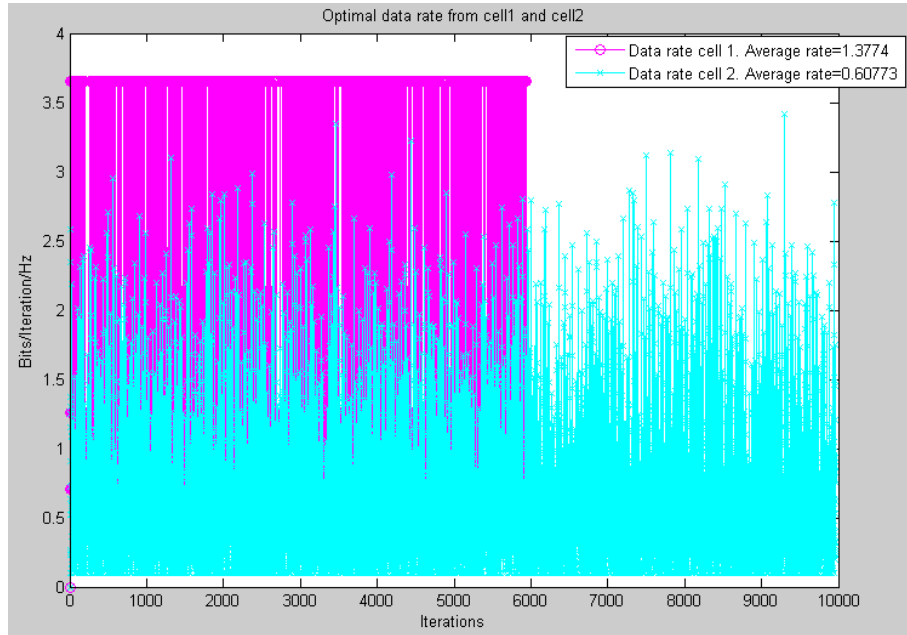


Figure 4.14: Optimal data rate from cell 1 and cell 2

As it was supposed in figure(4.14), the average data rates in both cells are the highest ones although the difference is small. In the case of cell 1 is just 0.01 bits/iteration/Hz more than in experiment 2, but in the case of cell 2 the improvement is higher, 0.17 bits/iteration/Hz more than in the precedent experiment.

On one hand, the data rate from cell 1 usually oscillates between two values: 4 bits/iteration/Hz and 0.5 bits/iteration/Hz. On the other hand the data rate from cell 2 takes more different values, as it can be appreciated in the picture.

If it is compared the data rate to the SINR, the differences are: the data rate obtained from SINR in cell 1 is 4.78 bits/iteration/Hz and from cell 2 is 2.46 bits/iteration/Hz. There is a huge gap between the estimated and real data rate.

It can be assumed that the data rate algorithm is quite conservative.

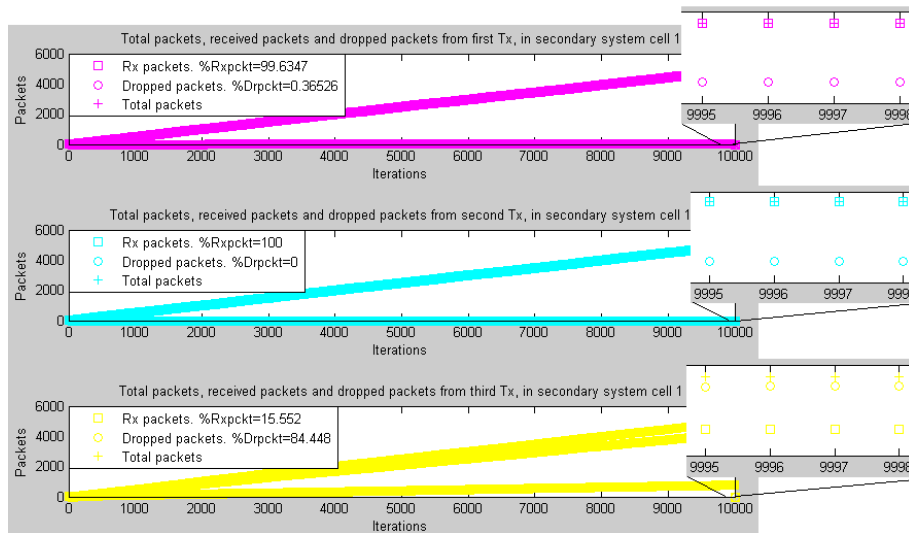


Figure 4.15: Received and dropped packets from cell 1

As it can be seen in the figure, the first and second transmitter has enhanced their received packets.

Actually, they have almost no dropped packets; just first transmitter has a percentage of dropped packets equal to 0.36%.

However, the third transmitter has decreased its percentage of received packets, from 46.14% to 15.55%.

There are some reasons that can explain this situation, on one hand, the fast fading can diminish the channel state and make it worse.

On the other hand, the other two transmitters can have better channel states and have a best metric when the scheduling algorithm has to select a secondary transmitter.

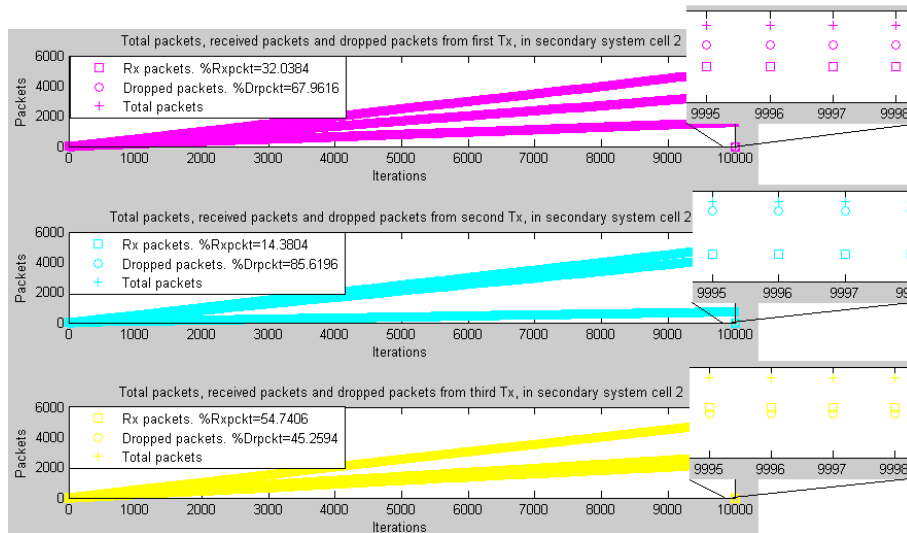


Figure 4.16: Received and dropped packets from cell 2

In this case, the improvement is not enough. The percentage of dropped packets is quite large in the second and first transmitter, and large in the third one.

However, the system has enhanced its behaviour and the improvement is a fact, if you compared to the previous experiment.

For example, the percentage of received packets has increased from 27.91% to 32.04% in the first transmitter, from 13.91% to 14.38% in the second one and from 41.62% to 54.74%.

As it can be seen in the two previous pictures, a larger buffer length enhances the percentage of received packets.

The improvement is enough if the system has a medium percentage of dropped packets. However, if that percentage is quite large, it must be included a reduction of packet arrival probability.

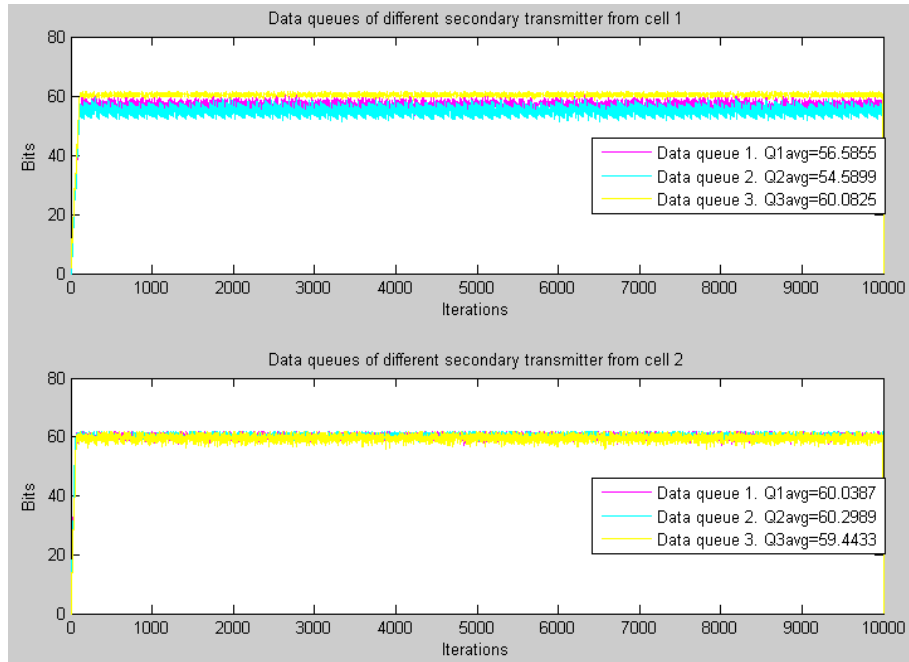


Figure 4.17: Data queues from cell 1 and cell 2

This figure shows the data queues that are stable and the ones that are unstable. For instance, the first average data queue is 56.58 bits, that average value is enough to keep stable the data queue and has a low level of dropped packets. The same situation happens for the second data queue, its value is even better, 54.59 bits. So all the packets are received.

On the other hand, the third transmitter has a average value close to the constraint, so its amount of dropped packets is higher.

For the second cell the situation is worse. The first and second transmitters have average values close to the constraint value so its amount of dropped packets is significant, as it can be seen in figure (4.16).

In contrast, the third transmitter has a better average data queue, so its level of received packets is higher.

4.2 Secondary users further from secondary receiver.

Once it has been known how the secondary system works when the transmitter are close to the receiver, the next step is to check the behaviour of the system when the transmitter are further away.

The main aim of these set of experiments is to know how the system works when the primary interference is more important, and the range of estimated data rate is more restricted.

It must be denoted that the relative positions of the different secondary transmitter of each cell have changed. Thereby, it makes no sense to compared Experiment 1 with Experiment 4 and the following cases.

The comparison will be made between the experiments of this section. However, some conclusions related to the different distance are also valid.

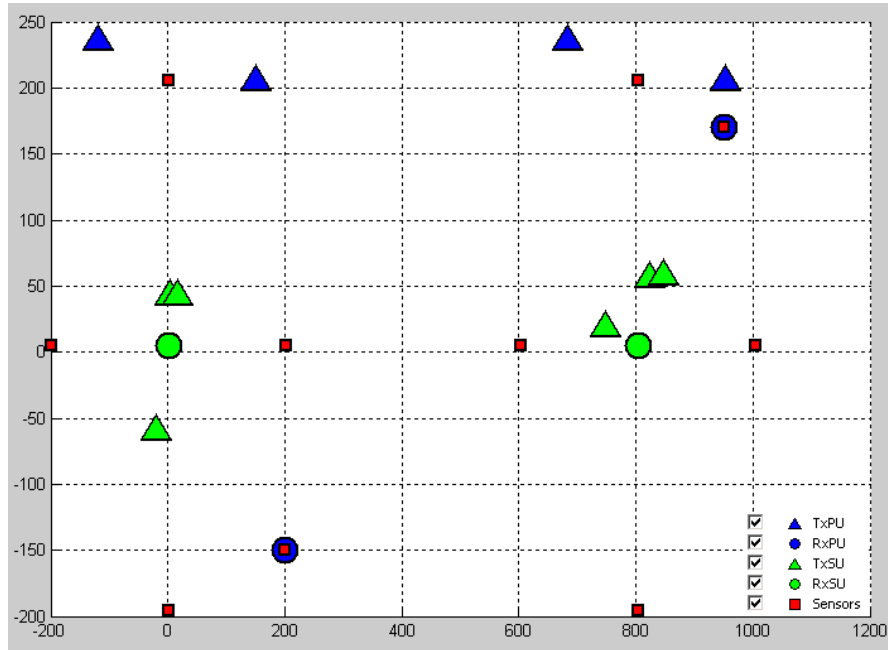


Figure 4.18: Second scenario implemented

The secondary users are fixed further, related to the pervious chapter, to the secondary user. Actually, the coverage radio cell used as a parameter is $r/5$, in the function that sets the positions. Where r is defined as the normal coverage radio cell.

During the following experiments all these parameters will be the same:

- Three secondary transmitters per cell, whose positions are fixed.
- Two primary transmitters per cell, whose positions are fixed.
- Five sensors per cell, whose positions are also fixed.
- Receiver Height, $h_b=30$ m.

- Transmitter height, $h_m=1.5$ m.
- Number of bits per packet, $\text{bpp}=2$.
- Transmission density power, $P_{tx}=1e-5$ W/Hz.
- Average interference in the own cell, $X_{av}=5.011e-18$ W/Hz (equal to -80 dBm over a bandwidth of 200 kHz)
- Neighbour average interference, $X_{avnc}=5.011e-21$ W/Hz (equal to -110 dBm over a bandwidth of 200 kHz)
- Number of iterations per experiment: 10000.

4.2.1 Experiment 4

This experiment is the same that Experiment 1, all of the parameters have the same values. The only difference is that the positions of the primary transmitters are further than the ones of the previous experiment.

However, in order to keep a method, the two parameters that can change will be described:

- Packet arrival probability, $p=0.1$.
- Data buffer size, $DBS=20$ bits.

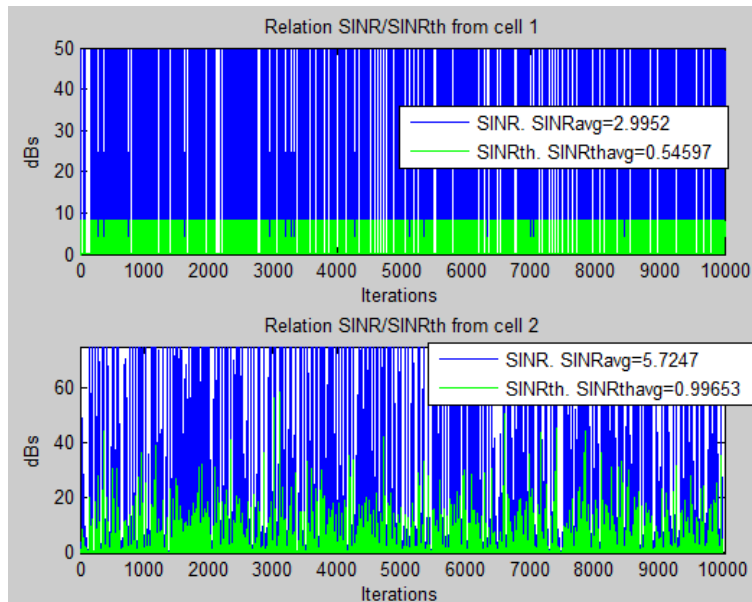


Figure 4.19: SINR and SINRth from cell 1 and cell 2

As it can be appreciated in the previous picture, all of the different values have decreased. In this case, the second cell has a better SINR_{th} , so it means a better average data rate.

As it can be seen in figure (4.18), the position of one of the secondary transmitter from cell 2 is closer than the rest ones, and even than the user from cell 1. It is quite probably that that user causes the higher SINR_{th} .

The relation between SINR and SINR_{th} is: for the first cell is 5.48 and for the second cell is 5.74. The quotient is quite similar in both cases. Nevertheless, the ratio is higher than the one calculated in Experiment 1. In that one, the values were 4.39 and 3.41.

Although most of the SINR values seem to be quite large, the average value avoids that possibility. Those values are a few, compared to all of the iterations made.

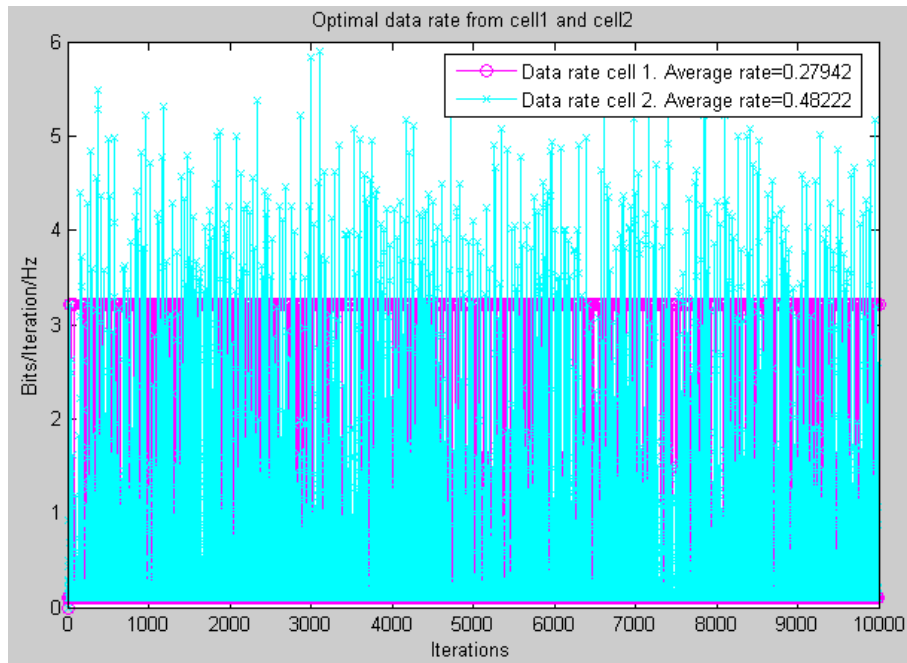


Figure 4.20: Optimal data rate from cell 1 and cell 2

This picture shows the average data rate for the cell 1 and for cell 2.

As it was explained before, the higher data rate is from cell 2, according to its higher average SINR_{th} .

One way of measuring which is the accuracy of data rate algorithm is to compared, the estimated average data rate seen above, to the real data rate obtained

from the SINR. In this case the real data rates are: 2 bits/iteration/Hz for the cell 1 and 2.74 bits/iteration/Hz for cell 2.

There is a difference of 1.73 bits/iteration/Hz in the cell 1 and 2.25 bits/iteration/Hz.

As it can be seen, the estimated data rate algorithm has not a good accuracy. There is a large difference between both average data rates.

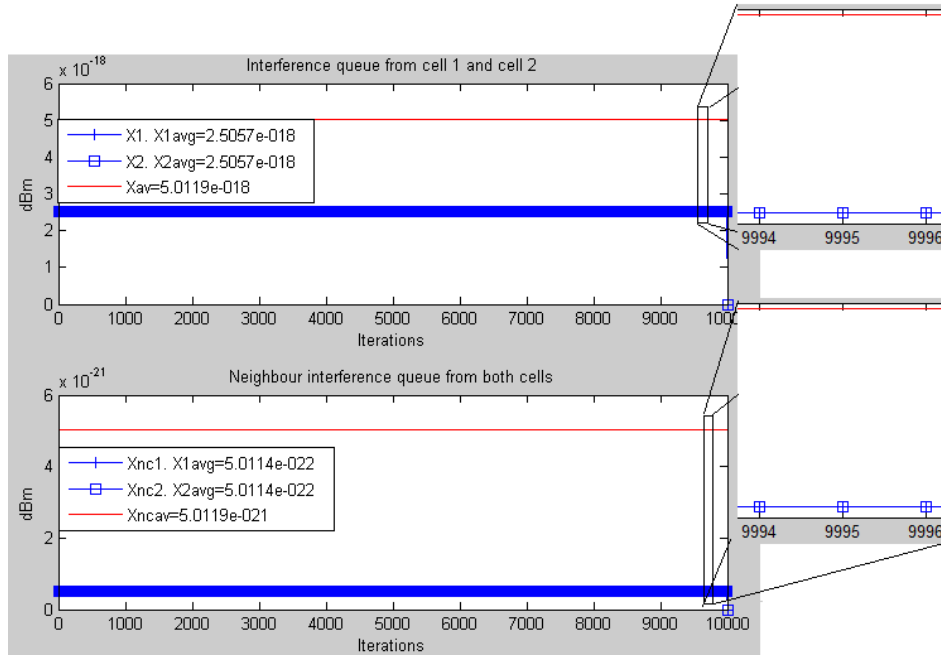


Figure 4.21: Interference queues from cell 1 and cell 2

As it was described in the previous chapter, the interference queue is also stable.

As it can be seen in the previous figure, the average interference value is over the interference queue level, so the interference from the secondary system is under control, in order to keep the normal running of the primary system.

The same situation happens in the neighbour interference.

In this case, the neighbour interference level is also lower than its average value, so the interference that one cell receive from the other is under its maximum constraint.

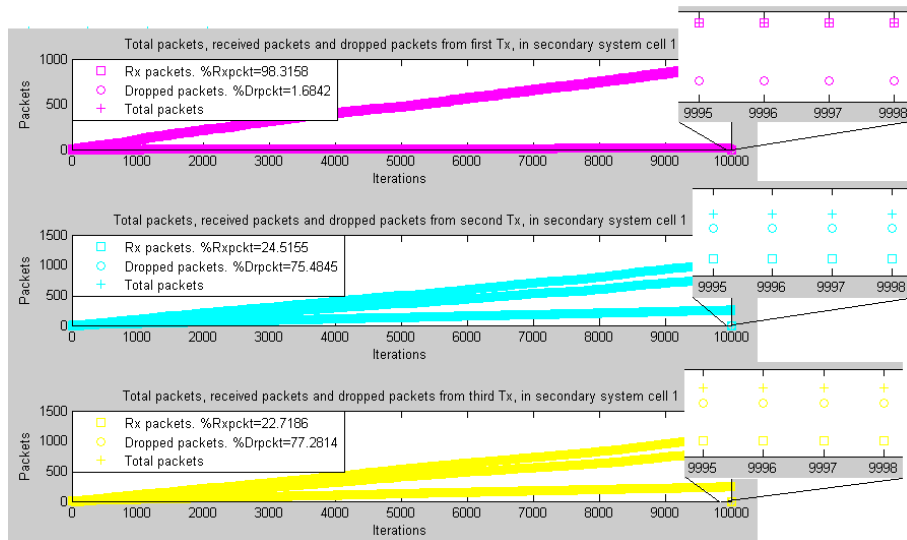


Figure 4.22: Received and dropped packets from cell 1

In this case, just the first transmitter has a good relationship between received and dropped packets. On the other hand, the second and the third one have quite bad relationship.

Two reasons can cause this situation.

First of all the average data rate is not as high as it should be. That situation causes that the data queues start to fill up and just the one who has the best channel will transmit. But if the overall data rate is similar to the input data rate, just one secondary user will keep its data queue in appropriate levels.

This is a logical assumption because if we compared the average input data rate, that is 0.2 bits/Hz, and the estimated data rate from cell 1, which is 0.28 bits/iteration/Hz, it can be checked that the overall data rate from cell 1 just can work with one cell in a suitable way, but not with more.

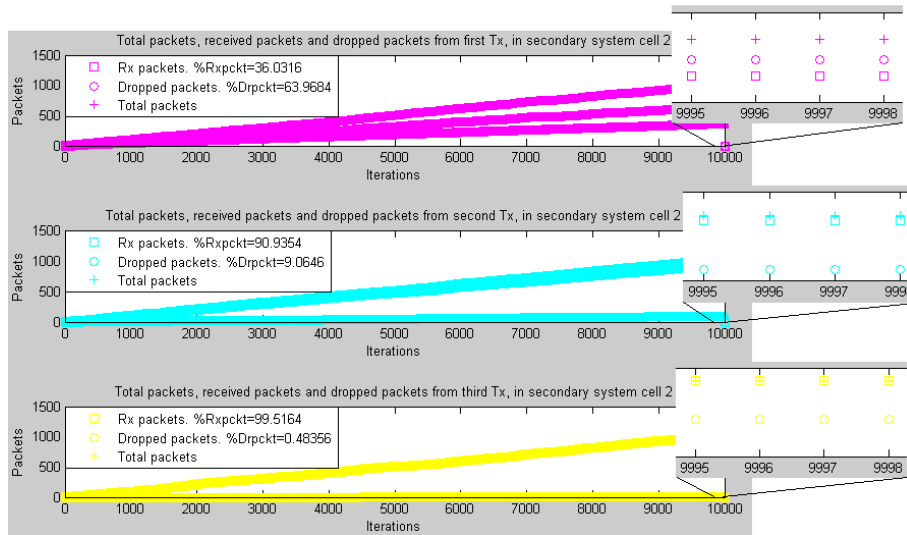


Figure 4.23: Received and dropped packets from cell 2

In this case the situation is almost the same, but the estimated data rate from cell 2 is higher than from cell 1. Thereby, two of the secondary transmitter have a satisfying relationship between received and dropped packets.

On the other hand, the first transmitter, which has the worst channel state, must drop most of the packets that receives.

If the buffer length would be higher, its metric in the scheduling algorithm would be better and it would transmit more.

Having an overall data rate larger than the sum of partial input data rates is essential in order to have stable queues.

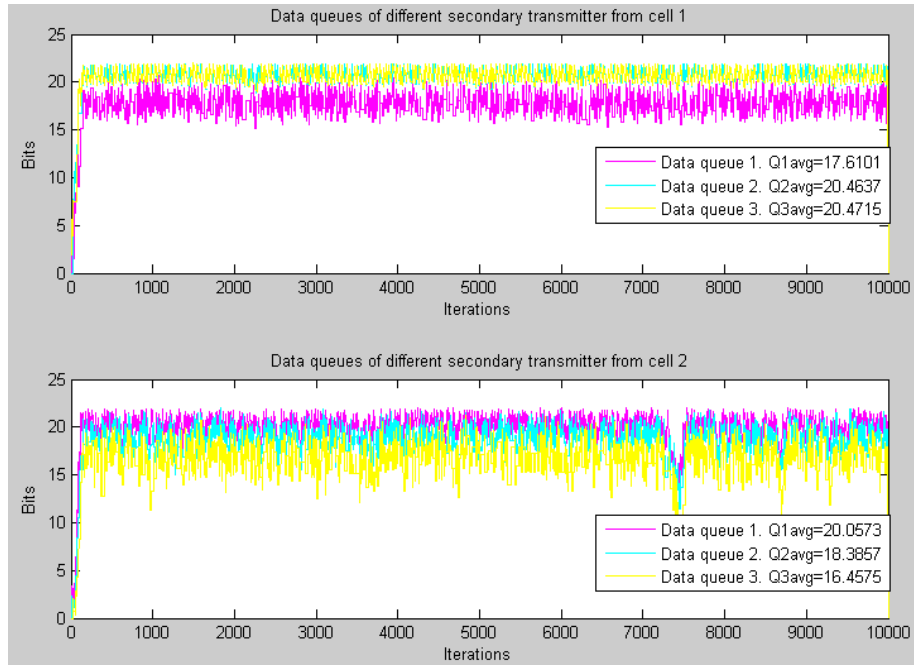


Figure 4.24: Data queues from cell 1 and cell 2

In this figure it can be shown the stable data queues.

As it can be seen in all these type of graphics, an average data queue value of 90% of the buffer length constraint it is enough to have a good relationship between received and dropped packets.

For example, average data queue from cell1 is 17.61, whose value is 88.05% of the constraint buffer length. With this average value, the percentage of received packets is 98%, which is an acceptable value.

In order to improve the average data rate, an increase of buffer length will be included in the following experiment.

4.2.2 Experiment 5

In this experiment it will be increased the data buffer length in order to increased the overall data rate and allow the system works better.

To achieve these objectives, the following parameter must change:

- Packet arrival probability, $p=0.1$.
- Data buffer size, $DBS=100$ bits.

In this experiment, the interference queue graphic will not be shown because the values are the same that in figure (4.21). In order to be as less repetitive as it is possible I have considered not showing it.

In the following figures, how this change affects the secondary system will be described.

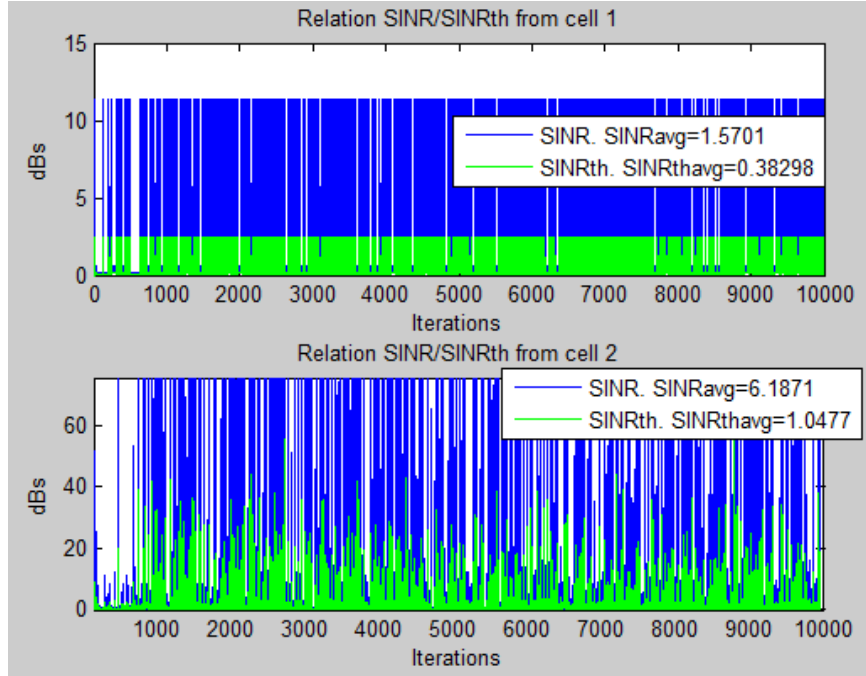


Figure 4.25: SINR and SINRth from cell1 and cell 2

In the previous picture, it can be appreciated that average values of cell 1 are lower than the same ones in Experiment 4.

Furthermore, the relation between SINR and SINR_{th} is lower than in the previous experiment. In this case the quotient is 4.13, compared to 5.43 from the preceding one.

On the other hand, if the second cell is analyzed, it can be seen that the relationship between SINR and SINR_{th} is higher, compared to Experiment 4. The value is 5.90 and the previous value was 5.77.

There is a small difference but it is completely different from what happen in cell 1.

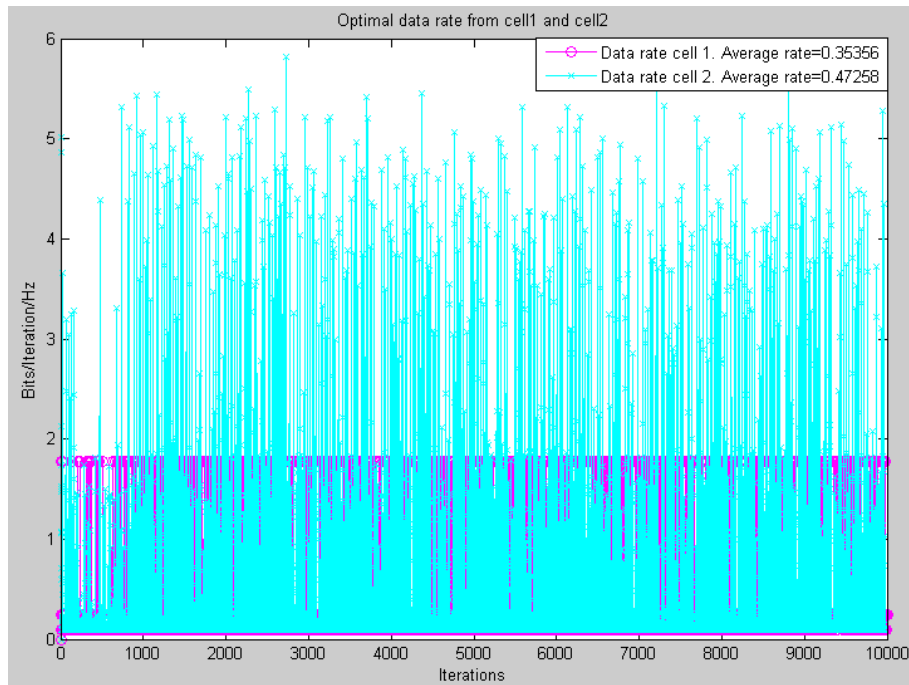


Figure 4.26: Optimal data rate from cell 1 and cell 2

In this case, the average data rate is higher in one cell and in the other one is almost the same.

Compared to previous precedent experiment, the average data rate has increased 0.07 bits/iteration/Hz and decreased 0.01 bits/iteration/Hz.

On the other hand, the real values of data rate are, according to the SINR from figure (SINR): 1.36 bits/iteration/Hz for cell 1 and 2.84 bits/iteration/Hz. There is a large difference between the real and estimated values.

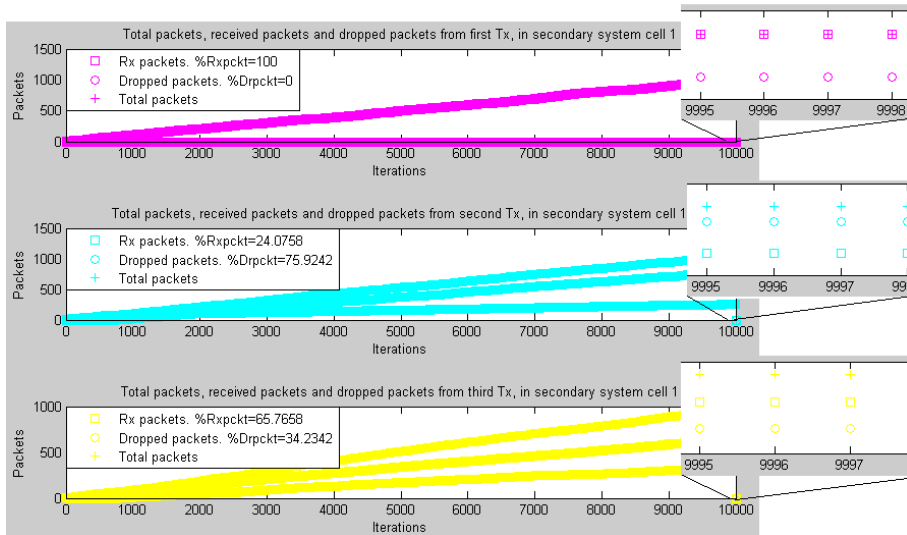


Figure 4.27: Received and dropped packets from cell 1

This picture shows how the overall data rate is distributed between the different transmitter according to its metric in the scheduling algorithm.

For example, the average data rate for cell 1 is 0.35 bits/iteration/Hz and the average packet arrival is 0.2 bits/iteration/Hz. Based on these two facts, it can be understood the figure.

The first transmitter from cell 1 has the best metric, so its received is 100%, its improvement is not large, because in the previous experiment, the percentage of received packets was 98%. The third transmitter has enhanced much more, its percentage of received packets goes from 22% to 65%. This improvement is also possible because the overall data rate has improved.

Finally, the second transmitter has improved too much, its metric of received packets is almost the same, because of its worse channel state and the small value of the general data rate.

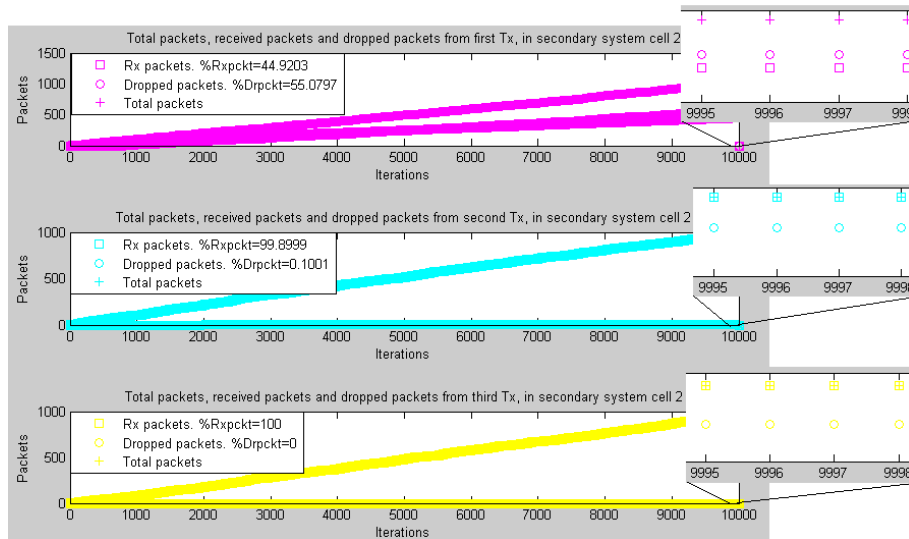


Figure 4.28: Received and dropped packets from cell 2

The situation is similar in the cell 2, but in this case there is an advantage; the overall data rate is higher than the one of cell 1. Thereby, a higher amount of packets will be received.

The average data rate is 0.47 bits/iteration/Hz, so two and a half transmitter could receive all of the packets because the average packet arrival is 0.2 bits/iteration/Hz.

Actually, this is what happens. The third and the second transmitters do not drop any packet, and the first one, due to its worse channel state, just can received the half part of all the packets that it receives.

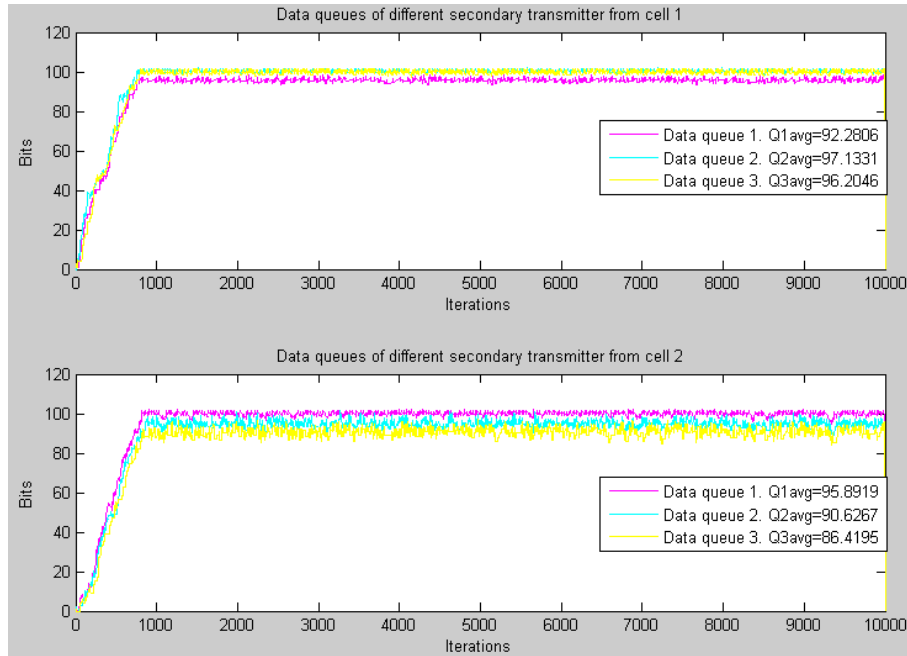


Figure 4.29: Data queues from cell 1 and cell 2

This figure is quite useful if it can be known the secondary transmitter that has the best channel state compared to the rest of the transmitter of each cell.

In this case the best transmitter from cell 1 is the first one. Its average data queue is lower. According to this, in the same conditions of data queue, the scheduling algorithm usually selects the first transmitter like the most suitable one, because its channel state is better.

On the other hand, the third and the second ones have close average data queue values, so their channel state must be quite similar.

For the cell 2 the situation is slightly different, the third transmitter has the best channel state. Afterwards, the second one has the best channel state and finally the first transmitter has the worst channel state in this cell.

4.2.3 Experiment 6

In this last experiment, the packet arrival probability will be slightly increased. In this way, the main aim is to analyze if the system is able to support this additional charge without losing a significant amount of packets.

The following parameters are defined like:

- Packet arrival probability, $p=0.1$.
- Data buffer size, $DBS=100$ bits.

In this experiment, the interference queue graphic will not be shown because the values are the same that in figure (4.21). In order to be as less repetitive as it is possible I have considered not showing it.

The different parameters that explain the behaviour of this new setting will be shown in the next figures.

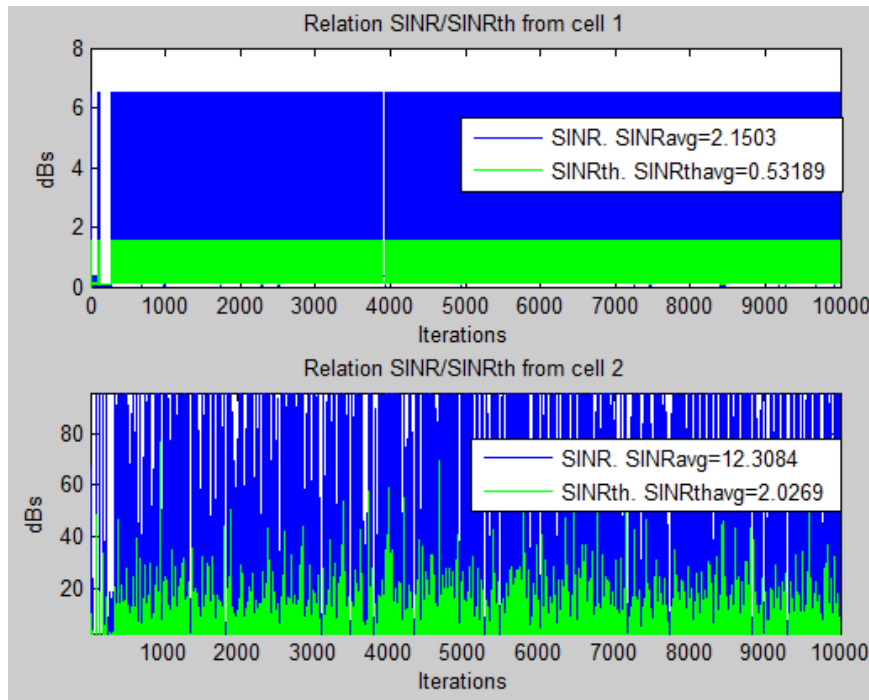


Figure 4.30: SINR and SINRth from cell 1 and cell 2

In this picture it can be seen that values of different SINR and $SINR_{th}$ have increased.

The main reason of this is the increase of packet arrival probability. In this case the data rate optimization algorithm is closer to the real value in the first cell than in the second one. POner algo mas!

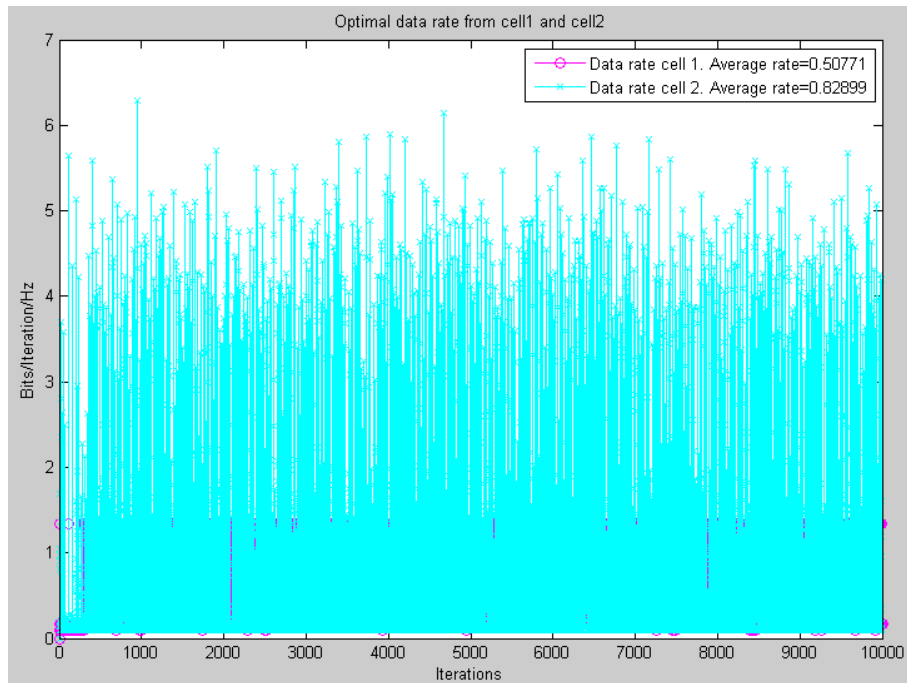


Figure 4.31: Optimal data rate from cell 1 and cell 2

As in the figure (4.30), the average data rate has also enhanced in both cells.

The real values of data rate are: 1.65 bits/iteration/Hz for the first cell and 3.73 bits/iteration/Hz. There is a higher deviation in the second cell than in the first one.

For the first cell, the improvement is 0.15 bits/iteration/cell and for the second one, the difference is 0.35 bits/iteration/Hz. This is a large improvement related to the previous experiment.

According to this, it seems to be an error in the measurements, because this new system has a better average data rate than the one of the precedent system, but any improvement has been made. Actually, the system supports a higher load.

It must be taken into account that channel states are based on probability distributions, so there is a range of values where the same system, with the same settings, can work better or worse. This is probably what has happened in this case.

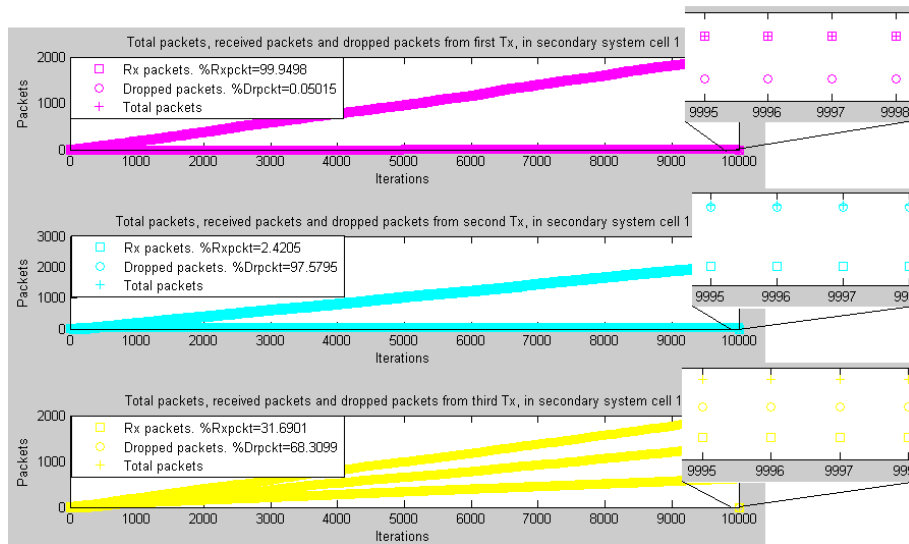


Figure 4.32: Received and dropped packets from cell 1

It must be denoted that the average packet arrival is 0.4 bits/iteration/cell. Compared to the average data rate, 0.5 bits/iteration/Hz, just one and a half receiver can accept all the packets that arrive.

Actually, this is what happens in the cell 1. Just the first transmitter can take most of the packets that arrive. Besides, the rest of the overall data rate is taken by the third transmitter, but it just can receive 31.69% of the total amount of packets. The second user almost transmit nothing.

Although the average data rate has improved, the average packet arrival has improved more, so the final situation is that the amount of dropped packets is higher than in Experiment 5

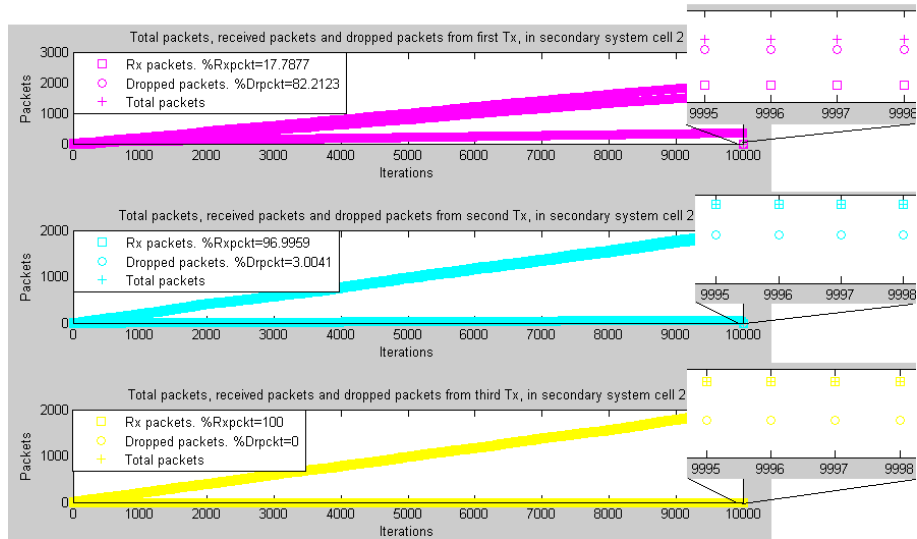


Figure 4.33: Received and dropped packets from cell 2

In the case of cell 2, the situation changes a bit. The average data rate is 0.82 bits/iteration/Hz. Compared to the average input data, two transmitters can receive most of the packets.

The previous figure shows this situation. The third and the second transmitter can receive almost all the packets, but the first one almost drops all the packets that receive.

The fact the second transmitter drops some packets is because its channel state is not as good as the weight of the data queue from first transmitter. Sometimes that weight is higher and the most suitable transmitter is the first one.

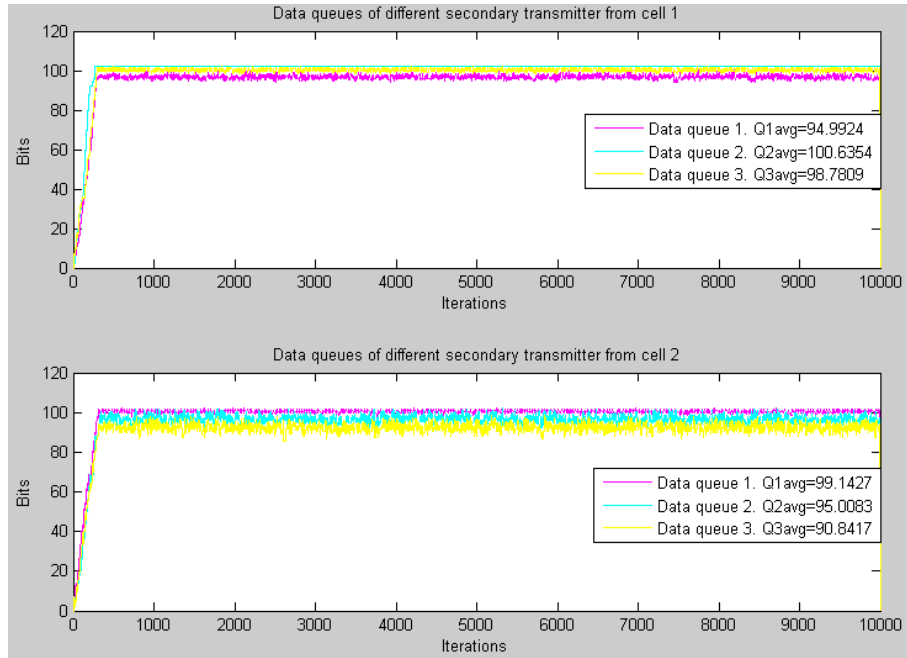


Figure 4.34: Data queues from cell 1 and cell 2

For the cell 1, the first transmitter has the best channel state, as it can be checked in figure (packets 1). In contrast, the third and the second transmitters have worse channel state, and because of that and the limited value of overall data rate, their average data queue is higher.

In the case of cell 2, the higher data rate allows better values of average data queue. The transmitter with the best channel state is the third one, followed by the second one and finally the first one.

4.3 Best value constraint for intercell interference

The last section of this chapter is focused on the computation of the most suitable value for the interference constraints, the own one and the intercell one.

During the previous sections the X_{av} and X_{av}^{NC} values were quite low, around -80 dBm and -110 dBm over a bandwidth of 200 kHz, respectively. Otherwise, they can be lower and find this minimum is the main aim of this section.

The method that has been implemented is to run a range of different interference

constraints and see when the interference queue starts to be higher than the interference constraints.

It will be done two measurements, according to the two scenarios that have been set during the previous sections.

4.3.1 Experiment 7

In this one, the scenario implemented in section 4.1 will be set.

The range of the interference constraints are:

- X_{av} = from $2.0119e-18$ W to $2.0119e-21$ W
- X_{av}^{NC} = from $5.0119e-23$ W to $5.0119e-26$ W

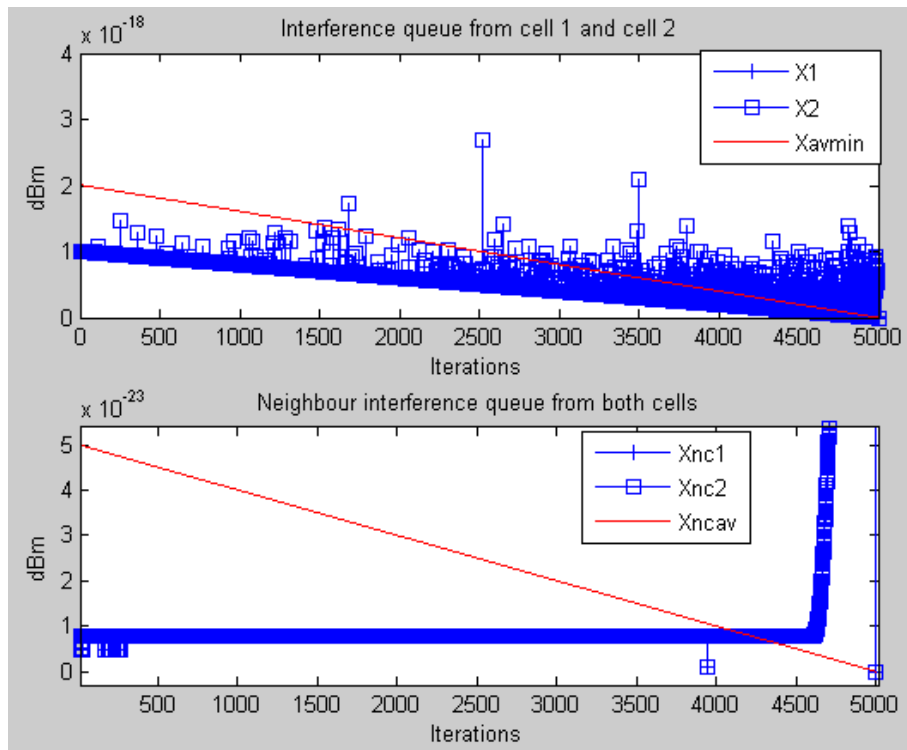


Figure 4.35: Interference queues from cell 1 and cell 2

As it can be seen in the previous picture, the range of interference constraints implemented has 30 dB. In the case of the intracell constraint, its value goes from -83 dBm to -113dBm.

The interference level starts to increase when up to iteration number 1000 and over the half number of iteration, the interference queue is unstable. So, it is in this specific range, from iteration 100 to iteration 2500 where the optimal intracell interference constraint is. In db terms, the optimal value would be between -95 dBm and -100 dBm.

The value used in the previous chapter was -80 dBm. This is a conservative value that let the system works appropriately.

The subplot is focused on the intercell interference constraint X_{av}^{NC} . Its range is also 30 dBm but its first value is -130 dBm, an extremely small value. It is must be denoted that the nose power level used is -120dBm.

The neighbour interference queue starts to be unstable up to iteration number 4200 and its increase is exponential in the last part of the graphic. The optimal intercell constraint is $8.1e-24$, in dB terms means -148 dBm. This value is much lower than the noise power level, so it does not affect the usual running of the neighbour cell.

Furthermore, the neighbour interference constraint used in the previous sections was quite conservative, because its value was -110 dBm.

4.3.2 Experiment 8

In this case, the scenario used in section 4.2, where the secondary transmitters are further away, will be implemented.

The range of the interference constraints are the following ones:

- X_{av} = from $2.0119e-18$ W to $2.0119e-21$ W
- X_{av}^{NC} = from $5.0119e-23$ W to $5.0119e-26$ W

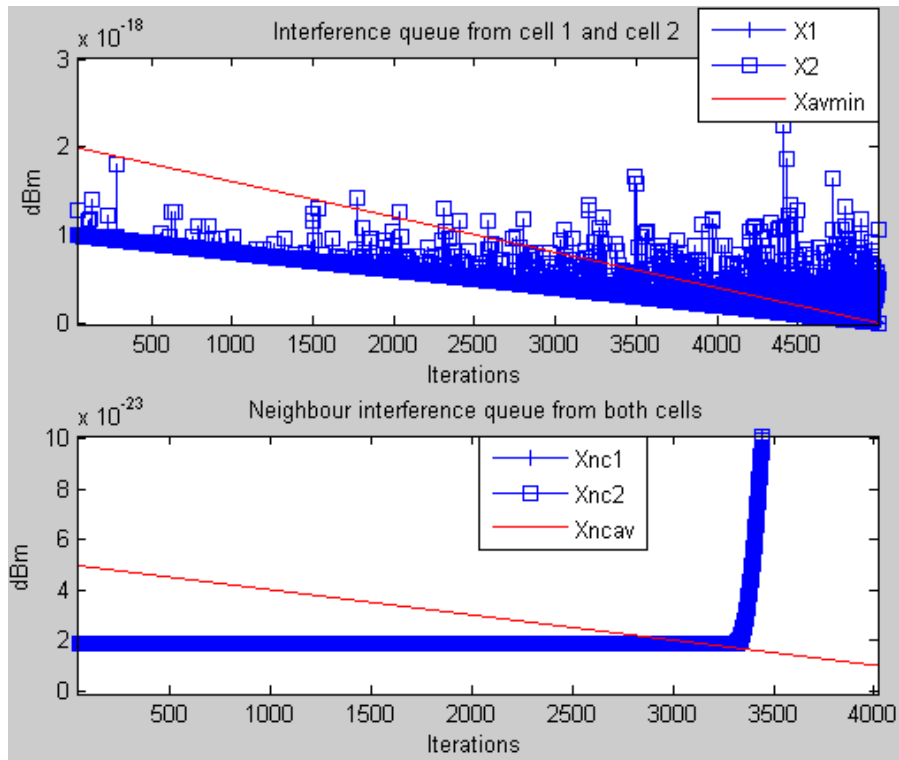


Figure 4.36: Interference queues from cell 1 and cell 2

In this case, the interference constraints are analyzed over the second scenario, the further one.

The rest of the parameters are the same, just change the distance between secondary transmitters and receiver.

As the precedent figure shows, the interference queue level increases previously. From the first value of interference constraint, -83 dBm, there are interference queue values close to the limit. Up to iteration number 1500 the interference level reaches the intracell interference constraint, and over iteration number 2500, the level is higher.

The optimal intracell interference constraint is between $1.5e-18$ W and $1e-18$ W, which means -95 dBm to -96 dBm. It is higher level than in the precedent experiment but the difference is small.

For the intercell interference constraint, it is possible to see that the limit and the interference level are the same around the iteration number 3000. The value associated to this iteration is $2e-23$. It means -144 dBm, which is a quite small value.

This interference constraint is 4 dBm than the one of the previous experiment.

Chapter 5

Conclusions and discussion

For the results of the simulations we have obtained several conclusions.

We have checked that the data rate optimization algorithm is quite conservative. In most of the simulations its estimated value is smaller than the real one, so there is a loss in the efficiency of the system.

Besides, when the different secondary transmitters are set further away from the secondary receiver, and consequently close to the primary system elements, the efficiency of the system is reduced. The interference from the primary system is larger and parameters like data rate reduce its value.

On the other hand the interference level from intracell and intercell is always quite controlled. Their values are under the interference constraints and that is a good point. We have considered higher values of X_{av} and X_{avc} than their optimal ones, but their values are quite suitable.

In order to improve the efficiency of the system, increasing the data queue size is a fine solution. We have checked that the system behaviour enhances and also its efficiency. Otherwise, in order to achieve the higher efficiency, the increase of data buffer size must be accompanied by a suitable value of packet arrival probability.

We must also take into account that all of the values that have been obtained from the several simulations are based on probabilistic distributions, so it is possible to find some results that do not fit precisely to what it is suppose to happen.

In conclusion, this work let have an overall view about how a distributed system based on cognitive radio is able to work.

For future work on this subject, it would be interesting to study more about the communication between fusion center and the rest of the elements, like transmitters and sensors. In this way, an improvement in the data rata optimization algorithm with would enhance the efficiency of the whole system.

Appendix A

Matlab code

A.1 Cell structure function

```
function cell=create_cell(P,r,S)

%This function sets most of the variables that work in the main program
%and calculates the distance matrix between all of the elements.

global numTxPU numTxSU numSensors hb hm sigmaS Xmin XminNC

PosTxPU=[P(1,1)-120,P(1,1)+150;P(2,1)+230,P(2,1)+200]

cell=struct('RxSU',P(:,1),'TxPU',PosTxPU,'TxSU',cell_point(r/10,numTxSU...
    ,P),'Sensors',S,c,zeros(1,numTxSU),'d',zeros(numTxSU,numSensors)...
    ,'e',zeros(1,numTxPU),int,zeros(1,numTxPU),'ecross',zeros(1,numTxPU...
    +numTxSU),'intcross',zeros(1,numTxPU+numTxSU),'c0',zeros(1,numTxSU)...
    ,'d0',zeros(numTxSU,numSensors),'e0',zeros(1,numTxPU),'int0',zeros...
    (1,numTxPU),'e0cross',zeros(1,numTxPU+numTxSU),'int0cross',zeros...
    (1,numTxPU+numTxSU),'Q',zeros(1,numTxSU),'rxpacks',zeros(1,numTxSU)...
    ,'droppacks',zeros(1,numTxSU),'X',[Xmin*ones(1,numSensors) XminNC]...
    ,'ropt',0,'whoTx',0,'whoTxOtherCell',0,'ack',0);

%----- Calculate distances between elements -----

    %Channel state between TxSU-RxSU, c0.
    distc=CalculateDistance(cell.TxSU,cell.RxSU);
    cell.c0=10.^(CalculateFixedLosses(distc',hb,hm,sigmaS)/10);
    %Channel state between TxPU-RxSU, int0.
    distint=CalculateDistance(cell.TxPU,cell.RxSU);
    cell.int0=10.^(CalculateFixedLosses(distint',hb,hm,sigmaS)/10);
    %Channel state between TxPU-RxS, e0.
    diste=CalculateDistance(cell.TxPU,cell.Sensors);
    cell.e0=10.^(CalculateFixedLosses(diste',hb,hm,sigmaS)/10);
    %Channel state between TxSU-RxS, d0.
    distd=CalculateDistance(cell.TxSU,cell.Sensors);
    cell.d0=10.^(CalculateFixedLosses(distd',hb,hm,sigmaS)/10);
```

A.2 Fixed losses function

```
function FL=CalculateFixedLosses(M,hb,hm,sigmaS)

%A distance matrix is an input, and Plane Earth loss and Shadowing are
%calculated based on it.
```

%The result is given in dB, but is changed to lineal in the main program

```

%Path loss
tam=size(M);
L=planeearthloss(M,hb,hm);
%Shadowing
SF=10*log10(lognrnd(0,sigmaS,tam(1),tam(2)));
%Fixed Losses
FL=-(L+SF);

```

A.3 Fast fading function

```
function cell=CalculateFastFading(cell)
```

%This function calculates the fast fading. It is added to shadowing and
 %space losses.
 %Rayleigh variables are amplitude values, so it must be squared, because
 %the system works with power values.

```

cell.c=(Rayleigh(size(cell.c0)).^2.*cell.c0);
cell.int=(Rayleigh(size(cell.int0)).^2.*cell.int0);
cell.e=(Rayleigh(size(cell.e0)).^2.*cell.e0);
cell.d=(Rayleigh(size(cell.d0)).^2.*cell.d0);
cell.ecross=(Rayleigh(size(cell.e0cross)).^2.*cell.e0cross);
cell.intcross=(Rayleigh(size(cell.int0cross)).^2.*cell.int0cross);

```

A.4 Arrival process function

```
function [Aq,cell]=ArrivalProcess(cell,prob,bits,controlqueue)
```

%This function simulates the packet arrival process. Returns number of
 %packets received and updates the data queue
 %Two variables, cell.rxpacks and cell.droppacks count the received and
 %dropped packets

```
global numTxSU
```

```
Aq=zeros(1,numTxSU);
```

```
for i=1:numTxSU
    %Bernoulli process

```

```

    aqaux=arrival(prob,bits);
    Aq(i)=aqaux;
    if aqaux~=0
        if cell.Q(i)<= (controlqueue*bits)
            %Add data to secondary transmitter's buffer
            cell.Q=dataqueue(cell.Q,0,aqaux,i);
            cell.rxpacks(i)=cell.rxpacks(i)+1;
        else
            cell.droppacks(i)=cell.droppacks(i)+1;
        end
    end
end
end

```

A.5 Scheduling second transmitter function

```

function cell=first_transmitter(cell)

global numTxSU NO Ptx numTxPU

J=zeros(1,numTxSU);
Pint=sum(Ptx.*cell.int) + sum(Ptx.*cell.intcross(1:numTxPU,:));
for i=1:numTxSU
    J(i)=cell.Q(i)*(log2(1+(cell.c(i)/(NO+Pint)))-sum(cell.X.*cell.d0(i,:)));
end
if b<=1
    cell.whoTx=0;
else
    [y,cell.whoTx]=max(J);
end
end

```

A.6 Data rate optimization function

```

function cell=DataRateOptimization(f,cell)

%This function finds the value of r that maximise the function fr.

global Ptx NO numTxPU

%Primary user interference coming from all cells
Ipu=sum(Ptx.*cell.int)+sum(Ptx.*cell.intcross(1:numTxPU));

```

```

r=0.1:0.01:20;
fr=r.*(1-exp(-(Ptx*cell.c(cell.whoTx)/(2.^r-1)-NO-Ipu)/f));
[rpos,y]=max(fr);
cell.ropt=r(y);

```

A.7 Data rate test function

```

function [cell,SINR,SINRth]=test_datarate(cell)

%This function checks if SINRth estimated is higher than SINR calculated in
%the secondary receiver

global NO Ptx numTxPU

I=sum(Ptx.*cell.int)+sum(Ptx.*cell.intcross(1:numTxPU))+...
    (Ptx*cell.intcross(numTxPU+cell.whoTxOtherCell));
SINRth=2^cell.ropt-1;
SINR=Ptx*cell.c(cell.whoTx)/abs(NO+I);

if SINRth<SINR
    cell.ack=1;
else
    cell.ack=0;
end
end

```

A.8 Main program

```

function [cell1,cell2]=main_program()

%-----
%Inicialization
[name,data]=textread('parameters.txt','%s%f','commentstyle','matlab');
[x,y]=textread('fixed_positions.txt','%f%f','commentstyle','matlab');

global numTxPU numTxSU numSensors NO Xmin Xav XavNC XminNC Ptx hb hm sigmaS

numTxPU=data(1);
numTxSU=data(2);
numSensors=data(3);
sigmaS=data(4);
hb=data(5);

```

```

hm=data(6);
prob=data(7);
bits=data(8);
controlqueue=data(9);
Ptx=data(10);
cont=data(11);
Xav=data(12);
XavNC=data(13);
work=data(14);
keep_param=data(15);

P=[x,y]';

Xmin=Xav/10;
XminNC=XavNC/10;
kb=1.3806e-23; %Boltzmann constant
T=300; %Room tempeture

%-----

%-----MAIN PROGRAM-----
%----- Basic Inicialization -----
D=CalculateDistance(P,P); %Distance between cells
r=D(1,2)/2; %r is the radio cell
close all
plottools('on')
hold on

% Calculating Dnc, loss and shadowing among cells.
if keep_param==0
    Dnc=10.^(CalculateFixedLosses(D(1,2),hb,hm,sigmaS)/10);
    f=Dnc;
end

%Calculating Noise Power constant

NO=kb*T; %equivalent to -120dBm over 200 kHz

for i=1:2
    % Create the sensor structure around the cell
    eval(['S' int2str(i) '=sensor_structure([P(:, ' int2str(i)...
        ') P(:, ' int2str(i+2) ')],r);']);
    % Creating cells
    eval(['cell' int2str(i) '=create_cell([P(:, ' int2str(i)...
        ') P(:, ' int2str(i+2) ')],r,S' int2str(i) ');']);
    %If it must keep the position of the elements

```



```

        if keep_param
            load fixed_parameters
            eval(['cell' int2str(i) '.TxSU=pos' int2str(i) '.TxSU;']);
            eval(['cell' int2str(i) '.c0=pos' int2str(i) '.c0;']);
            eval(['cell' int2str(i) '.d0=pos' int2str(i) '.d0;']);
            eval(['cell' int2str(i) '.e0=pos' int2str(i) '.e0;']);
            eval(['cell' int2str(i) '.int0=pos' int2str(i) '.int0;']);
            eval(['cell' int2str(i) '.e0cross=pos' int2str(i) '.e0cross;']);
            eval(['cell' int2str(i) '.int0cross=pos' int2str(i) '.int0cross;']);
        else
            eval(['cell' int2str(i) '.d0=[cell' int2str(i)...
                '.d0 Dnc*ones(numTxSU,1)];']);
        end
        eval(['representation(cell' int2str(i) ');']); % Plotting Cells
    end

hold off
grid

if keep_param==0
    [cell1,cell2]=cross_interference(cell1,cell2);
else
    f=pos1.d0(1,end);
end
% Adding fast fading
for i=1:2
    eval(['cell' int2str(i) '=CalculateFastFading(cell' int2str(i) ');']);
end

%-----Output Inicialization-----

Ropt1=zeros(1,1);
Ropt2=zeros(1,1);
SINR1out=zeros(1,1);
SINRth1out=zeros(1,1);
SINR2out=zeros(1,1);
SINRth2out=zeros(1,1);
Qout1=zeros(work,numTxSU);
Qout2=zeros(work,numTxSU);
Xout1=zeros(work,2);
Xout2=zeros(work,2);
rxpacks1=zeros(work,numTxSU);
droppacks1=zeros(work,numTxSU);
rxpacks2=zeros(work,numTxSU);
droppacks2=zeros(work,numTxSU);

```

```

%-----Main looping-----

while cont<work

%--Arrival process and Choosing the most suitable secondary transmitter

    for i=1:2
        eval(['[Aq,cell' int2str(i) '=ArrivalProcess(cell' int2str(i)...
            ',prob,bits,controlqueue);']); % Adding fast fading
        eval(['cell' int2str(i) '=first_transmitter(cell' int2str(i) ');']);
    end

    cell1.whoTxOtherCell=cell2.whoTx;
    cell2.whoTxOtherCell=cell1.whoTx;

%----- Data rate optimization and testing data rate -----

    for i=1:2
        if eval(['cell' int2str(i) '.whoTx~=0']);
            eval(['cell' int2str(i) '=DataRateOptimization(f,cell' int2str(i)...
                ');']);
            eval(['Ropt' int2str(i) '= [Ropt' int2str(i) ',cell' int2str(i)...
                '.ropt'];']);
            eval(['[cell' int2str(i) ',SINR' int2str(i)...
                ',SINRth' int2str(i) '=test_datarate(cell' int2str(i) ');']);
            eval(['SINR' int2str(i) 'out=[SINR' int2str(i)...
                'out,SINR' int2str(i) ');']);
            eval(['SINRth' int2str(i) 'out=[SINRth' int2str(i)...
                'out,SINRth' int2str(i) ');']);
        end
    end

%-----Updating parameters-----

    for i=1:2
        eval(['cell' int2str(i) '=interferqueue(cell' int2str(i) ');']);
        if eval(['cell' int2str(i) '.ack==1']);
            eval(['cell' int2str(i) '.Q=dataqueue(cell' int2str(i)...
                '.Q,cell' int2str(i) '.ropt,0,cell' int2str(i) '.whoTx);']);
            eval(['cell' int2str(i) '.ack=0;']);
            eval(['cell' int2str(i) '.ropt=0;']);
            eval(['cell' int2str(i) '.whoTx=0;']);
            eval(['cell' int2str(i) '.whoTxOtherCell=0;']);
        end
    end

    eval(['cell' int2str(i) '=CalculateFastFading(cell' int2str(i) ');']);

```

```

%----- Output -----

    Qout1(cont,:)=cell1.Q;
    Qout2(cont,:)=cell2.Q;
    Xout1(cont,:)=sum(cell1.X(1:numSensors),cell1.X(numSensors+1));
    Xout2(cont,:)=sum(cell2.X(1:numSensors),cell2.X(numSensors+1));
    rxpacks1(cont,:)=cell1.rxpacks;
    droppacks1(cont,:)=cell1.droppacks;
    rxpacks2(cont,:)=cell2.rxpacks;
    droppacks2(cont,:)=cell2.droppacks;

%-----

    cont=cont+1;
end

pos1=cell1;
pos2=cell2;
if keep_param==0
    save('fixed_parameters','pos1','pos2');
end
SINR1av=mean(SINR1out(2:end));
SINRth1av=mean(SINRth1out(2:end));
SINR1av=mean(SINR2out(2:end));
SINRth2av=mean(SINRth2out(2:end));

Ropt1avg=mean(Ropt1(2:end));
Ropt2avg=mean(Ropt2(2:end));
for i=1:numTxSU
    eval(['Qout1' int2str(i) 'avg=mean(Qout1(:, ' int2str(i) '));']);
    eval(['Qout2' int2str(i) 'avg=mean(Qout2(:, ' int2str(i) '));']);
    eval(['percentage_rxpacks1' int2str(i) '=rxpacks1(end-1,' int2str(i)...
        ')/(rxpacks1(end-1,' int2str(i)')+droppacks1(end-1,' int2str(i) '))*100;']);
    eval(['percentage_droppacks1' int2str(i) '=droppacks1(end-1,' int2str(i)...
        ')/(rxpacks1(end-1,' int2str(i)')+droppacks1(end-1,' int2str(i) '))*100;']);
    eval(['percentage_rxpacks2' int2str(i) '=rxpacks2(end-1,' int2str(i)...
        ')/(rxpacks2(end-1,' int2str(i)')+droppacks2(end-1,' int2str(i) '))*100;']);
    eval(['percentage_droppacks2' int2str(i) '=droppacks2(end-1,' int2str(i)...
        ')/(rxpacks2(end-1,' int2str(i)')+droppacks2(end-1,' int2str(i) '))*100;']);
end

%This function shows all the figures
graphics()

```

Bibliography

- [1] Joseph Mitola III. *Cognitive radio - An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, Royal Institute of Technology (KTH), 8 May, 2000.
- [2] José María Llorca Beltrán y Alberto Bayo Moya. *Simulator of characterization radio channel losses in Matlab*. Specialization project in Wireless Communications, Norwegian University of Science and Technology (NTNU), December, 2009.
- [3] Saunders, Simon R.; Aragón-Zavala, Alejandro. *Antennas and Propagation for Wireless Communication Systems (2nd Edition)*, John Wiley & Sons, 2007.
- [4] Goldsmith, Andrea. *Wireless Communications*. Cambridge University Press, 2005.
- [5] Pål Håland. *Queue management and interference control for cognitive radio*. Master degree thesis, Norwegian University of Science and Technology (NTNU), June, 2009.
- [6] Kansanen, Kimmo. *Sensing based interference and data queue back-pressure approach for cognitive radio*. Norwegian University of Science and Technology (NTNU).