A Decentralized Algorithm for Radio Resource Management in Heterogeneous Wireless Networks with Dynamic Number of Mobiles

Julio Rojas-Mora*, Habib B.A. Sidi*, Rachid El-Azouzi*, Yezekael Hayel*
*LIA, Université d’Avignon
Avignon, France
Email: [julio.rojas, habib.sidi, rachid.elazouzi, yezekael.hayel]@univ-avignon.fr

Abstract

Overlapping coverage of multiple base stations provides a multiple degree of freedom problem for mobiles which have to choose between different wireless technologies. This paper presents a framework to achieve an efficient dynamic and decentralized radio resource management in heterogeneous wireless network scenarios. Each mobile has to decide to which technology it wants to be connected to. In such problems of user-network association, we consider non-cooperative games and we show the existence and uniqueness of Nash equilibrium under some assumptions on the utility functions. We design a fully distributed algorithm that can be employed for converging the system to a Nash equilibrium. The novelties of the proposed algorithm are that, under dynamic environment (mobiles may leave or join the system), this algorithm is still robust and provides fast convergence to Nash equilibrium. Moreover, if there is a new arrival, each mobile already connected to a base station always stays connected which avoids repeated vertical handovers. Also, the new arrival runs the proposed algorithm in order to find the best base station to be connected to. For the departures, the algorithm reaches a Nash equilibrium using as few vertical handovers as possible, with as few mobiles as possible; increasing the speed of convergence. Finally, we apply our algorithm for different overlapping situations between heterogeneous wireless networks.

I. INTRODUCTION

In the world of wireless networks, the emergence of new technologies induce that mobiles will have the possibility to have access to different technologies at the same location. Indeed, with the growing offers for 3G+ access everywhere from telephony providers, more and more users equipped with laptops, netbooks or new generation smartphones can have simultaneously access to WiFi and 3G. Moreover, the emergence of wide area wireless networks based on WiMAX or LTE will complicate the allocation problem for the mobiles. Over the last years there has been a growing interest on how users should be assigned to exploit those resources more efficiently. Therefore, there have been many proposals based on joint Radio Resource Management (RRM) to efficiently share the scarce network resource in order to take the advantage of all the heterogeneous resources available. But resource management for QoS provisioning in heterogeneous wireless network is a complex task because of heterogeneity of policies and mechanisms used for QoS provisioning in different wireless networks, along-with highly probabilistic behavior of network traffic. Moreover, the presence of the multiple radio access technology provides roaming capability in different wireless networks through vertical handovers. Those handover operations may cause significant degradation to QoS provisions.

Mechanisms for optimizing and controlling mobiles in an area where there is coexistence of wireless technologies is studied in [5] with cognitive radios WiFi cells inside a WiMAX cell. Another work is described in [2] with the technology 802.11(e) and a central controller. Those studies propose a centralized approach. Some papers propose a joint radio resource (JRRM) mechanism to achieve an efficient usage of a joint pool of resources. For example, [7] and [8] propose a framework for a JRRM based on fuzzy neural methodology and reinforcement learning algorithms. Reinforcement learning techniques have been first applied in a wireless network for studying optimal power control
mechanism in [6]. The advantage of that mechanism is twofold. First, the algorithm is fully distributed. Each agent does not need a lot of information to update his decision, he needs only his perceived utility which depends on other players actions but can be easily obtained (for example if we consider that the utility depends on QoS metrics like the throughput and/or the delay). Second, it has been theoretically proved that this decentralized mechanism, if it converges, converges to a Nash equilibrium. Numerous applications of this algorithm or small variants have been proposed in the literature: spectrum sharing in a cognitive network [15], routing protocols in an ad hoc network [12], repartition of traffic between operators [1] and pricing [10]. Mainly, those studies consider fixed number of players. But, the algorithm takes a certain amount of iterations to reach a situation close to a Nash equilibrium. Moreover, depending on the system, the number of users playing the game can evolve in time very quickly. This is the case with our user-network association problem in which mobiles are leaving and joining the system dynamically. This is the main novelty of our work compared to all other works that always consider a reinforcement learning with fixed number of players.

In this paper, we consider a cell with two co-localized radio access technologies (RAT). Since in most practical scenarios, distributed algorithms are preferred over centralized ones (centralized algorithms tend to be complex and not easily scalable), we address the association problem through a fully distributed algorithm. We assume that the mobiles take alone the decision about which technology to be connected to. The utility of a user is given by the throughput perceived which depends on the number of users in the system as well as the channel condition. We model the problem as a non-cooperative game where the players are the mobiles and the strategy of a player is the choice of the technology. Whenever the system state changes (new arrivals or departures of players), every player remaining in the system, try to maximize his utility function.

Paper organization and Contribution: In this paper we present a framework to achieve an efficient dynamic and decentralized radio resource management in heterogeneous wireless network scenarios. A description of the model is given in section II. A detailed equilibrium analysis is provided in section III. Our analysis shows that there exist at most two Nash equilibria and we characterize sufficient conditions to have uniqueness of Nash equilibrium. In section IV, we design a fully distributed algorithm that can be employed for convergence to a pure Nash equilibrium. The major contributions of this paper are listed as follow:

- The novelty of the proposed algorithm is that under a dynamic environment (variable number of players), this algorithm is still robust and provides fast convergence to pure Nash equilibrium.
- Based on these rules (described in section V-C), if there is a new arrival, each mobile in the system always stays connected to a single base station which avoids repeated vertical handovers.
- We also study the impact of overlapping area on the performance of this algorithm.

The performances of our mechanism in a dynamic environment are described with several simulations and statistics in section V. Finally, we conclude the paper in section VI.

II. Problem Statement

In a context of heterogenous networks, we consider the presence of two different technologies in a common network area. This network structure has been frequently issued in the literature, and come to meet the increasing demand of mobile communications with high data rates. We can differentiate two main scenarios of coexistence. First, we consider a geographical area divided into two neighboring cells with different technologies as depicted in Figure 1. This scenario is an application of a heterogeneous system made of a wide UMTS cell and several small Wifi cells. In this scenario, the two technologies are not totally overlapped. That kind of scenario is called complementary network architecture because the two technologies are increasing the total coverage (the Wifi cells extend the coverage of wide UMTS or WiMAX cells). Users are spread into the cells and some of them are located out of the area of coexistence. Those later are connected to a single base station(technology) and have a heavy influence on the dynamic game of users inside the overlapping area.

The second scenario introduces a concurrent networks architecture: for example, the inter-networking between mobile WiMAX and HSxPA systems, with co-localized base stations as shown in Figure 2. Here, the mobiles are always in presence of the two coexisting technologies. Many of the radio access technologies in next generation
heterogeneous networks are characterized by higher rates and frequency range coupled with a relatively low coverage in order to ensure the optimization of distribution and reuse of radio resources. This scenario is thus an application of network tight coupling architecture and can be applied in a context of small cells in restricted areas or city centers.

In both scenarios, a mobile can be faced to a decision problem. A mobile may decide on which base station or technology to be connected to. This decision can be based on individual performance. Mobile performance at each base station/technology is determined by a QoS metric which depends on the number of the mobiles connected as well as the physical rate being used by the technology chosen. Every mobile would like to find the technology that maximize his individual performance. But, as this performance depends on the actions of the other mobiles, the system can be described as a non-cooperative game.

III. GAME THEORETIC MODEL

A. Static environment

We consider two systems: system 1 and system 2. At each slot, every mobile decides on which system to be connected to. Let \( N \) be the total number of users in the area of coexistence system. We define by \( n_1 \) (resp. \( n_2 \)) the number of users connected to system 1 (resp. 2). For every user, we consider that the utility function is equal to the throughput perceived by the user. The throughput is determined by the number of users as well as physical rate being used by the technology chosen. We assume also that the mobiles connected to the same system will receive the same throughput (the game is symmetric). This means that the utility function of any mobile depends only on the number of user in the system. This type of non-cooperative game is a congestion game [16]. Let \( U_i \) be the utility function of a user connected to the system \( i \). As the game is symmetric, a Nash equilibrium (NE) \((n_1^*, n_2^*)\) is given by the two conditions:

\[
U_1(n_1^*) \geq U_2(n_2^* + 1) \\
U_2(n_2^*) \geq U_1(n_1^* + 1)
\]

The previous definition means that none user connected to a system has an incentive to move to the other system. Now we consider the following assumptions:

Assumption

- \( U_i \) is bounded and decreasing,
- there exists an integer \( n_i^{th} \) such that \( U_i \) is strictly decreasing for all \( n \geq n_i^{th} \).

The last assumption immediately implies that, in any NE \((n_1^*, n_2^*)\) such that \( n_i^* \geq n_i^{th} \), at least one inequality in (1) or (1) is a strict inequality. We have the following proposition saying that there exists always a Nash equilibrium.

Proposition 1: For each \( N \), there exist a Nash equilibrium.
Proof: Without loss of generality, we assume that \( U_1(1) \geq U_2(1) \). Let \( n_{th} = \max\{n : U_1(n) \geq U_2(1)\} \). If \( n \leq n_{th} \), then the partition \((n,0)\) is a NE. We first show that there exists a NE for \( N = 2 \). Indeed, if \( U_1(2) \geq U_2(1) \), then \((2,0)\) is a NE, otherwise \((1,1)\) is a NE. Using the method of mathematical induction, we assume that there exists a NE \((n_1^*, n_2^*)\) for \( N \), and show that the above proposition is true for \( N + 1 \). To this end, we prove that the following relations hold:

\[
\text{If } U_1(n_1^* + 1) \geq U_2(n_2^* + 1) \text{ then } (n_1^* + 1, n_2^*) \text{ is a NE} \quad (3)
\]

\[
\text{If } U_1(n_1^* + 1) \leq U_2(n_2^* + 1) \text{ then } (n_1^*, n_2^* + 1) \text{ is a NE} \quad (4)
\]

We shall only prove (3), since (4) is symmetric. Given that \((n_1^*, n_2^*)\) is a NE for \( N \), then

\[
U_2(n_2^*) \geq U_1(n_1^* + 1) \geq U_1(n_1^* + 2). \tag{5}
\]

where the last inequality follows from the monotonicity of \( U_1 \). With the assumption in (3), it has been established that \((n_1^* + 1, n_2^*)\) is a NE.

The following proposition characterize the number of equilibria and sufficient conditions to have uniqueness of NE.

Proposition 2: For each \( N \), let \((n_1^*, n_2^*)\) is the NE. The non-cooperative game has one or two Nash equilibrium. Furthermore,

1. If \( U_1(n_1^*) \geq U_2(n_2^* + 1) \) and \( U_2(n_2^*) > U_1(n_1^* + 1) \), then \((n_1^*, n_2^*)\) is the unique N.E.
2. If \( U_1(n_1^*) = U_2(n_2^* + 1) \), then there are two Nash equilibria: \((n_1^*, n_2^*)\) and \((n_1^* - 1, n_2^* + 1)\)
3. If \( U_1(n_1^* + 1) = U_2(n_2^*) \), then there are two Nash equilibria: \((n_1^*, n_2^*)\) and \((n_1^* + 1, n_2^* - 1)\)

Proof: First we start to show that for each \( k \geq 2 \) the partition \((n_1^* - k, n_2^* + k)\) is not a NE. Let us assume that is not true, i.e., there exists a \( k \geq 2 \) such that \((n_1^* - k, n_2^* + k)\) is a Nash equilibrium. Thus we have

\[
U_2(n_2^* + k) \geq U_1(n_1^* - k - 1) > U_1(n_1^*) \geq U_2(n_2^* + 1) \tag{6}
\]

This therefore contradicts the fact that the function \( U_2 \) is a decreasing function in \( n \). In the same way, we may show that \((n_1^* + k, n_2^* - k)\) is not a NE.

1. To this end, we shall only prove that \((n_1^* + 1, n_2^* - 1)\) is not a Nash equilibrium. Assume that \((n_1^* + 1, n_2^* - 1)\) is a NE. Thus,

\[
U_2(n_2^* + 1) > U_1(n_1^* + 1) > U_2(n_2^* + 1)
\]

This therefore contradicts the fact that the decreasing function \( U_2 \) is a decreasing in \( n \).

2. We have

\[
U_2(n_2^* + 1) = U_1(n_1^*) \quad (7)
\]

\[
U_1(n_1^* - 1) \geq U_1(n_1^*) \geq U_2(n_2^* + 1) \geq U_2(n_2^* + 2) \tag{8}
\]

which prove that \((n_1^* - 1, n_2^* + 1)\) is a NE. It is easy to show that from equality (7), \((n_1^* + 1, n_2^* - 1)\) is not a NE.

3. The proof is similar to that of (2).

The main focus of our paper is to adapt a totally decentralized algorithm to a stochastic environment of players. This point is very important for our networking scenario and architecture. In such system, new users arrive according to a stochastic arrival process, and each user has a finite sized file to transmit. A user leaves the system when the entire file is transmitted. Then every mobile stays in the area of coexistence following a random amount of time. This sojourn time will depend on the throughput assigned to him. We mention that the non-cooperative game is played as a sequence of one stage static games where each one, we have proved, have at least a Nash equilibrium (see proposition 2).
The algorithm we have used is based on a reinforcement of mixed strategies. The players are synchronized such that the decision of all players (playing a pure strategy) induce the utility perceived for each one.

In [13] we can found the original algorithm in which we based our work. It has been proved that for a fixed number of players if this algorithm converges, it will always do to a Nash Equilibrium. But in mobile telecommunications systems, mobility and user’s activity is such that the number of users evolves rapidly.

Nonetheless, with two objectives in mind, we will try to apply this algorithm, with as few modifications as possible, when the number of users in the system (its state) is dynamic. The first one is to confirm whether this modified algorithm can be used to induce the system to be at the Nash equilibrium as frequently as possible. The second one is to make the algorithm as much distributed as possible, meaning that we would like to get from the base stations just the essential information on the state of the system or even no information at all.

Let \( N_t \) be the total number of users in the system at time \( t \). Given a set of strategies \( C = \{1, 2\} \), each player \( p \in \{1, \ldots, N_t\} \) chooses the pure strategy \( c = 1 \) with probability \( \beta_t(p) \) (and conversely chooses the strategy \( c = 2 \) with probability \( 1 - \beta_t(p) \)).

As the utility perceived by user \( p \) at time \( t \) depends on his strategy as well as that of the other users, his utility function can be expressed as:

\[
u_t^{(p)} = \mathbb{1}_{\{c_t(p)=1\}} \cdot U_1(n_t^1) + \mathbb{1}_{\{c_t(p)=2\}} \cdot U_2(N_t - n_t^1)
\]

where \( n_t^1 \) is the number if users that chose \( c = 1 \) (conversely, \( N_t - n_t^1 \) is the number if users that chose \( c = 2 \))

A reinforcement learning approach is used to update \( \beta_t(p) \) according to \( u_t^{(p)} \):

\[
\beta_t(p) = \beta_{t-1}(p) + b \cdot \left( \mathbb{1}_{\{c_t(p)=1\}} - \beta_{t-1}(p) \right) \cdot u_t^{(p)}.
\]

The reinforcement learning calculations are carried out by each user until a threshold \( \epsilon \) on consecutive results of probability \( \beta \) is not surpassed. Reaching individual convergence means that each user has no incentive on changing strategies, and there is no need to keep expending energy on calculations.

However, there are some cases where a user that has already converged, needs to restart his calculation process. In [4], even though it is not explicitly stated, it can be inferred that information about arrivals and departures is distributed to every user from a centralized entity. When this information reaches a user that has converged, his probability is reset.

We will keep the idea of having users restart calculations from a point \( \alpha \) (conversely, \( 1 - \alpha \)) close to zero (conversely, one) for users that have already converged to strategy \( c = 1 \) (conversely, \( c = 2 \)). Instead of broadcasting the information about the state of the system to all users, we define different methods that users who already converged use to estimate when an event occurs and, accordingly, restart calculations. In other words, our algorithm is defined such that users can adapt themselves to changes in the system.

The first method involves comparing at time \( t \), the utility obtained by a user in a window of the last \( n \) iterations, against a distinctive pattern that indicates the occurrence of an event. This pattern is composed of \( n - 2 \) points with the same utility value and the remaining 2 points with a different one. If the latest two points in the window have a smaller value than the remaining have, we can say there has been an arrival to the technology selected by the user. If, however, these points have a bigger value, we can say there has been a departure on the technology selected by the user. Depending on the size of the window used, false positives are more or less frequently signaled.

The method for detecting changes will be applied to infer changes in both, the technology to which the user is connected, as well as the other technology. In the former, the user will detect a reduction of the available throughput that induce a possible change in strategy. In the latter, a reduction of the noise level signals a rise in available throughput and an evaluation of a change in strategy. The user’s decision about making a handover will be taken using this information.

Detecting changes in noise, on the technology the user is not connected to, requires probing it, increasing energy consumption. It is for this reason that our second method tries to capture the departure rate of the system, making users that have converged to restart calculations periodically.
Algorithm 1 Dynamic Distributed Algorithm.

1) Initialize $\beta_{t-1}^{(p)}$ as starting probability for new players in $P$.
2) For each player $p$:
   a) If player $p$ has converged and restarting conditions are met, then:
      i) If $\beta_{t-1}^{(p)} \approx 1$ set $\beta_{t-1}^{(p)} = 1 - \alpha$.
      ii) If $\beta_{t-1}^{(p)} \approx 0$ set $\beta_{t-1}^{(p)} = \alpha$.
   b) If player $p$ has converged and restarting conditions are not met, move to player $p + 1$
   c) Player $p$ performs a choice, over $C$, according to $\beta_{t-1}^{(p)}$.
   d) Player $p$ updates his probability $\beta_{t-1}^{(p)}$ according to his choice using (10).
   e) If $|\beta_{t}^{(p)} - \beta_{t-1}^{(p)}| < \epsilon$ then player $p$ has converged.
3) Remove players that departed, make $t = t + 1$ and go to step 1.

This policy might be costly in terms of energy consumption, as users in both technologies are going to restarting frequently, so the third policy we use does not require that a user who already converged to restart calculations, leaving the choice of technology to users arriving to the system.

After a departure, the system might be left in a different point than the Nash equilibrium, until the arrival of new users returns the system to equilibrium. We will evaluate the impact of this policy in the system’s performance, but we expect faster convergence as well as a drastic reduction in the number of vertical handovers.

In the next section, we will observe by simulation what’s the behavior of our distributed algorithm in a dynamic environment with a coexistence area between two wireless wireless technologies.

V. Simulations

A. Applications

In our simulations, we consider the coexistence between WiMAX and HSDPA systems. From those simulations we are mainly interested in showing the convergence of the algorithm and it performances. The use of WiMAX and HSDPA technologies would only help to setup a network model. We consider a fixed capacity $s_w$ (resp. $s_h$) for the WiMAX (resp. HSDPA) system. In WiMAX systems, the available throughput is gradually shared between users, depending on the number of available subcarriers. Considering there is no intercell interference, we assume that the global system capacity $s_w$ is constant. The utility $u_{t}^{(w)}$ perceived by every user in WiMAX at time $t$ is given by:

$$u_{t}^{(w)} = \frac{s_w}{(n^t_w \cdot \max(s_w, s_h))}. \quad (11)$$

On the other hand, in HSDPA systems, the throughput per user is mainly affected by intra-cell interferences, created by the increasing number of users connected to the system. The type of modulation and coding scheme used, intercells interference and the distance factor are also taken into consideration. The throughput allocated to each user $p$, is computed for a given value of the signal to interference-and-noise ratio (SINR), estimated by:

$$SINR_p = \frac{g_p \cdot P_p}{\sigma^2 + \sum_{j \neq p} g_j P_j \delta_{jp}} \quad (12)$$

where inter-cells interferences are not considered, $g_p, P_p$ respectively the channel gain and transmission power of user $p$, $\delta_{jp}$ the orthogonality factor and $\sigma^2$ the additive background noise. The throughput for user $p$ is then,

$$T_p = R_p f(SINR_p), \quad (13)$$

with $R_p$ the user’s transmission rate and $f(.)$ the cumulative distribution function of the efficiency [17] estimated by $f(SINR_p) \approx (1 - \exp(-SINR_p))^M$, where $M$ is the packet length.
We will assume that the users are identical and the global throughput is a decreasing function of the number of users present in the system, that will be shared between those users. The utility perceived by each user connected to the HSDPA system at time $t$ is:

$$u_i^{(h)} = s_h f(SINR_{i_h}) / \left( n_h \cdot \max (s_w, s_h) \right).$$ (14)

Given this configuration of the simulations, we will look at the impact of the variation of the overlapping region between the different technologies on the convergence of the algorithm. More, we will consider different size of the area of coexistence, from the superposition of the cells to adjacent cells, as depicted in Figure 3.

### B. Metrics

In order to evaluate the performance of the algorithm, we define two metrics. Those metrics will allow us to compare the different restarting mechanisms inside the algorithm. To this purpose we calculate, over a sliding non overlapping window $h$ of 900 iterations of each simulation, a weighted average of the samples obtained using the kernel smoothing method:

$$KS(t) = \frac{\sum_i f(i) K\left( \frac{t-i}{h/2} \right)}{\sum_i K\left( \frac{t-i}{h/2} \right)}; \quad K(u) = \frac{3}{4} \left( 1 - u^2 \right) \cdot \mathbb{1}_{\{|u| \leq 1\}},$$ (15)

where $t$ is the point in the middle of the window and $i \in h$ are the iterations in the window.

The first metric used is:

$$PNE(t) = KS(t) : f(i) = \mathbb{1}_{\{NE\}},$$ (16)

i.e. the average proportion of iterations the system is at the Nash equilibrium in the window whose middle point is the iteration $t$. This metric allows to compare how often the policies make the algorithm reach the Nash equilibrium.

The second one is:

$$PUC(t) = KS(t) : f(i) = \frac{\sum_p \mathbb{1}_{\{p \in C_i\}}}{N_i},$$ (17)

where $C_i$ is the set of users running the reinforcement learning algorithm at each iteration $i \in h$, and $N_i$ is the number of users in the system at each iteration $i \in h$. Therefore, $PUC$ measures the average proportion of users running the reinforcement learning algorithm in the window whose middle point is the iteration $t$. This metric allows to compare how often the policies make the users perform the reinforcement learning algorithm.

A good balance between both metrics is desired.
C. Simulation scenario

In our model, we have considered that users who are trying to send files with exponentially distributed sizes, arrive following an exponentially distributed inter-arrival time rounded to the closest iteration. There are no simultaneous arrivals, since the iteration when the next arrival happens, is calculated with each new arrival.

The sojourn time will depend on the throughput assigned to each user, which means they are being served by a M/M/1 queue with Processor Sharing (PS) discipline.

This simulation scenario has the following structure:

1) Change of state detection (Case):
   - Case 1: In this case, every user that have converged is actively detecting the restarting pattern on either of the two technologies using a window of 10 iterations.
   - Case 2: In this case, users actively detect the restarting pattern using a window of 10 iterations on the technology in which they are connected with a forced periodic restarting that follows the departure rate. In this case, users that detect the pattern will restart with probability 0.5.
   - Case 3: In this case users do not restart after individual convergence.

2) We used three different sets of rates with constant load $\rho = 0.95$, by taking $\lambda \in \{1/6000, 1/4500, 1/3000\}$ and setting $\mu$ accordingly.

3) The percentage of overlapping area of both technologies to the total area covered by both base stations, $\alpha$, was taken in $\{25\%, 50\%, 100\%\}$.

For each simulation, 10 independent runs were made. Each independent run was composed of 1000000 iterations of the algorithm.

We started each simulation with 5 users. The maximum available throughput for each technology is fixed at $S = \{s_h = 2.5, s_w = 3\}$. We have picked an acceleration parameter $b = 0.3$, a convergence threshold $\epsilon = 10^{-6}$ and restarting probability $\alpha = 0.5$.

D. Results and analysis

First, we will see how the different strategies to restart calculations work. Figure 4 shows the behavior of users who follow the pattern of utility to detect changes in the state of the system (Case 1). On the left side (iteration 18600), users in WiMAX detect a change on the system (most probably an arrival on the non-overlapped region covered by the WiMAX base station) and restart calculations, converging most of them in less than 500 iterations. On the right side (iteration 20800), there is a detection of a change in the system by the users that are using HSDPA. They restart accordingly and then one chooses to stay in HSDPA and the other picks WiMAX.

An interval of a simulation of Case 2 is shown on Figure 5. Users restart when feeling changes of utility on the technology they are using, but also some of them restart periodically following the departure rate. Here, we can see that the blue and red users restart when they reach the time mark (the period is completed), and both of them go to WiMAX. Later, between iterations 6000 and 6500, there seems to be a change in the system by the users that are using HSDPA. They restart accordingly and then one chooses to stay in HSDPA and the other picks WiMAX.

On Figure 6 we can see that $PNE$ is inversely proportional to both $\alpha$ and the rates, but is not very sensitive to the policy used in the algorithm. The smallest the overlapped area, the smallest the number of users playing the game and, therefore, the better performance obtained. The same behavior is observed with the change of rates, but in this case higher rates imply faster changes in the state of the system and, therefore, worst performance is achieved. As for the policy used, the best performance is attained making users feel changes of state in both technologies (Case 1), while letting users that have converged stay in the selected technology for the remainder of their call (Case 3) offers the worst performance in general, although margins of difference are not quite dramatic.

On the other hand, differences in $PUC$ for changes in any of the parameters are big (see Figure 7). For policies 1 and 2, the biggest the overlapped area, the biggest the proportion of users performing calculations at any given iteration. Policy 3 does not follow this pattern of behavior, but this might be due to the difference of at least one order of magnitude with respect to the other two policies, which makes absolute differences very small. Another cause for the differences might be that the number of users in the overlapped region is small in with $\alpha = 25\%$, but
they are responsible to get the system to the Nash equilibrium, making them more active and taking them longer to converge. As the size of the region increases ($\alpha = 50\%$), the number of users grows proportionally, but they are not able to easily converge. Finally, when the area is completely overlapped ($\alpha = 100\%$), all users can play the game and converge quickly, leaving the responsibility to make the system reach the Nash equilibrium to new users. Changes in rates do not create the regular patterns of changes for $PUC$ that were observed in $PNE$.

**VI. CONCLUSIONS AND FUTURE WORKS**

From our simulations we can conclude that we can follow the algorithm in [13], originally developed for a fixed state (number of players in the system), and use it for a dynamic environment (variable number of players due to an M/M/1 queue with PS discipline) with excellent levels of convergence to the Nash equilibrium (above 90\%) in scenarios with low rates of arrivals and departures ($\lambda = 1/6000$), independently of the size of the overlapped area. But the algorithm decreases its performance with more users, due to higher rates or bigger overlapped area. Nonetheless, convergence to the Nash equilibrium is not very sensitive to the policy used in the algorithm.

On the other hand, the policy of not doing anything in the case of arrivals clearly has the best performance.
with respect to the proportion of users calculating at any given iteration (one order of magnitude smaller than the other policies), which coupled with its fairly good convergence (close to the other policies) makes it the best strategy tested. This means a totally decentralized algorithm with no information broadcasted by the base station, can be used by each user to collectively reach a Nash equilibrium reducing the total number of vertical handovers performed.

**REFERENCES**


