POLITECNICO DI TORINO

Master thesis in Communications and computer networks engineering

On the Effectiveness of Small Cells in Some Simple HetNet Scenarios

Student:

Emanuele Cabibbo

First Examiner:

Ajmone Marsan Marco Giuseppe

Academic year 2018/19

Abstract

The rapid succession of different mobile network generations and the increase of the mobile traffic, has given rise to a scenario in which different cells are overlaid each other in order to increase the capacity and to avoid a lack of resources on the most crowded regions, like, e.g. a dense urban environment.

So, a process of network densification is required. The latter can be realized with well-positioned micro cells inside a macro cell. Indeed, the micro cells are disposed in order to cover very crowded regions, the so-called hot spots, and to unload the macro cell as much as possible.

This master thesis has the purpose to inspect what, one or more micro cells insertion, will provoke in a macro cell; showing how, even in simple cases, with just one or two micro cells, the impact of the small cells parameters can strongly effects the macro cell performance, not always in an easily predictable way. During the entire work, the cells are modelled as queues that produce a so-called network of queues.

The tools utilized are: a software (Tangram-II) to obtain the data from the different models (one model for each micro-macro cells arrangement has been created), and a calculus environment (FreeMat), similar to Matlab. The OS utilized is a linux distribution: Ubuntu 12.04, 32-bit.

Contents

List of Figures

1 Introduction

Since the dawn of time, since the human species come into the world, the need for communicate among us has been fundamental. So, over the years, new methods of communication have been explored, utilizing always new technologies more and more complex. Since the pigeon, to the smoke signals, to the horns, to the telegraph, telephone and so on so forth. Today, the networking science allows to have, concerning the voice service, very accurate and with an high quality level, communications, and to experience new types of human interconnections through the, so loved, messaging services, such as, to name just a few: Whatsapp, Telegram and Messenger; or video calls, thanks to which, is always possible, not just to hear the voice, but even to see again the desired person on the other side of the world.
But, in general, what's a telecommunication system made of? Its basic elements are: a **Transmitter**, that converts the information to a signal, a **Transmission medium**, that carries the signal, therefore, the infos, and a **Receiver**, that converts back the signal to useful informations.
In a first approximation, however, it's possible to distinguish 2 kind of communication: **wired** or **wireless**. The first, usually, faster and more efficient but with the need for a physical installation of, e.g., a twisted pair copper wire; the second, slower and with a very higher data loss with respect the wired one, just for the fact that, the signal goes through the air, giving, though, the obvious advantage of no need for any wire installation. Another important distinction can be done basing on the method utilized to transmit the signal. Indeed, the latter can be accomplished by an **analog** or a **digital** signal. About the analog one, the signal is varied continuously together with the information, being inevitably affected by a noise (this last, more or less hard to remove, depending on whether the noise is additive or not), whereas, in the digital one, the info is first encoded in a set of discrete values, then sent and decoded at the receiver, making the resistance to noise very high [1]. So, even if, the complexity, in a digital system, increase, the disadvantages are almost negligible with respect its advantages, keeping in mind the qualitative leap that the telecommunication services did in the last years. In our days, indeed, the most predominant technologies are always digital, allowing us comforts, that were just unthinkable with an analog system.
The data communication and networking science have a long history behind, and, in particular, the mobile network importance in the last few years has been exponentially amplified due to his increased utilization that grows every single year. So, it's easily understandable how, even the weight of the design choices of this big network, has significantly increased, and this, bring with it, the need for real "mobile network architects".
Just think that, during only 40 years, 5 different mobile protocols have followed, passing from the first mobile generation (*1G*) to the fifth (*5G*), from the simple, poor, low quality level call, to the world wide web at our hand, from the Roman empire to our days in a heartbeat. So, during all the introduction, starting from the roots, through the historical overview [4], we will trace it to the body of a mobile network, trying to understand its different "languages" (means the protocols) adopted between early 80s till the present.

1.1 Historical background

- **1835:** Samuel Morse develops a *telegraph system* and some year later send the first message, in Morse code, from Washington, D.C., to Baltimore, Maryland. This new technology, composed of dots and dashes, transmitted over copper wires, quickly spread out and miles of telegraph lines were born.

- **'30s:** The first telephone networks have already appeared at the end of the XIX century, Teletypewriter Exchange Service in 1930, and, in that years, all the telegraph services were collected under the public switched telephone network (*PSTN*), that by the second half of the XX century, it could carry international voice call.

- **'50s:** The demand to network the computers is growing more and more. Indeed, up to that point, these terminals where just standalone machines, operating separately. The turning point came right in these years with the IBM succeeding to link 2 digital devices over the analog PSTN using acoustic couplers and telephone sets. These latter operating at 300 bits per second. All in a sudden, voice and data network begin to merge.

- **1962:** The digital communication come to being and the first fax over PSTN is transmitted thanks to the invention of a device, called *modem* (short for modulator/demodulator). This latter has the task to turn the digital data, at one end terminal, to analog data, that crosses through the wire, and finally, to recover the original digital data again at the other end. Digital communication starts to be more convenient, reliable and cheaper than the analog one; its quick diffusion is inevitable.

- **early '60s:** A new transmission technique is invented: *packet switching*, and it will completely replace the *circuit switching*. This second one, indeed, allowed the communication, creating a real electric circuit, through a series of telephony switches. This method was very expensive, because of the exclusive reservation of the circuit by the user, and very inefficient, in fact, just about the 50% of the resources were exploited. In the first one, instead, the data are first digitalized and chopped in packets, then sent from the source to the destination, allowing the coexistence of more conversations in just a single circuit and a more efficient utilization of the network. All these innovations will bring, for the first time, to the idea of computers communicating over a unique, common data, communication mesh.

- **'70s and '80s:** The military project *ARPANET*, predecessor of the modern internet, is born, and one of the most important protocols is created: Transmission Control Protocol/Internet Protocol (*TCP/IP*). Moreover, make their first appearance The Integrated Services for Digital Networks (*ISDN*), set of communication standards, supporting voice, video and data, circuit based, but even allowing packet switching; other technologies, such as, Frame Relay (FR) and Asynchronous Transfer Mode (ATM), and the first analog mobile phone system (*1G*).

- **'90s and modern days:** Different mobile phone generation have followed each other, improving the security of the system (*2G*), introducing new techniques as *CDMA* (*3G*), new resources implementations, higher and higher frequencies, efficiency and bandwidth exploitation (*4G, 5G*).

1.2 Structure of a mobile network

Let's introduce some concepts: a modern mobile network is a communication network where the last link is wireless. The network is made up of different **Base Transceiver Stations (BTS)**, even called, more simply, Base Stations (BS) (**Figure 1**), which are nothing but, antennas, or better known as *transceivers* (because of their capability to transmit and receive at the same time), that cover a specific area called **cell**.

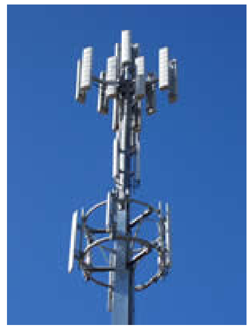


Figure 1: Ensemble of antennas forming a base station

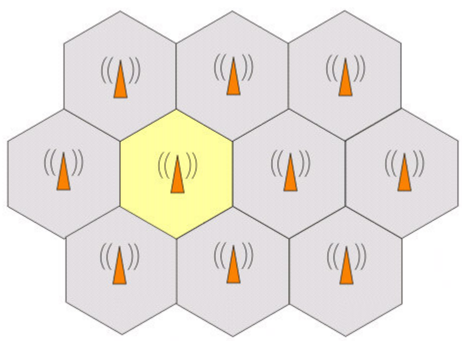


Figure 2: Representation of a cellular network

The latter is classified basing on its coverage (e.g. a micro-cell rarely overcome the radius of 100 m as opposed to the macro cells, that can reach even few kilometers) and theoretically represented with an hexagonal shape, so that, an ensemble of well disposed BS, called network, will produce an image similar to an hive (**Figure 2**). In the figure above the BS are represented in orange, while, the single cell, with the hexagonal grey area.

On one hand, this structure is very practical, on the other hand, during the years, a big problem has been faced because of this organization: interference; and besides, how could have been avoided, since our task is to cover, with radio waves, a large amount of surface without interruptions and below a certain power, at the transceivers, that would be harmful for human beings? For those reasons, usually, a different frequency is assigned to all the adjacent cells, making them use a different *channel* of information. So, the idea is to choose a bunch of different frequencies all assigned to adjacent cells and make then repeat the scheme of those chosen frequencies over and over for all the other further cells. The image below will better explain the concept.

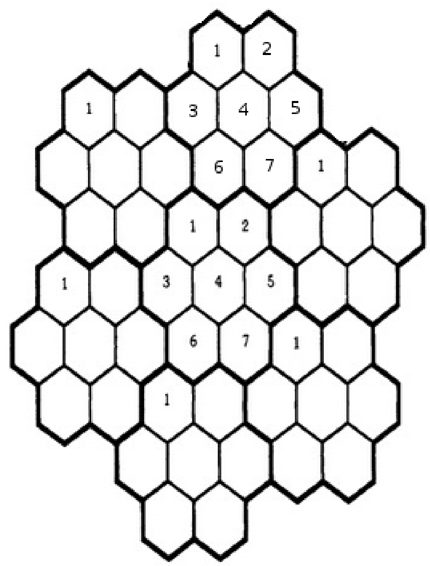


Figure 3: Frequency reuse on a cellular network

Now, a question arise: how can a user pass from one cell to another, i.e. which component make possible an **handover**? The answer is the **Base Station Controller (BSC)**. To give an idea, it can be thought as the brain of the network, connecting the **Mobile Switching Center (MSC)** (in charge of the call set-up/release, routing etc.), to the BTS. And once we got a small overview about a cellular network, we could think: which *language* is used to make users communicate each other? There is more than one way to make this happen? All questions discussed in the next paragraph.

1.3 Protocols and technologies utilized

As already said above, in the last years different protocols have followed each other, let's take a closer look and discuss the technologies exploited by them:

1G: Developed in the late 70s, it's an analog system, the only one [1], that allows just a voice service with poor quality and security (not data encryption supported). The most important technology utilized is *FDMA* (short for Frequency Division Multiple Access), thanks to which, users can access to the same transmission channel without interfering. Indeed, the entire band is divided in smaller sub-bands, one for each user, that can in this way, exploit the network concurrently with all the others (**Figure 4**).

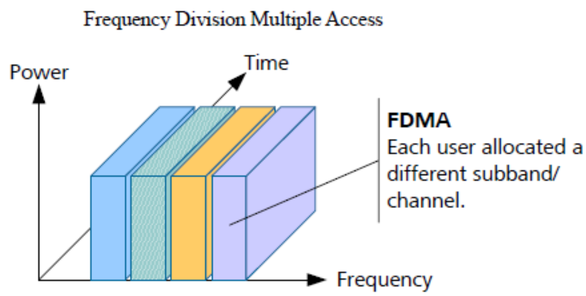


Figure 4: FDMA technique

2G: Digital transmission make income. As said before, the voice is first converted in digital signal, so, transmitted, then reconverted in analog; this the main idea, and the advantages of utilizing this technology are enormous. Launched on the GSM standard with a digital encryption. Its main technology is *TDMA* (Time Division Multiple Access), concept very similar to FDMA, but, this time, the divisions of the resources is operated in time and not in frequency. So, users now will share the same frequency channel but in different time slots (**Figure 5**). It also introduces the SMS (Short Message Service) and MMS (multimedia messages) services. With 2G the union between data and voice begins, especially with the *2.5G* (GPRS: General Packet Radio Service), implementing packet switching, and with *2.75G* (EDGE: Enhanced Data Rates for GSM Evolution), introducing the *8PSK* (8-Phase Shift Keying) encoding and, so, an enhancement of the data rate. But what's the 8PSK? In fact, it's possible to encode the symbols, that have been obtained by sampling the signal, in different ways. In 8PSK we are exploiting 3 bits making the symbols encodable in 8 different ways just shifting the phase.

[1] All the other protocols are digital

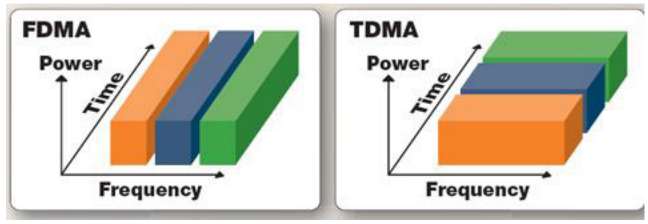


Figure 5: TDMA technique

3G: *UMTS* (Universal Mobile Telecommunications Service) is a typical 3G standard. The latest UMTS release is *HPSA+* (Evolved High Speed Packet Access) capable of reaching theoretically 56 Mbit/s in downlink. From now on, an higher bit rate will be possible, allowing a smoother web surfing. The most innovative technology introduced is *CDMA* (Code Division Multiple Access). This is more complex with respect FDMA and TDMA, indeed it will distinguish different users, not through the frequency or the time, but utilizing pseudo-random or orthogonal codes (called *chip codes*) used to modulate (at the transmitter) and demodulate (at the receiver) the signal (**Figure 6**). When an orthogonal or a pseudo-random code is used? The first, when we can transmit on a synchronous CDMA, indeed, being the mobile-to-base link synchronized, we can exploit the mathematical property of orthogonality to make no interference among them; the second, on an asynchronous CDMA, where we are forced to use statistically uncorrelated codes, given the impossibility of a perfectly synchronization.

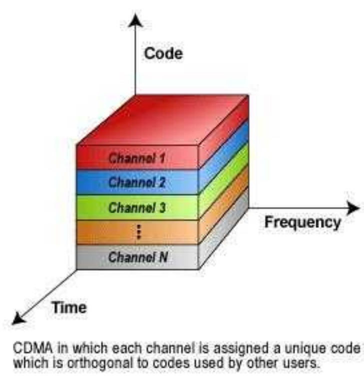


Figure 6: CDMA technique

4G: Developed with the *LTE* (Long Term Evolution) standard and after with LTE-A (LTE-Advanced). This is a revolutionary standard implementing a lot of new technologies, the most important: *re-organization of the resource block*. In time, a resource block is made up of 7 symbols, and 12 sub-carriers in frequency. 7 symbols make a slot in time, 2 slots make a subframe and 5 subframe an entire frame. The not divisible element is called Resource Element (**Figure 7**); *OFDMA* (Orthogonal Frequency Multiple Access), practically an enhancement of FDMA, allowing to better exploit the bandwidth eliminating small portion of frequencies used in FDMA to interval the different sub-bands: the guard bands; *MIMO* (Multiple Input Multiple Access) which multiply the capacity of a radio link using multiple transmitting and receiving antennas.

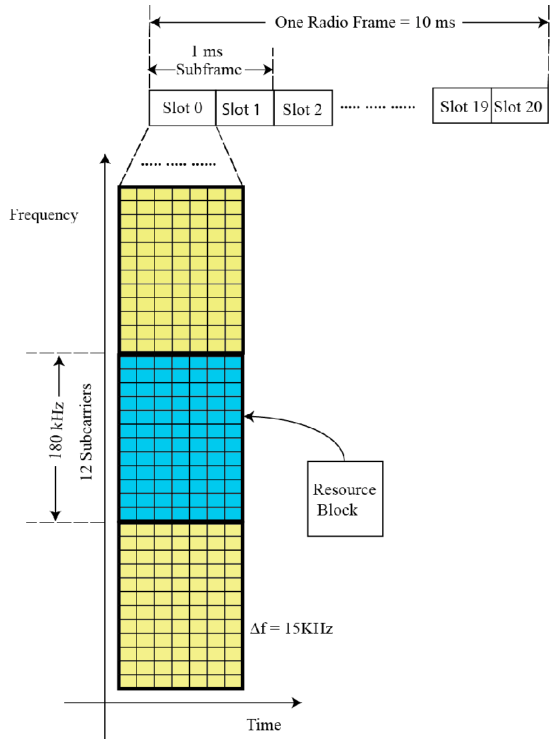


Figure 7: LTE resources scheme

5G: It uses very high frequencies (defining 2 bands: less than 6GHz and greater than 24GHz frequencies) and, therefore, millimiter waves and very high beam directivity, because of its very high susceptibility to be shielded. Moreover, an higher number of antennas is required, and *Edge Computing*, which brings the calculation closer to the location where they are needed, is used; thanks to it, the latency is reduced. *Massive MIMO* is also implemented: a MIMO using even more antennas.

1.4 Thesis goals

Today, who knows how many times you experienced a slowdown of your connection in a very busy area without knowing the reason, without knowing that even the provider resources have an end, without knowing that the overcrowding of the cells is behind this phenomenon and it is one of the biggest issue that afflicts the networks: indeed, it causes the decrease of the available bandwidth, and so, of the bit rate exploitable by each single user. Just to make the concept clearer: we can think about the resources as a cake; the more the people who wants the cake, the more pieces of cake are needed, and the less will be the piece that each of the guest will eat. So, in our case, the more the people utilizing a network the less bandwidth allocated for each of them, meaning, less bit rate for everyone in the network.

Moreover, on top of that, the techniques and the protocols utilized (like 4G, 5G, etc.) are continuously evolving and even mixing among them, making the complexity of the systems higher and higher. Clearly, a rapid scenarios succession have occurred, arriving till our days where the **HetNet** (Heterogeneous Networks), that are networks in which different cells, adopting different technologies, coexist, are commonplace.

In conclusion then, this thesis will try to shed light, through simple HetNet case studies, on the options to choose in order to make the network fast and reliable even when the system is overloaded. In particular, a software (Tangram-II) will be exploited, because of the big amount of calculus and complexity growing more and more, gradually changing the different cases. The insertion of one or more micro-cells, on the so called hotspots (crowded areas inside the coverage ray), in a macro-cell will be analyzed. Furthermore, the results will give us interesting data on the understanding of the real impact of the insertion of these micro-cells in the system (in terms of capacity, probability of loss etc.).

2 Presentation

2.1 Utilized tools

In general, during all the entire work, it is necessary to evaluate the theoretical results, therefore, a modelling and a good calculation environment are required. The tools chosen are, respectively, Tangram-II and FreeMat.

2.1.1 Tangram-II installation

In broad terms, the instructions at *http://www.land.ufrj.br/tools/tools.html* have been followed. To do so, I partitioned an external HDD installing a 32 bit Linux distribution (in this case ubuntu 12.04), because Tangram is incompatible with 64 bit operating systems. Giving, then, the necessary user permission to modify the *etc/profile.d* system folder, I created the necessary file to configure Tangram-II and executed the commands by terminal. Finally I restarted the system, otherwise an installation is required for each time the program is started.

2.1.2 Tangram-II environment

Tangram-II includes mainly 2 modules [2]. Below its main menu (**Figure 8**).

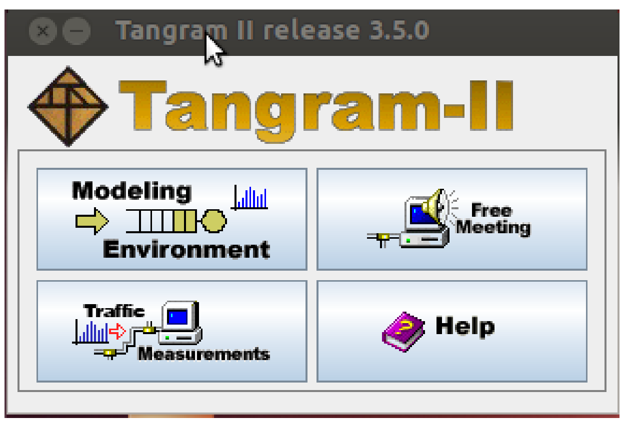


Figure 8: Tangram-II main menu

2.1.3 Modelling Environment

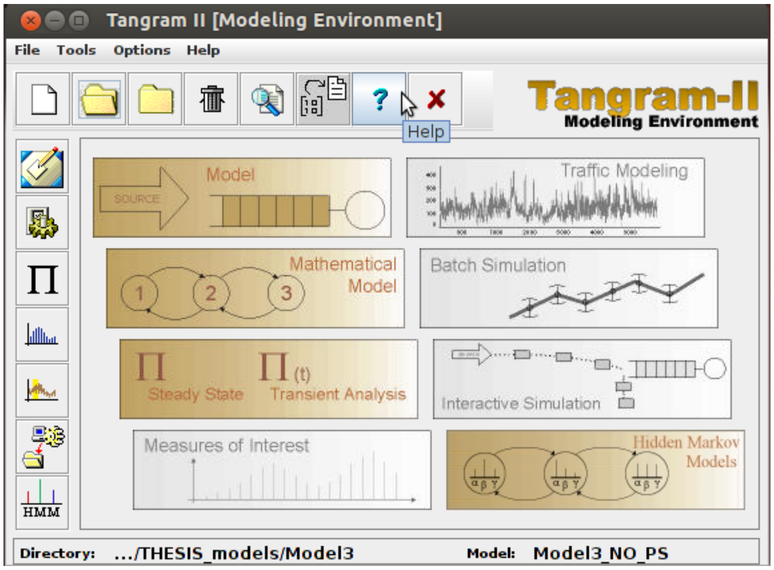


Figure 9: Tangram-II Modelling Environment

Divided in specific sections:

Model Specification: it allows to define a model represented by a collection of objects, that interacts through messages. It uses the TGIF tool to draw the module.

Generate the Mathematical Model: if all the events in the model are exponentially distributed, the Markov Chain is generated. Even certain non-Markovian models can be solved by Tangram-II.

Mathematical Solution: in this case, the model is solved analytically. The Analytical Model Solution Module implements several solution techniques for obtaining steady state and transient behaviours.

Simulation: a discrete events simulator, in batch or interactive mode, and a rare event simulation module are presents. In the interactive mode the user can observe the evolution of the model through animations.

Measures of interest: it allows to compute different measures of interest for the user, like: probability mass function of state variables, probability of a set of states and conditional probabilities. The results can even be plotted using GNUplot.

Traffic Modelling: this module computes first and second order statistics from Markovian models or from traces by the simulator or imported from a file.

2.1.4 Traffic Engineering

It contains:

Traffic Generator:

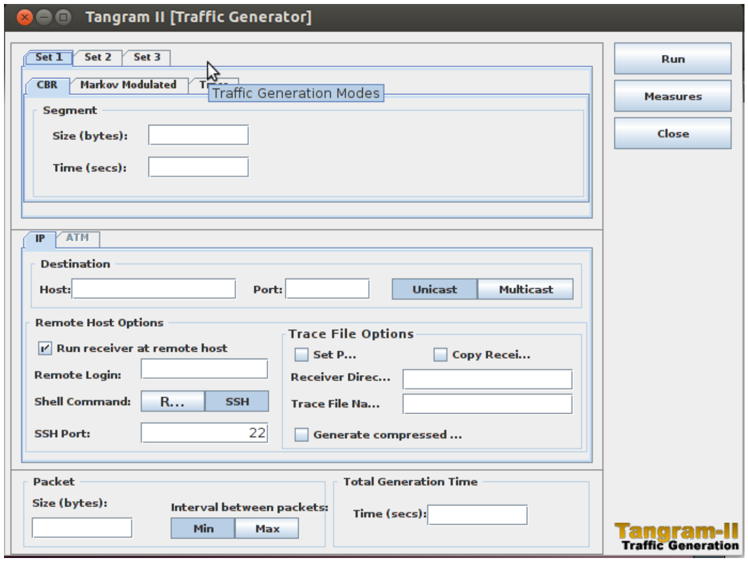


Figure 10: Tangram-II Traffic Generator

Teletraffic engineering's main goal is to be able to predict, with a certain accuracy, the effect generated by the traffic injected, through applications, into the network, and so, to establish if the required QoS (short for Quality of Service) is reached. Indeed with the increasing demand for multimedia applications, become essential to know all the internal characteristics of the network, in order to improve its efficiency and the overall quality of those applications.
The Traffic Generator section, then, is a powerful tool for discovering basic network characteristics like jitter, loss and consecutive loss and even many not-basic ones, like: bottleneck bandwidth, one-way delay, bottleneck buffer size, drop rate, etc. The Traffic Generator developed is able, accordingly to the user specifications, to implement UDP/IP and native ATM.

Traffic Modelling:

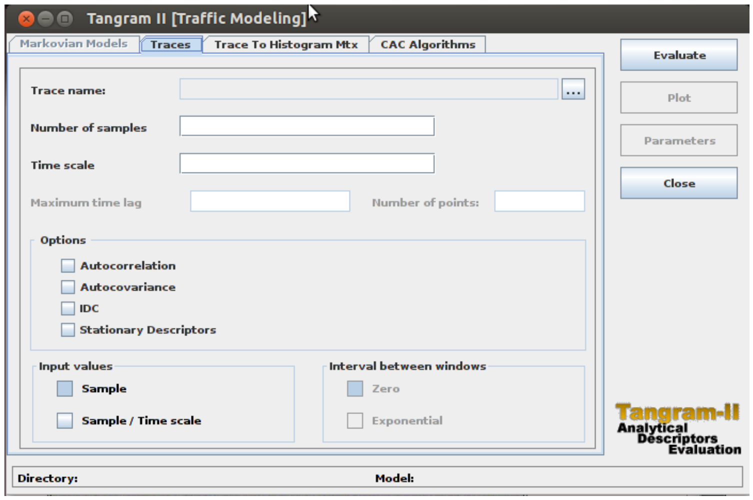


Figure 11: Tangram-II Traffic Modelling

it provides a set of tools to perform a few experiments. It is able to:

- use statistics from real traces
- choose from different traffic models
- calculate descriptors from the models to be able to match parameters and/or verify statistical differences from the model to the measured data
- create a "complete" performance model which includes the traffic model and the resources under study
- solve the model via simulation or analysis
- conduct experiments with traffic generators over a laboratory environment

2.1.5 FreeMat

It is a free environment for engineering and scientific prototyping and data processing [3]. It's similar to MATLAB from Mathworks or IDL from Research Systems but open source. The version used is FreeMat 4.0 for 32 bit operating systems.

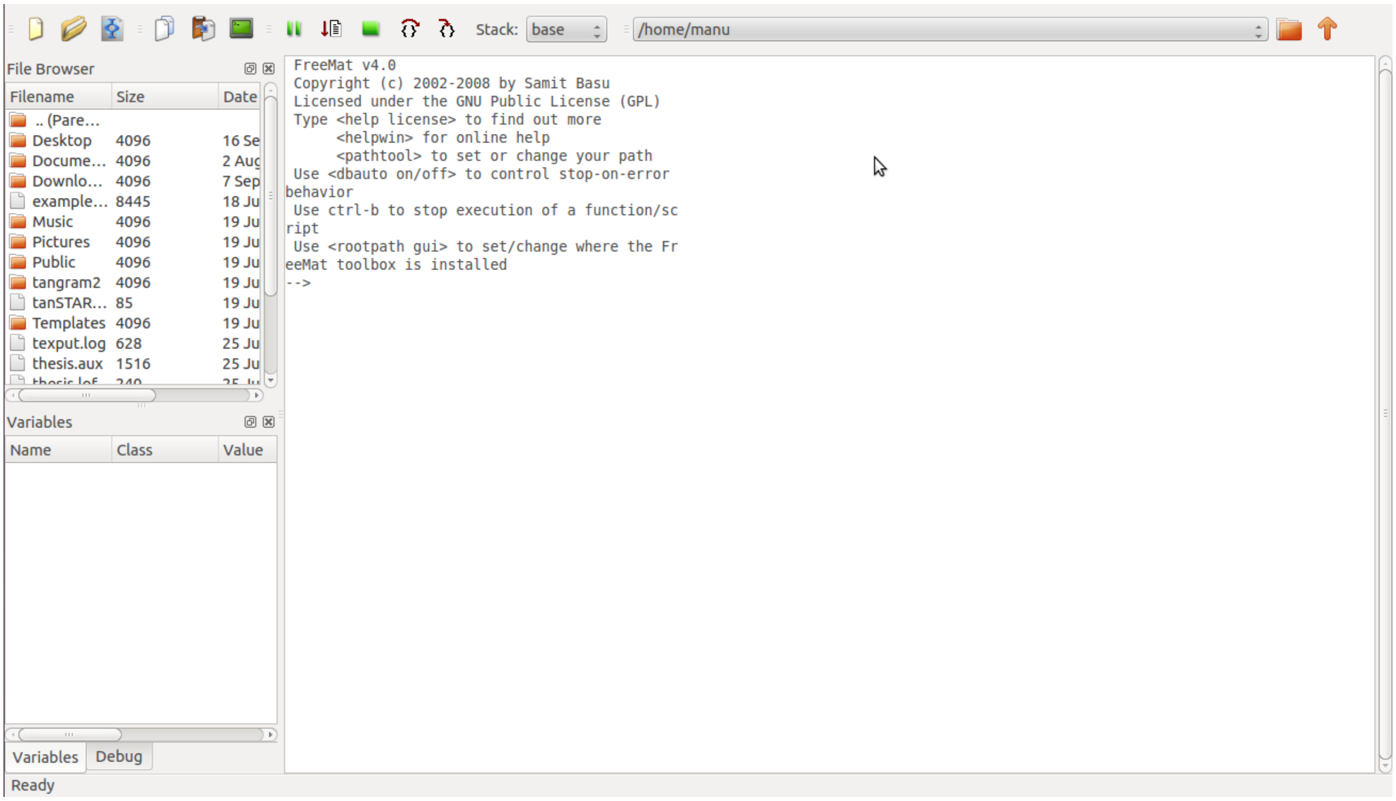


Figure 12: FreeMat menu

In the image above it's shown the FreeMat window with the menu options above, the file browser and the variables window on the left, and the console at the center.

2.2 The cases studied

The following cases, with increasing complexity, are analyzed

2.2.1 Two cells: a macro and a micro one, no processor sharing.

This is the simplest of the cases and it's composed by two cells, called cell A and cell B, respectively, macro and micro cell [5]. The micro cell is strategically positioned, internally to the macro, in order to absorb the traffic peak of an hotspot (**Figure 13**). Indeed, the users can generate service requests that have to be served by the cells, provided that they have enough resources. When there is a lack of resources, in other words, when all the servers that make the cell are busy, the user can't be served and a loss occur. Referring to the previous sentence, in fact, each single cell is modelled as a M/M/m/0, which means, Markovian input and outputs, with a certain number of servers (each server can serve one user), and without a queue, therefore, a lossy system by definition. When a customer is under service, an handover or the completion of the service can happen. In the first case, when the request can't be accommodated (because the cell, to which the request is directed, is full) it is refused, that is, a loss. In the macro cell we can receive an handover from either the other systems or the micro cell, whereas, in the micro cell we can receive just an handover from the macro cell and not from another system, by construction. Moreover, being the micro cell embracing a crowded spot, it's logical to think it with a larger capacity, and with a newer and more efficient technology, than the macro cell's one (e.g. 4G adopted by the macro cell and 5G by the micro one). Another reasonable consideration, given by the reduced dimension of the micro cell, is to choose the time spent in the micro cell lower than the macro one, assuming an "homogeneous" user mobility that doesn't depend on the cell the users are in.

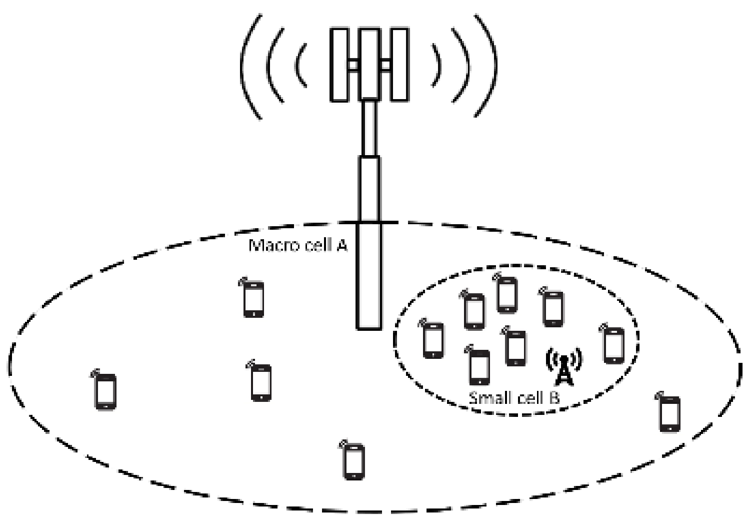


Figure 13: A system with a micro cell inside a macro one

2.2.2 Three cells: a macro and two micro ones, no processor sharing.

Here, 2 micro cells are present inside the macro cell, so 2 sub-cases arise

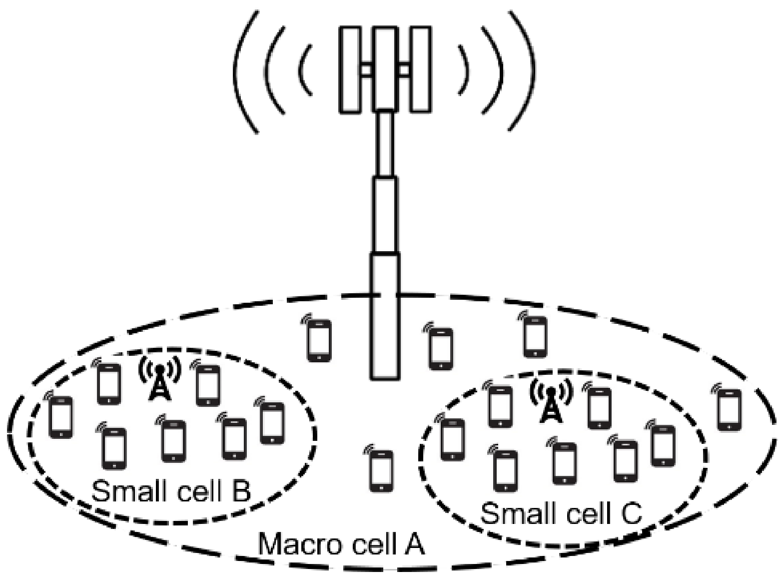


Figure 14: Two disjointed micro cells inside a macro one

1. The 2 micro cells are positioned disjointly, so that, a user is always forced to go through the macro cell in order to reach the other micro cell (**Figure 14**). Similarly, here, all the considerations done in section 2.2.1 are valid even for the second micro cell.

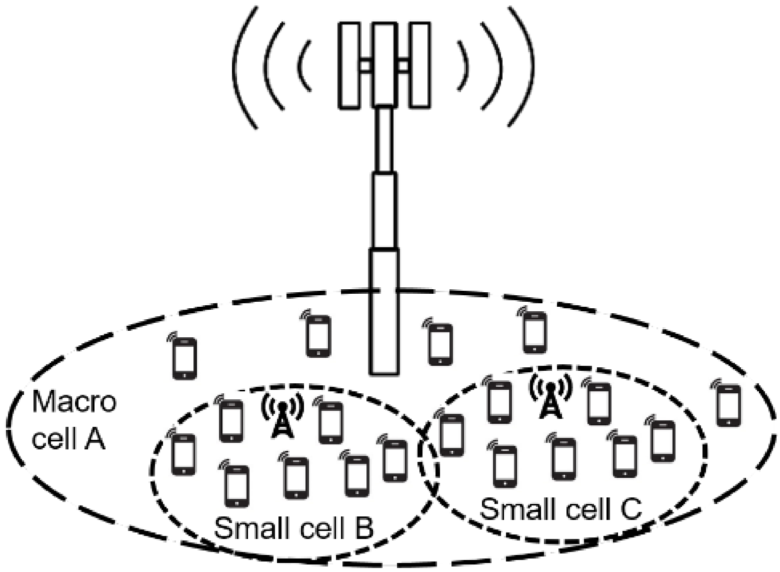


Figure 15: Two joined micro cells inside a macro one

2. The 2 micro cells are placed in a contiguous way in order to directly allow the passage of a user from one to the other small cell (**Figure 15**). Therefore, the only difference with respect the previous cases is that the handover can even happen from one micro cell to the other micro one in addition to the micro-to-macro handover.

2.2.3 Two cells: a macro and a micro one, with processor sharing.

This case may seems identical to the one of the section 2.2.1; indeed, the situation is equal to the one depicted in **Figure 13** but the modelling concept of the cell is completely different.
In fact, all the previous cases come from a **circuit-like** vision of the serving cells, from a no processor sharing idea; if we have to represent a more modern system we should keep in mind a processor sharing (abbreviated with **PS**) scheme, that means, to work in **packet mode**, but what's the difference? Up to now the system reserved to each user a predefined and fixed "slot" of resources, now, the resources are not divided in slots in advance: if just one user is being served I'll give him, potentially, the 100% of the resources available in the server, not just a small portion; to increasing of the customers, I'll divide accordingly my entire resources.
A trivial example can be done to explain the concept: in a party some friends have to eat a cake and the number of guests isn't known. In a no-processor-sharing view the cake is divided before in a certain number of pieces, when the pieces of cake finish, no one else can eat; in a processor-sharing view, the cake isn't divided in advance, but, gradually that the guests arrive to the party: if there is only one guest, potentially, he can eat all the cake, when a second guest arrives they can have half of the cake for each and so on so forth.
It's easily understandable that, this way to think the cell is more complicated but even more efficient, because of its capacity to adapt the resources basing on the customers' number.

2.2.4 Three cells: a macro and two micro ones, with processor sharing.

This model implements the same ideas and concepts introduced in section 2.2.3. The only difference is in the total number of cells: two micro cells are inserted inside the macro one for a total of three cells.
Obviously, like in 2.2.2, the two micro cells can be jointed or disjointed. Both cases are handled.

3 Case 1

Here, the scenarios introduced in subsection 2.2.1 will be discussed.
As you might imagine, the first thing to deal with, is the creation of the model in Tangram-II; to do this, Tangram uses the third part program Tgif.
At a first approximation, I decided to look just at an isolated system, made up of just a macro and a micro cell, not surrounded by many other identical systems (section 3.1.1). Then, I inserted this model in a wider context full of other systems influencing one another (section 3.1.2).
Let's start looking at the general structure

3.1 The model

In general, we can have arrivals at cell A or at cell B

- Arrivals at cell A are made up of 3 components:
  1. the exogenous arrivals representing the requests generated by the users in the macro cell
  2. the handovers coming from the small cell B
  3. the handovers coming from other macro cells in other systems (their rate is set to zero in the following section 3.1.1)
- Arrivals at cell B comprise:
  1. the exogenous arrivals modelling the users' requests in the micro cell
  2. the handovers coming from the macro cell

Obviously, all these arrivals are characterized by a rate (λ_A, λ_{BA}, λ_{HA}, λ_B, λ_{AB}), in particular, the arrivals from the small cell B to cell A and vice versa correspond to a fraction of the departures from cell B or A.
About the services, instead, we notice how, regarding both cells, completion time depends on 2 main factors:

1. the time necessary to complete the request issued by the user (called T_S)
2. the time spent by the user in the cell, in other words, the **dwell time** (denoted by T_D)

These times are assumed to have all an exponential distribution with a certain rate (μ_A and μ_B, for T_{SA} and T_{SB}, μ_{hA} and μ_{hB}, for T_{DA} and T_{DB}).
Therefore, from the previous assumption, the total rate of 2 negative exponential independent random variables is given by their sum, and so, the average service time for the cell A:

$$E[S_A] = \frac{1}{\mu_A + \mu_{hA}} = \frac{1}{\mu_{tA}}$$

where $\mu_{tA} = \mu_A + \mu_{hA}$, the total rate, said before

The nominal load of the queue A is given by:

$$\rho_A = \frac{\lambda_A}{\mu_A}$$

whereas, its normalized nominal load:

$$\rho_{nA} = \frac{\lambda_A}{N_A \mu_A}$$

Properly changing the various indexes, last 3 formulas are valid even for cell B.
Speaking about the cell A, a service, then, can go through:

1. the completion of the service with probability $P_{1A} = \frac{\mu_A}{\mu_{tA}}$
2. the handover towards B with probability $P_{2A} = \frac{\beta \mu_{hA}}{\mu_{tA}}$
3. the handover towards another macro cell, external to this system, with probability $P_{3A} = \frac{(1-\beta)\mu_{hA}}{\mu_{tA}}$ where β is equal to the probability of an handover from the macro to the small cell.

whereas, for cell B:

1. the completion of the service with probability $P_{1B} = \frac{\mu_B}{\mu_{tB}}$
2. the handover towards A with probability $P_{2B} = \frac{\mu_{hB}}{\mu_{tB}}$

3.1.1 Unbalanced system

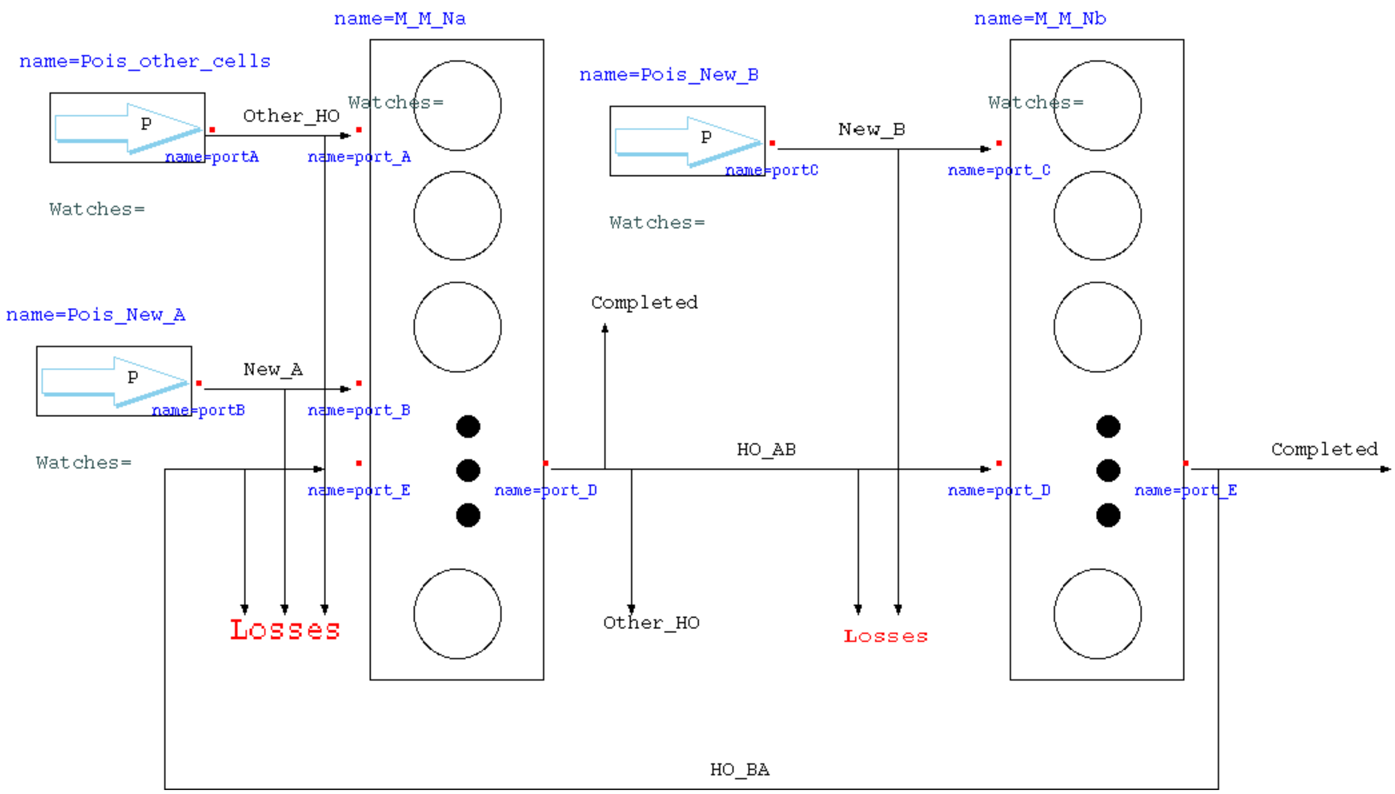


Figure 16: Unbalanced Tangram model for the case in Figure 13

Thanks to the fact that we can easily trace the model to a **Continuous-Time Markov Chain** (CTMC), which can be solved with standard techniques, we can give a mathematical expression to measurements like the probability of loss.
Indeed, the average number of customers on both cells is:

$$E[n_A] = \sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} k_A \pi_{k_A,k_B} \qquad E[n_B] = \sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} k_B \pi_{k_A,k_B}$$

so, the loss probabilities:

$$P_{lossA} = \frac{\sum_{k_B=0}^{N_B} (\lambda_A + \lambda_{hA} + k_B \mu_{hB}) \pi_{N_A,k_B}}{\sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} (\lambda_A + \lambda_{hA} + k_B \mu_{hB}) \pi_{k_A,k_B}} \tag{3.1}$$

$$P_{lossB} = \frac{\sum_{k_A=0}^{N_A} (\lambda_B + k_A \mu_{hA} \beta) \pi_{k_A,N_B}}{\sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} (\lambda_B + k_A \mu_{hA} \beta) \pi_{k_A,k_B}} \tag{3.2}$$

Note that, all those considerations, are given by a circuit-like view of the BS. Newer BS can work in packet mode, dividing the resources among active services without giving a predefined portion to each user (i.e. theoretically, if just a single user is active in a modern BS, it will exploit the entire capacity).
Therefore, in conclusion, the Markovian model must account for those differences, but, the expressions for the expected number of customers and the loss probabilities, remain the same as in the formulas described before.
Now, referring to the **Figure 16**, we can distinguish different objects in the model:

- **M_M_Na**: represents the macro cell made up of Markovian arrivals and outputs, services exponentially distributed with a certain rate, and a maximum number of **Na** servers equal to 16. When all the cell's servers are busy, a loss occur.
- **M_M_Nb**: the micro cell, is implemented like the M_M_Na, except for the maximum number of servers, reasonably set to 64, given that, the micro cell has to serve a crowded hotspot.
- **Pois_New_A**, **Pois_New_B** and **Pois_other_cells**: the 3 exogenous Poisson sources. Notice that, in this unbalanced case (where basically "unbalanced" mean, like said above, not inserted in a network of other identical systems), the generation rate of Pois_other_cells source is set to zero
- **the directional arrows**: keeping in mind **Figure 13**, mean: how a user can go from one cell to the other, the exogenous arrivals and the losses.

3.1.2 Balanced system

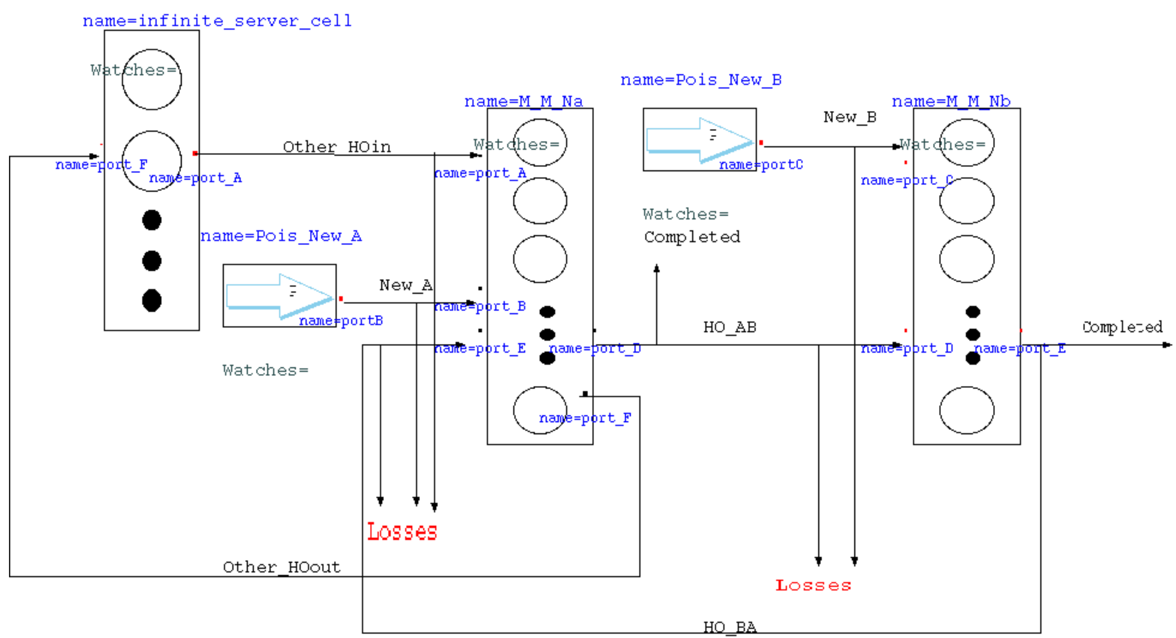


Figure 17: Balanced Tangram model for the case in Figure 13

This model is very similar to the previous one, its only difference is the implementation of the handovers coming from other systems, and then, the introduction of the **infinite_server_cell** (**Figure 17**): a dummy cell inserted with just the task of simulating these exogenous arrivals, with a certain rate, similar to the service rate of the handovers exiting from cell A. Indeed, instead of iteratively derive the handover entering in cell A, with a fixed point iteration, setting the rate equal to the served users, going out from **port_F**, the presence of a cell, relatively slow (with lower service rate) comparing to the others, spaces in time this kind of arrivals and well approximate these users.
Then, the loss probabilities for cells A and B:

$$P_{lossA} = \frac{\sum_{k_B=0}^{N_B} \sum_{k_x=0}^{N_x} (\lambda_A + k_x \mu_x + k_B \mu_{hB}) \pi_{N_A,k_B,k_x}}{\sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} \sum_{k_x=0}^{N_x} (\lambda_A + k_{x\mu_x} + k_B \mu_{hB}) \pi_{k_A,k_B,k_x}} \tag{3.3}$$

$$P_{lossB} = \frac{\sum_{k_A=0}^{N_A} \sum_{k_x=0}^{N_x} (\lambda_B + k_A \mu_{hA} \beta_{AB}) \pi_{k_A,N_B,k_x}}{\sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} \sum_{k_x=0}^{N_x} (\lambda_B + k_A \mu_{hA} \beta_{AB}) \pi_{k_A,k_B}} \tag{3.4}$$

Moreover, the cell, actually, has not an infinite number of servers, but just 10 users can be served all at once. The last implementation choice is due to the fact that, if the cell would have been a true infinite server one, theoretically, an infinite number of states would have been created, so a Tangram error would have been displayed.

3.2 Code implementation

3.2.1 Tangram-II

Basically, I need for 2 types of objects: the cell and the Poisson source; furthermore, few conceptual operations are necessary in the code, here, the most significant lines common to each model

1. **The generations of the arrivals**:

```
Events=

 event= Packet_Generation(EXP, RATE)
 condition= (TRUE)
 action= {
          msg(portB, all, 0);
        };
```

Figure 18: Arrivals generation inside the Poisson sources

Code (**Figure 18**) written inside the 3 Poisson sources, defining the event **Packet_Generation** with time of occurrence exponentially distributed, having the constant **RATE** has rate. These events are all sent to **portB** or **portC**, basing on which Poisson source we are considering.

```
Messages=
    msg_rec=port_A
    action={
              int busy;
              busy=Busy_Server;
              if(Busy_Server<MaxNumServers)
                 busy=Busy_Server+1;
              set_st("Busy_Server",busy);
           };
    msg_rec=portB
    action={
              int busy;
              busy=Busy_Server;
              if(Busy_Server<MaxNumServers)
                 busy=Busy_Server+1;
              set_st("Busy_Server",busy);
           };
    msg_rec=portE
    action={
              int busy;
              busy=Busy_Server;
              if(Busy_Server<MaxNumServers)
                 busy=Busy_Server+1;
              set_st("Busy_Server",busy);
           };
```

Figure 19: Arrivals to the cells

```
Events=
    event=Service(EXP,Busy_Server*TOT_SERVICE_RATE)
    condition=(Busy_Server>0)
    action=
             {
               int busy;
               busy=Busy_Server-1;
               msg(portD,all,0);
               set_st("Busy_Server",busy);
             }; prob = P_HOab;

             {
               int busy, comp;
               busy=Busy_Server-1;
               set_st("Busy_Server",busy);
             }; prob = P_COMP;

            {
              int busy;
              busy=Busy_Server-1;
              msg(port_F,all,0);
              set_st("Busy_Server",busy);
            }; prob = P_HO;
```

Figure 20: Cells services

2. **The arrivals**: Both, **portB** and **portC**, the handovers coming from the microcell and the ones coming from the other systems, are all connected to the message reception section of the cells through the lines displayed in **Figure 19**. Notice that **portC** is not present in the image because this code is extrapolated from cell A, and just the cell B new arrivals get trough this port. Obviously, the concept is the same for each port: I specify the entrance port, define

a local variable equal to a global one and containing the number of busy servers, then, just if this latter is under the maximum value (e.g. 16 for the macro cell), I increment the local variable and set the value of the global one equal to it.

3. **The services**: Their implementation is showed in **Figure 20**, where an event, with an exponentially distributed time, called **Service**, is defined. Depending on the different probabilities a server is freed and, eventually, e.g. in the case of an handover from cell A to cell B, a new message is directed to the specified port in order to make the cell B receive the handover (trivially, the message represents the user moving from cell A to cell B).

3.2.2 Freemat

After generating, using Tangram-II, all the files (**Figure 21**), containing all the probability of each state, I isolated just the needed data thanks to the help of the **LibreOffice Calc** software, then I used them to obtain the probabilities of loss of the different cells.
The main code is repeated for each point on the X axis ($\lambda_A = 1, 5, 10, 15$) in order to find the loss probability with that certain exogenous arrival rate for cell A. So, I just implemented the formulas (3.1) and (3.3) basing on whether the system is balanced or not (**Figure 22** and **Figure 23**).

```
Model1_balanced_PS.IM.PROB1
#Measure of Interest: PROB1
#PMF[CellA.Customers,CellB.Customers,infinite_server_cell.Busy_Server]
1	2.9175158340e-13	#(0,0,0)
2	4.0046610685e-14	#(0,0,1)
3	2.6778114292e-15	#(0,0,2)
4	1.1481562312e-16	#(0,0,3)
5	3.5041936366e-18	#(0,0,4)
6	8.0487087093e-20	#(0,0,5)
7	1.4547619878e-21	#(0,0,6)
8	2.1775290704e-23	#(0,0,7)
9	2.8382522248e-25	#(0,0,8)
10	0.0000000000e+00	#(0,0,9)
11	0.0000000000e+00	#(0,0,10)
12	9.1697100075e-14	#(0,1,0)
13	1.2520997134e-14	#(0,1,1)
14	8.3353145354e-16	#(0,1,2)
15	3.5610191529e-17	#(0,1,3)
16	1.0836540798e-18	#(0,1,4)
17	2.4834937092e-20	#(0,1,5)
18	4.4833776913e-22	#(0,1,6)
19	6.7106515019e-24	#(0,1,7)
20	8.7567912776e-26	#(0,1,8)
21	0.0000000000e+00	#(0,1,9)
22	0.0000000000e+00	#(0,1,10)
23	5.2277202741e-14	#(0,2,0)
24	7.0133825483e-15	#(0,2,1)
25	4.6096506057e-16	#(0,2,2)
26	1.9538861071e-17	#(0,2,3)
27	5.9198295435e-19	#(0,2,4)
28	1.3537526934e-20	#(0,2,5)
29	2.4431685386e-22	#(0,2,6)
30	3.6629674521e-24	#(0,2,7)
```

Figure 21: Example of a file, containing the state probabilities, generated by Tangram-II

```
%Ploss code (final version)
clc
clear all
close all

%data
lambda_A=[1 5 10 15];
PlossA=zeros(1,4);
miu_hB=0.1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load probabilities
pi_kakb=load('Model1_NO_PS.IM.PROB1');
pi_Nakb=pi_kakb(1041:1105,1);

%create kb vector
kb=[0:64]';

%formula numerator
num=sum((ones(65,1).*lambda_A(1) + kb.*miu_hB).*pi_Nakb);

%formula denominator and Ploss
kb=[0:64]';
kb=repmat(kb,17,1);

den=sum((ones(1105,1).*lambda_A(1) + kb.*miu_hB).*pi_kakb);
PlossA(1)=num/den;
```

Figure 22: FreeMat code: Probability of loss for cell A (unbalanced system)

```
%Ploss code (final version)
clc
clear all
%close all

%data
lambda_A=[1 5 10 15];
PlossA=zeros(1,4);
miu_hB=0.1;
mux=0.01;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load probabilities
pi_kakb=load('Model1_NO_PS.IM.PROB1');
pi_Nakb=pi_kakb(11441:12155,1);

%create kb, kx vectors
kx=[0:10];
kx=repmat(kx',65,1);
kb=[0:64]';
kb=reshape(repmat(kb',11,1),715,1);

%formula numerator
num=sum((ones(715,1).*lambda_A(1) + mux.*kx + kb.*miu_hB).*pi_Nakb);

%formula denominator and Ploss
kx=[0:10];
kx=repmat(kx',1105,1);
kb=[0:64]';
kb=repmat(reshape(repmat(kb',11,1),715,1),17,1);

den=sum((ones(12155,1).*lambda_A(1)+ mux.*kx + kb.*miu_hB).*pi_kakb);
PlossA(1)=num/den;
```

Figure 23: FreeMat code: Probability of loss for cell A (balanced system)

In the figures above it's shown the code (for both, balanced and unbalanced system) for $\lambda_A = 1$. The great FreeMat capability of handle the arrays has been exploited here, much simplifying the code; however, the latter is even the hardest part because a lot of arrays has to be managed. Moreover, it has to be noticed how, just changing from unbalanced to balanced the same model, the complexity of the code rapidly grows due to different dimensioning operations. Indeed, for each operation, the vectors are modelled on the data in order to make the dimensions match each other, avoiding FreeMat errors. Finally all the four probabilities are plotted using a logarithmic scale on the Y axis.

3.3 Results case 1

Values set:

In this small paragraph I'll declare all the values set to obtain the following graphs for the models 3.1.1 and 3.1.2

- $\beta = 1/3$
- **M_M_Na**
  1. Dwell rate (μ_{hA}): $\mu_{hA} = 1$
  2. Service rate (μ_A): $\mu_A = 1$
  3. Max number of servers (Na): $Na = 16$
- **M_M_Nb**
  1. Dwell rate (μ_{hB}): $\mu_{hB} = 0.1$
  2. Service rate (μ_B): $\mu_B = 1$
  3. Max number of servers (Nb): $Nb = 64$
- **Pois_New_A**
  1. rate (λ_A): λ_A variable, its values are $1, 5, 10$ or 15
- **Pois_New_B**
  1. rate (λ_B): $\lambda_B = 20$
- **Pois_other_cells**
  1. rate (λ_{HA}): λ_{HA} variable. it is $\lambda_{HA} = 0$ when the system is unbalanced, and equal to the output of **infinite_server_cell** when the systems is balanced.
- **infinite_server_cell** [2]
  1. Total rate (μ_{hx}): $\mu_{hx} = 1$
  2. Max number of servers (Nx): $Nx = 10$

[2] This object substitutes **Pois_other_cells** in the balanced system

Obtained plots:

First of all, from the unbalanced system, I obtained the probability of loss of the cell A.

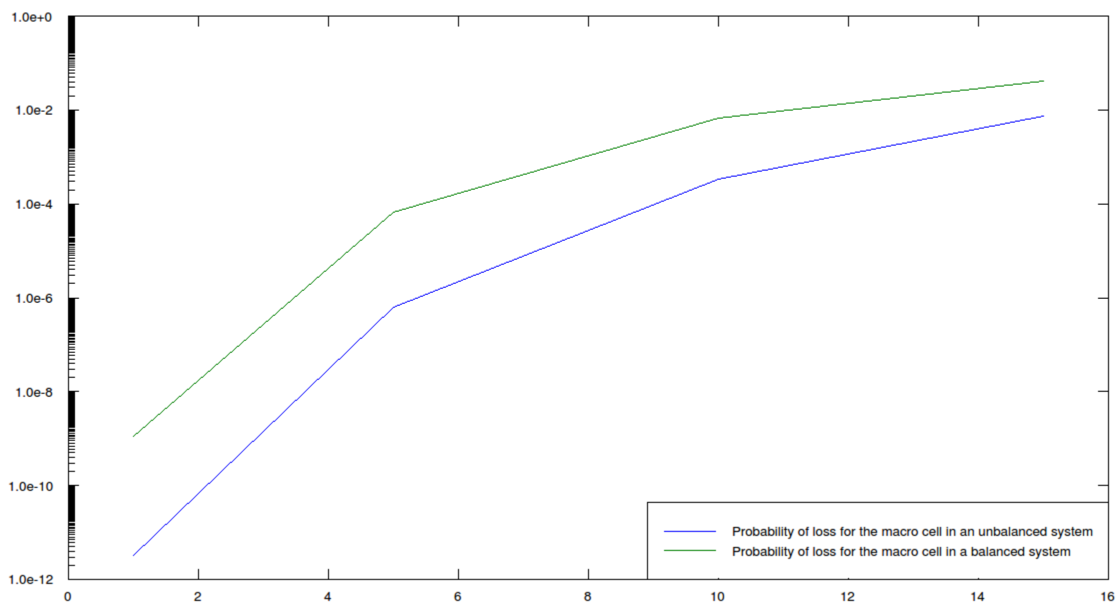


Figure 24: Probability of loss for cell A: balanced versus unbalanced system

On the X axis are represented the exogenous arrivals to the cell A on a linear scale, on the Y axis the probabilities on a logarithmic scale.
As expected, increasing the rate of the **Pois_New_A**, on the X axis, a logarithmic shape is obtained. Indeed, to an increase of the amount of users, entering in a given time in the cell, corresponds a bigger and bigger probability of losing a customer, that however is always smaller compared to the balanced system (**Figure 24**), in which, the additional arrival rate, coming from the other systems, is not equal to zero.
The same happens for the cell B probability of loss.

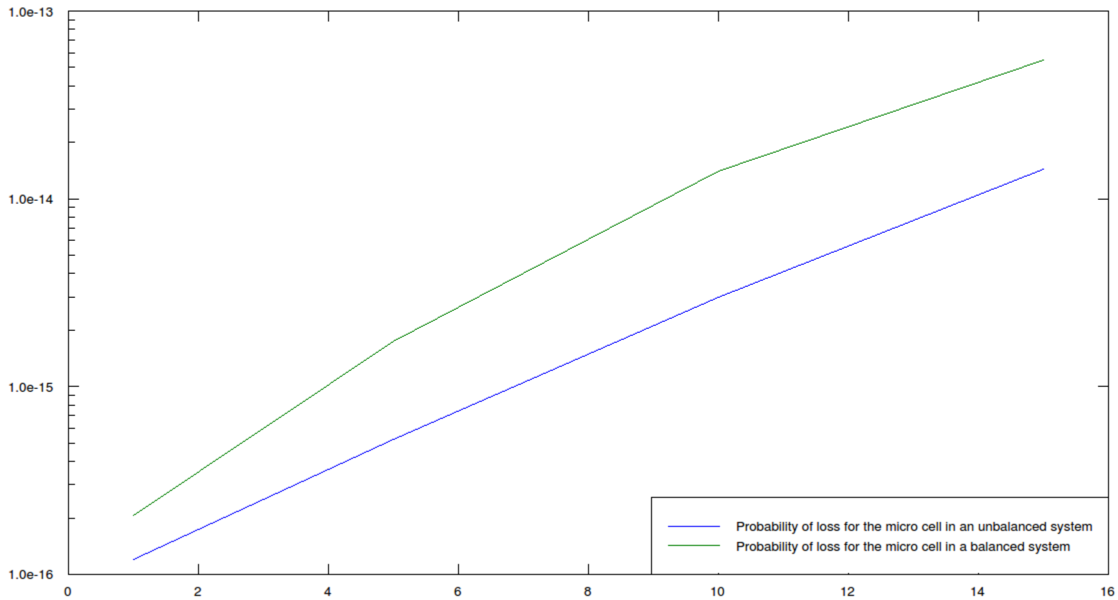


Figure 25: Probability of loss for cell B: balanced versus unbalanced system

The cell B has a lower probability to lose a packet (**Figure 25**) because of its bigger capacity and of the minor impact of the cell A exogenous arrivals on the micro cell. Moreover, the difference between the balanced and the unbalanced case is less relevant than the previous case. This is due to the fact that the most arrivals of the cell B are coming from **Pois_New_B**, representing the new exogenous arrivals to cell B, and not from the handovers coming from the other systems. Notice that, about the blue curve, the probability is so small that the line seems to have a linear trend instead of a logarithmic one.

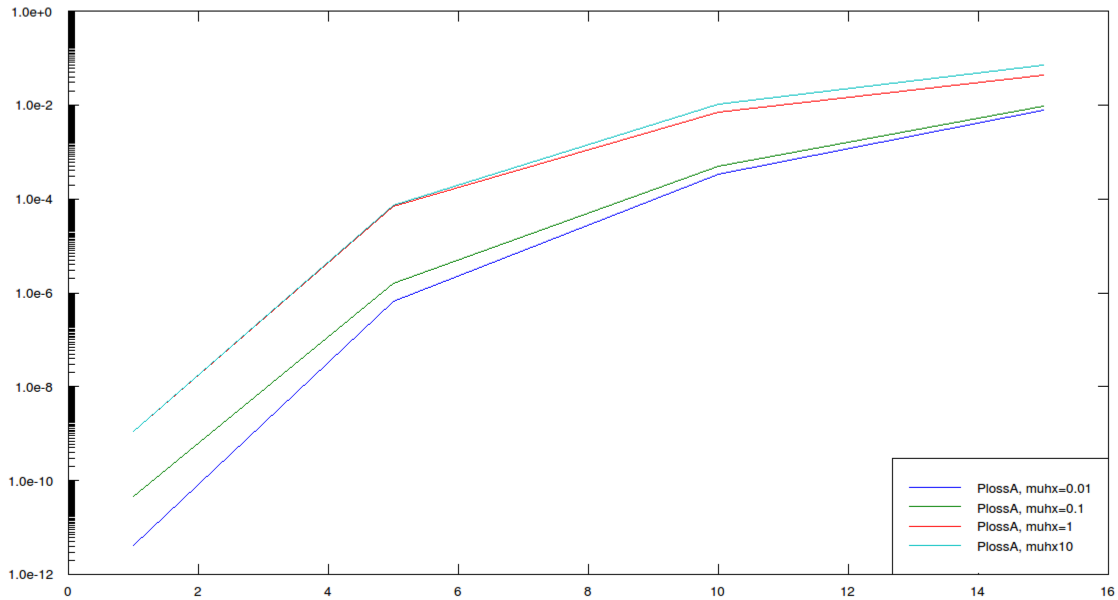


Figure 26: Probability of loss for cell A increasing μ_{hx}

In **Figure 26**, I plotted another interesting experiment showing how the probability of loss for the cell A, raising the μ_{hx} value (it assumes the values $0.01, 0.1, 1, 10$), increases more and more. This is a logical consequence of the increase of customers entering the cell A: the higher the speed of the services by **infinite_server_cell** the higher λ_{HA} and the higher the total cell A load and, with that, its losses.
Consequently, in the cell B, increasing the **infinite_server_cell** rate, the effect is very similar (**Figure 27**) to the previous one due to the increase of the handovers from the macro to the micro cell. Even notice how the differences between the figures in the plot becomes bigger and bigger rising the cell B load.

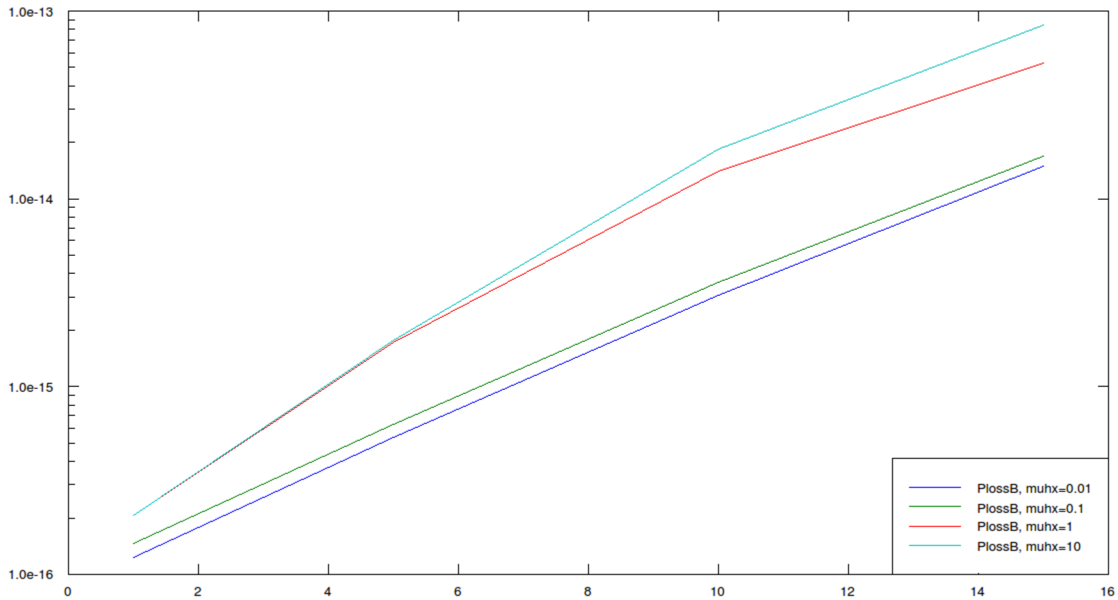


Figure 27: Probability of loss for cell B increasing μ_{hx}

4 Case 2

Here, the cases in subsection 2.2.2 are handled.

4.1 Disjoined micro cells

4.1.1 Model and code implementation

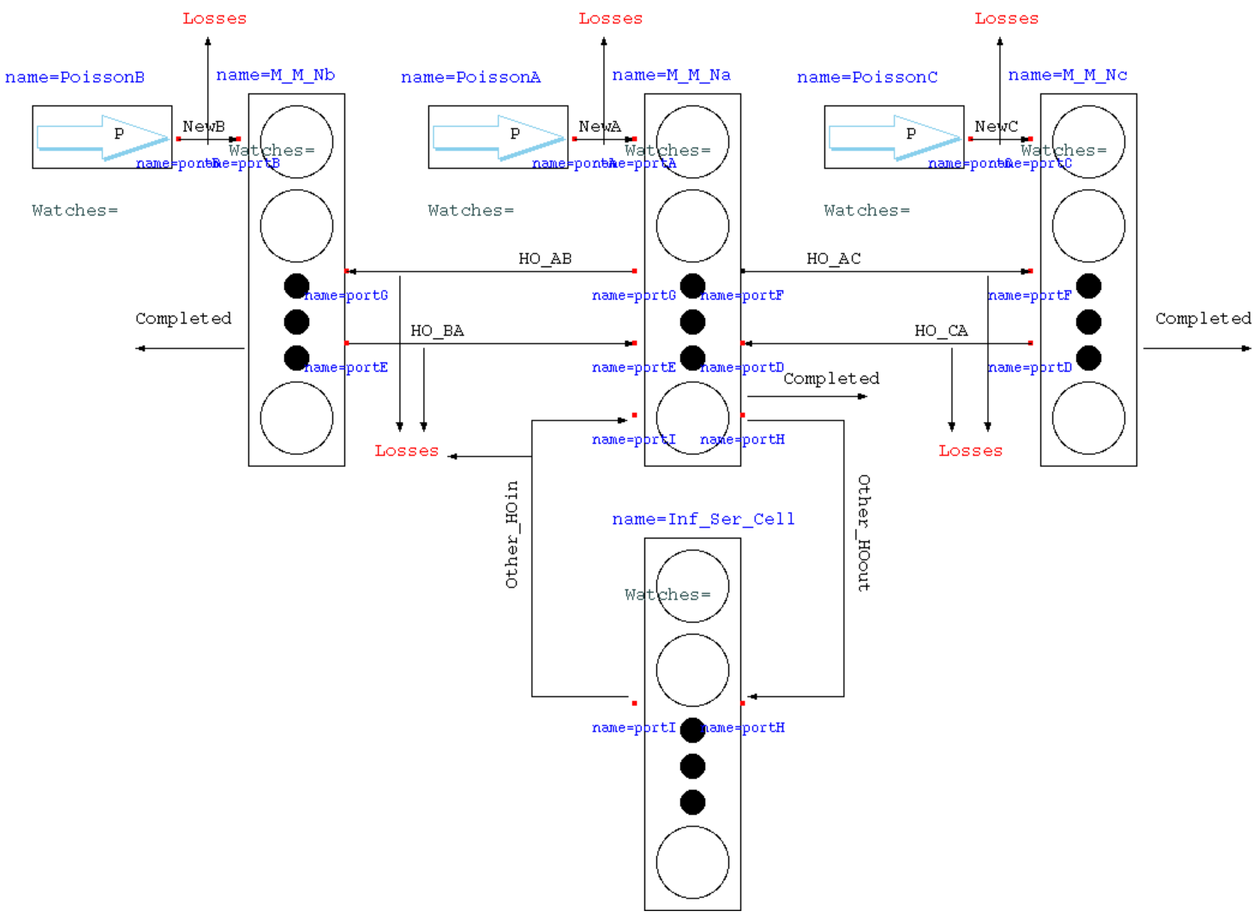


Figure 28: Tangram model for the case in Figure 14

Inspiring to the cell B, the cell C have been constructed. All the formulas introduced in section 3.1.1 are still valid. Moreover, for the loss probabilities:

$$P_{lossA} = \frac{\sum_{k_B=0}^{N_B} \sum_{k_C=0}^{N_C} \sum_{k_x=0}^{N_x} (\lambda_A + k_x\mu_x + k_B\mu_{hB}\beta_{BA} + k_C\mu_{hC}\beta_{CA})\pi_{N_A,k_B,k_C,k_x}}{\sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} \sum_{k_C=0}^{N_C} \sum_{k_x=0}^{N_x} (\lambda_A + k_x\mu_x + k_B\mu_{hB}\beta_{BA} + k_C\mu_{hC}\beta_{CA})\pi_{k_A,k_B,k_x}}$$

$$P_{lossB} = \frac{\sum_{k_A=0}^{N_A} \sum_{k_C=0}^{N_C} \sum_{k_x=0}^{N_x} (\lambda_B + k_A\mu_{hA}\beta_{AB} + k_C\mu_{hC}\beta_{CB})\pi_{k_A,N_B,k_C,k_x}}{\sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} \sum_{k_C=0}^{N_C} \sum_{k_x=0}^{N_x} (\lambda_B + k_A\mu_{hA}\beta_{AB} + k_C\mu_{hC}\beta_{CB})\pi_{k_A,k_B,k_C,k_x}}$$

$$P_{lossC} = \frac{\sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} \sum_{k_x=0}^{N_x} (\lambda_C + k_A\mu_{hA}\beta_{AC} + k_B\mu_{hB}\beta_{BC})\pi_{k_A,k_B,N_C,k_x}}{\sum_{k_A=0}^{N_A} \sum_{k_B=0}^{N_B} \sum_{k_C=0}^{N_C} \sum_{k_x=0}^{N_x} (\lambda_C + k_A\mu_{hA}\beta_{AC} + k_B\mu_{hB}\beta_{BC})\pi_{k_A,k_B,k_C,k_x}}$$

Notice that, here β_{BC} and β_{CB} are equal to zero.

Tangram-II

The model in **Figure 17** has been reformatted just because of visual matters, furthermore it is extended, bearing in mind the **Figure 14**, with another micro cell **M_M_Nc** and a new Poisson source **PoissonC**.
The structure of the code, for the 2 new objects (the Poisson source and the cell), remains identical to the lines of code for the Poisson sources and the cells of the previous models.

FreeMat

Similarly to **Case 1**, the structure of the code remain the same, but the length of the files changes, and with it, the dimensioning of the arrays. The piece of code for $\lambda = 1$ for cell A, B and C, are reported below (**Figure 29**, **Figure 30** and **Figure 31**).

```
%Ploss code (final version)
clc
clear all
format long
%data
lambda_A=[1 5 10 15];
PlossA=zeros(1,4);
miu_hB=0.1;
miu_hC=0.1;
miux=1;
BetaBA=1/2;
BetaCA=1/2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load probabilities
pi_kakbkc=load('Model2_NO_PS.IM.PROB1');
pi_Nakbkc=pi_kakbkc(50865:54043,1);
%create kb kc kx vectors
kb=[0:1:16]';
kb=reshape(repmat(kb',187,1),3179,1);
kc=[0:1:16]';
kc=repmat(reshape(repmat(kc',11,1),187,1),17,1);
kx=[0:10];
kx=repmat(kx',289,1);
%formula numerator
num = sum((ones(3179,1).*lambda_A(1) + kx.*miux + kb.*miu_hB*(BetaBA) + kc.*miu_hC*(BetaCA)).*pi_Nakbkc);
%formula denominator and Ploss
kx=[0:10];
kx=repmat(kx',4913,1);
kb=[0:1:16]';
kb=repmat(reshape(repmat(kb',187,1),3179,1),17,1);
kc=[0:1:16]';
kc=repmat(reshape(repmat(kc',11,1),187,1),289,1);

den=sum((ones(54043,1).*lambda_A(1) + kx.*miux + kb.*miu_hB*(BetaBA) + kc.*miu_hC*(BetaCA)).*pi_kakbkc);
PlossA(1)=num/den;
```

Figure 29: FreeMat code: Probability of loss for cell A

```
%Ploss code (final version)
clc
clear all
format long
%data
lambda_A=[1 5 10 15];
PlossB=zeros(1,4);
lambda_B=5;
miu_hA=1;
miu_hC=0.1;
miux=1;
betaAB=1/6;
betaCB=1/2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load probabilities
pi_kakbkc=load('Model2_NO_PS.IM.PROB1');

for i=0:16
    pi_kaNbkc(1+(i*187):187+(i*187))=pi_kakbkc(2993+(i*3179):3179+(i*3179));
end
pi_kaNbkc=pi_kaNbkc';

%create ka kc vectors
kc=[0:16]';
kc=repmat(reshape(repmat(kc',11,1),187,1),17,1);
ka=[0:16]';
ka=reshape(repmat(ka',187,1),3179,1);
%formula numerator
num = sum((ones(3179,1).*lambda_B + ka.*miu_hA.*(betaAB) + kc.*miu_hC.*(betaCB)).*pi_kaNbkc);
%formula denominator and Ploss
ka=[0:16]';
ka=reshape(repmat(ka',3179,1),54043,1);
kc=[0:16]';
kc=repmat(reshape(repmat(kc',11,1),187,1),289,1);
den=sum((ones(54043,1).*lambda_B + ka.*miu_hA*(betaAB) + kc.*miu_hC*(betaCB)).*pi_kakbkc);
PlossB(1)=num/den;
```

Figure 30: FreeMat code: Probability of loss for cell B

```
%Ploss code (final version)
clc
clear all
format long
%data
lambda_A=[1 5 10 15];
PlossC=zeros(1,4);
lambda_C=5;
miu_hA=1;
miu_hB=0.1;
miux=1;
betaAC=1/6;
betaBC=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load probabilities
pi_kakbkc=load('Model2_NO_PS.IM.PROB1');

for i=0:288
    pi_kakbNc(1+(i*11):11+(i*11))=pi_kakbkc(177+(i*187):187+(i*187));
end
pi_kakbNc=pi_kakbNc';

%create ka kb vectors
kb=[0:16]';
kb=repmat(reshape(repmat(kb',11,1),187,1),17,1);
ka=[0:16]';
ka=reshape(repmat(ka',187,1),3179,1);
%formula numerator
num = sum((ones(3179,1).*lambda_C + ka.*miu_hA.*(betaAC) + kb.*miu_hB.*(betaBC)).*pi_kakbNc);
%formula denominator and Ploss
kb=[0:16]';
kb=repmat(reshape(repmat(kb',187,1),3179,1),17,1);
ka=[0:16]';
ka=reshape(repmat(ka',3179,1),54043,1);
den=sum((ones(54043,1).*lambda_C + ka.*miu_hA*(betaAC) + kb.*miu_hB*(betaBC)).*pi_kakbkc);
PlossC(1)=num/den;
```

Figure 31: FreeMat code: Probability of loss for cell C

4.1.2 Results case 2, disjoined micro cells

Values set:

- **M_M_Na**
  1. $\beta_{AB} = \beta_{AC} = 1/6$
  2. Dwell rate (μ_{hA}): $\mu_{hA} = 1$
  3. Service rate (μ_A): $\mu_A = 1$
  4. Max number of servers (Na): $Na = 16$
- **M_M_Nb**
  1. $\beta_{BA} = 1$
  2. Dwell rate (μ_{hB}): $\mu_{hB} = 10$
  3. Service rate (μ_B): $\mu_B = 1$
  4. Max number of servers (Nb): $Nb = 16$
- **M_M_Nc**
  1. $\beta_{CA} = 1$
  2. Dwell rate (μ_{hC}): $\mu_{hC} = 10$
  3. Service rate (μ_B): $\mu_C = 1$
  4. Max number of servers (Nb): $NC = 16$
- **Pois_New_A**
  1. rate (λ_A): λ_A variable, its values are $1, 5, 10$ or 15
- **Pois_New_B**
  1. rate (λ_B): $\lambda_B = 5$
- **Pois_New_C**
  1. rate (λ_C): $\lambda_C = 5$
- **infinite_server_cell**
  1. Total rate (μ_{hx}): $\mu_{hx} = 1$
  2. Max number of servers (Nx): $Nx = 10$

The cell C probabilities of having an handover towards the cell A or completing the service, are chosen identical to the ones of the cell B, in order to have a symmetric system.

Obtained plots:

About the probabilities, same considerations as in **Case 1** can be done. In the figures below the shape of the loss probability for the cell A, B and C:

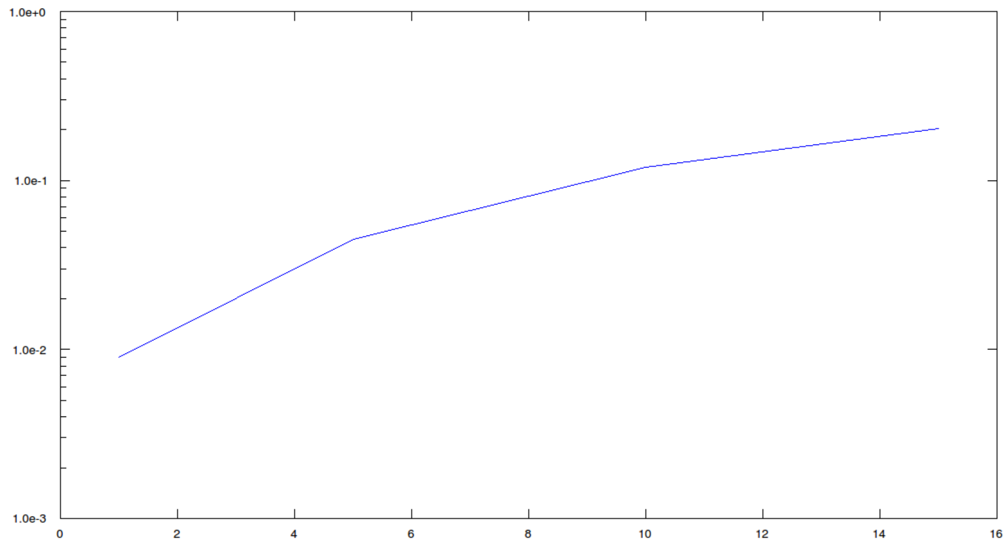


Figure 32: Probability of loss for the cell A

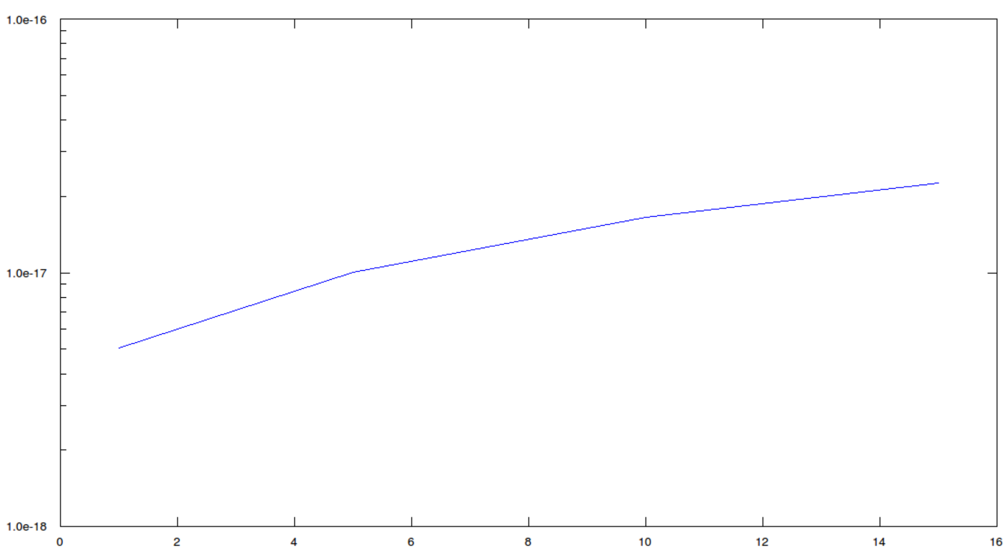


Figure 33: Probability of loss for the cells B and C

In **Figure 32** the losses are considerably high; indeed, the probability of having an handover from one of the two small cell to the macro one is very high, and so, almost the entire traffic going out from the micro cells is added to the arrival rate of the macro.
The opposite phenomenon is, instead, shown in **Figure 33**: the probability of loss for the cells B and C is very low because of the small load of both 2 cells. In fact, the probability of having an handover from the cell A to one of the 2 micro cells is very unlikely, so the major component of the micro cells' arrival rate are just the new arrivals (represented in the model by the objects **Pois_New_B** and **Pois_New_C**). Moreover, note that, obviously, because of the symmetry of the system, the cell B an C probability of loss, have the same identical trend.
Let's now make some experiment on the probability of loss of every cell, fixing the micro cells' dwell rate and setting $\mu_{hB} = \mu_{hC} = 0.1$.

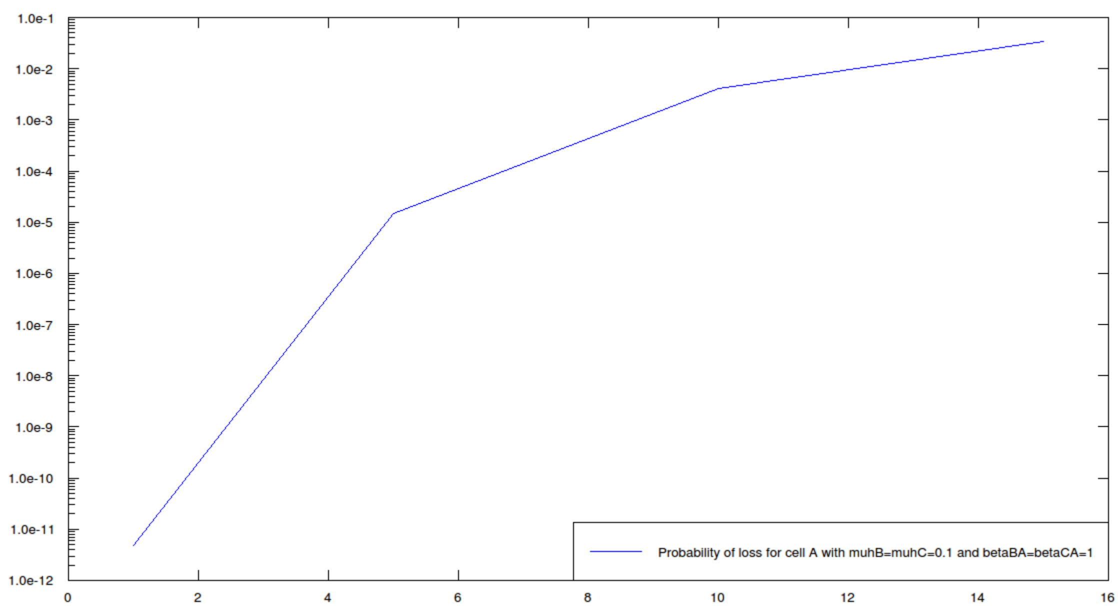


Figure 34: Probability of loss for the cell A, $\mu_{hB} = \mu_{hC} = 0.1$

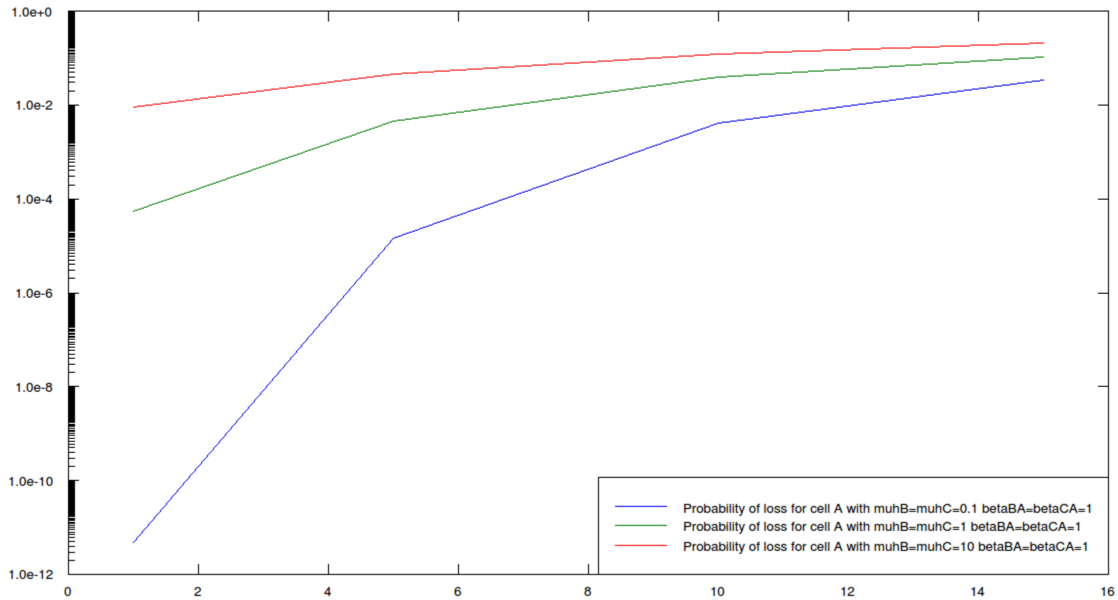


Figure 35: Probability of loss for the cell A varying the service rate of the micro cells

The 2 pictures above regards the cell A, in particular, in **Figure 34** we can appreciate how the probability of loss decreased comparing to **Figure 32**, this is due to the drastic change in the micro cells' dwell rate. Indeed in this case, being $\mu_{hB} = \mu_{hC} = 0.1$, the customers' mobility is very low and the cell A load decreases (and with it the probability of loss) with respect the previous case in which $\mu_{hB} = \mu_{hC} = 10$; the same phenomenon is shown in **Figure 35**, in which it's gradually increased the micro cells' service rate.

About cell B and C, in **Figure 36** below, the probability of loss is always represented with $\mu_{hB} = \mu_{hC} = 0.1$. The losses are much higher than the macro cell because of the micro cells' slower service rate and their greater load.
In **Figure 37** we observe the 3 curves that, even if they maintain more or less the same shape, change very much in terms of order of magnitude. This is due to the increasing dwell rate, and so, even the service rate, that inevitably make the probability of loss decreasing more and more.

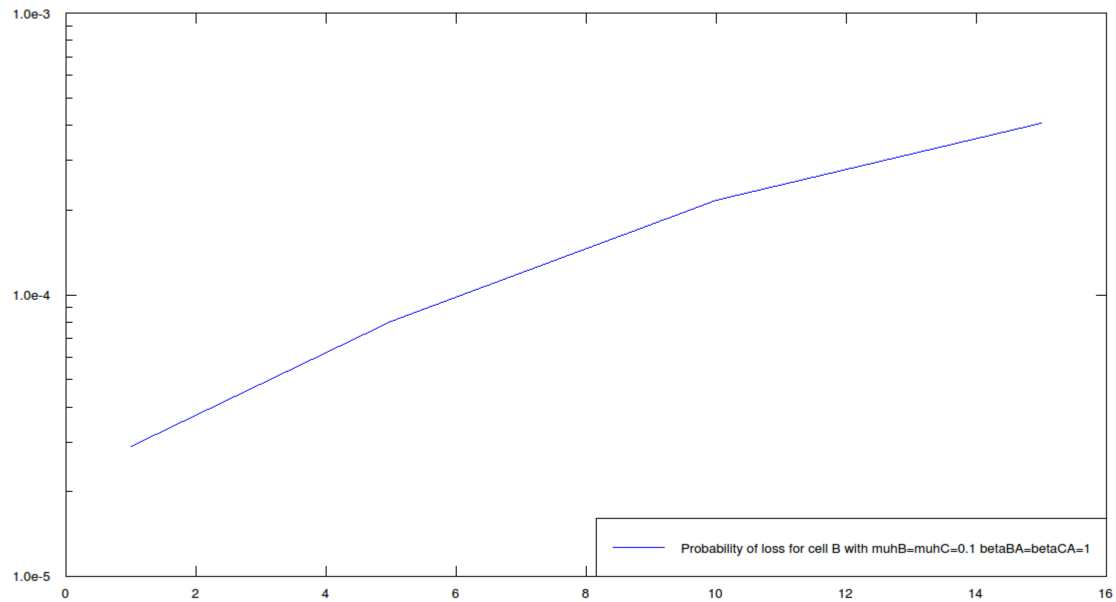


Figure 36: Probability of loss for the cells B and C, $\mu_{hB} = \mu_{hC} = 0.1$

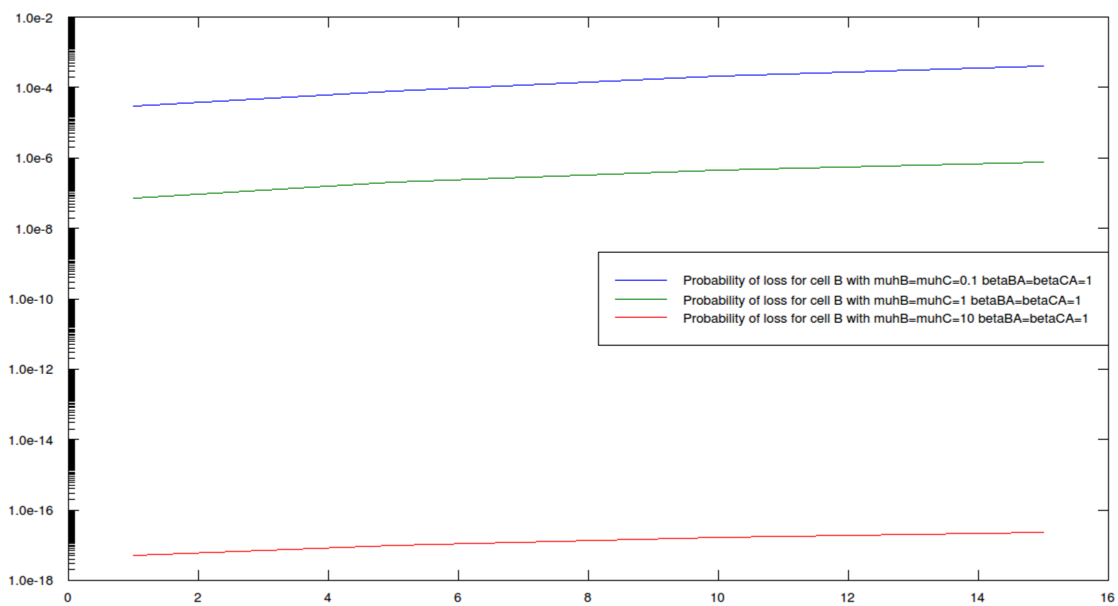


Figure 37: Probability of loss for the cells B,C varying the dwell rate of the micro cells

4.2 Joined micro cells

4.2.1 Model and code implementation

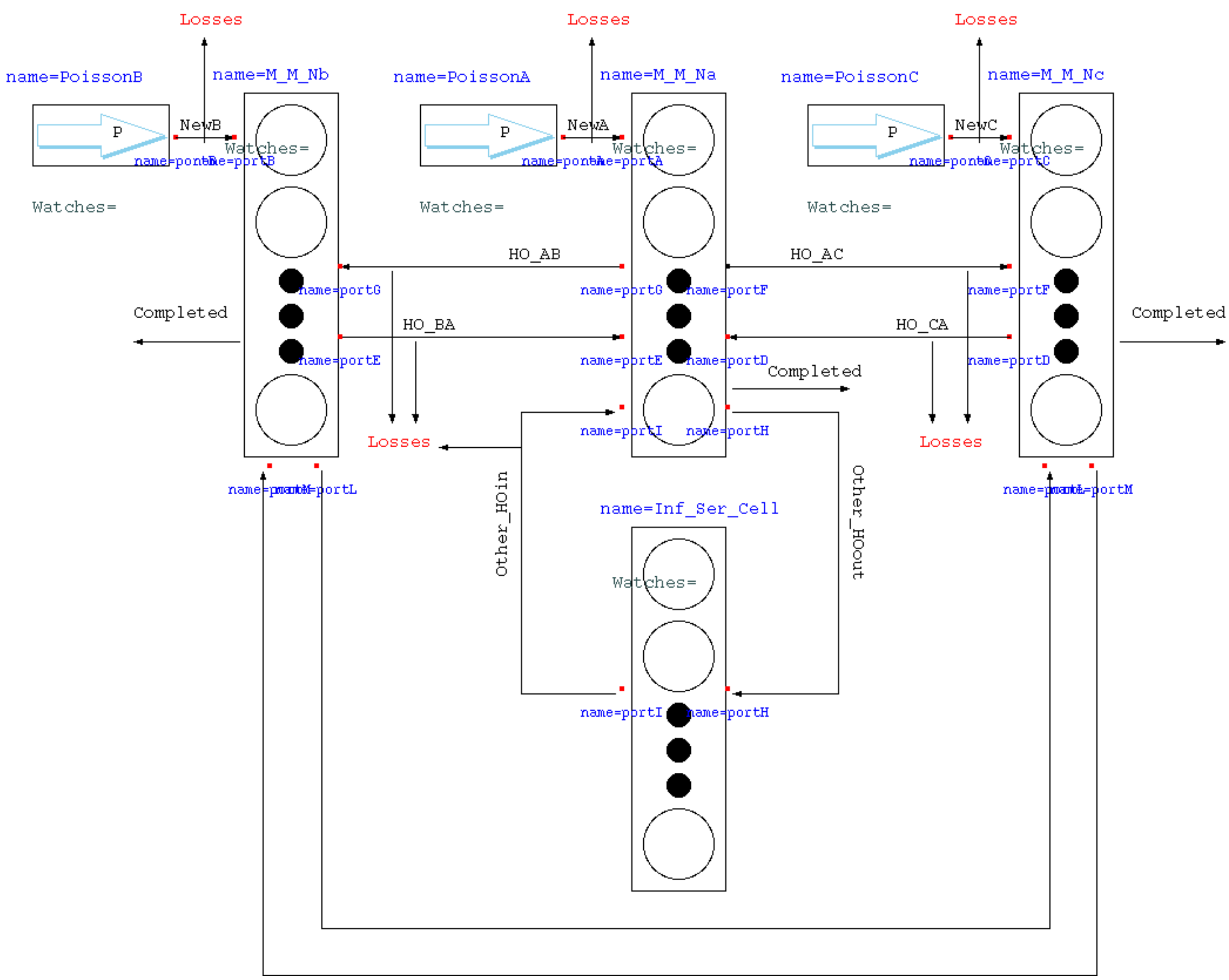


Figure 38: Tangram model for the case in Figure 15

Tangram-II

Same model and code of **Figure 28** but, referring to the **Figure 15**, the micro cells can now communicate each other (**Figure 38**) (β_{BC} and β_{CB} are now not equal to zero).

FreeMat

A very similar code to **Case 2** has been written (**Figure 39**, **Figure 40** and **Figure 41**).

```
%Ploss code (final version)
clc
clear all
format long
%data
lambda_A=[1 5 10 15];
PlossA=zeros(1,4);
miu_hB=0.1;
miu_hC=0.1;
miux=1;
BetaBA=1/2;
BetaCA=1/2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load probabilities
pi_kakbkc=load('Model3_NO_PS.IM.PROB1');
pi_Nakbkc=pi_kakbkc(50865:54043,1);

%create kb kc kx vectors
kb=[0:1:16]';
kb=reshape(repmat(kb',187,1),3179,1);
kc=[0:1:16]';
kc=repmat(reshape(repmat(kc',11,1),187,1),17,1);
kx=[0:10];
kx=repmat(kx',289,1);
%formula numerator
num = sum((ones(3179,1).*lambda_A(1) + kx.*miux + kb.*miu_hB*(BetaBA) + kc.*miu_hC*(BetaCA)).*pi_Nakbkc);
%formula denominator and Ploss
kx=[0:10];
kx=repmat(kx',4913,1);
kb=[0:1:16]';
kb=repmat(reshape(repmat(kb',187,1),3179,1),17,1);
kc=[0:1:16]';
kc=repmat(reshape(repmat(kc',11,1),187,1),289,1);

den=sum((ones(54043,1).*lambda_A(1) + kx.*miux + kb.*miu_hB*(BetaBA) + kc.*miu_hC*(BetaCA)).*pi_kakbkc);
PlossA(1)=num/den;
```

Figure 39: FreeMat code: Probability of loss for cell A

```
%Ploss code (final version)
clc
clear all
format long
%data
lambda_A=[1 5 10 15];
PlossB=zeros(1,4);
lambda_B=5;
miu_hA=1;
miu_hC=0.1;
miux=1;
betaAB=1/6;
betaCB=1/2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load probabilities
pi_kakbkc=load('Model3_NO_PS.IM.PROB1');

for i=0:16
    pi_kaNbkc(1+(i*187):187+(i*187))=pi_kakbkc(2993+(i*3179):3179+(i*3179));
end
pi_kaNbkc=pi_kaNbkc';

%create ka kc vectors
kc=[0:16]';
kc=repmat(reshape(repmat(kc',11,1),187,1),17,1);
ka=[0:16]';
ka=reshape(repmat(ka',187,1),3179,1);
%formula numerator
num = sum((ones(3179,1).*lambda_B + ka.*miu_hA.*(betaAB) + kc.*miu_hC.*(betaCB)).*pi_kaNbkc);
%formula denominator and Ploss
ka=[0:16]';
ka=reshape(repmat(ka',3179,1),54043,1);
kc=[0:16]';
kc=repmat(reshape(repmat(kc',11,1),187,1),289,1);

den=sum((ones(54043,1).*lambda_B + ka.*miu_hA*(betaAB) + kc.*miu_hC*(betaCB)).*pi_kakbkc);
PlossB(1)=num/den;
```

Figure 40: FreeMat code: Probability of loss for cell B

```
%Ploss code (final version)
clc
clear all
format long
%data
lambda_A=[1 5 10 15];
PlossC=zeros(1,4);
lambda_C=5;
miu_hA=1;
miu_hB=0.1;
miux=1;
betaAC=1/6;
betaBC=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load probabilities
pi_kakbkc=load('Model3_NO_PS.IM.PROB1');

for i=0:288
    pi_kakbNc(1+(i*11):11+(i*11))=pi_kakbkc(177+(i*187):187+(i*187));
end
pi_kakbNc=pi_kakbNc';

%create ka kb vectors
kb=[0:16]';
kb=repmat(reshape(repmat(kb',11,1),187,1),17,1);
ka=[0:16]';
ka=reshape(repmat(ka',187,1),3179,1);
%formula numerator
num = sum((ones(3179,1).*lambda_C + ka.*miu_hA.*(betaAC) + kb.*miu_hB.*(betaBC)).*pi_kakbNc);
%formula denominator and Ploss
kb=[0:16]';
kb=repmat(reshape(repmat(kb',187,1),3179,1),17,1);
ka=[0:16]';
ka=reshape(repmat(ka',3179,1),54043,1);

den=sum((ones(54043,1).*lambda_C + ka.*miu_hA*(betaAC) + kb.*miu_hB*(betaBC)).*pi_kakbkc);
PlossC(1)=num/den;
```

Figure 41: FreeMat code: Probability of loss for cell C

4.2.2 Results case 2, joined micro cells

Values set:

- **M_M_Na**
  1. $\beta_{AB} = \beta_{AC} = 1/6$
  2. Dwell rate (μ_{hA}): $\mu_{hA} = 1$
  3. Service rate (μ_A): $\mu_A = 1$
  4. Max number of servers (Na): $Na = 16$
- **M_M_Nb**
  1. $\beta_{BA} = 1/2$
  2. Dwell rate (μ_{hB}): $\mu_{hB} = 10$
  3. Service rate (μ_B): $\mu_B = 1$
  4. Max number of servers (Nb): $Nb = 16$
- **M_M_Nc**
  1. $\beta_{CA} = 1/2$
  2. Dwell rate (μ_{hC}): $\mu_{hC} = 10$
  3. Service rate (μ_C): $\mu_C = 1$
  4. Max number of servers (Nb): $NC = 16$
- $\beta_{BC} = \beta_{CB} = 1/2$
- **Pois_New_A**
  1. rate (λ_A): λ_A variable, its values are $1, 5, 10$ or 15
- **Pois_New_B**
  1. rate (λ_B): $\lambda_B = 5$
- **Pois_New_C**
  1. rate (λ_C): $\lambda_C = 5$
- **infinite_server_cell**
  1. Total rate (μ_{hx}): $\mu_{hx} = 1$
  2. Max number of servers (Nx): $Nx = 10$

Obtained plots:

The probability of loss for the cells considering $\mu_{hB} = \mu_{hC} = 1$ and $\beta_{BA} = \beta_{CA} = 0.5$ (**Figure 42**).

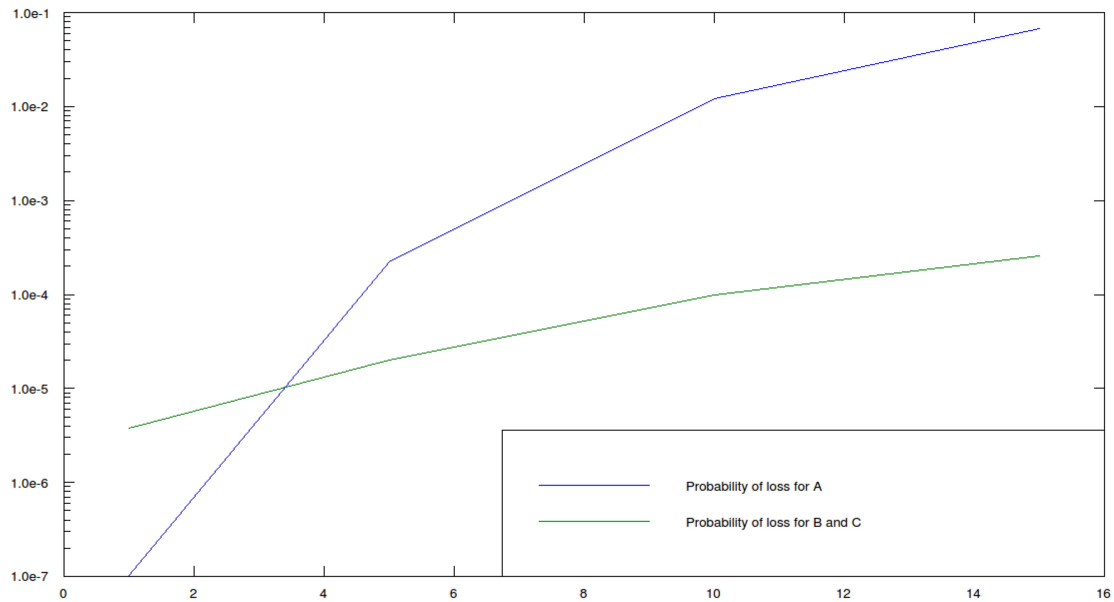


Figure 42: Probability of loss for the cells A, B and C

Being now the model more complex, different significant results have been obtained. In particular, the variations of the loss probabilities of each cell, changing some parameter, is studied.
4 main variables are considered: $\beta_{BA}, \beta_{CA}, \mu_{hB}, \mu_{hC}$.

FIRST EXPERIMENT

What's the maximum variation on the probabilities that we can obtain from these parameters? Let's consider the 2 cases:

1. $\mu_{hB} = \mu_{hC} = 0.1$ and $\beta_{BA} = \beta_{CA} = 0$
2. $\mu_{hB} = \mu_{hC} = 10$ and $\beta_{BA} = \beta_{CA} = 1$

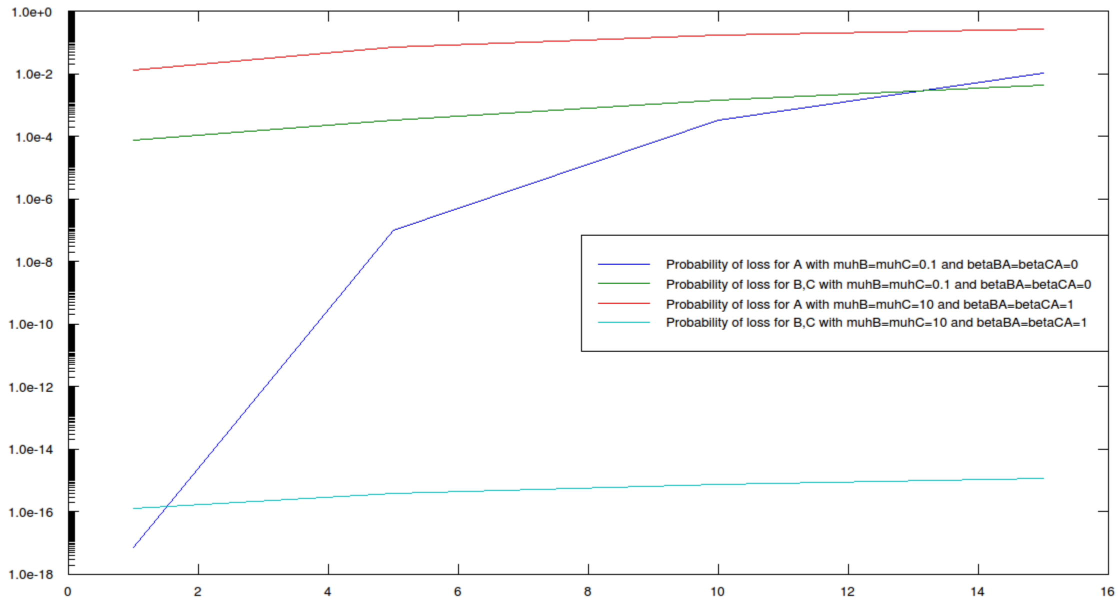


Figure 43: Graphical representation of the probability of loss for the cells A, B and C in case 1 and 2

In **Figure 43**, immediately, we can notice how much different is the behaviour of the macro with the micro cell.
The first has a variable variation, indeed, for an higher exogenous arrival rate (λ_A) the difference on the probability between these 2 extreme cases decrease, meanwhile, the second one seems to have a constant variation on the probability either for $\lambda_A = 1$ or for $\lambda_A = 15$.
This is due to the fact that for the macro cell, in case 2, the exogenous contribution to the arrivals becomes almost negligible because of the intake due to the handovers, that occur with probability equal to 1. And on top of that, the micro cells have an high service rate and are continuously sending customers, affecting the cell A's load, that is maintained high for all the entire experiment.
The cell B and C probability of loss, instead, just changes because of the dwell rate variation.

SECOND EXPERIMENT

I examined the losses for the cell A, fixing the dwell rates of the 2 micro cells but gradually incrementing the entering load coming from the latter.

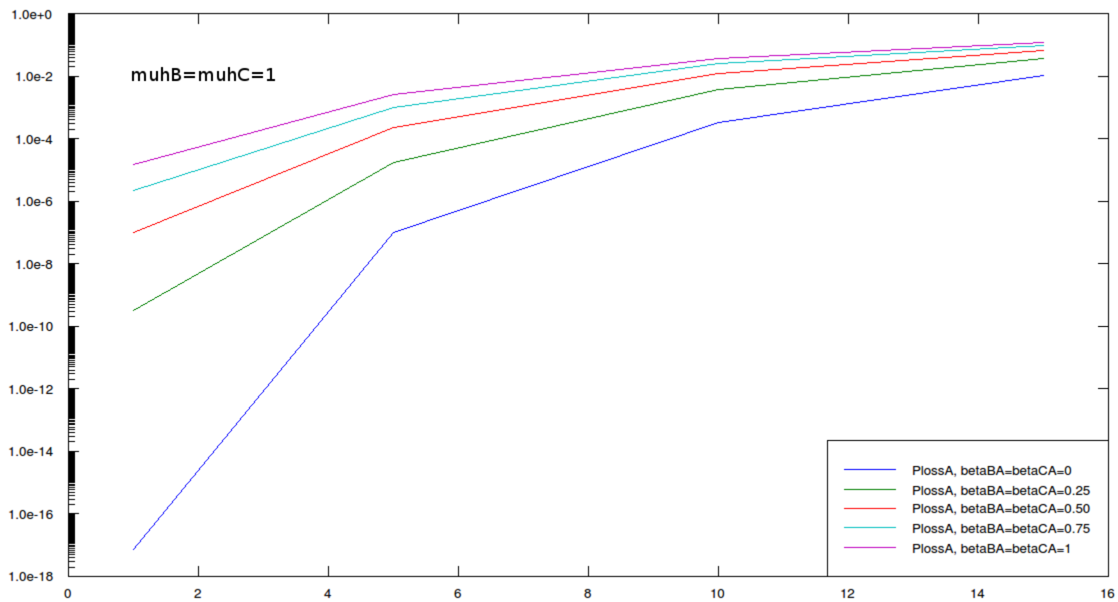


Figure 44: Probability of loss for the cell A incrementing β_{BA} and β_{CA}

Obviously, the cell A probability of loss increase gradually with β_{BA} and β_{CA}

THIRD EXPERIMENT

This time I highlighted how the losses of the cell A increase, fixing the probability of having an handover from the micros to the macro cell, but rising the dwell rates of the 2 micros, and so, their total service rate. So, from the point of view of the small cells: the probability of an handover towards A is always the same but the service is faster and faster, that is, more and more customers are served, increasing the incoming load of the macro cell A.

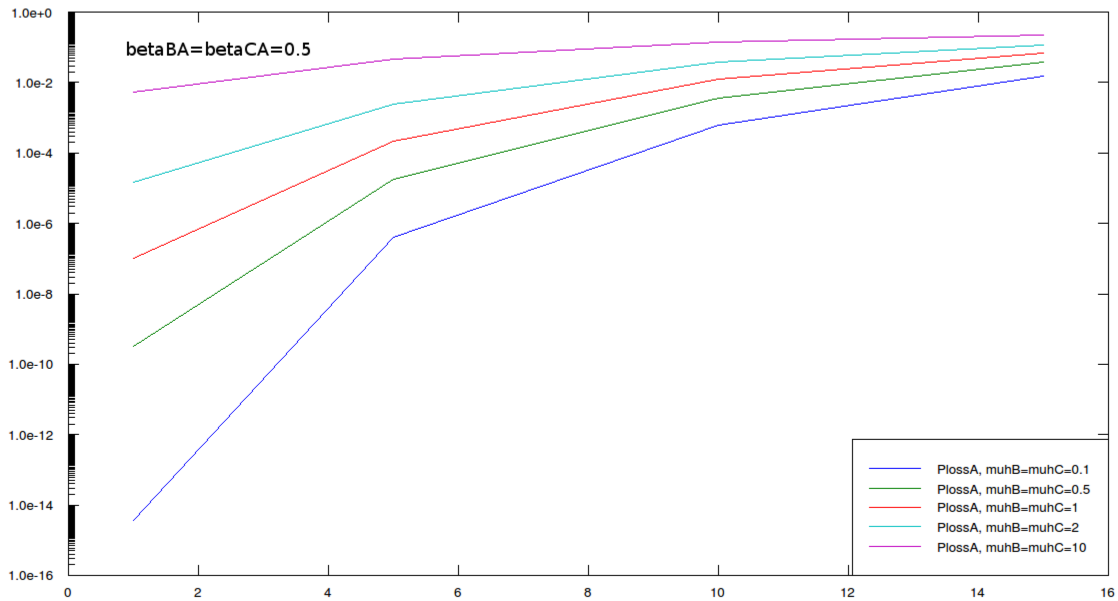


Figure 45: Probability of loss for the cell A incrementing μ_{hB} and μ_{hC}

FOURTH EXPERIMENT

Similar to the second experiment but this time we look at the probability of loss for the cell B and C.

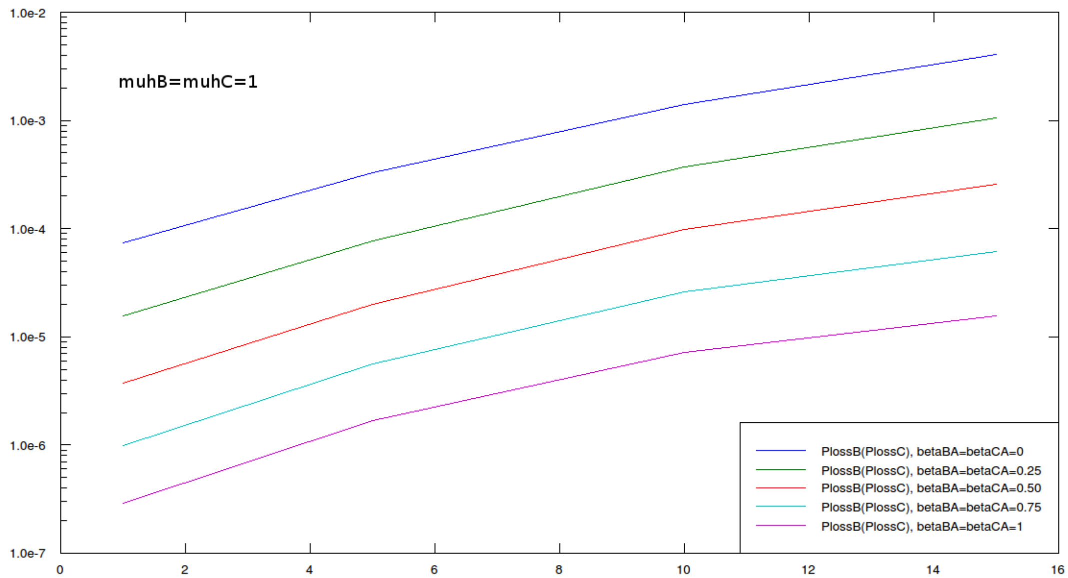


Figure 46: Probability of loss for the cells B and C incrementing β_{BA} and β_{CA}

The difference is evident: the probability, for the micro cells, to send a customer to the cell A is bigger and bigger; in reply, the macro cell will rarely send back a customer to one of the 2 micro cells because of its high probability to complete a service or to have an handover towards other systems. The probability of loss for the micros have no alternative but to decrease every time that β_{BA} and β_{CA} are increased.

FIFTH EXPERIMENT

Similar to the third experiment but this time we look at the probability of loss for the cell B and C. Furthermore, the phenomenon is very similar to the previous experiment. The only difference is that, here, the amount of customers sent to the cell A increase because we increase the service rate of the micro cells and not their probabilities of having an handover towards A.

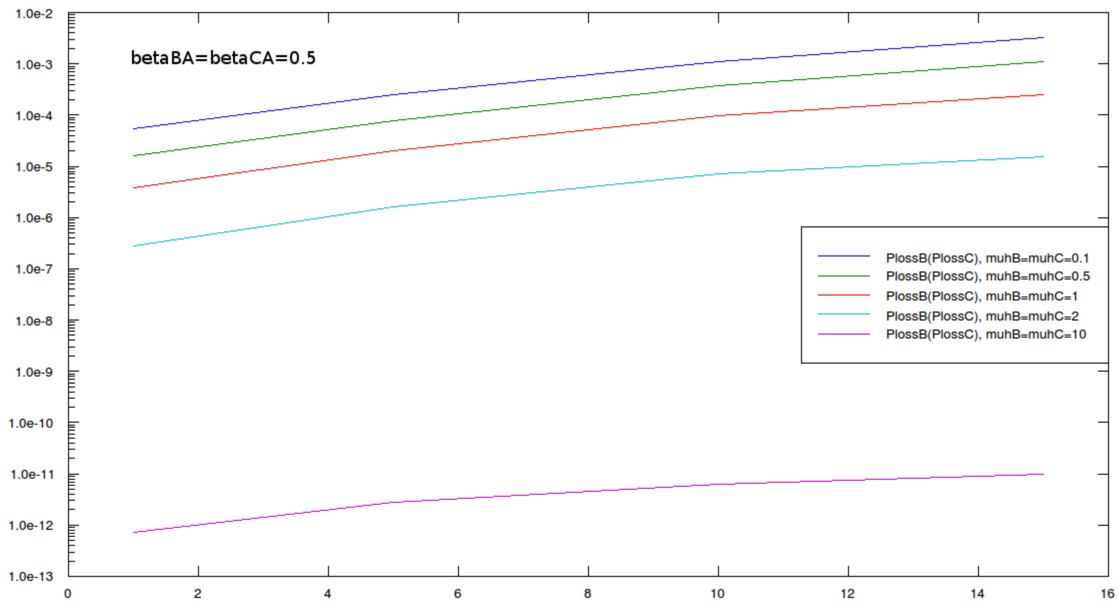


Figure 47: Probability of loss for the cells B and C incrementing μ_{hB} and μ_{hC}

5 Case 3

Let's examine more in detail the case introduced in 2.2.3.

5.1 The model

Here, the 2 cells should be considered as M/M/1/0s with processor sharing (PS) with a limit on the maximum number of active services (N_A or N_B basing on the cell considered).
According to this model, then, services always proceeds at maximum rate: $N_A\mu_A$ for the macro and $N_B\mu_B$ for the micro cell. When there are more than a customer in service ($k_A > 1 \vee k_B > 1$) the rate for each user becomes $\frac{N_A\mu_A}{k_A}$ or $\frac{N_B\mu_B}{k_B}$, considering even the dwell rate, then, for cell A, the total rate would be:

$$\mu_{tA}(k_A) = \frac{N_A\mu_A}{k_A} + \mu_{hA}$$

hence, the probabilities:

1. completion of the service with probability $P_{1A}(k_A) = \dfrac{\frac{N_A\mu_A}{k_A}}{\mu_{tA}(k_A)} = \dfrac{N_A\mu_A}{N_A\mu_A + k_A\mu_{hA}}$
2. handover towards B with probability $P_{2A}(k_A) = \dfrac{\beta\mu_{hA}}{\mu_{tA}(k_A)} = \dfrac{k_A\beta\mu_{hA}}{N_A\mu_A + k_A\mu_{hA}}$
3. handover towards another macro cell, external to this system, with probability

   $$P_{3A}(k_A) = \frac{(1-\beta)\mu_{hA}}{\mu_{tA}(k_A)} = \frac{k_A(1-\beta)\mu_{hA}}{N_A\mu_A + k_A\mu_{hA}}$$

regarding the cell B

$$\mu_{tB}(k_B) = \frac{N_B\mu_B}{k_B} + \mu_{hB}$$

Probabilities:

1. completion of the service with probability $P_{1B}(k_B) = \dfrac{\frac{N_B\mu_B}{k_B}}{\mu_{tB}(k_B)}$
2. handover towards A with probability $P_{2B}(k_B) = \dfrac{\mu_{hB}}{\mu_{tB}(k_B)}$

It's important to notice how the dependency, for the rates and for the routing probabilities, on the number of active services, may be a problem for a software like Tangram-II, which has a solution approach not based on the construction of the detailed underlying Markovian model. Indeed, if we try to implement the above routing probabilities we will receive an error stating that it's impossible to use a float number (the variable in which the routing probability is stored) as variable, but should be declared as a constant, i.e. the routing probabilities can't change on every iteration. Because of this, is fundamental, either for cell A or cell B, to upper and lower bound the service rate:

$$k_A(\mu_A + \mu_{hA}) \leq \mu_{tA}(k_A) \leq N_A(\mu_A + \mu_{hA})$$

$$k_B(\mu_B + \mu_{hB}) \leq \mu_{tB}(k_B) \leq N_B(\mu_B + \mu_{hB})$$

The routing probabilities for cell A:

$$\frac{\mu_A}{\mu_A + \mu_{hA}} \leq P_{1A}(k_A) \leq \frac{N_A\mu_A}{N_A\mu_A + \mu_{hA}}$$
$$\frac{\beta\mu_{hA}}{N_A\mu_A + \mu_{hA}} \leq P_{2A}(k_A) \leq \frac{\beta\mu_{hA}}{\mu_A + \mu_{hA}}$$
$$\frac{(1-\beta)\mu_{hA}}{N_A\mu_A + \mu_{hA}} \leq P_{3A}(k_A) \leq \frac{(1-\beta)\mu_{hA}}{\mu_A + \mu_{hA}}$$

The routing probabilities for cell B:

$$\frac{\mu_B}{\mu_B + \mu_{hB}} \leq P_{1B}(k_B) \leq \frac{N_B\mu_B}{N_B\mu_B + \mu_{hB}}$$
$$\frac{\mu_{hB}}{N_B\mu_B + \mu_{hB}} \leq P_{2B}(k_B) \leq \frac{\mu_{hB}}{\mu_B + \mu_{hB}}$$

In the end then, in this case and in the next one, the same routing probabilities of the previous models: 2 cells and 3 cells model, are utilized.
Obviously, given that the macro cell is of an older generation than the micro one, could have been considered even the intermediate case in which the macro cell is modelled as an M/M/m/0 and the micro one as an M/M/1/0-PS.

5.1.1 Unbalanced system

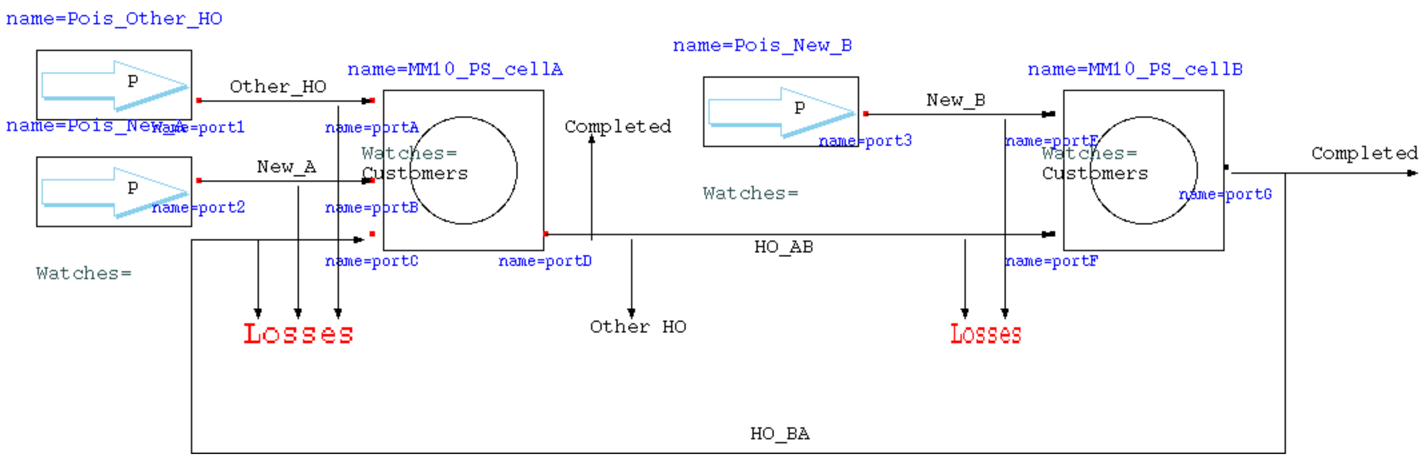


Figure 48: Unbalanced Tangram model for the case in Figure 13, with PS

The same loss probabilities of section 3.1.1 are used.

5.1.2 Balanced system

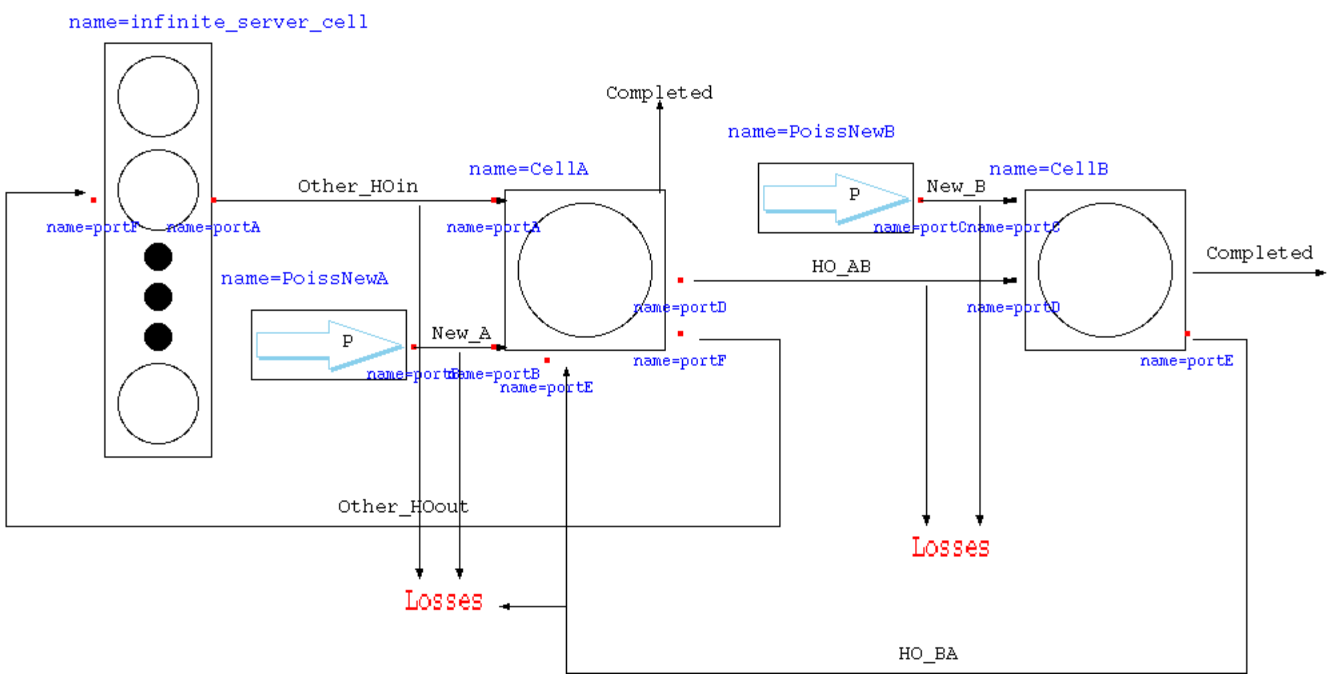


Figure 49: Balanced Tangram model for the case in Figure 13, with PS

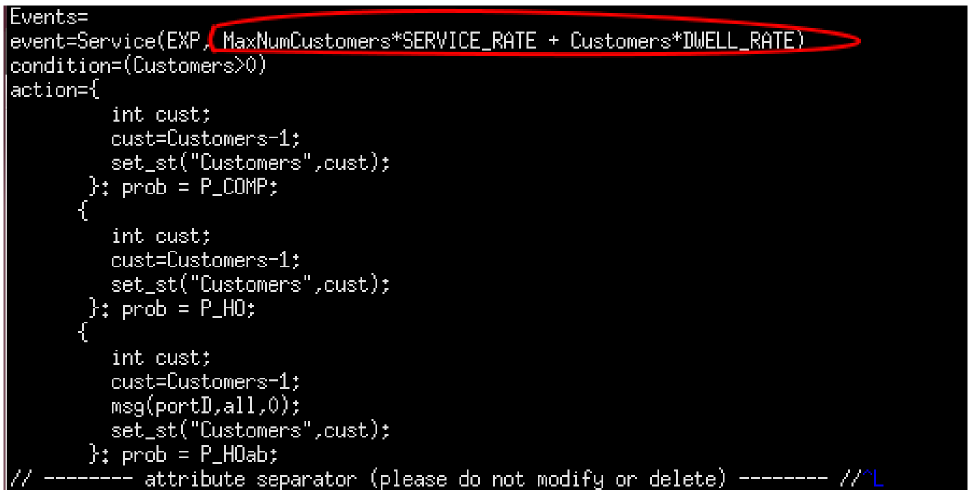
The formulas in section 3.1.2 are used.

5.2 Code implementation

The code is similar to the other 2 cases, following, just the differences have been reported.

5.2.1 Tangram-II

- **The services:** In the image below we can observe the difference in the service rate with respect the no-PS models. The new rate is circled in red.



```
Events=
event=Service(EXP, MaxNumCustomers*SERVICE_RATE + Customers*DWELL_RATE)
condition=(Customers>0)
action={
         int cust;
         cust=Customers-1;
         set_st("Customers",cust);
       }; prob = P_COMP;
      {
         int cust;
         cust=Customers-1;
         set_st("Customers",cust);
       }; prob = P_HO;
      {
         int cust;
         cust=Customers-1;
         msg(portD,all,0);
         set_st("Customers",cust);
       }; prob = P_HOab;
// -------- attribute separator (please do not modify or delete) -------- //^L
```

Figure 50: Tangram-II services implementation for PS-models

5.2.2 FreeMat

For both, unbalanced and balanced system, the same identical (opportunely changing the names of the file to load) code presented in section 3.2.2, has been utilized.

5.3 Results case 3

Values set:

- $\beta = 1/3$
- **M_M_Na**
  1. Dwell rate (μ_{hA}): $\mu_{hA} = 1$
  2. Service rate (μ_A): $\mu_A = 1$
  3. Max number of servers (Na): $Na = 16$
- **M_M_Nb**
  1. Dwell rate (μ_{hB}): $\mu_{hB} = 0.1$
  2. Service rate (μ_B): $\mu_B = 1$
  3. Max number of servers (Nb): $Nb = 64$
- **Pois_New_A**
  1. rate (λ_A): λ_A variable, its values are $1, 5, 10$ or 15
- **Pois_New_B**
  1. rate (λ_B): $\lambda_B = 20$
- **Pois_other_cells**
  1. rate (λ_{HA}): λ_{HA} variable. it is $\lambda_{HA} = 0$ when the system is unbalanced, and equal to the output of **infinite_server_cell** when the systems is balanced.
- **infinite_server_cell** [3]
  1. Total rate (μ_{hx}): $\mu_{hx} = 1$
  2. Max number of servers (Nx): $Nx = 10$

[3] This object substitutes **Pois_other_cells** in the balanced system

Obtained plots:

Similarly to Case 1, let's proceed comparing the unbalanced and balanced system but with PS now. The probability of loss for the macro cell:

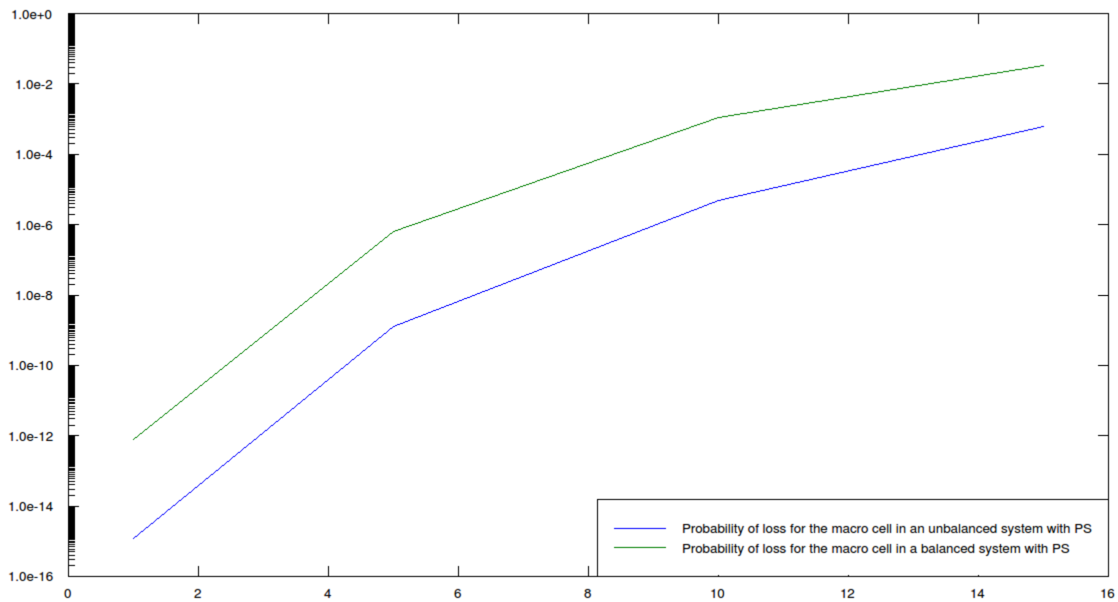


Figure 51: Probability of loss of the cell A: balanced versus unbalanced system

Obviously, the probability trend to be higher in the balanced system is maintained.

All the data used up to now, aren't ideal to represent the cell B probability of loss, indeed, its probability is so low that either the computer sensitivity represents it with zero. For this reason λ_B will be set to 40 in order to increment the traffic in the system, and, especially, for cell B **Figure 52**.

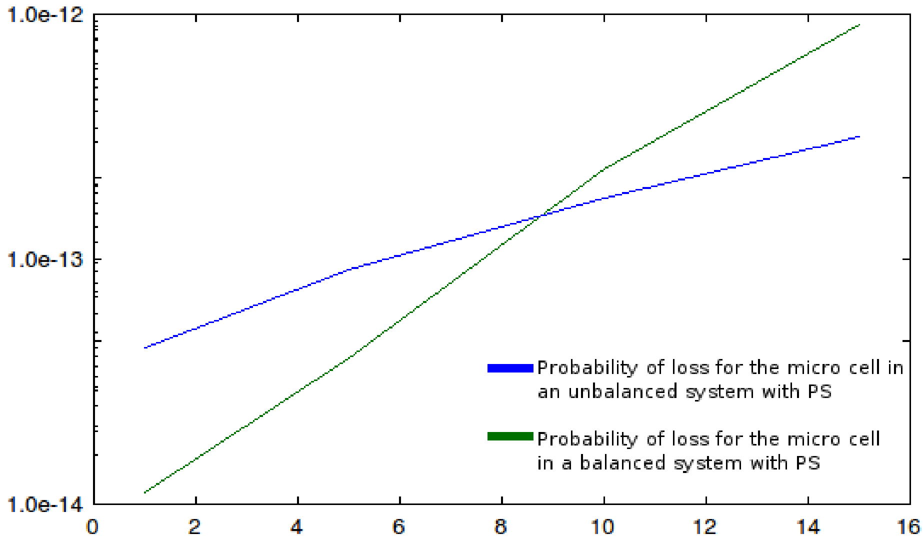


Figure 52: Probability of loss of the cell B: balanced versus unbalanced system

In the figures above we can now appreciate the probability, even if this is still very small in both the plots.

But What's the real advantage of using a system with the PS? Let's now compare, through a series of experiments, the probability of loss for cell A and cell B with and without PS for both: balanced and unbalanced system.

FIRST EXPERIMENT

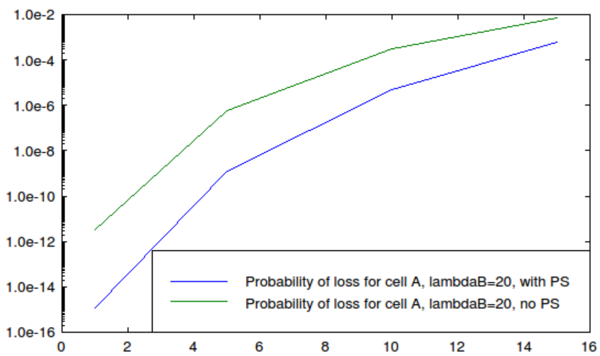


Figure 53: Comparison between a PS and a no-PS cell A (unbalanced system)

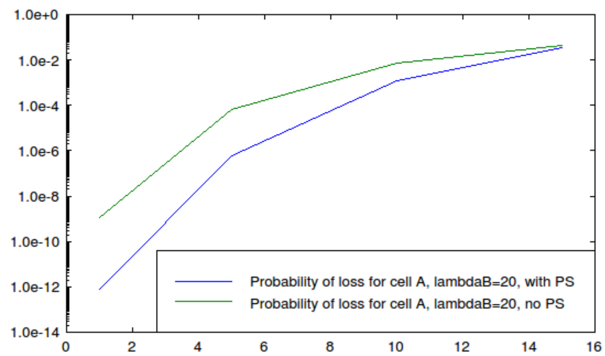


Figure 54: Comparison between a PS and a no-PS cell A (balanced system)

We can notice how in the unbalanced system (**Figure 53**) the difference between PS and no-PS is still relevant when the nominal load is at $\lambda_A = 15$, whereas, in the balanced system (**Figure 54**), the difference for an high load is almost nullified. Surely, in the unbalanced case the two curves will touch each other but at an higher nominal load; this is probably due to the fact that, in the balanced case, the circulating traffic in the system is always higher than the unbalanced one and so the **Figure 54** is just anticipating what will happen for the **Figure 53** for $\lambda_A = 20, 25, etc...$

SECOND EXPERIMENT

Here, I just tried to examine how the probability of loss for cell A changes increasing λ_B more and more.

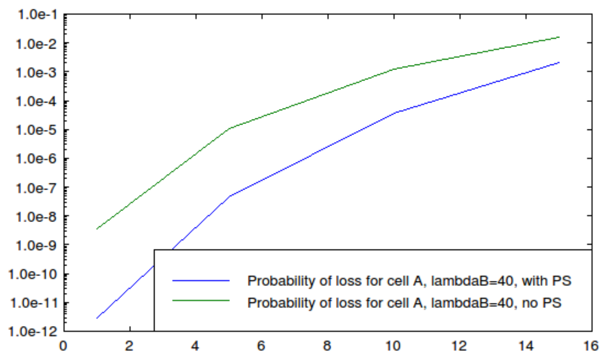


Figure 55: Probability of loss of the cell A (unbalanced system)

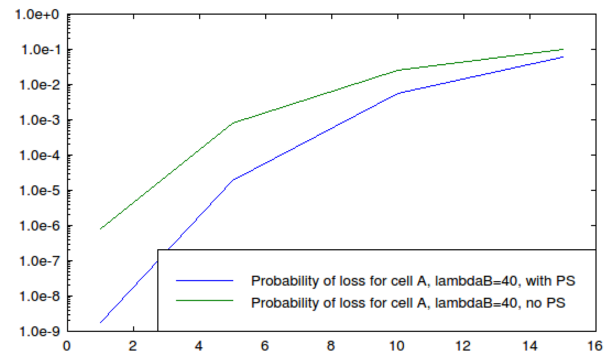


Figure 56: Probability of loss of the cell A (balanced system)

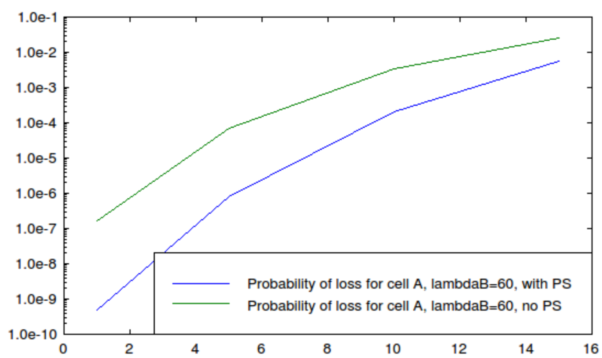


Figure 57: Probability of loss of the cell A (unbalanced system)

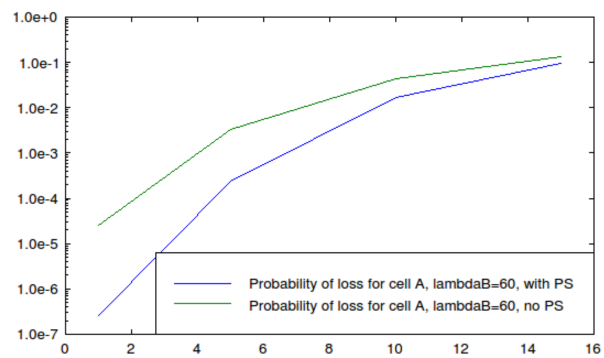


Figure 58: Probability of loss of the cell A (balanced system)

Horizontally comparing the figures above, we can make the same consideration of the experiment one. So, let's read the images in a vertical way, that mean, the two unbalanced and then the two balanced, always changing the λ_B rate. As expected the plots are very similar among them; therefore, to A, the cell B exogenous arrivals are not so relevant, in the sense that, a very big variation in λ_B is necessary to make the probability of loss of cell A considerably change, either for the unbalanced or the balanced case. But, in our case, with first $\lambda_B = 40$ and then $\lambda_B = 60$, how much is this variation? The answer is found in the sixth experiment.

THIRD EXPERIMENT

Up to now, we have seen the effect of the PS for the macro cell, and what's the impact on the micro one? We can appreciate the huge difference in the images below: in both cases, unbalanced and balanced, between PS and no-PS we have almost a difference of ten order of magnitude of the probability of loss. But why in the macro cell we have such a small difference comparing to the micro one? This is probably related to the fact that, although being the micro cell dwell rate minor than the macro cell, the maximum number of customers in the cell B is set to 64 instead of the 16 of the cell A. And being the service rate, about the PS case, dependent on the **MaxNumCustomers** variable on the code, just this, is already sufficient to make the micro performance much more better than the macro one
So, right the difference in the number of customers and in the dwell rate between the two cells find is explanation in the fact that the cell B should be more powerful, like has been noted, than the A one; B has, indeed, to provide service for an hotspot.

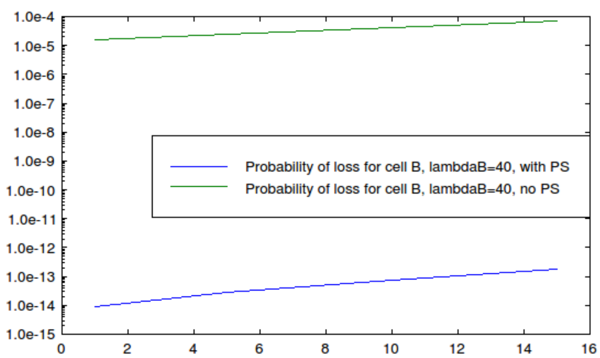


Figure 59: Comparison between a PS and a no-PS cell B (unbalanced system)

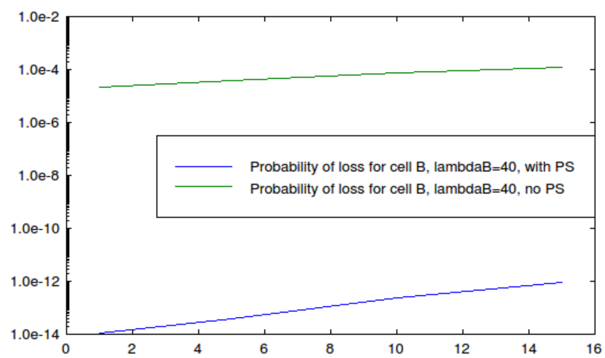


Figure 60: Comparison between a PS and a no-PS cell B (balanced system)

FOURTH EXPERIMENT

Increasing λ_B ($\lambda_B = 60$) with respect to the previous experiment we have a reduction of the difference between PS and no-PS. Now, they are distant about three order of magnitude due to the rising of the micro cell exogenous arrivals that strongly impact the performance of the cell B, indeed, at the moment, the cell has to roughly serve, in average, the double of the customers in a time unit with respect before.

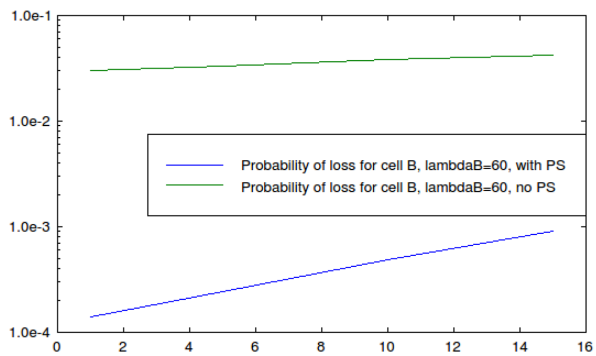


Figure 61: Comparison between a PS and a no-PS cell B (unbalanced system)

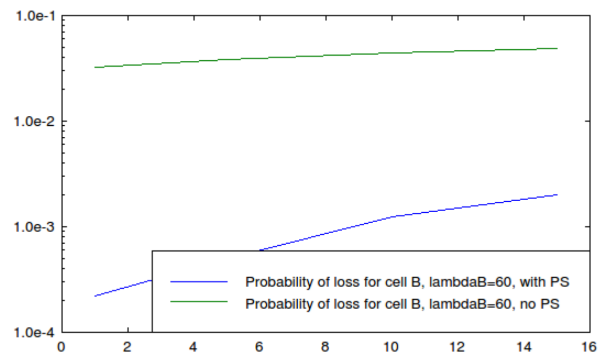


Figure 62: Comparison between a PS and a no-PS cell B (balanced system)

FIFTH EXPERIMENT

In this experiment, we will talk just about the processor sharing system.

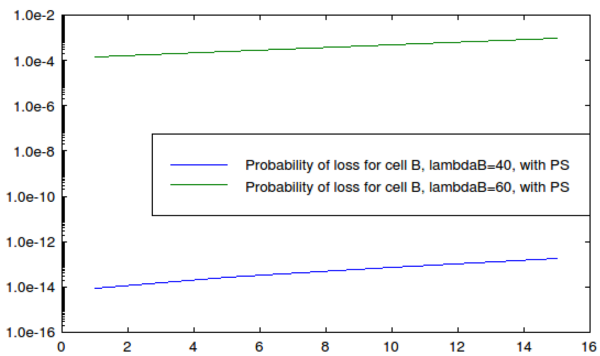


Figure 63: Probability of loss for cell B incrementing the cell B exogenous arrival (unbalanced system)

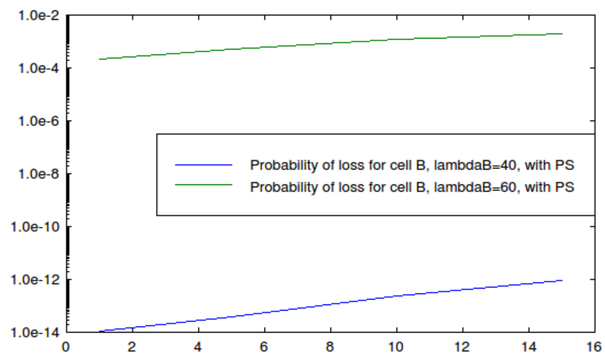


Figure 64: Probability of loss for cell B incrementing the cell B exogenous arrival (balanced system)

In the images above, a huge distance, similar to the one of the third experiment, divides the two curves, underlining how much the exogenous arrivals strongly influence the cell B probability of loss. Passing from $\lambda_B = 40$ to $\lambda_B = 60$ is like passing from a system with PS to a no-PS one with $\lambda_B = 40$ depicted in the third experiment.

SIXTH EXPERIMENT

Here I answer to the last question did in the second experiment: how much the probability of loss for the cell A is influenced by cell B exogenous arrivals rate? Going from $\lambda_B = 40$ to $\lambda_B = 60$, the answer is from almost 2 order of magnitude, for a smaller nominal load ($\lambda_A = 5$), to about half order of magnitude, for a bigger nominal load ($\lambda_A = 15$).

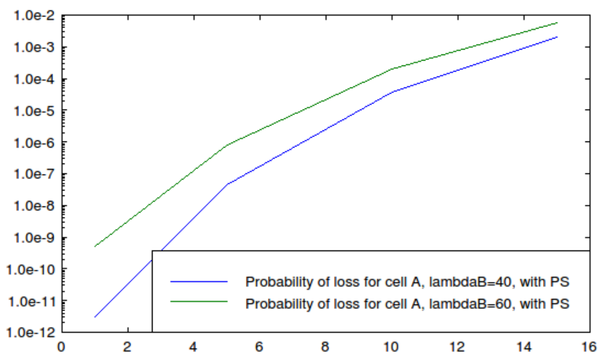


Figure 65: Probability of loss for cell A incrementing the cell B exogenous arrival (unbalanced system)

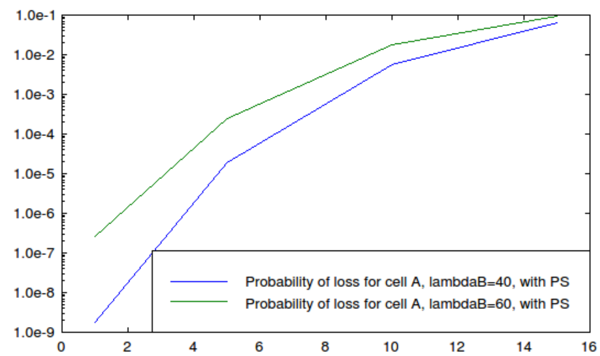


Figure 66: Probability of loss for cell A incrementing the cell B exogenous arrival (balanced system)

6 Case 4

Let's now go more in deep about section 2.2.4.
From now on, we'll get rid of the unbalanced model; indeed, up to here we should be completely conscious of the fact that the unbalanced system is very similar to the "real one" but having to handle less traffic its performances are always better than the balanced one, that means, in general, a lower loss probability.

6.1 Disjoined micro cells

6.1.1 Model and code implementation

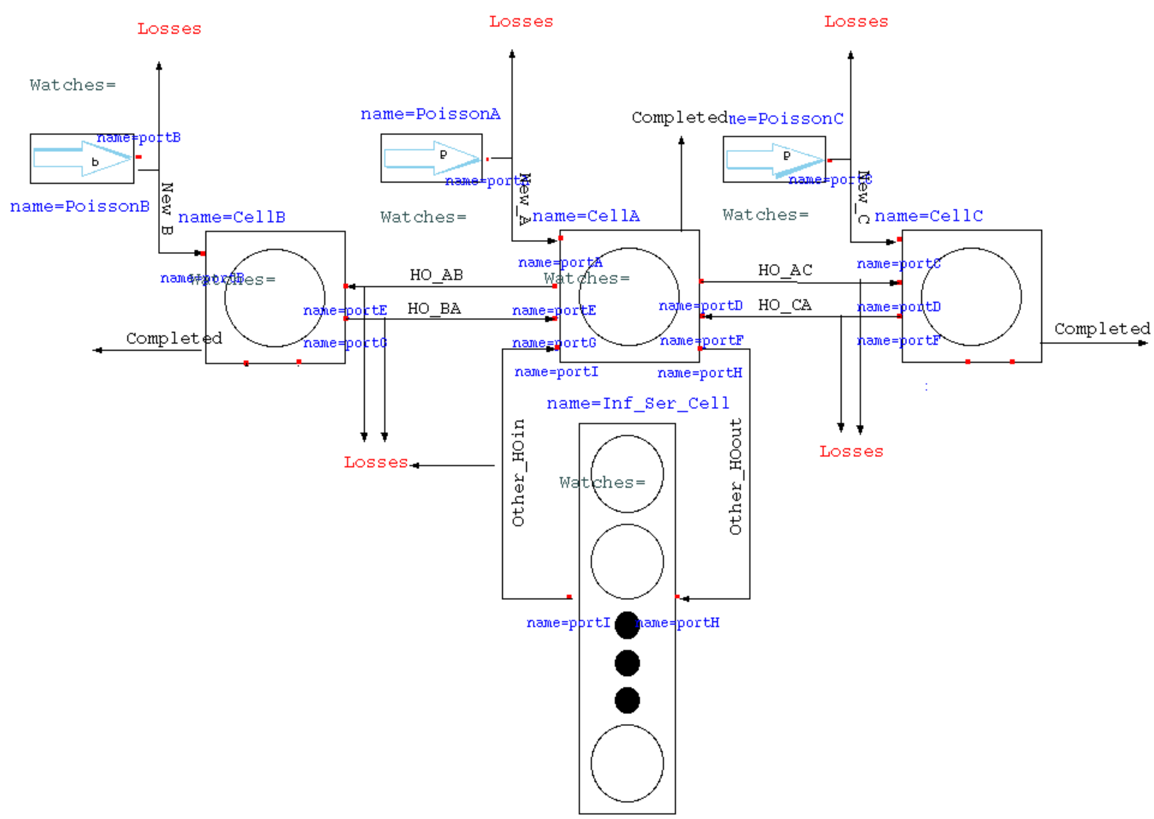


Figure 67: Tangram model for the case described in section 2.2.4

The same formulas used in Case 2, disjoined micro cells, have been exploited.

Tangram-II & FreeMat
The codes either for Tangram-II or FreeMat have been maintained the same of Case 2 with disjoined micro cells.

6.1.2 Results case 4, disjoined micro cells

Values set:

- **M_M_Na**
  1. $\beta_{AB} = \beta_{AC} = 1/6$
  2. Dwell rate (μ_{hA}): $\mu_{hA} = 1$
  3. Service rate (μ_A): $\mu_A = 1$
  4. Max number of servers (Na): $Na = 16$
- **M_M_Nb**
  1. $\beta_{BA} = 1$
  2. Dwell rate (μ_{hB}): $\mu_{hB} = 10$
  3. Service rate (μ_B): $\mu_B = 1$
  4. Max number of servers (Nb): $Nb = 16$
- **M_M_Nc**
  1. $\beta_{CA} = 1$
  2. Dwell rate (μ_{hC}): $\mu_{hC} = 10$
  3. Service rate (μ_B): $\mu_C = 1$
  4. Max number of servers (Nb): $NC = 16$
- **Pois_New_A**
  1. rate (λ_A): λ_A variable, its values are $1, 5, 10$ or 15
- **Pois_New_B**
  1. rate (λ_B): $\lambda_B = 5$
- **Pois_New_C**
  1. rate (λ_C): $\lambda_C = 5$
- **infinite_server_cell**
  1. Total rate (μ_{hx}): $\mu_{hx} = 1$
  2. Max number of servers (Nx): $Nx = 10$

Obtained plots:

Let's have a look to the shape of the loss probability for each cell, using the data listed in the previous page.

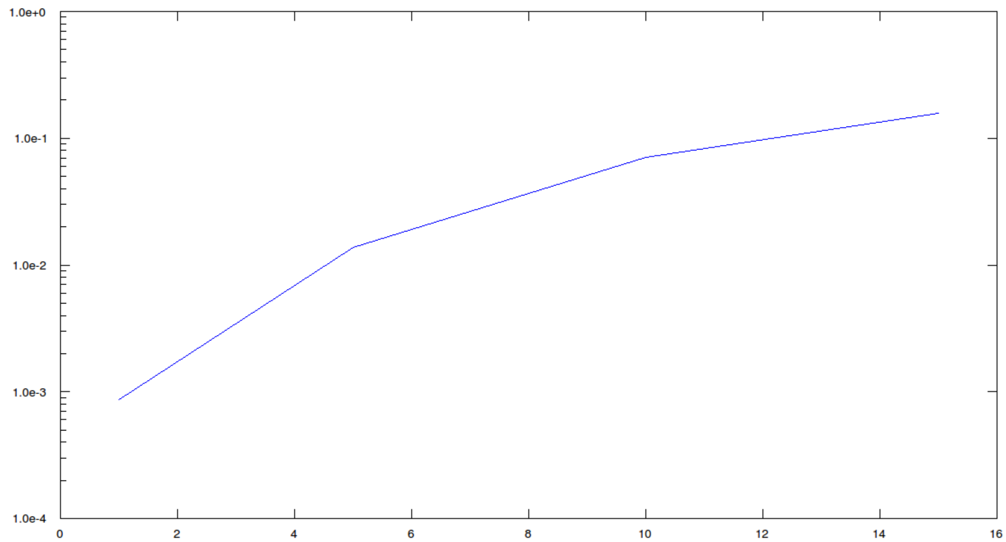


Figure 68: Probability of loss for the cell A

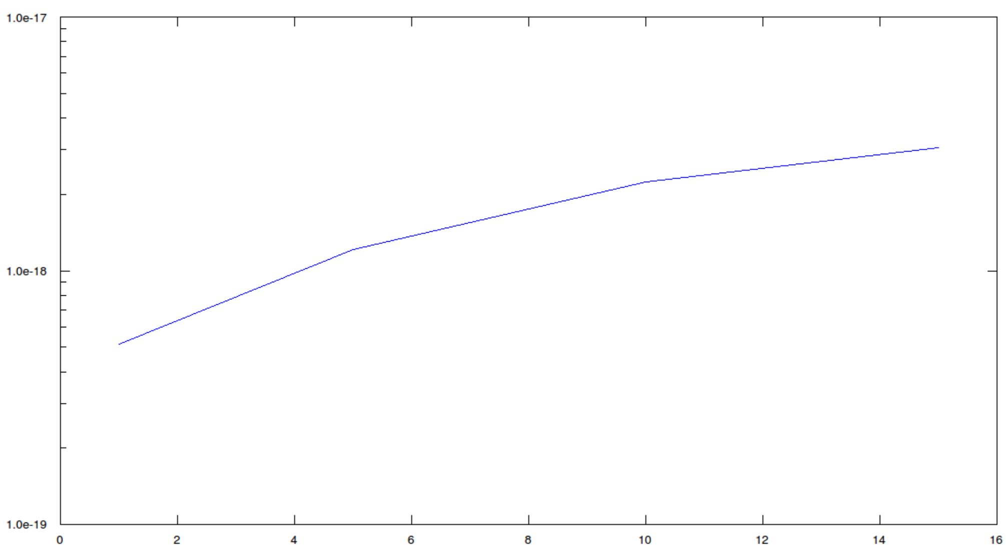


Figure 69: Probability of loss for the cells B and C

Let's set now $\mu_{hB} = \mu_{hC} = 0.1$ like in the results section of Case 2.

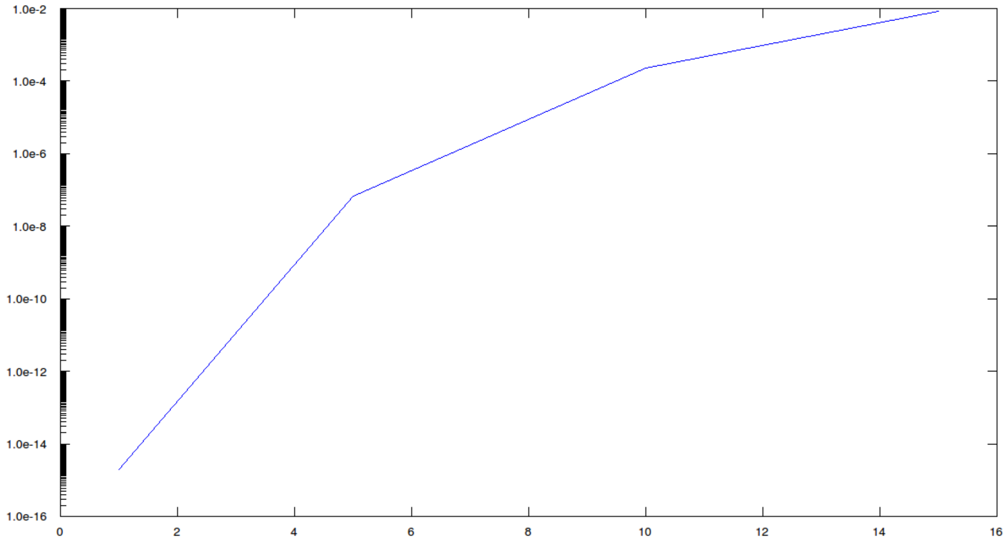


Figure 70: Probability of loss for the cell A

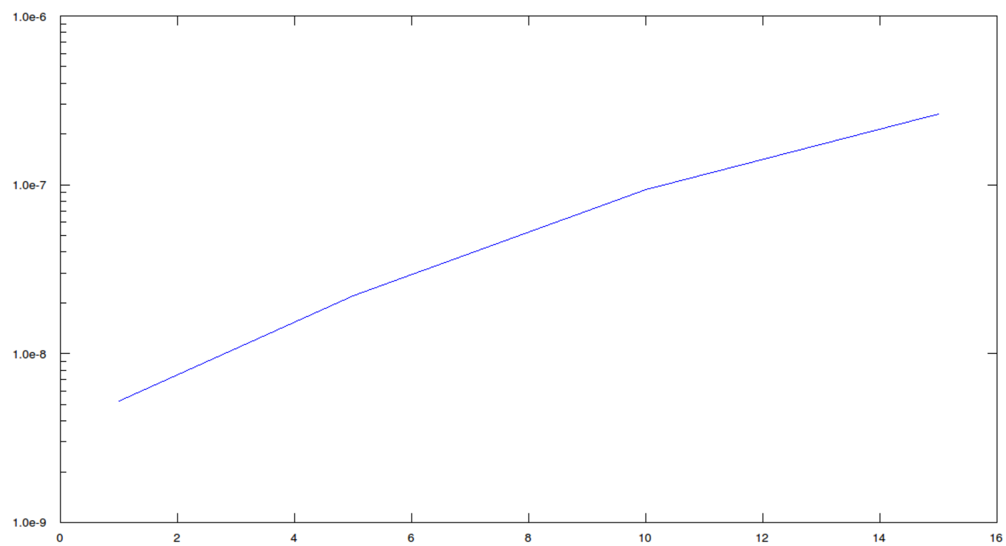


Figure 71: Probability of loss for the cells B and C

We can notice how much the probabilities are lowered in **Figure 70** and **Figure 71** with respect **Figure 68** and **Figure 69**. This is because of the lower dwell rate, that means a bigger dwell time, that means a longer time spent by each user in the micro cells, in other words, low mobility users. This is an important results then, we confirmed that with a lower mobility by the users the overall system performances get better. The last conclusion could appear wrong if we just consider the micro cells: indeed, being these cells occupied for a longer time their loss probability, obviously, increase; but again, if we consider the entire model the probability of loss for the macro will be lowered a lot and the micro cells' loss probability will increase, but, nevertheless, under an acceptable level.

Now, some plots varying the dwell rates.

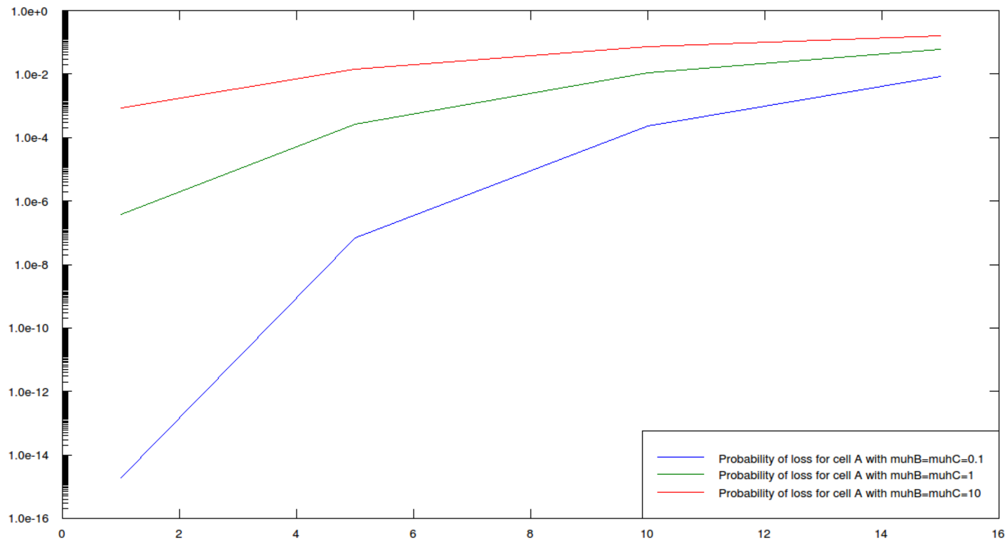


Figure 72: Probability of loss for the cell A, varying the micro cells' dwell rate

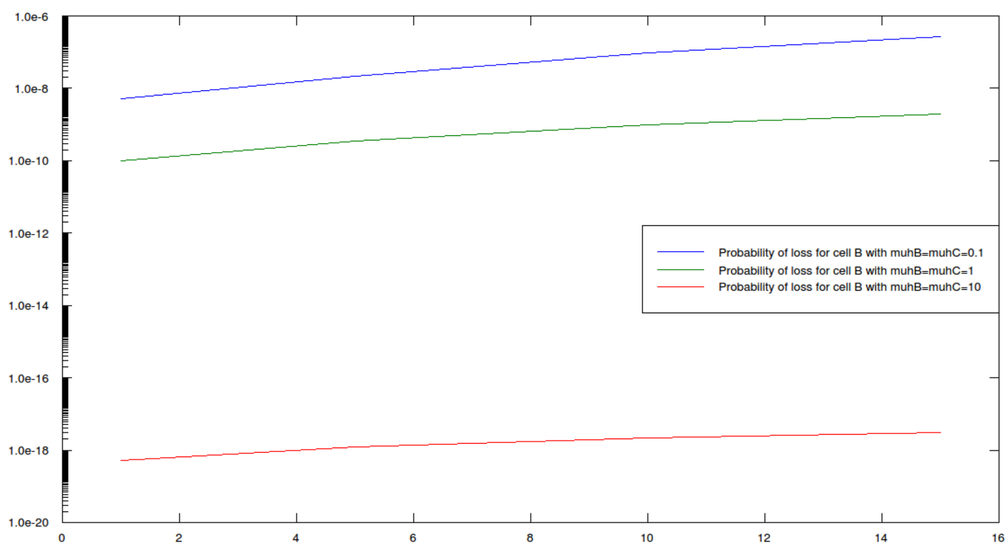


Figure 73: Probability of loss for the cell B, varying the micro cells' dwell rate

6.2 Joined micro cells

6.2.1 Model and code implementation

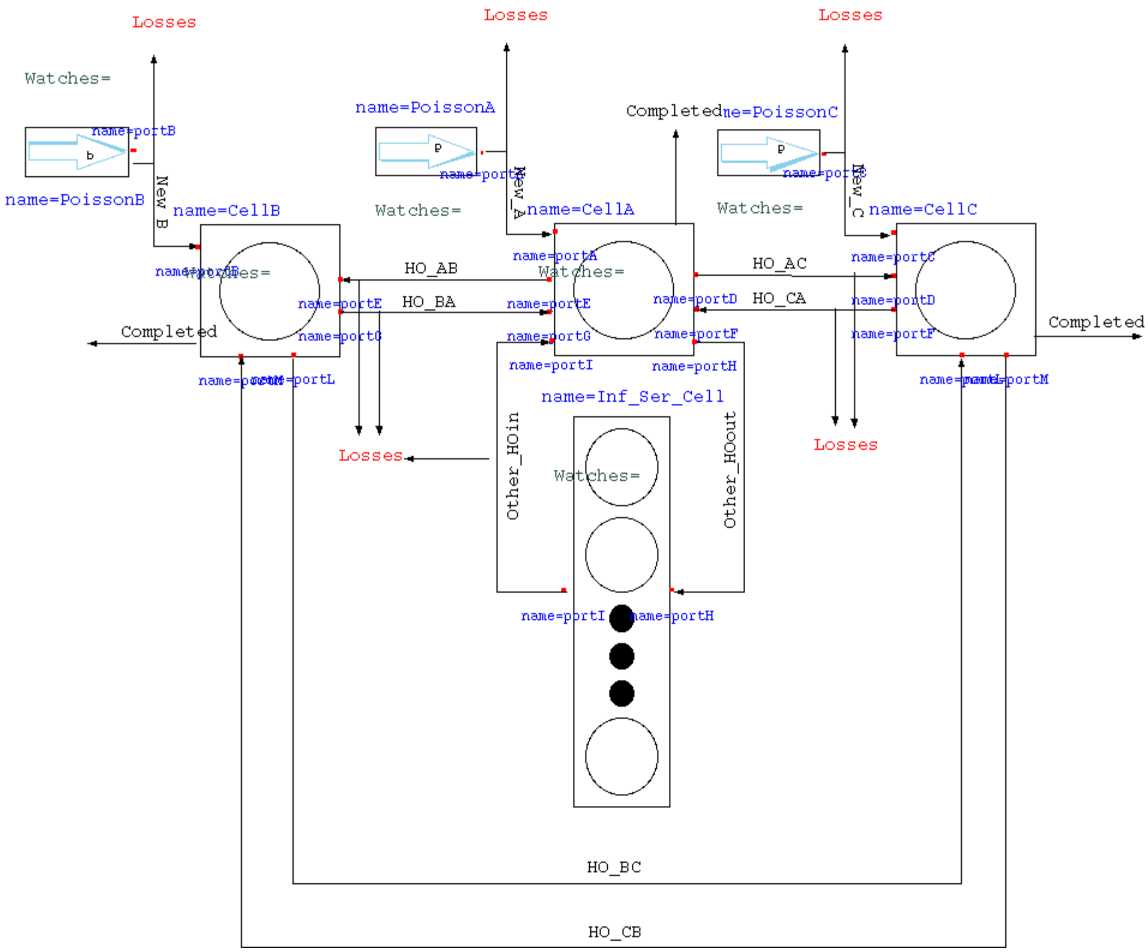


Figure 74: Tangram model for the case described in section 2.2.4

The same formulas used in Case 2, joined micro cells, have been exploited.

Tangram-II & FreeMat

The codes either for Tangram-II or FreeMat have been maintained the same of Case 2 with joined micro cells.

6.2.2 Results case 4, joined micro cells

Values set:

- **M_M_Na**
  1. $\beta_{AB} = \beta_{AC} = 1/6$
  2. Dwell rate (μ_{hA}): $\mu_{hA} = 1$
  3. Service rate (μ_A): $\mu_A = 1$
  4. Max number of servers (Na): $Na = 16$
- **M_M_Nb**
  1. $\beta_{BA} = 1/2$
  2. Dwell rate (μ_{hB}): $\mu_{hB} = 1$
  3. Service rate (μ_B): $\mu_B = 1$
  4. Max number of servers (Nb): $Nb = 16$
- **M_M_Nc**
  1. $\beta_{CA} = 1/2$
  2. Dwell rate (μ_{hC}): $\mu_{hC} = 1$
  3. Service rate (μ_C): $\mu_C = 1$
  4. Max number of servers (Nb): $NC = 16$
- $\beta_{BC} = \beta_{CB} = 1/2$
- **Pois_New_A**
  1. rate (λ_A): λ_A variable, its values are $1, 5, 10$ or 15
- **Pois_New_B**
  1. rate (λ_B): $\lambda_B = 5$
- **Pois_New_C**
  1. rate (λ_C): $\lambda_C = 5$
- **infinite_server_cell**
  1. Total rate (μ_{hx}): $\mu_{hx} = 1$
  2. Max number of servers (Nx): $Nx = 10$

Obtained plots:

Here, to obtain a better precision I increased the number of points on the X axis, indeed, now, λ_A values are: λ_A = 1-2-5-7.5-10-15. In **Figure 75** I plotted together the loss probabilities for A, B and C.

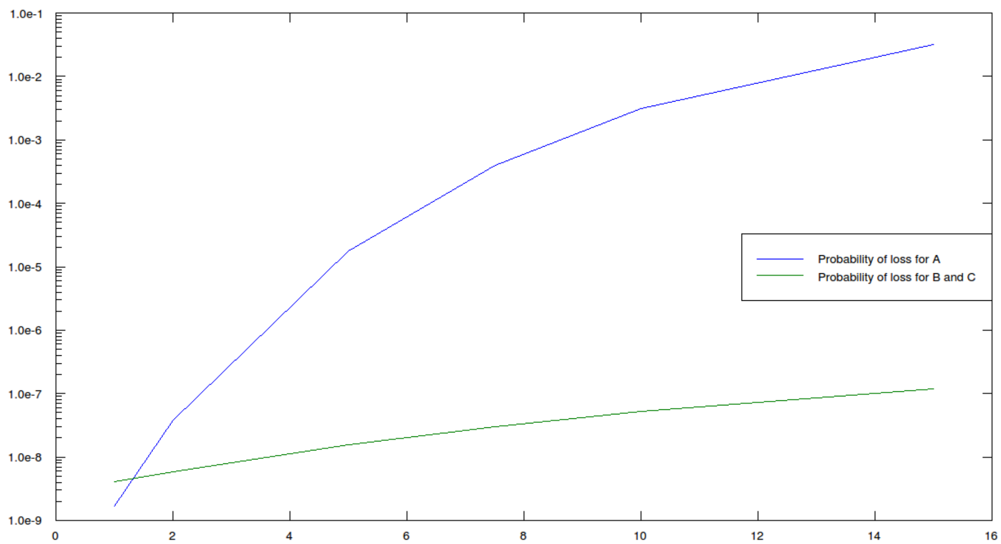


Figure 75: Probability of loss for the cell A,B and C

Setting $\mu_{hB} = \mu_{hC} = 0.5$ and making some experiment varying the handover probabilities and the dwell rates, I obtained the four images below.

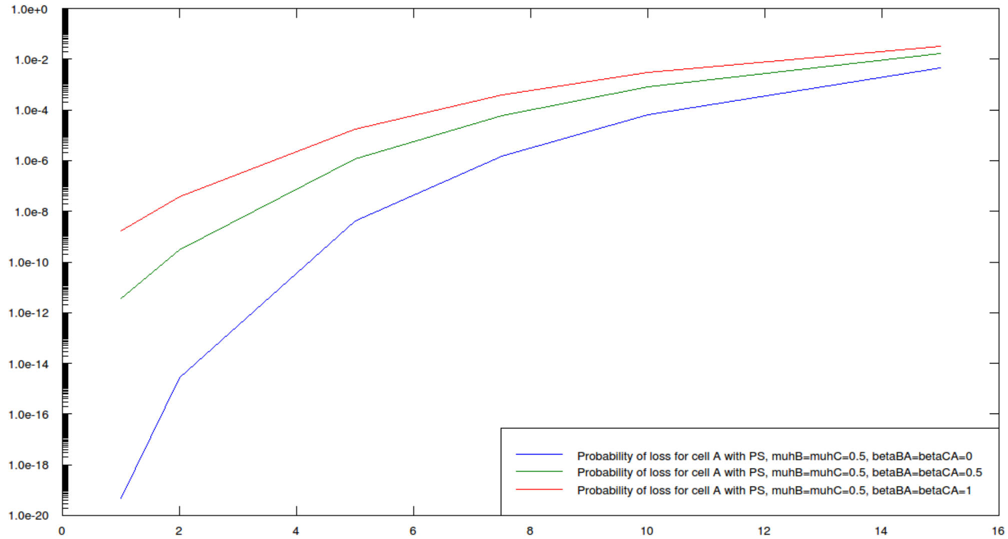


Figure 76: Probability of loss for the cell A, incrementing β_{BA} and β_{CA}

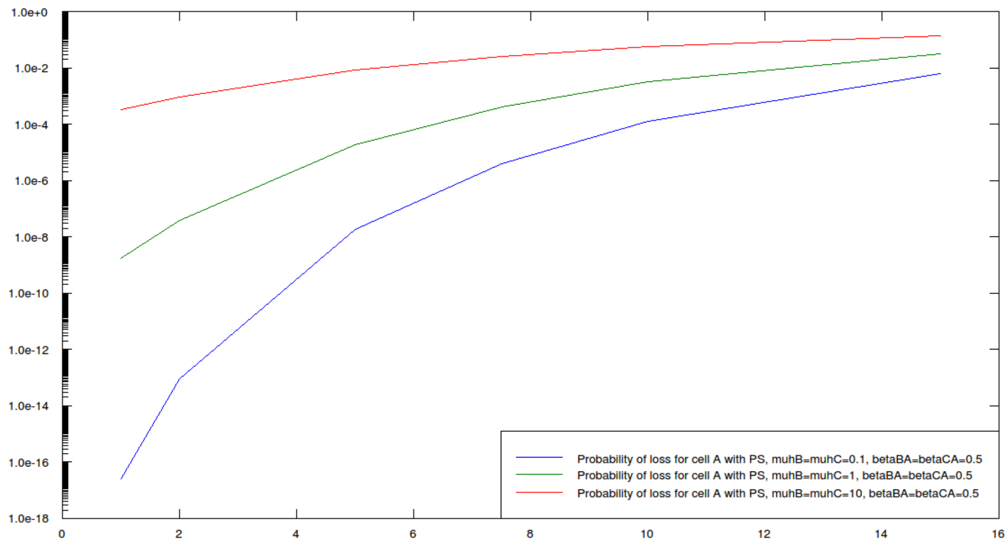


Figure 77: Probability of loss for the cell A, varying the micro cells' dwell rate

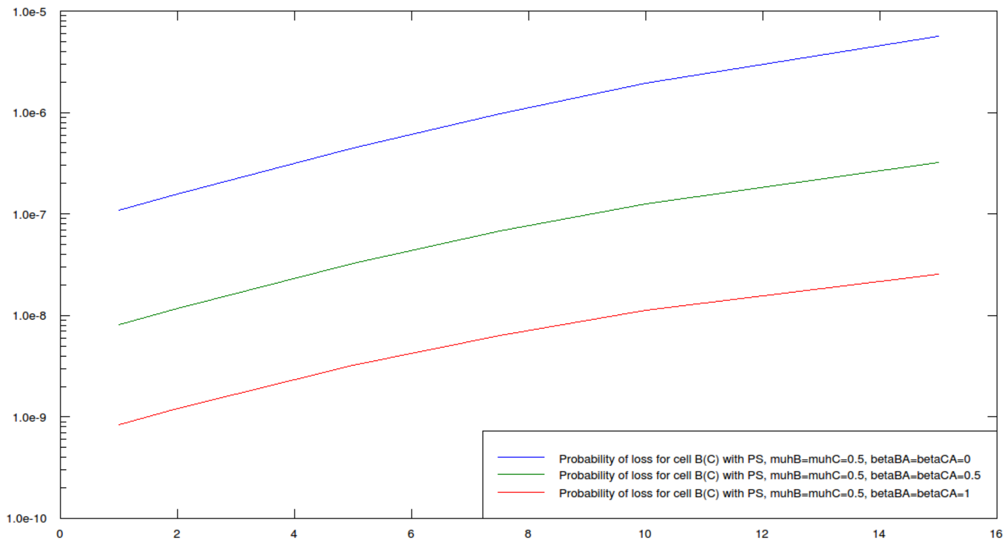


Figure 78: Probability of loss for the cell B, incrementing β_{BA} and β_{CA}

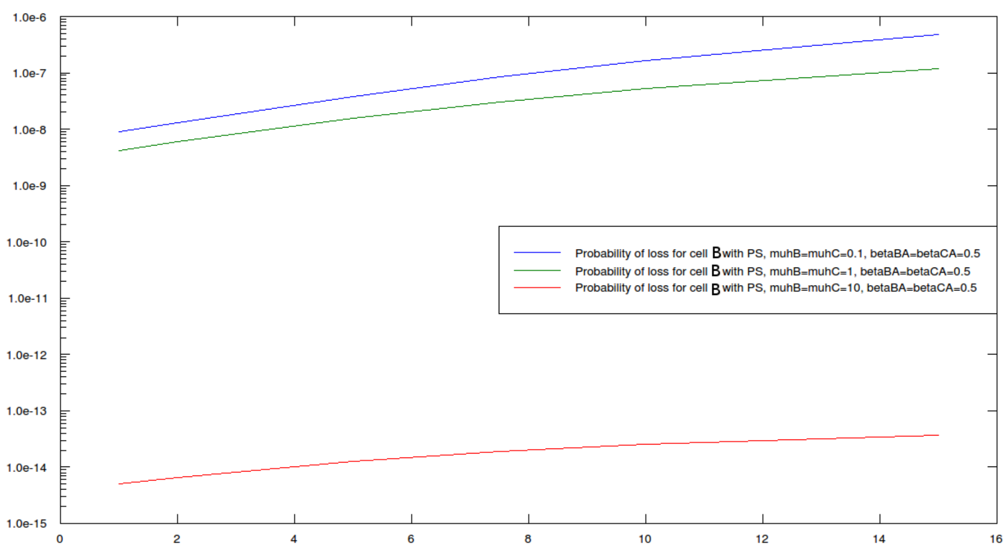


Figure 79: Probability of loss for the cell B, varying the micro cells' dwell rate

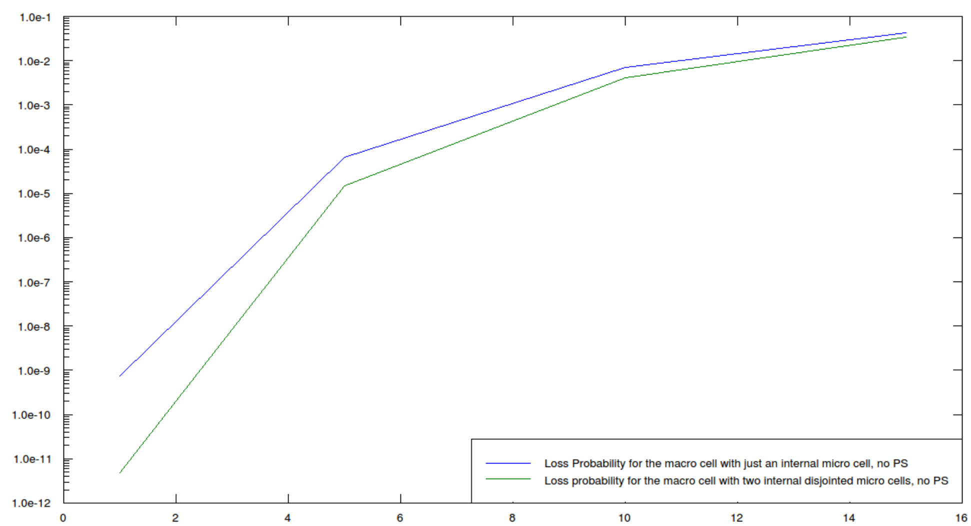
Observe how the overall system performances are always better when the two micro cells are more active, that means that can communicate each other ($\beta_{BA} = \beta_{CA} \neq 1$), and the users have a low mobility in the micro cells.

7 Conclusions

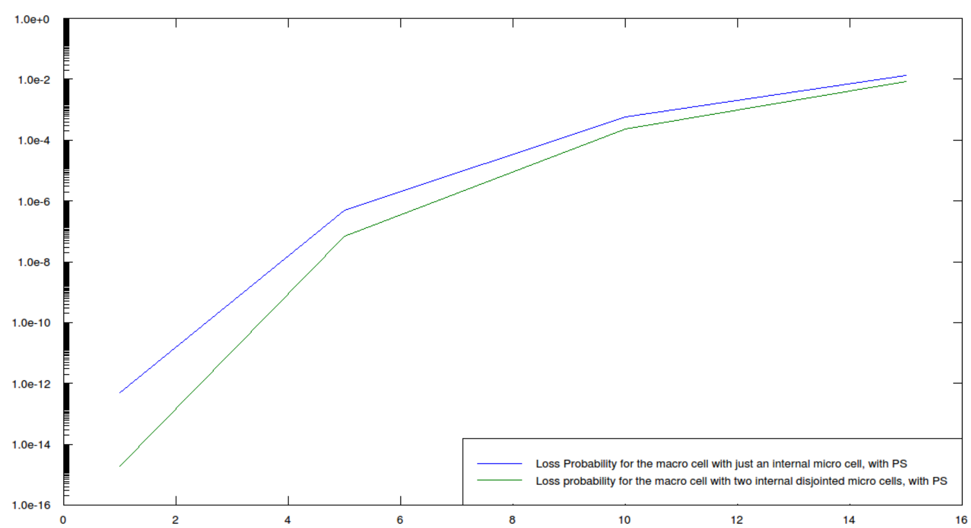
So, at the end, have we reached our prefixed initial goals?
Let's proceed with order.
Note that in this section will be marked as a better model the one which has better performance in terms of loss probability, without consider any other parameter. Moreover, just the more realistic cases, like the balanced systems, will be considered.

7.1 One micro or two micro disjoined cells?

Well, we have seen how the Case 1 already improves just adding another micro cell. The probability of loss for cell A with one micro cell and no PS (**Figure 80-(a)**) goes from about 10^{-9} to 10^{-1}, considering the case with disjoined micro cells the probability of loss for cell A can go from 10^{-11} to 10^{-2}. About the processor sharing case: the loss probability get lower than no-PS model one. So, we have already a relevant improvement: definitely **two micro disjoined cells**.



(a) Case 1: without PS

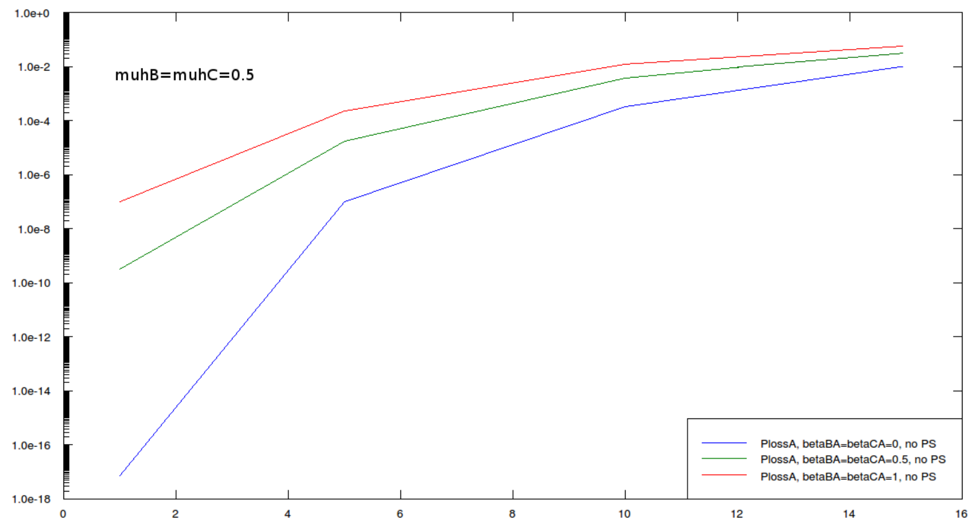


(b) Case 2: with PS

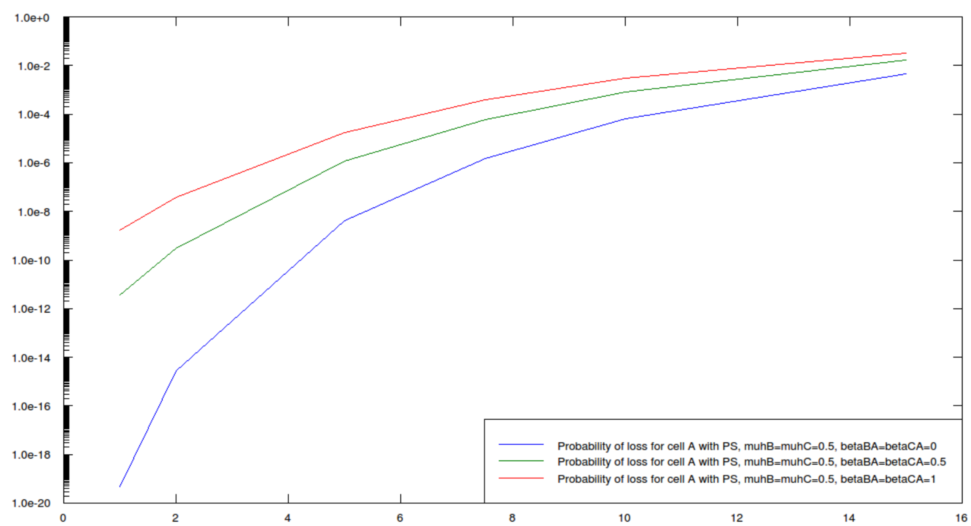
Figure 80: Comparison Case 1 and Case 2 (Disjoined micro cells): without PS (a) and with PS (b)

7.2 Joined or disjoined micro cells?

Clearly, for the macro cell, the joined micro cells case is the best solution, Indeed, in **Figure 81**, the highest curve, the $\beta_{BA} = \beta_{CA} = 1$ case (red line), represents the two disjointed micro cells situation. The same consideration is valid for the PS case.



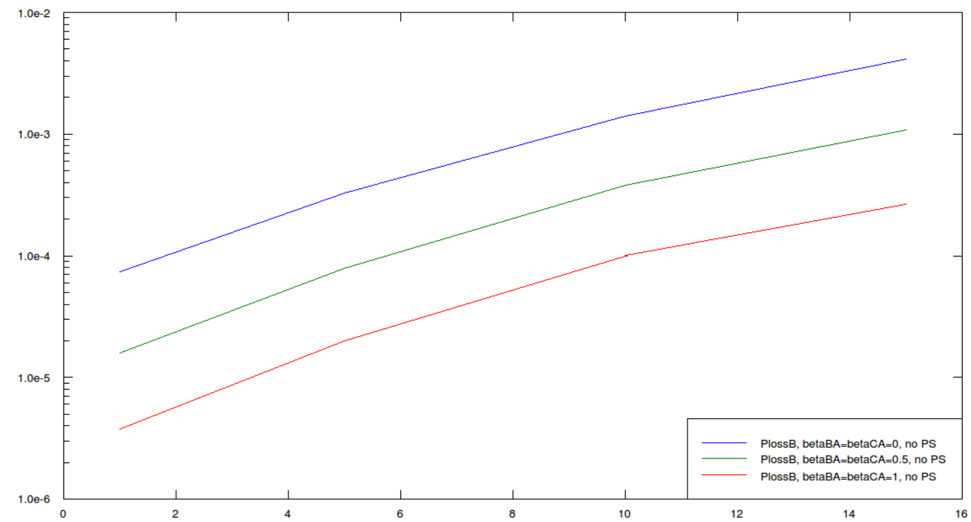
(a) without PS



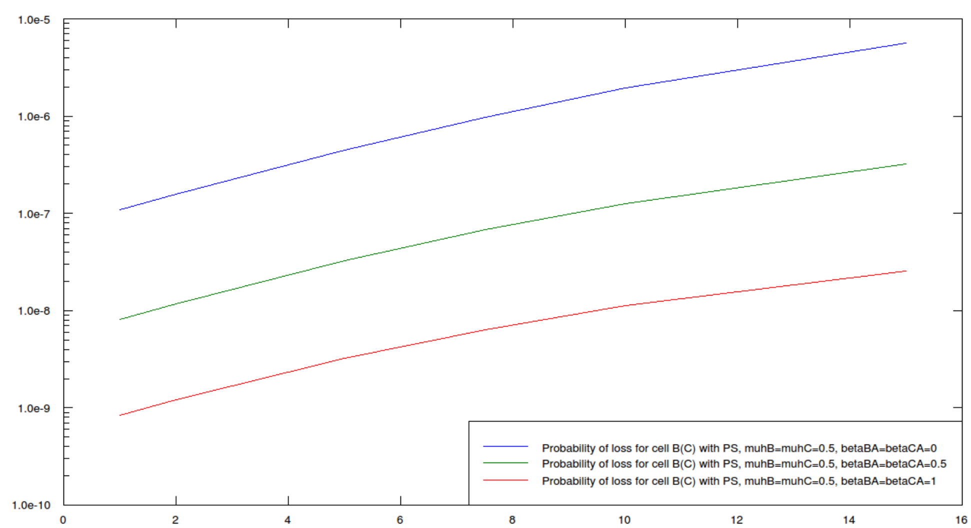
(b) with PS

Figure 81: Comparison Case 2: Disjoined micro cells and Case 2: Joined micro cells with PS (a) and without PS (b)

Now, let's have a look to the micro cells (**Figure 82**)



(a) without PS



(b) with PS

Figure 82: Comparison Case 2: Disjoined micro cells and Case 2: Joined micro cells with PS (a) and without PS (b)

In the previous image (**Figure 82**) the disjoined micro cells case seems to be a bit better: this case, represented with a red line, is, indeed, the lower bound in the figure, for both: PS and no PS case. By the way, especially for lower arrival rates, there are big advantages for the macro cell in using the joined micro cells model (e.g. for $\lambda_A = 1$ the macro cell probability of loss can improve up to about ten order of magnitude, instead, the micro cells probability of loss can get worse up to about just two order of magnitude). So, in conclusion, between 2 disjointed or **joined micro cells**, considering the entire system, it's definitely better the second model, because, regarding the micro cells, a small variation will happen but, regarding the macro cell, there are slightly less losses.

7.3 PS or not PS?

This is an easier question with respect the previous ones. Without any doubt the **PS model** is more performant in all the previous cases and for the following systems (**Figure 83** and **Figure 84**).

<u>Two cells: 1 macro - 1 micro case</u>

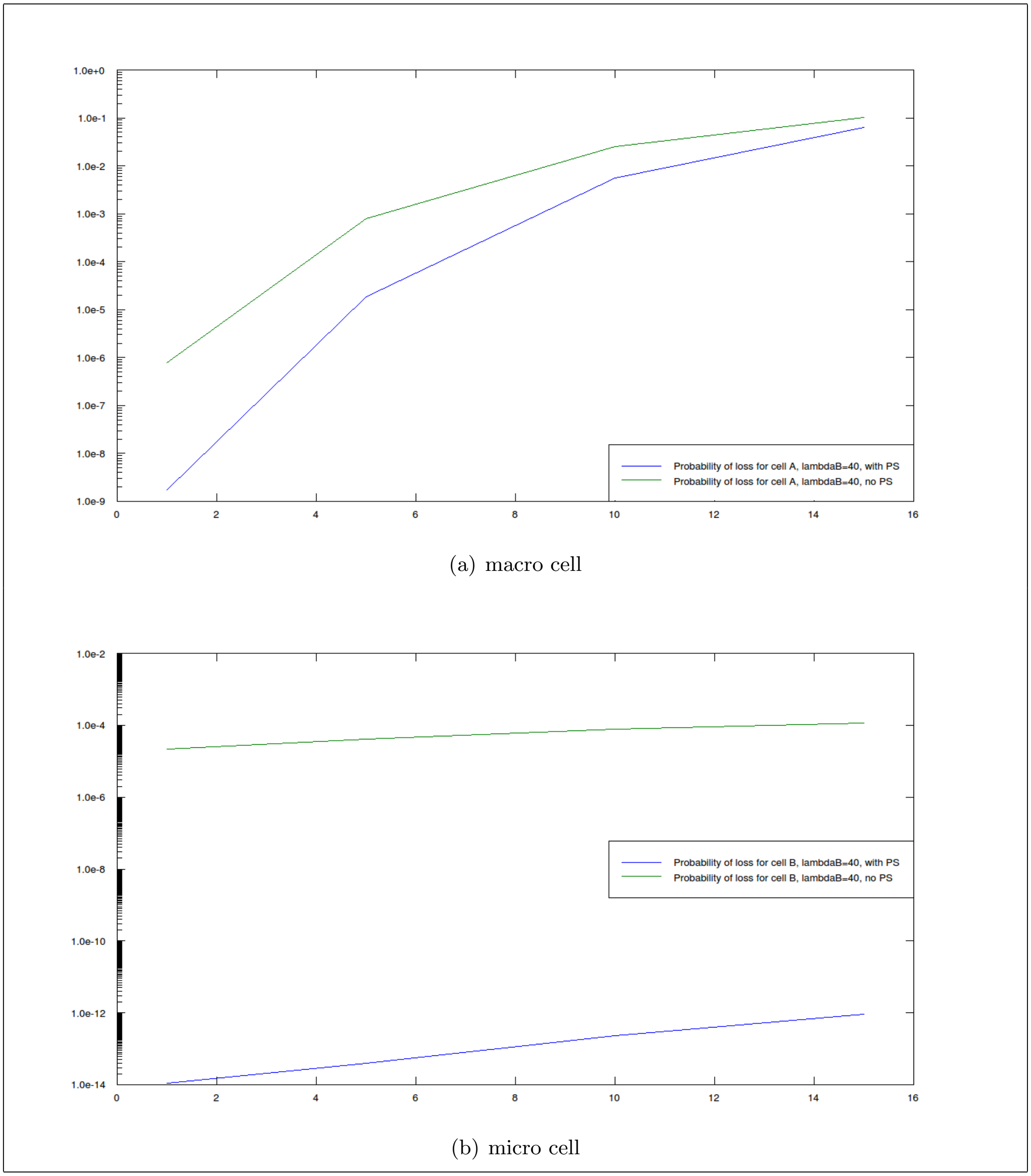
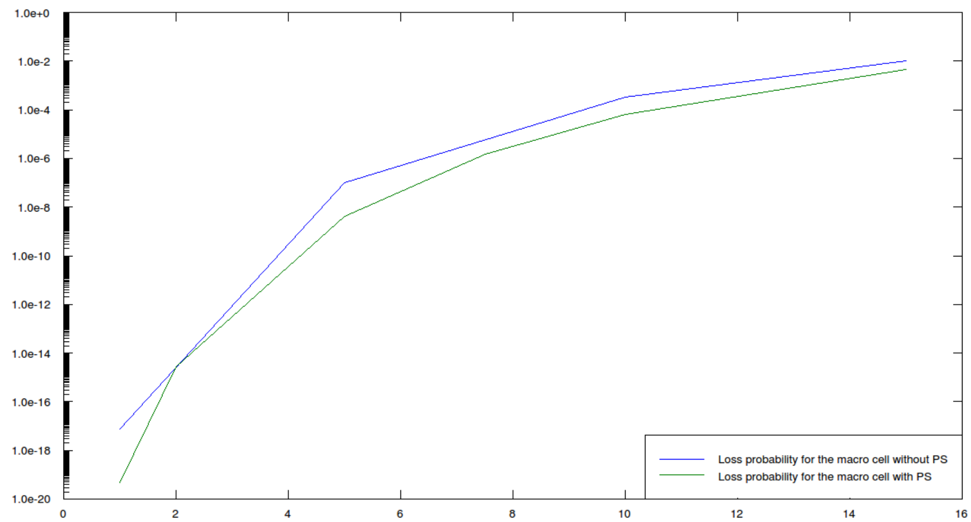


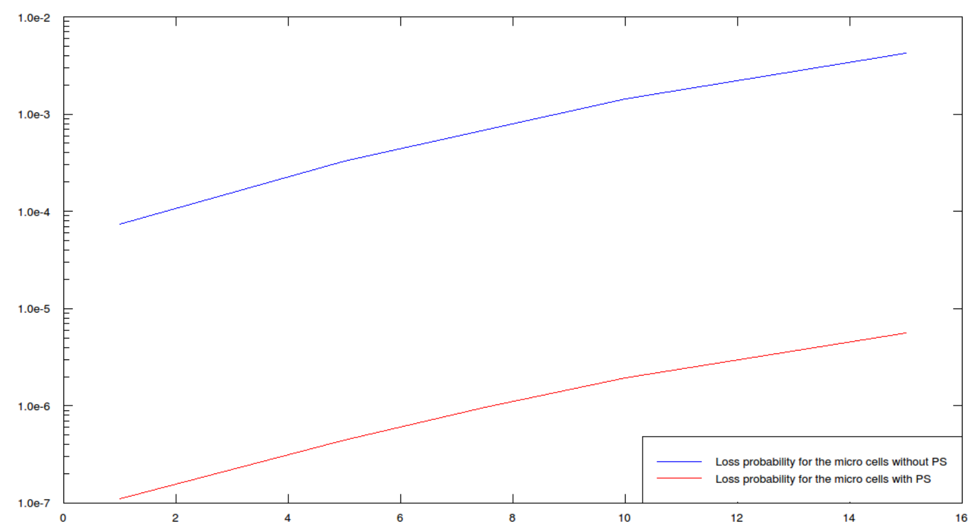
Figure 83: Comparison Case 1 and Case 3: for the macro cell (a) and for the micro cell (b)

Three cells: 1 macro - 2 micro case

Let's try to set some critical parameter for the micro cells, choosing $\beta_{BA} = \beta_{CA} = 0$, in order to have the maximum exchange between the two micro cells, and $\mu_{hB} = \mu_{hC} = 0.5$. The PS model is still better than the no-PS one?



(a) macro cell



(b) micro cell

Figure 84: Comparison Case 2 and Case 4: for the macro cell (a) and for the micro cell (b)

The answer to the last question is still yes, even if the difference between the 2 cases is reduced.

7.4 Best model?

In conclusion, we experienced how the micro cells strongly affects the total performance. In particular, big improvements happen when:

1. micro cells have a considerably bigger, with respect the macro cell, number of maximum customers that can be served
2. macro and micro cells implemented with the processor sharing method (there are small improvements even considering just one cell with PS, given that the PS model always improves the system)
3. In the system there are users with low mobility

So, what's the best model?
We can firmly state that **the processor sharing system with two jointed micro cells** inside the macro one **is the best solution** in terms of performance.

8 Acknowledgements

When I enrolled at Politecnico I already knew it would have been hard, not only for the subjects, but, actually, because this choice would have meant to ”grow up”.
Just today I understand what does it really mean.
At that time, I knew I would have grown up, I would have changed, but without knowing the meaning of ”growing up”, the how and the why. Now, I realize.
My opinion is that, for the most part of things that we achieve in life, we tend to emphasize on how hard it was to accomplish whatever goal; actually, as I see it, it’s almost useless to speak about the difficulty of something because of the subjectivity of the concept. It’s much more helpful to dialogue on your motivations, your tasks and on the determination that you are willing to put to achieve something. I’m not saying that was easy for me, but just that I had predefined and well known goals to reach; and in spite of everything, I did it, thanks to my will and determination.
The hardest part was not about Politecnico, but to be entirely independent, daily getting by in each situation and to start over in life. Again, I did it.
But, I would be a foul to don’t be thankful to the people that inspired, loved and encouraged me till today, and that will be always present in my life whatever happens.
For these reasons, I thank my parents: my dad, that thought me to be honestly ambitious, and my mother, that thought me the infinite beauty of humility rising upon injustice and ignorance. I thank my brother, because everyday tries to show me the beauty of a kind world. I thank all my relatives, my uncles and cousins; and especially, a tireless lovely woman: my grandmother; a real teacher of humbleness and owner of a good heart: my aunt Lucia, and a tenacious and strong soul: my aunt Orietta. I dedicate a good part of this work to my grandfather, today no longer among us, that this can be somehow a crumb of reply to a lovely glance never returned with an hug. And obviously, I thank all my travel fellows: Gianni, my sincere partner in crimes; Luca, the most honest and affable friend; Luigi, good truly fellow of these last two years; Gaetano, loyal friend; Raffaele, witness of a faithful and true friendship; Livio and Salvo, my video-games associates and lovely guys; Giulia, one of the most beautiful mind; Salvo, my personal roommate that makes me smile since already five years; Julia, crazy and lovely fresh girl; Stefania, sensitive bearer of beautiful feelings; Margherita, sweet flatmate; and many others, like: Chiara, Simone, Manuel and Simona.
At the end, that’s true, it hasn’t been easy for me, but today I’m stronger, because, if I look back, I knew there is another feeling that inspires me in addition to the determination: the pride.
Not the pride of being here, but how I got here: thanks to all of you.
Let’s cheers to a new brighter start.

References

[1] https://en.wikipedia.org/wiki/Telecommunication.

[2] http://www.land.ufrj.br/tools/tools.html.

[3] http://freemat.sourceforge.net/.

[4] Jones and Bartlett. *"The evolution of data network"*. http://samples.jbpub.com/9781284059274/9781284059274_CH01_draft_5448_1.pdf.

[5] Marco Ajmone Marsan and Fatemeh Hashemi. "On the Effectiveness of Small Cells in Some Simple HetNet Scenarios". In: (2018).