



**HAL**  
open science

# A Distributed Algorithm for Fair and Efficient User-Network Association in Multi-Technology Wireless Networks

Pierre Coucheney, Corinne Touati, Bruno Gaujal

► **To cite this version:**

Pierre Coucheney, Corinne Touati, Bruno Gaujal. A Distributed Algorithm for Fair and Efficient User-Network Association in Multi-Technology Wireless Networks. [Research Report] RR-6653, 2008, pp.32. inria-00322403v1

**HAL Id: inria-00322403**

**<https://inria.hal.science/inria-00322403v1>**

Submitted on 17 Sep 2008 (v1), last revised 7 Sep 2009 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*A Distributed Algorithm for Fair and Efficient  
User-Network Association in Multi-Technology  
Wireless Networks*

Coucheney, Pierre and Touati, Corinne and Gaujal, Bruno

N° 6653

May 2008

Thème NUM

*R*apport  
de recherche



# A Distributed Algorithm for Fair and Efficient User-Network Association in Multi-Technology Wireless Networks

Coucheney, Pierre and Touati, Corinne and Gaujal, Bruno

Thème NUM — Systèmes numériques  
Équipe-Projet MESCAL

Rapport de recherche n° 6653 — May 2008 — 29 pages

**Abstract:** Recent mobile equipment (as well as the norm IEEE 802.21) now offers the possibility for users to switch from one technology to another (vertical handover). This allows flexibility in resource assignments and, consequently, increases the potential throughput allocated to each user.

In this paper, we design a fully distributed algorithm based on trial and error mechanisms that exploits the benefits of vertical handover by finding fair and efficient assignment schemes. On the one hand, mobiles gradually update the fraction of data packets they send to each network based on the rewards they receive from the stations. On the other hand, network stations send rewards to each mobile that represent the impact each mobile has on the cell throughput.

This reward function is closely related to the concept of marginal cost in the pricing literature. Both the station and the mobile algorithms are simple enough to be implemented in current standard equipment.

Based on tools from evolutionary games, potential games and replicator dynamics, we analytically show the convergence of the algorithm to solutions that are efficient and fair in terms of throughput. Moreover, we show that after convergence, each user is connected to a single network cell which avoids costly repeated vertical handovers.

Several simple heuristics based on this algorithm are proposed to achieve fast convergence. Indeed, for implementation purposes, the number of iterations should remain in the order of a few tens. We also compare, for different loads, the quality of their solutions.

**Key-words:** Distributed Algorithms, Hybrid Wireless Networks, Evolutionary Games, Potential Games, Replicator Dynamics, Vertical Handover, Fairness.

## Un algorithme distribué pour une association utilisateur-réseau efficace et équilibrée dans les réseaux sans fils multi-technologiques

**Résumé :** Les équipements mobiles récents (tels que définis dans la norme IEEE 802.21) permettent aux usagers de basculer d'une technologie à l'autre (ce que l'on nomme "handover vertical"). Cela permet de la souplesse dans l'allocation des ressources et, par conséquent, augmente potentiellement les débits alloués aux usagers.

Dans cet article, nous concevons un algorithme distribué qui procède par tâtonnement pour trouver une association utilisateur-réseau efficace et équilibrée, afin d'exploiter les bénéfices du "handover vertical". D'une part, les mobiles mettent jour par jour la proportion de paquets de données qu'ils envoient sur chaque réseau à partir de l'utilité que leur transmet la station de base. D'une autre part, les stations de base des réseaux calculent et envoient l'utilité des mobiles qui représente l'impact que chaque mobile a sur le débit global du réseau.

Cette fonction d'utilité est rapprochée de l'idée du coût marginal dans la littérature sur la tarification. Aussi bien l'algorithme de la station de base que celui du mobile sont suffisamment simples pour être implémentés dans les équipements standards actuels.

À partir de méthodes des jeux évolutionnaires, des jeux de potentiel et de la dynamique de réplique, nous montrons de manière analytique la convergence de l'algorithme vers une solution efficace et équilibrée en terme de débit. De plus, nous montrons qu'une fois l'équilibre atteint, chaque utilisateur est connecté à un unique réseau ce qui permet de supprimer le coût du "handover vertical".

Plusieurs heuristiques reposant sur cet algorithme sont proposées afin d'obtenir une convergence rapide. En effet, pour des raisons d'ordre pratique, le nombre d'itérations doit demeurer de l'ordre de quelques dizaines. Nous comparons alors la qualité des solutions fournies dans divers scénarios.

**Mots-clés :** Algorithmes distribués, réseaux sans-fils hétérogènes, interconnexion de réseaux, théorie des jeux évolutionnaires, jeux de potentiels, dynamique de réplique, handover vertical, équilibre.

## 1 Introduction

The overall wireless market is expected to be served by six or more major technologies (GSM, UMTS, HSDPA, WiFi, WiMAX, LTE). Each technology has its own advantages and disadvantages and none of them is expected to eliminate the rest. Moreover, radio access equipment is becoming more and more multi-standard, offering the possibility of connecting through two or more technologies concurrently, using the norm IEEE 802.21. Switching between networks using different technology is referred to as *vertical handover*. This is currently done in UMA, for instance, which gives an absolute priority to WiFi over UMTS whenever a WiFi connection is available. In this paper, in contrast, we address the problem of computing an optimal association by providing a distributed algorithm which is both fair to all users and optimal in terms of overall throughput. Here are the contributions of the paper.

- First, we propose a distributed algorithm with guaranteed convergence to a non-cooperative equilibrium. This algorithm is based on an iterative mechanism: at each time epoch the mobile nodes adapt the proportion of the traffic they send on each network, based on some reward measure they receive from the network. This work is in line with some recent work on learning of Nash equilibria (see, for instance, [1] [2]).
- Second, based on tools from potential games, we show that, by appropriately setting up the reward measure, the resulting equilibria can be made efficient or fair.
- Last, we show that the obtained equilibrium is always pure: after convergence each user is associated to a single technology.

To validate our results, we propose several practical implementations of the algorithm and assess their performance in the practical setting of a geographical area covered by a global WiMAX network overlapping with several local IEEE 802.11 (also called WiFi) cells. We suppose that each user can multi-home, that is to say split her traffic between her local WiFi network and the global WiMAX cell, in order to maximize her reward (to be defined later).

The integration of WiFi and UMTS or WiFi and WiMAX technologies has already received some attention in the past.

There is a family of papers looking for solutions using Markov or Semi-Markov Decision Processes [3, 4]. Based on Markovian assumptions upon the incoming traffic, these works provide with numerical solutions, so as to optimize some average or discounted reward over time. Yet, because of the complexity of the system at hand (the equations of the throughput in actual wireless systems are not linear, and not even convex), important simplifying assumptions need to be made, and the size of the state space quickly becomes prohibitive to study real systems. Moreover, these methods require to precisely know the characteristics of the system (e.g. in terms of

bandwidth achieved in all configurations, interference impact of one cell over the neighboring ones), data which are hardly available in practice.

Our approach is rather orthogonal as we seek algorithms that converge towards the optimal allocation, using real-time measurements rather than off-line data. Such an approach follows game theory frameworks. There has been recent work that, based on evolutionary games [5], provide with optimal equilibria. Evolutionary games [6, 7], or the closely-related population games, are based on Darwinian-like dynamics. The evolutionary game literature is now mature and includes several so-called population dynamics, which model the evolution of the number of individuals of each population as time goes by. In our context, a population can be seen as a set of individuals adopting the same strategy (that is to say choosing the same network cell in the system and adopting identical network parameters). Recent work [5] have shown that, considering the so-called *replicator dynamics*, an appropriate choice of the fitness function (that determines how well a population is adapted to its environment) leads to efficient equilibria, which is quite interesting. However they do not provide with algorithms that follow the replicator dynamics (and hence converge to the equilibria). Additionally they do not justify the use of evolutionary games. Indeed, such games assume a large number of individuals, each of them having a negligible impact on the environment and the fitness of others. This assumption is not satisfied here, where the number of active users in a given cell is on the order of a few tens. The arrival or departure of a single one of them hence significantly impacts the throughput allocated to others. As the number of players is limited, we are hence dealing with another kind of equilibria, namely the Nash Equilibria.

The third trend of papers this article is in line with, concerns Nash equilibria learning mechanisms. In the context of load balancing, a few algorithms (see, for instance [1, 2]) have been shown to converge to Nash Equilibria. Interestingly enough, it has been pointed out that this class of algorithms has similar behavior and convergence properties as replicator dynamics in evolutionary game theory. It is to be noted that the main weakness of these algorithms is that they may converge to *mixed* strategy Nash equilibria, that is to say to equilibria where each user randomly picks up a decision at each time epoch. Such equilibria are unfortunately not interesting in our case, as they amount to perpetual handover between networks.

Finally, there is a growing interest in measuring or analyzing the efficiency of Nash Equilibria. The most famous concept is certainly the “price of anarchy” [8]. Let us also mention the more recent SDF (Selfish Degradation Factor) [9]. We will show in the following that the Nash Equilibria our algorithm converges to are always optimal with respect to these two performance measures. In addition, it has interesting fairness properties. Indeed, we show how our algorithm can be tuned so as to converge to  $\alpha$ -fair points (defined in cooperative game theory, see [10]), for arbitrary value of

the parameter  $\alpha$ . This wide family of fairness criteria includes in particular the well-known max-min fairness and proportional fairness and can be generalized so as to cover the Nash Bargaining Solution point [11].

In the present paper, we hence propose to make use of the previous works in evolutionary games on heterogeneous network, with additional fairness considerations, while proposing methods based on works on Nash learning algorithms that can be implemented on future mobile equipments. In addition, our work present a novel result which is that our algorithm converges to *pure* (as opposed to mixed) equilibria, preventing undesired repeated handovers between stations.

## 2 Framework and Model

In this section, we present the model and the objective of this work while introducing the notations used throughout the paper.

### 2.1 Interconnection of Heterogeneous wireless networks

We consider a set  $\mathcal{C}$  of network cells, that can be of various technologies (WiFi, WiMAX, UMTS, LTE...), and a fixed set  $\mathcal{I}$  of active users. Any of the  $|\mathcal{I}|$  users<sup>1</sup> can connect to a specific subset of these cells and network technologies, depending on her geographical location, wireless equipment and operator subscription.

### 2.2 User throughput and cell load

By *throughput*, we refer to the rate of useful information available for a user, in a given network, sometimes also called *goodput* in the literature.

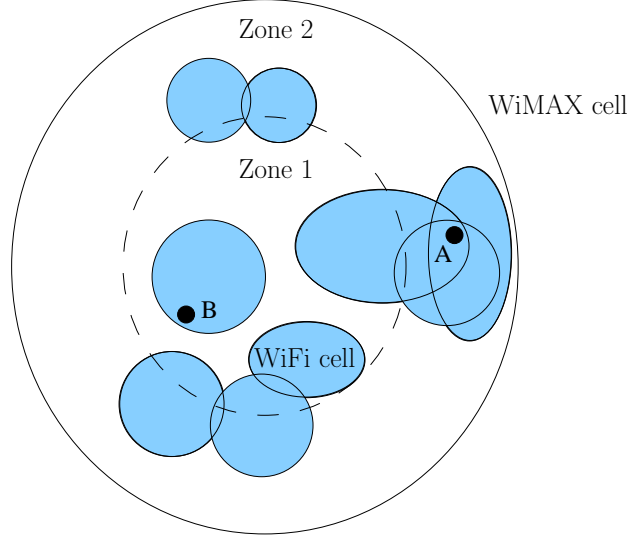
The throughput obtained by an individual on a network depend on both her own parameters and the ones of others. These parameters include geographical position (interference and attenuation level) as well as wireless card settings (coding schemes, TCP version, to cite a few). As commonly done in previous papers [3, 4], we discretize the cells of networks into *zones* of identical throughput. This means that users in the same zone will receive the same throughput. We denote by  $\mathcal{Z}_c$  the set of zones in cell  $c$ . The distribution of users and their number in the zones of a network is called *load* of the network (see Fig 1).

More formally, we suppose that each user has a set of network cells she can connect to, and a specific zone associated to each of them<sup>2</sup>. An admissible choice  $a_i$  for user  $i$  is a pair  $a_i = (a_i^c, a_i^z) \in \mathcal{A}_i$ . The set of all

<sup>1</sup>In the following we use the term *users* and *mobiles* interchangeably.

<sup>2</sup>Note that in some cases a user may have a choice of two zones of a given cell, corresponding to 2 different sets of network parameters. The framework allows this by setting two different choices with identical cells associated to them but different zones.





**Figure 1:** An heterogeneous wireless system consisting of a single MAN (Metropolitan Area Network, e.g. WiMAX) cell and a set of partly overlapping LAN (Local Area Networks, e.g. WiFi) hot-spots (in grey). As user  $B$  (in zone 1) is closer to the WiMAX antenna, it can use a more efficient coding scheme than  $A$  (in zone 2) (for instance QAM 16 instead of QPSK). Zones are represented with a dash line, as opposed to cells, with full lines.

possible choices is  $\mathcal{A} \stackrel{\text{def}}{=} \otimes_{i \in \mathcal{I}} \mathcal{A}_i$ . The user's decision is denoted by  $A_i \stackrel{\text{def}}{=} (A_i^c, A_i^z)$ . Then, we denote  $A$  the vector of users decisions  $A \stackrel{\text{def}}{=} (A_i)_{i \in \mathcal{I}}$ , and call it an admissible *association*. Hence, an association is a function from the set of users to the set of possible choices.

For each association  $A$ , the *load* on zone  $z$  of network  $c$  is denoted by  $l_{(c,z)}(A)$ , the number of users in this zone using cell  $c$ :

$$\forall c \in \mathcal{C}, z \in \mathcal{Z}_c, \quad l_{c,z}(A) \stackrel{\text{def}}{=} \sum_{i \in \mathcal{I}} \delta_{A_i, (c,z)},$$

with  $\delta$  the Kronecker delta ( $\delta_{a,b} = 1$  if  $a = b$  and 0 otherwise). Hence, the load  $l_c$  on cell  $c$  is a vector of size  $|\mathcal{Z}_c|$  whose components correspond to the load on a particular zone:  $l_c(A) \stackrel{\text{def}}{=} (l_{c,z}(A))_{z \in \mathcal{Z}_c}$ .

**Assumption 1.** *The throughput  $t_{c,z}$  of cell  $c$  in zone  $z$  is a function depending only of the vector load  $l_c(A)$  of cell  $c$ .*

With this notations, the throughput received by user  $i$  when she takes decision  $A_i = a_i$  is  $t_{a_i}(l_{a_i^c}(A))$ .

### 2.3 Pure versus mixed strategies

As opposed to multi-homing between WiFi systems (see [12, 5]), multi-homing between different technologies (e.g. WiFi and WiMAX) induces several complications: the different technologies may have different delays, have different packet sizes or coding systems,... and re-constructing the messages sent by the mobiles may be hazardous. Hence, while each user can freely switch between the networks cells she has access to, we aim at algorithms that converge - after a transitional state - to equilibria in which each user uses a single network (so as to avoid the cumbersome handover procedure). These are called *pure strategy equilibria* (see Section 3).

Yet, during the convergence phase, each mobile is using *mixed strategies*<sup>3</sup>. Then, the experienced throughput needs to be considered in terms of expectations. In this case,  $q_i$  is a vector of probabilities where  $q_{i,a_i}$  is the probability for mobile  $i$  to use choice  $a_i \in \mathcal{A}_i$ . The global strategy set is the matrix  $Q \stackrel{\text{def}}{=} (q_i)_{i \in \mathcal{I}}$  while  $A_i$  and  $T_i$  are random variables corresponding respectively to the decision and the throughput of user  $i$  (depending on probabilities  $Q$ ).

### 2.4 Efficiency and Fairness

In our approach, we consider elastic or data traffic. Then, the Quality-of-Service (QoS) experienced by each mobile user is its experienced throughput. We are hence interested in seeking equilibria that are optimal (in the sense of Pareto) in terms of throughput. Such equilibria are matrices  $Q$  such that one cannot find another strategy  $Q'$  that increase the expected throughput of a user without decreasing that of another one:  $\forall Q' \neq Q, \exists i \in \mathcal{I}$  s. t.  $\mathbb{E}_{Q'}[T_i] > \mathbb{E}_Q[T_i], \Rightarrow \exists j \in \mathcal{I}, \mathbb{E}_{Q'}[T_j] < \mathbb{E}_Q[T_j]$ .

We design a fully distributed algorithm that converges to points which are not only Pareto optimal but also  $\alpha$ -fair. The class of  $\alpha$ -fair points [10], achieves

$$\max_Q \sum_{i \in \mathcal{I}} \mathbb{E}_Q[G(T_i)] \text{ with } G(x) \stackrel{\text{def}}{=} \frac{x^{1-\alpha}}{1-\alpha}. \quad (1)$$

In the case of pure strategies, for each mobile  $i$  such that  $A_i = a_i$ , then  $\mathbb{E}_Q[T_i] = t_{a_i}(l_{a_i^c}(A))$ . So, we aim at building an algorithm that converges to a set of decisions  $A^* = (A_i^*)_{i \in \mathcal{I}}$  that reaches

$$\max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} G(t_{a_i}(l_{a_i^c}(A))).$$

When  $\alpha = 0$ , the corresponding solution is a social optimum. When  $\alpha$  tends to one, the solution is a proportional fair point (or Nash Bargaining

<sup>3</sup>The formal definition is given in Section 3. Roughly speaking, this means in our context that each packet emitted by each mobile is sent over a randomly chosen cell, according to some probability matrix.

Solution) and when  $\alpha$  tends to infinity, it converges to a max-min fair point. The parameter  $\alpha$  hence allows flexibility in choosing between fully efficient versus fair allocation, while ensuring Pareto optimality.

Finally, it is well-known that selfish behavior in the use of resources (networks) may lead to inefficient use, in case of congestion for example. To circumvent this, we introduce some reward functions that are notified to users. Thus, instead of competing for throughput, we consider an algorithm reflecting a non-cooperative game between users that compete for maximizing their rewards. We will give an explicit closed-form of the reward function in Section 4. As in the throughput case, the reward on a zone of a cell  $r_c$  only depends on the load on that cell. When  $A_i = a_i$ , we denote by  $r_{a_i}(l_{a_i^c}(A))$  the reward received by user  $i$  (as for the throughput, the reward received by user  $i$  also depends on the choices of the other mobiles of the system, as reflected in the association vector  $A$ ). In the case of mixed strategies, we denote by  $R_i$  the random variable corresponding to the reward of user  $i$  (depending on probabilities  $Q$ ). Its mean is  $\mathbb{E}_Q[R_i]$ .

### 3 Nash Learning Algorithm

In this section, we build an iterative algorithm with guaranteed convergence to the Nash equilibrium of the system. To this end, we start by recalling the definition of Nash equilibria (Section 3.1), then we present the learning algorithm (Section 3.2) as well as the reward functions that allows us to obtain fair and efficient convergence points.

#### 3.1 Nash Equilibria

In this subsection, we first recall the definitions of Nash equilibria and some fundamental properties.

Firstly, let us introduce a useful notation: let  $x$  be a vector. We denote by  $x_s$  the  $s$  component of vector  $x$ , and by  $x_{-s}$  the set of the other components. It follows that up to some re-ordering,  $x = (x_s, x_{-s})$ . For instance, we can now rewrite the load on cell  $c$  as  $l_c = (l_{c,z}, l_{c,-z})$ .

We also introduce  $e_{a_i}$  a vector of size  $\mathcal{Z}_{a_i^c}$  defined by

$$e_{a_i}[s] \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } s = a_i^z, \\ 0 & \text{otherwise.} \end{cases}$$

**Definition 1 (Pure strategies Nash Equilibria).** *An association scheme  $A = (A_i)_{i \in \mathcal{I}}$  such that  $\forall i, A_i = a_i$  on the set  $\mathcal{I}$  of players on the set  $\mathcal{C}$  of cells is a pure Nash Equilibrium for reward  $r$  if,  $\forall i \in \mathcal{I}, \forall a'_i \neq a_i, r_{a_i}(l_{a_i^c}(A)) \geq r_{a'_i}(l_{a'_i^c}(A) + e_{a'_i})$ .*

Now, suppose that instead of choosing  $a$ , the strategy of each user  $i$  is the choice of a vector probability  $q_i$  over the possible choices. Each component

$q_i$  of the vector is the probability with which she takes action  $a_i \in \mathcal{A}_i$ . This is called a *mixed* strategy. Equivalently, a mixed strategy of a user is the choice of the percentage of packets or sessions she sends to any of the possible networks she has access to. Let us denote by  $\mathcal{S}_i$  the set of strategies for user  $i$ :

$$\mathcal{S}_i = \{q_i \in [0, 1]^{|A_i|} \text{ s.t. } \sum_{a_i \in \mathcal{A}_i} q_{i,a_i} = 1\},$$

and  $Q \in \mathcal{S} = \otimes_{i \in \mathcal{I}} \mathcal{S}_i$  the strategy matrix for all users:  $Q = (q_i)_{i \in \mathcal{I}}$ . One can notice that  $\mathcal{S}_i$  is a convex and compact subset of  $\mathbb{R}^{|A_i|}$ .

**Definition 2 (Mixed strategies Nash Equilibria).** *A set of  $|\mathcal{I}|$  probability vectors  $q_i$  of size  $|\mathcal{C}|$  is a Nash Equilibrium if for reward  $R$ ,*

$$\forall i, \forall q'_i \neq q_i, \mathbb{E}_{q_i, q_{-i}}[R_i] \geq \mathbb{E}_{q'_i, q_{-i}}[R_i].$$

As the number of users *and* their set of choices (networks) is finite, this is a *finite game* so that it admits (at least one) mixed strategy Nash Equilibria [13].

### 3.2 The Nash Learning Algorithm

A Nash learning algorithm is an iterative algorithm on the strategy set that converges to a Nash equilibrium. Based on [1], we consider Algorithm 1, where  $b_i(\sigma) \in \mathbb{R}$  is the algorithm step for user  $i$  at epoch  $\sigma$ . Since  $\forall i \in \mathcal{I}$ ,  $q_i$  must remain in the strategy space  $\mathcal{S}_i$ , the step size is constrained by:  $m \leq b_i(\sigma) r_{c,z}(\sigma) \leq 1$ , where  $m = \max_{a_i} \left( \max \left( -\frac{q_{i,a_i}}{1 - q_{i,a_i}} - \frac{1 - q_{i,a_i}}{q_{i,a_i}} \right) \right) \leq 0$ .

---

#### Algorithm 1 Learning Algorithm

---

Initialize arbitrarily vectors  $q_i(0)$  for all users

At each time epoch  $\sigma$ , **forall** user  $i$  **do**

Take decision  $\bar{a}_i \stackrel{\text{def}}{=} A_i(\sigma)$  with probability  $q_i(\sigma)$

Receive reward  $r_{\bar{a}_i}(l_{\bar{a}_i}(A(\sigma)))$

Update strategy vector:

$$\forall a_i \in \mathcal{A}_i, q_{i,a_i}(\sigma+1) \leftarrow q_{i,a_i}(\sigma) + b_i(\sigma) r_{a_i}(l_{a_i}(A(\sigma))) (\delta_{a_i^c, a_i^c} - q_{i,a_i}(\sigma)) \quad (2)$$


---

Eq. 2 determines the update mechanism, which we call *system dynamics*. Observe that this dynamics can also be put in the matrix form:

$$Q(\sigma + 1) = Q(\sigma) + b(\sigma)H(Q(\sigma)), \quad (3)$$

where  $Q(\sigma) \stackrel{\text{def}}{=} (q_i(\sigma))_{i \in \mathcal{I}} \in \mathcal{S}$  denotes the state of users at instant  $\sigma$ , and  $b(\sigma) = (b_i(\sigma))_{i \in \mathcal{I}}$ . Function  $H(\cdot)$  represents the update specified by Algorithm 1, and does not depend on step size  $b(\sigma)$ .

**Reward function** It is known [14] that introducing some costs in the rewards of users can increase the efficiency of the Nash equilibrium. An option is to apply the *marginal cost pricing*, which asserts that each user on a network should pay a tax balancing the loss of throughput caused by his presence.

According to the random decision  $A$ , the reward function for user  $i$  is defined as:

$$r_{A_i}(l_{A_i^c}(A)) = G(t_{A_i}(l_{A_i^c}(A))) - \sum_{j \neq i} \delta_{A_j^c, A_i^c} \left( G(u_{i,j}(A)) - G(v_j(A)) \right), \quad (4)$$

where  $\delta$  is the Kronecker delta and  $u_{i,j}$  and  $v_j$  are the following functions from the set of associations to  $\mathbb{R}$ :

$$\begin{cases} u_{i,j}(A) = t_{A_j}(l_{A_j^c}(A) - e_{A_i}), \\ v_j(A) = t_{A_j}(l_{A_j^c}(A)). \end{cases}$$

One can notice that  $u_{i,j}(A)$  is equal to the throughput of user  $j$  when removing user  $i$  of the system. Then,  $v_j(A) - u_{i,j}(A)$  represents the loss of throughput for user  $j$  caused by the presence of user  $i$  in the same network. Finally, the interpretation of the reward function is the following:

- $G(t_{A_i}(l_{A_i^c}(A)))$  is the  $\alpha$ -fair throughput obtained by user  $i$ ,
- $\sum_{j \neq i} \delta_{A_j^c, A_i^c} (G(u_{i,j}(A)) - G(v_j(A)))$  is the total loss of  $\alpha$ -fair throughput for others users on cell  $A_i^c$  caused by the presence of user  $i$ .

We show, in the following section, the convergence of the algorithm to an efficient Nash equilibrium.

## 4 Algorithm Properties

In this section, we study the Nash learning algorithm presented in Section 3. We start by showing its convergence, in a probabilistic way, using replicator dynamics (Section 4.1). Then, we prove that, using our reward function, the equilibrium attained is efficient (Section 4.2), and that it is a pure Nash equilibrium (Section 4.3).

### 4.1 Replicator Dynamics

In the following study, we consider  $b \in \mathbb{R}$  defined by:  $b \stackrel{\text{def}}{=} \sup_{\sigma} \max_{i \in \mathcal{I}} b_i(\sigma)$ , and  $Q^b(\cdot)$ , the piecewise-constant interpolation matrix of  $Q(\sigma)$ , defined by:  $Q^b(\theta) \stackrel{\text{def}}{=} Q(\sigma)$ ,  $\forall \theta \in [\sigma b, (\sigma + 1)b)$ . Now consider  $\{Q^b(\cdot); b > 0\}$  and the limit  $Q(\cdot)$  of this sequence as  $b \rightarrow 0$ .

Recall that a sequence of random variables  $(A_t)_{t \in \mathbb{R}}$  *weakly converges* to a random variable  $A$  if for any continuous and bounded function  $f$ :  $\mathbb{E}[f(A_t)] \rightarrow \mathbb{E}[f(A)]$ . Then, we have:

**Theorem 1.** *The sequence  $\{Q^b(\cdot)\}$  of interpolated processes weakly converges, as  $b \rightarrow 0$ , to  $Q(\cdot)$ , which is the (unique) solution of Cauchy problem*

$$\frac{dQ}{d\theta} = \Phi(Q), \quad Q(0) = Q_0 \quad (5)$$

where  $\Phi : \mathcal{S} \rightarrow \mathbb{R}^{|\mathcal{Z}| \times |\mathcal{A}|}$  is given by (with  $H$  as defined in (3))  $\Phi(Q) = \mathbb{E}_Q[H(Q)]$ .

*Proof.* This theorem is a particular case of a general result due to Kushner (Theorem 3.2 in [15]). See also [2] for a discussion in this scenario.  $\square$

**Corollary 1.** *When  $b \rightarrow 0$ , the sequence  $\{q_{i,a_i}^b(\cdot)\}$  weakly converges to the solution of the differential equation*

$$\frac{dq_{i,a_i}}{d\theta} = q_{i,a_i}[f_{i,a_i}(Q) - \bar{f}_i(Q)], \quad (6)$$

with  $f_{i,a_i}(Q) = \mathbb{E}_Q[R_i | A_i = a_i]$  and  $\bar{f}_i(Q) = \sum_{a'_i \in \mathcal{A}_i} q_{i,a'_i} f_{i,a'_i}(Q)$ .

*Proof.* Using the fact that  $1 - q_{i,a_i} = \sum_{a'_i \neq a_i} q_{i,a'_i}$ ,

$$\begin{aligned} \mathbb{E}_Q[H(Q)]_{i,a_i} &= q_{i,a_i}(1 - q_{i,a_i})\mathbb{E}_Q[R_i | A_i = a_i] - \sum_{a'_i \neq a_i} q_{i,a_i} q_{i,a'_i} \mathbb{E}_Q[R_i | A_i = a'_i] \\ &= q_{i,a_i} \left( \sum_{a'_i \neq a_i} q_{i,a'_i} \mathbb{E}_Q[R_i | A_i = a_i] - \sum_{a'_i \neq a_i} q_{i,a'_i} \mathbb{E}_Q[R_i | A_i = a'_i] \right) \\ &= q_{i,a_i} \left( f_{i,a_i}(Q) - \sum_{a'_i \in \mathcal{A}_i} q_{i,a'_i} \mathbb{E}_Q[R_i | A_i = a'_i] \right) \\ &= q_{i,a_i} (f_{i,a_i}(Q) - \bar{f}_i(Q)). \end{aligned}$$

Then, by (5), the result holds.  $\square$

This relation is well-known in evolutionary game theory, under the name *replicator dynamics*. Indeed, in an evolutionary game,  $q_{i,a_i}$  can be interpreted as the fraction of population  $i$  using choice  $a_i$ . Then, if the set of individuals choosing  $a_i$  is adapted to the surrounding landscape (i.e. its fitness is higher than the average one:  $f_{i,a_i}(Q) > \bar{f}_i(Q)$ ) it would develop (increase in terms of number of individuals:  $\frac{dq_{i,a_i}}{d\theta} > 0$ ) while badly adapted sets of individuals decrease in size and might eventually disappear.

## 4.2 Convergence Properties

Let  $V$  be a dynamics of the form:  $\frac{dQ}{d\theta} = V(Q)$ . We say that  $\bar{Q}$  is a *stationary point* for the dynamics  $V$  if  $V(\bar{Q}) = 0$ . Also,  $\bar{Q}$  is *asymptotically stable* if there exists a neighborhood  $U$  of  $\bar{Q}$  such that:

$$Q(\theta) \in U \Rightarrow Q(\theta) \longrightarrow \bar{Q} \text{ when } \theta \longrightarrow \infty.$$

From Eq. (6), one can notice that any pure strategy (i.e. any vector  $q_i$  having a component equal to 1) is stationary. Also, from a folk theorem (see [2, 6]):

- All Nash equilibria are stationary points.
- All asymptotically stable stationary points are Nash equilibria.

Hence, the dynamics (6), (and hence the learning algorithm when  $b$  goes to 0) can only converge to a Nash equilibrium of the system.

Following an idea of [12], we now show that the specified reward function (4) ensures that the algorithm 1 converges toward an efficient and fair association scheme.

Consider the function  $F : \mathcal{S} \longrightarrow \mathbb{R}$  which is the summation on all the users of their expected  $\alpha$ -fair throughput:

$$F(Q) = \sum_{i \in \mathcal{I}} \sum_{a_i \in \mathcal{A}_i} q_{i,a_i} \mathbb{E}_Q[G(T_i) | A_i = a_i]. \quad (7)$$

Then, we have the following proposition:

**Proposition 1.** *F is a potential function for the dynamics (6), i.e. :*

$$\forall i \in \mathcal{I}, \forall a_i \in \mathcal{A}_i, f_{i,a_i}(Q) = \frac{\partial F}{\partial q_{i,a_i}}(Q).$$

*Proof.* Let us first differentiate the function  $F$ :

$$\frac{\partial F}{\partial q_{i,a_i}}(Q) = \mathbb{E}_Q[G(T_i) | A_i = a_i] + \sum_{j \neq i} q_{j,a_j} \frac{\partial \mathbb{E}_Q[G(T_j) | A_j^c = a_j^c]}{\partial q_{i,a_i}}.$$

Recall:

$$r_{A_i}(l_{A_i^c}(A)) = G(t_{a_i}(l_{A_i^c}(A))) - \sum_{j \neq i} \delta_{A_j^c, A_i^c} \left( G(u_{i,j}(A)) - G(v_j(A)) \right).$$

Assume  $a_i^c = c$  and let us denote by  $\mathcal{L}_c$  the set of possible loads on cell  $c$ . Hence,

$$\begin{aligned}
f_{i,a_i}(Q) &= \mathbb{E}_Q[R_i | A_i = a_i] \\
&= \mathbb{E}_Q[G(T_i) | A_i = a_i] \\
&\quad - \sum_{j \neq i} \delta_{A_j, A_i^c} (\mathbb{E}_Q[G(u_{i,j}(A)) | A_i = a_i] - \mathbb{E}_Q[G(v_j(A)) | A_i = a_i]) \\
&= \mathbb{E}_Q[G(T_i) | A_i = a_i] \\
&\quad - \sum_{j \neq i} \sum_{\bar{l}_c \in \mathcal{L}_c} G(t_{a_j}(\bar{l}_c)) (\mathbb{P}[l_c = \bar{l}_c + e_{a_i}, A_j^c = c | A_i^c = c] - \mathbb{P}[l_c = \bar{l}_c, A_j^c = c | A_i^c = c]) \\
&= \mathbb{E}_Q[G(T_i) | A_i = a_i] \\
&\quad - \sum_{j \neq i} q_{j,a_j} \sum_{\bar{l}_c \in \mathcal{L}_c} G(t_{a_j}(\bar{l}_c)) (\mathbb{P}[l_c = \bar{l}_c | A_i^c \neq c, A_j^c = c] - \mathbb{P}[l_c = \bar{l}_c | A_i^c = A_j^c = c]) \\
&= \mathbb{E}_Q[G(T_i) | A_i = a_i] + \sum_{j \neq i} q_{j,a_j} \frac{\partial \mathbb{E}_Q[G(T_j) | A_j^c = a_i^c]}{\partial q_{i,a_i}}.
\end{aligned}$$

□

The concept of potential functions in game theory was proposed by Dov Monderer and Lloyd Shapley [16] and is now well-studied (see for instance [17]). For intuition, consider a state  $Q$  at which  $f_{i,c}(Q) > f_{i,d}(Q)$ . At such a state, user  $i$  has a greater fitness using cell  $c$  than using cell  $d$ , so that he would prefer to switch to cell  $c$ . But  $\frac{\partial F}{\partial q_{i,c}}(Q) - \frac{\partial F}{\partial q_{i,d}}(Q) = f_{i,c}(Q) - f_{i,d}(Q) > 0$ , so this profitable strategy change leads to a marginal increase in potential.

**Theorem 2.** *Using the reward function (4), the dynamics (6), and hence Algorithm 1 converge to an asymptotically stable state that maximizes the total expected  $\alpha$ -fair throughput.*

*Proof.* First, the potential function  $F$  is a global Lyapunov function for the dynamics, i.e.  $\langle V(Q), \nabla F(Q) \rangle \geq 0, \forall Q \in \mathcal{S}$  where:

- $V(Q)$  is the dynamics (6):  $\frac{dq_{i,a_i}}{d\theta} = (V(Q))_{i,a_i}$ ,
- $\nabla F(Q) = (\frac{\partial F}{\partial q_{i,a_i}}(Q))_{i,a_i}$ ,
- $\langle \cdot, \cdot \rangle$  is the Euclidian inner product.



Indeed:

$$\begin{aligned}
\langle V(Q), \nabla F(Q) \rangle &= \sum_{i \in \mathcal{I}} \sum_{a_i \in \mathcal{A}_i} (V(Q))_{i, a_i} f_{i, a_i}(Q) \\
&= \sum_{i \in \mathcal{I}} \sum_{a_i \in \mathcal{A}_i} q_{i, a_i} [f_{i, a_i}(Q) - \bar{f}_i(Q)] f_{i, a_i}(Q) \\
&= \sum_{i \in \mathcal{I}} \left[ \sum_{a_i \in \mathcal{A}_i} q_{i, a_i} f_{i, a_i}(Q)^2 - \left( \sum_{a_i \in \mathcal{A}_i} q_{i, a_i} f_{i, a_i}(Q) \right)^2 \right].
\end{aligned}$$

Now, since  $\sum_{a_i \in \mathcal{A}_i} q_{i, a_i} = 1$ , by Jensen's inequality we have that the term inside the brackets is non-negative and is equal to zero if and only if  $f_{i, a_i}(Q) = \bar{f}_i(Q)$ . This means that  $(V(Q))_{i, a_i} = 0$ . Thus, the summation is also non-negative and is zero if and only if  $Q$  is a stationary point for the dynamics  $V$ .

Since  $F$  is continuous on the compact set  $\mathcal{S}$ , the maximum of  $F$  on this set is reached, then we infer that the dynamics converge to an asymptotically stable state which is a local maximum of the potential function.

We denote by  $\hat{Q}$  the equilibrium attained by the dynamics. Since  $\hat{Q}$  is a stationary point for the dynamics, we have:

$$\begin{cases} f_{i, a_i}(\hat{Q}) = \bar{f}_i(\hat{Q}) \\ \text{or} \\ \hat{q}_{i, a_i} = 0. \end{cases} \quad (8)$$

Now consider the constrained optimization problem,

$$\max_{Q \in \mathcal{S}} F(Q). \quad (9)$$

We call  $Q^*$  a solution of this problem, then  $F(Q^*) \geq F(\hat{Q})$ . The corresponding Lagrangian function is ( $\mu_{i, a_i} \geq 0$  and  $\lambda_i \in \mathbb{R}$ ):

$$L(Q, \lambda, \mu) = F(Q) - \sum_{i \in \mathcal{I}} \lambda_i \left( \sum_{a_i \in \mathcal{A}_i} q_{i, a_i} - 1 \right) + \sum_{i \in \mathcal{I}} \sum_{a_i \in \mathcal{A}_i} \mu_{i, a_i} q_{i, a_i}.$$

Therefore, problem (9) can be rewritten as the unconstrained optimization problem  $\max_Q \left( \min_{\lambda, \mu} L(Q, \lambda, \mu) \right)$  and the dual problem is

$$\min_{\lambda, \mu} \left( \max_Q L(Q, \lambda, \mu) \right).$$

We have the following inequality:

$$\min_{\lambda, \mu} \max_Q L(Q, \lambda, \mu) \geq \max_Q \min_{\lambda, \mu} L(Q, \lambda, \mu). \quad (10)$$

Now, using the fact that function  $f$  derives from  $F$ , we can see that the stationary point conditions (8) are Kuhn-Tucker first order conditions of the Lagrange dual function, with  $\hat{\lambda}_i = \overline{f}_i(\hat{Q})$  and  $\hat{\mu}_{i,a_i} = \overline{f}_i(\hat{Q}) - f_{i,a_i}(\hat{Q}) \geq 0$ . This implies that the dynamics converges to the solution of the Lagrange dual problem. Since the stationary conditions hold,  $L(\hat{Q}, \hat{\lambda}, \hat{\mu}) = F(\hat{Q})$ , and by (10) we have:  $F(\hat{Q}) \geq F(Q^*)$ . Consequently,  $F(\hat{Q}) = F(Q^*)$ .  $\square$

This proves our claim of subsection 3.2: introducing some appropriate cost in the reward functions of users leads to an efficient and fair equilibria.

### 4.3 Pure Strategies versus Mixed Strategies

To avoid excessive cumbersome handover procedures, we want the algorithm to converge to a pure strategy. In this section, we give a formal proof of such a property.

**Proposition 2.** *Algorithm 1 converges to a pure Nash equilibrium.*

*Proof.* In every cell, we add a dummy zone to whom a user belongs if he could not use this cell originally. The throughput received in this dummy zone is zero, and its load has no impact on the throughput of the others zones. Using this trick, we can assume that every users have the same set  $\mathcal{C}$  of cells with no loss of generality.

If the step of the algorithm is sufficiently small, the algorithm has the same asymptotic behaviour than the replicator dynamics:

$$\frac{dq_{i,a_i}}{d\theta} = q_{i,a_i}(f_{i,a_i}(Q) - \overline{f}_i(Q)), \quad (11)$$

hence it suffices to show that this dynamics converge to pure strategies.

First, we consider the case of two mobiles following this dynamics. We are going to show that the projection of the trajectory of this dynamics to any two cells, is the trajectory of a replicator dynamics which converges to a pure strategy.

Suppose that the dynamics (11) converge to the stable equilibrium  $Q^* = (q_1^*, q_2^*)$  which maximizes the function  $F(Q) = \sum_{c \in \mathcal{C}} q_{1,c} f_{1,c}(Q) + \sum_{c \in \mathcal{C}} q_{2,c} f_{2,c}(Q)$ .

Let  $c \in \mathcal{C}$  and  $d \in \mathcal{C}$ . We now consider the dynamics restricted to these two cells when strategies on the others cells are fixed to their value in matrix  $Q^*$  (this is equivalent, in multihoming, to fixing the part of packets sent by the two cells). Let us denote by  $\alpha_i$  the fraction of throughput sent by all cells but  $c$  and  $d$ :  $\alpha_i = 1 - q_{i,c}^* - q_{i,d}^*$ . If  $(1 - \alpha_i) = 0$ , then  $q_{1,c}^* = q_{1,d}^* = 0$ . In this case, we cannot have  $q_{2,c}^* q_{2,d}^* \neq 0$ . Indeed, in this case,  $f_{2,c}(Q^*) = r_{2,c}(2)$  and  $f_{2,d}(Q^*) = r_{2,d}(2)$  ( $r_{(c,i)}(i)$  denotes the reward on cell  $c$  on the zone of user  $i$ , when only user  $i$  is on the cell), so the function  $F$  is maximized if

user 2 sends the throughput where he has the best reward, i.e. to cell  $c$  if  $r_{2,c}(2) > r_{2,d}(2)$  (we suppose that  $r_{2,c}(2) \neq r_{2,d}(2)$ , i.e. the throughput on the cell  $c$  and on the cell  $d$  are not equal, what is checked in practice). For now, we consider that  $\alpha_1$  and  $\alpha_2$  are not equal to 1.

We denote by  $\bar{Q} = (\bar{q}_{i,c}, \bar{q}_{i,d})_{i \in \mathcal{I}}$  the vector of strategies of the restricted dynamics. Since  $\bar{q}_{i,d} = 1 - \bar{q}_{i,c}$ , we see that we can write the restricted dynamics as follow:

$$\frac{d\bar{q}_{i,c}}{d\theta} = \bar{q}_{i,c}(1 - \bar{q}_{i,c})(f_{i,c}(\bar{Q}) - f_{i,d}(\bar{Q})), \quad (12)$$

where  $f_{i,c}(\bar{Q}) = r_{(i,c)}(i)((1 - \bar{q}_{j,c})(1 - \alpha_j) + \alpha_j) + r_{(i,c)}(i,j)(\bar{q}_{j,c}(1 - \alpha_j))$ , and  $r_{(i,c)}(i)$  denotes the reward on cell  $c$  on the zone of user  $i$ , when only user  $i$  is on the cell, and  $r_{(i,c)}(i,j)$  when user  $j$  is also on the cell.

We denote by  $\bar{Q}^*$  the stable equilibrium reached by the restricted dynamics. We now show that  $\bar{Q}^*$  is a vector of pure strategies. Suppose it is not: at least the strategy of one user is mixed. Next, we will find a contradiction with the fact that the linearization of the dynamics around a mixed equilibrium is necessarily unstable. In fact, if the Jacobian matrix (see just after) admits an eigenvalue with a strictly positive real part then the equilibrium is unstable, hence the dynamics will not converge to this point.

So, let us linearize the dynamics (12) around  $\bar{Q}^*$  by computing the Jacobian matrix at this point,  $M = (M_{i,j})_{(i,j) \in \mathcal{I}^2}$ , which is defined by:

$$M_{i,j} = \frac{\partial}{\partial \bar{q}_{j,d}} \frac{d\bar{q}_{i,c}}{d\theta}(\bar{Q}^*).$$

We want to show that this matrix has an eigenvalue with a strictly positive real part.

First, one can notice that if  $\bar{q}_{1,c}^*$  is a pure strategy (assume  $q_{1,c} = 1$ ), then  $\bar{q}_{2,c}^*$  is also a pure strategy. Indeed,  $f_{2,c}(\bar{Q}^*) = r_{2,c}(2)$  and  $f_{2,d}(\bar{Q}^*) = -r_{2,d}(1)$ , and since the two are not equal in general, by the conditions of stationarity (8) we infer that  $\bar{q}_{2,c}^* = 0$  or 1. Thus, without loss of generality, we can consider the case where both users have a mixed strategy.

According to the conditions of stationarity (8),  $\forall i \in \{1, 2\}$ ,  $f_{i,c}(\bar{Q}^*) = f_{i,d}(\bar{Q}^*)$ , then  $\forall i \in \{1, 2\}$ ,  $M_{i,i} = 0$ .

Now, we show that symmetric terms of the matrix have the same sign ( $M_{1,2}M_{2,1} \geq 0$ ):

$$M_{1,2} = \bar{q}_{1,c}^*(1 - \bar{q}_{1,c}^*)(1 - \alpha_2)(r_{(1,c)}(1, 2) - r_{(1,c)}(1) + r_{(1,d)}(1) - r_{(1,d)}(1, 2))$$

and

$$M_{2,1} = \bar{q}_{2,c}^*(1 - \bar{q}_{2,c}^*)(1 - \alpha_1)(r_{(2,c)}(1, 2) - r_{(2,c)}(2) + r_{(2,d)}(2) - r_{(2,d)}(1, 2)).$$

Since  $\bar{q}_{1,c}^*(1 - \bar{q}_{1,c}^*)$ ,  $\bar{q}_{2,c}^*(1 - \bar{q}_{2,c}^*)$  and  $(1 - \alpha_1)$ ,  $(1 - \alpha_2)$  are positive, it suffices to prove that:

$$\begin{aligned} r_{(1,c)}(1, 2) - r_{(1,c)}(1) + r_{(1,d)}(1) - r_{(1,d)}(1, 2) \\ = r_{(2,c)}(1, 2) - r_{(2,c)}(2) + r_{(2,d)}(2) - r_{(2,d)}(1, 2). \end{aligned}$$

In fact, by the formula of the reward function:

$$\begin{aligned} r_{(1,c)}(1, 2) - r_{(1,c)}(1) + r_{(1,d)}(1) - r_{(1,d)}(1, 2) \\ = t_{(1,c)}(1, 2) - (t_{(2,c)}(2) - t_{(2,c)}(1, 2)) - t_{(1,c)}(1) \\ \quad + t_{(1,d)}(1) - t_{(1,d)}(1, 2) + (t_{(2,d)}(2) - t_{(2,d)}(1, 2)) \\ = t_{(2,c)}(1, 2) - (t_{(1,c)}(1) - t_{(1,c)}(1, 2)) - t_{(2,c)}(2) \\ \quad + t_{(2,d)}(2) - t_{(2,d)}(1, 2) + (t_{(1,d)}(1) - t_{(1,d)}(1, 2)) \\ = r_{(2,c)}(1, 2) - r_{(2,c)}(2) + r_{(2,d)}(2) - r_{(2,d)}(1, 2). \end{aligned}$$

Therefore, the Jacobian matrix is:

$$\begin{pmatrix} 0 & M_{1,2} \\ M_{2,1} & 0 \end{pmatrix}.$$

And the positive eigenvalue is  $\sqrt{M_{1,2}M_{2,1}}$ . This is zero if  $r_{(1,c)}(1, 2) - r_{(1,c)}(1) + r_{(1,d)}(1) - r_{(1,d)}(1, 2) = 0$  which means that the reward in cells  $c$  and  $d$  are identical for user 1, which is not the case in practice (it could happen if this user cannot connect to these cells, hence he would have a reward equal to zero, but in this case, we would have  $\alpha_1 = 1$  in order to maximize the function  $F$ ). Finally, the Jacobian matrix admits an eigenvalue with a strictly positive real part. And, as this matrix has an eigenvalue with a strictly positive real part, the equilibrium is unstable for the dynamics, which means that our algorithm will not converge to such a state.

Since  $Q^*$  is a stable equilibrium, necessarily  $(\frac{1}{1 - \alpha_i}(q_{i,c}^*, q_{i,d}^*))_{i \in \{1,2\}}$  are stable strategies for the restricted dynamics. Hence, for user  $i \in \{1,2\}$ ,  $q_{i,c}^* = 0$  or  $q_{i,d}^* = 0$ .

Since for every choice of two cells among the  $|\mathcal{C}|$  in the restriction of the dynamics (11), at least one component of each user's strategy is equal to zero, we conclude that only one component is not zero, hence it corresponds to a pure strategy.

Now, let us consider the case where more than two users are involved in the dynamics (11). Suppose that the dynamics converge to the state  $Q^*$ . Also suppose that the strategy of a user (say user 1) is not pure. We are going to show that by fixing all users to their strategy in  $Q^*$ , except two of them (1 and another), the dynamics followed by the two is similar to the preceding one (involving only two users), and then converge to pure strategies which will be a contradiction.

The general idea of the proof is to see the impact of other users as a fix amount of reward taken away from all cells.

In order to do so, suppose we fix the strategy of all the users to  $Q^*$ , except for users 1 and 2. We denote by  $X$  the submatrix of  $Q^*$  where lines 1 and 2 are removed. Let us consider a new system for users 1 and 2 only whose dynamics is given by ( $i = 1$  or  $2$ ),

$$\frac{dq_{i,a_i}}{d\theta} = q_{i,a_i}[\phi_{i,a_i}(q_1, q_2) - \bar{\phi}_i(q_1, q_2)], \quad (13)$$

where  $\phi_{i,a_i}(q_1, q_2) =$

$$\sum_{\ell \in L} \left( \gamma(\ell) G(t_{a_i}(\ell)) - \sum_{j \neq i} (\gamma(\ell) q_{j,a_i} (G(t_{a_j}(\ell + e_{a_i})) - G(t_{a_j}(\ell))) \right),$$

and

$$\bar{\phi}_i(q_1, q_2) = \sum_{a'_i \in \mathcal{A}_i} q_{i,a'_i} \phi_{i,a'_i}(q_1, q_2).$$

where  $\gamma(\ell)$  is the probability that the load on cell  $a_i$  is  $\ell$ , taking into account that the load on cell  $a_i$  coming from users  $3, \dots, I$  is  $x_{3,a_i} + \dots + x_{I,a_i}$ . (for example this probability is 0 if  $\ell < x_{3,a_i} + \dots + x_{I,a_i}$ ).

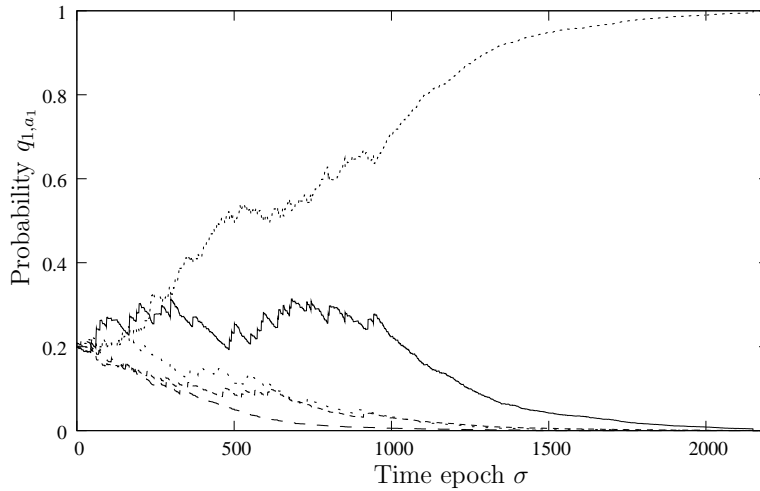
Since  $\phi_{i,a_i}(q_1, q_2) = \mathbb{E}_{q_1, q_2}[R_i | A_i = a_i]$ , then using Theorem 1, We claim that the limit dynamics of the original system is the same as this new one for users 1 and 2.

Using the new rewards, we can apply the same method as for two users (users 1 and 2): let us look at cells  $c$  and  $d$ . We define as before  $\alpha_1$  and  $\alpha_2$ . If  $\alpha_2 = 1$ , we can't say anything about the strategy of 1, because we cannot discard the case where  $\phi_{1,c}(q_1, 0) = \phi_{1,d}(q_1, 0)$  since the modified reward values live in a continuous set, depending on  $X$  (this was physically impossible with the unmodified rewards). To solve this case, it suffices to notice that the equality  $\phi_{1,c}(q_1, 0) = \phi_{1,d}(q_1, 0)$  can only hold if another user (say 3) has a non zero component on network  $c$  or  $d$ . In this case we rather select users 1 and 3 and the problem disappears since  $\alpha_3 \neq 0$ . Finally, the strategy of user 1 must be pure.  $\square$

Consider a typical run of algorithm 1 over a system made of 10 users with 5 choices over 10 networks. The figure displays for one user, the probabilities of choosing each of the 5 possible choices  $a_i$ . As the user has  $|\mathcal{A}_i| = 5$  possible choices, at time epoch 0, each choice has probability 0.2. Then, at  $\sigma$  grows, all the probabilities tend to 0 except 1 which corresponds to the optimal action at the pure Nash Equilibrium.

From Theorem 2 and Proposition 2, we can finally conclude:

**Theorem 3.** *The algorithm 1 with rewards 4 converges to a pure allocation which corresponds to the  $\alpha$ -fair solution of the system.*



**Figure 2:** Convergence of the probability values for each of the 5 possible choices of one user.

## 5 Implementation and Validation

This section is devoted to implementation issues and shows the numerical tests that were performed so as to study several possible practical heuristics based on the algorithm.

First, notice that in Algorithm 1, users only need to know the reward on her current cell to compute her step size and new strategy vector. Also, each base station only needs her own load to compute the rewards, hence allowing for a fully distributed algorithm.

During the execution of the algorithm, each user  $i$  stays connected to all possible cells in  $\mathcal{A}_i$  to avoid costly handovers. At each time slot (typically, frames are sent every 40 ms for video transmission), each user executes Algorithm 1 independently, updates her probability vector, makes a choice (*i.e.* a couple (cell, zone) according to this vector) and sends a packet to the corresponding base station.

Meanwhile, each base station measures the throughputs in all its zones and computes the corresponding rewards. Then, it sends to every user their individual reward.

Once user  $i$  reaches a pure strategy, she informs the cells and disconnects from all of them but the chosen one.

Each cell waits for all users connected to her to converge before asking them to monitor their reward. From then on, any variation of the load is due to an arrival or departure in the cell. Hence, upon detection of a change of her reward, each user resets the algorithm, with a new probability vector.

In the previous theoretical sections, convergence of the algorithm have been shown when the step size  $b$  tends to 0. Here, we present several simple heuristics with different step size computation methods. While the convergence step should be small enough to ensure convergence, larger values are preferable for increasing the algorithm speed. Hence, appropriate trade-offs need to be examined.

In the first subsection, we present the different heuristics (Subsection 5.1). We then present the scenario to be simulated (in terms of number of users and network topology) (Subsection 5.2). To perform the tests, realistic throughputs need to be chosen for different combinations of loads, i.e. values of  $t_l$  for each possible load  $l$ . We provide such values in Subsection 5.3. We then compare the results obtained by the different heuristics, in terms of efficiency (the quality of the solution) and convergence speed (Subsection 5.4). Finally we briefly comment in Subsection 5.5 and 5.6 on the impact of fairness on the resulting association and on the impact of an arrival or departure of a user on the system.

## 5.1 The Different Heuristics for the Steps

Each heuristic actually consists of two parts:

*A stopping test* As time increases, the probabilities of choosing each action tends either to 0 or 1. So as to speed up convergence, we consider thresholds  $\epsilon_m$  and  $\epsilon_M$  such that:

$$\forall i \in \mathcal{I}, \forall a_i \in \mathcal{A}_i, \begin{cases} q_{i,a_i}(\sigma + 1) \leftarrow 0 & \text{if } q_{i,a_i}(\sigma) < \epsilon_m \\ q_{i,a_i}(\sigma + 1) \leftarrow 1 & \text{if } q_{i,a_i}(\sigma) > 1 - \epsilon_M. \end{cases}$$

When one of this operation is done, the strategies are normalized to remain in the strategy set  $\mathcal{S}$ , and to preserve the condition  $\sum_{a_i \in \mathcal{A}_i} q_{i,a_i} = 1$ . In the tests, we fix  $\epsilon_m = 0.05$  and  $\epsilon_M = 0.3$ .

*A step size computation* : different schemes to compute  $b_i(\sigma)$  are considered.

### 5.1.1 Constant Step Size (CSS)

In this heuristic, the step size is predefined and constant throughout time:  $\forall i \in \mathcal{I}, \forall \sigma, b_i(\sigma) = b$ . For low values (CSS<sub>L</sub>), typically  $b = 0.01$ , the algorithm converges in almost all cases to the optimal solution, but at the cost of a high number of iterations. For high values (CSS<sub>H</sub>), typically  $b = 1$ , the convergence and the optimality are not guaranteed anymore. Intermediate values (CSS<sub>M</sub>), typically  $b = 0.1$ , are possible compromises.

### 5.1.2 Constant Update Size (CUS)

At each time epoch, each user computes the maximum step size so that the change of probabilities for all choices, is bounded by a predefined value  $\Delta_q$  (fixed to 0.1 in the experiments):

$$\forall i \in \mathcal{I}, \forall a_i \in \mathcal{A}_i, \quad \text{abs}(q_{i,a_i}(\sigma + 1) - q_{i,a_i}(\sigma)) \leq \Delta_q.$$

By bounding the update of every user, this scheme yields smooth changes in the strategy vectors and hence can be expected to follow the behavior of the differential equations.

### 5.1.3 Decreasing Step Size (DSS)

The underlying idea of this scheme is to use a few iterations with large steps before using some smaller step sizes. Indeed, a big step size lets actions associated to large rewards to quickly get high probabilities of occurrence. Since the algorithm converges to a Nash Equilibrium regardless of the initial conditions, using a few large steps amounts in changing the initial conditions so as to get close to extrema points, and hence to possible pure strategies Nash Equilibria. Then, the following iterations with smaller step sizes correspond to a good approximation of the  $CSSL$  algorithm. These steps confirm (or infer) the fact that the extremal point closer to the one obtained after the first iterations is (or not) a Nash Equilibrium.

We consider two variants of the decreasing step size mechanism. The first one is a cyclic decreasing step size (DSSSA) (in the experiments,  $b = 3/(\sigma \bmod 10)$ ). During each cycle a Nash equilibrium candidate is tested. This is inspired from simulated annealing approaches.

The second variant (DSSCSS) is a decreasing step size phase followed by a constant large step size (in the experiments,  $b = 4/\sigma$  if  $\sigma < 120$  and  $b = 4$  otherwise). The underlying idea is that the first phase would stabilize a certain number of users. Then, a large step size should improve the convergence speed of the others to their respective preferable choices.

## 5.2 System Scenario

We consider a simple scenario of an operator providing subscribers with a service available either through a large WiMAX cell or a series of WiFi hot spots.

For each simulation, a topology is chosen randomly, according to 3 parameters (the number of users, the number of WiFi hot spots and the number of possible choices for each user  $|\mathcal{A}_i|$ ). More precisely, for each user:

- The first choice is the WiMAX cell and one of the 8 possible zones (as defined in Section 5.3), picked at random (uniformly).



- All other  $|\mathcal{A}_i| - 1$  choices are one of the  $|\mathcal{C}| - 1$  cells, picked up according to a uniform law. As explained in Section 5.3, we do not consider zones in the WiFi cells.

The strategy vector is initialized with equal probabilities:  $\forall i \in \mathcal{I}, \forall a_i \in \mathcal{A}_i, q_{i,a_i}(0) = 1/|\mathcal{A}_i|$ .

### 5.3 Throughput of TCP sessions in WLAN and WiMAX

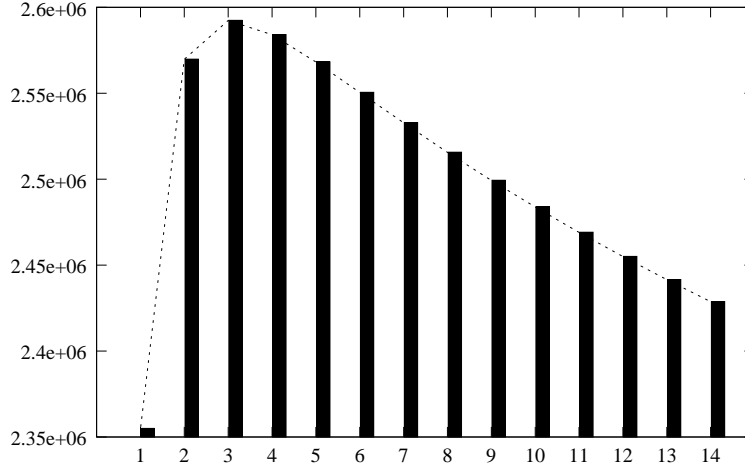
Computing the throughput experienced by a packet in a wireless environment is extremely hard due to the complexity of the physical system (as opposed to wired system, where the physical medium is separated from the outside world, and hence has reliable properties, the wireless link quality changes at every instant, due to the environment: air quality, buildings and physical obstacles, etc). Therefore, actual closed formula available in the literature were obtained using strong assumptions on the outside world and do not refer to throughput of a single packet but of means of flows. Indeed, as the number of packets in any connection is large, the flow is usually approximated as a fluid.

In addition, the useful throughput of a connection, also called *goodput* depends on the network protocol. Roughly speaking, two main elements have strong impact on the achieved goodput: first is the physical system, which depends on the technology in terms both of maximum capacity and multiplexing technology, second is the transport protocol. In this simulation study, we consider the case of TCP flows for which good throughput approximations are available in the literature. Yet, the use of UDP flows, or a mixture of TCP and UDP flows do not impact the performance of the algorithm. (Note that allowing users to use either TCP or UDP protocol for their transmission amounts, in the algorithm, to considering an additional zone in the network cell.)

**Equations of throughput in WiFi cells** Based on [18], we consider that the throughput of connection  $i$  is

$$t_i(l_c) = \frac{L_{TCP}}{l_c(T_{DATA} + T_{ACK} + 2T_{TBO}(l_c) + 2T_W(l_c))}$$

where  $L_{TCP} = 8000$  bits is the size of a TCP packet,  $T_{ACK}$  is the raw transmission times of TCP ACK (approximately 1.091 ms),  $T_{DATA}$  the raw transmission times of a TCP data packet (about 1.785 ms). Then,  $T_W$  and  $T_{TBO}$  are the mean total time lost due to collisions and backoffs respectively. These depend on the collision probability of each packet, and hence on the load of the network. This collision probability can be numerically obtained via a fixed point equation given in [18]. Figure 3 displays the throughput of a WiFi cell, as a function of the load.



**Figure 3:** Capacity of a WiFi cell as a function of its load (in bit/s). The maximum is reached with 3 users.

**WiMAX** As opposed to WiFi, the WiMAX technology uses OFDMA multiplexing. Hence, each user receives a certain number of carriers which are converted into a certain amount of throughput depending on the chosen modulation and coding scheme, which greatly depends on the link quality at the receiver side. We consider a fair sharing in terms of carriers [19], *i.e.* if  $p$  users are present in the WiMAX cell, each of them will receive  $NbSCarriers/p$  sub-carriers, similarly to processor sharing. Hence, the goodput experienced by a user in zone  $z$  (corresponding to a coding scheme) is roughly the fraction  $1/\sum_{z \in \mathcal{Z}_c} l_{c,z}$  of the throughput she would obtain if she were alone in the cell.

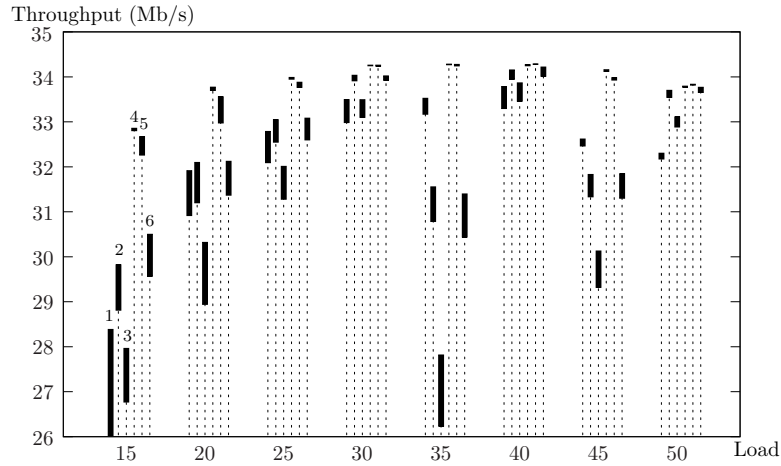
For a single user within the WiMAX cell, we follow experimental values obtained in [20] for IEEE WiMAX 802.16d for its eight zones:

Modulation	QAM64 3/4	QAM64 2/3	QAM16 3/4	QAM16 1/2
TCP goodput	9.58	8.88	6.80	4.50
Modulation	QPSK 3/4	QPSK 1/2	BPSK 3/4	BPSK 1/2
TCP goodput	3.37	2.21	1.65	1.08

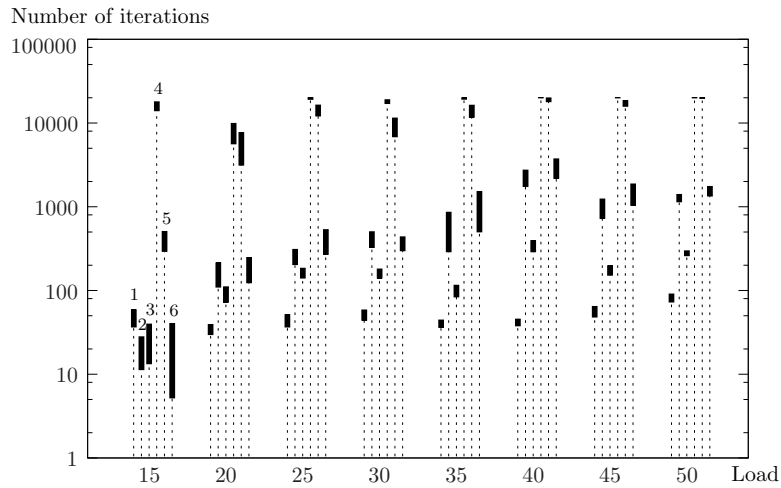
## 5.4 Comparisons between Heuristics

Figure 4 displays the performance (in terms of global throughput) obtained by the six heuristics ( $CUS$ ,  $DSSSA$ ,  $DSSCSS$ ,  $CSS_L$ ,  $CSS_M$  and  $CSS_H$  resp.) as a function of the total number of users  $|\mathcal{I}|$ . For a given load, all heuristics have been tested on the same topology to allow a fair comparison.

The small constant step size ( $CSS_L$  with  $b = 0.01$ ), provides the best performance. It is even tested optimal for the small values of  $|\mathcal{I}|$ , up to 20.



**Figure 4:** Average performance of the heuristics ( $CUS$ ,  $DSSSA$ ,  $DSSCSS$ ,  $CSS_L$ ,  $CSS_M$  and  $CSS_H$  resp.) with different loads (with 5% confidence intervals).



**Figure 5:** Average number of iterations before convergence of heuristics ( $CUS$ ,  $DSSSA$ ,  $DSSCSS$ ,  $CSS_L$ ,  $CSS_M$  and  $CSS_H$  resp. ) for different loads (with 5% confidence intervals).

Most heuristics stay within 10 % of the optimal (except for DSSCSS whose performance can be poor). Also note that the total capacity of the system is less than 36 ( $10 * 2.6$  (WiFi) + 9.58 (WiMAX)) Mbit/s. Thus the best heuristic is always within 5 % of the optimal. Finally, it should be noted that the medium constant step size ( $CSS_M$ ) with  $b = 0.1$  is always very close to the best ( $CSS_L$ ) and that the constant update size ( $CUS$ ) performs better and better when the number of users grows.

As for the number of iterations, it varies widely between the different heuristics, even on a logarithmic scale (see Figure 5). The  $CUS$  heuristic is a clear winner here (with an average number of iterations never above 80). Meanwhile,  $CSS_L$  does not always converge within the limit of 20,000 iterations set in the program.

Under high loads,  $CUS$  provides the best compromise with very fast convergence and reasonable performance. Under light load, the constant step size of medium size ( $CSS_M$ ) is also an interesting choice, for its performance is almost optimal and its number of iterations remains below 100.

## 5.5 Impact on Fairness

Consider the following scenario: a set of 20 users, each having 3 available choices among 10 cells. The WiMAX cell is numbered 0 and its 8 zones are numbered from 0 to 7. The set of choices of the users are  $\mathcal{A} =$

$$\begin{array}{lll} \{\{0, 1\}, \{8\}, \{1\}\} & \{\{0, 5\}, \{6\}, \{4\}\} & \{\{0, 1\}, \{6\}, \{9\}\} \\ \{\{0, 2\}, \{2\}, \{6\}\} & \{\{0, 3\}, \{8\}, \{9\}\} & \{\{0, 6\}, \{4\}, \{9\}\} \\ \{\{0, 7\}, \{3\}, \{6\}\} & \{\{0, 4\}, \{1\}, \{2\}\} & \{\{0, 6\}, \{6\}, \{9\}\} \\ \{\{0, 5\}, \{3\}, \{4\}\} & \{\{0, 6\}, \{3\}, \{1\}\} & \{\{0, 7\}, \{9\}, \{6\}\} \\ \{\{0, 3\}, \{8\}, \{1\}\} & \{\{0, 6\}, \{4\}, \{7\}\} & \{\{0, 6\}, \{9\}, \{5\}\} \\ \{\{0, 0\}, \{6\}, \{5\}\} & \{\{0, 5\}, \{4\}, \{1\}\} & \{\{0, 6\}, \{6\}, \{4\}\} \\ \{\{0, 3\}, \{3\}, \{4\}\} & \{\{0, 3\}, \{8\}, \{4\}\}. & \end{array}$$

The optimal association scheme, for  $\alpha = 0$  (efficient scheme) and  $\alpha = 2$  (fair schemes) are respectively:

$$\begin{aligned} A_{\text{eff}} &= \{2, 1, 2, 1, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 0, 2, 1, 1, 1\}, \\ A_{\text{fair}} &= \{0, 1, 0, 1, 0, 2, 1, 2, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 0, 1\} \end{aligned}$$

resulting in throughputs of:

$$\begin{aligned} T_{\text{eff}} &= 0.824, 1.225, 0.824, 1.225, 1.225, 1.225, 0.824, 1.225, 0.824, 1.225, \\ &\quad 0.824, 0.824, 0.824, 2.245, 2.246, 9.58, \quad 0.824, 1.225, 0.824, 1.225. \\ T_{\text{fair}} &= 2.22, \quad 1.225, 2.22, \quad 1.225, 1.125, 1.225, 1.225, 1.225, 1.225, 1.225, \\ &\quad 2.245, 1.225, 1.225, 2.246, 1.225, 1.225, 1.225, 1.225, 1.125, 1.225. \end{aligned}$$

The efficient scheme achieves a total throughput of 31.29 Mb/s. The fair scheme suffers a degradation of slightly less than 10%, with a total

throughput of 28.34 Mb/s. Yet a closer look at the figures indicates that the efficient scheme leads to high differences between users (user 1 only obtains a throughput of 0.8 Mb/s while user 16 is granted 9.58 Mb/s). As for the fair association scheme, on the other hand, all users benefit from throughputs higher 1.1 Mb/s. As in bandwidth allocation mechanisms in wired systems[11], the parameter  $\alpha$  hence allows to finely tune the compromise between maximum global throughput and fairness between users.

To understand these differences, let us compare the loads between the associations:

$$L_{\text{eff}}^{\text{wifi}} = \{3, 2, 3, 2, 1, 2, 1, 2, 3\}, \quad L_{\text{fair}}^{\text{wifi}} = \{1, 2, 2, 2, 2, 2, 1, 2, 2\}.$$

From Fig. 3, one can see that the maximum capacity for the WiFi cells is obtained for a load of 3 users. Hence, the efficient scheme tries to obtain as many cells with load 3 as possible. Meanwhile, the WiMAX capacity is maximal when its users all belong to zone 0. Hence, such users are automatically associated to this cell (in our case there is only one such user, which obtains a throughput of 9.58 Mb/s).

On the other hand, the fair scheme tries to find balanced association schemes. Hence, the loads of the different WiFi cells are close to one another<sup>4</sup> (here ranging between 1 and 2) and the WiMAX cell is associated to some users belonging to efficient zones. Their number is chosen so as to obtain similar performance as for users remaining in the WiFi cells.

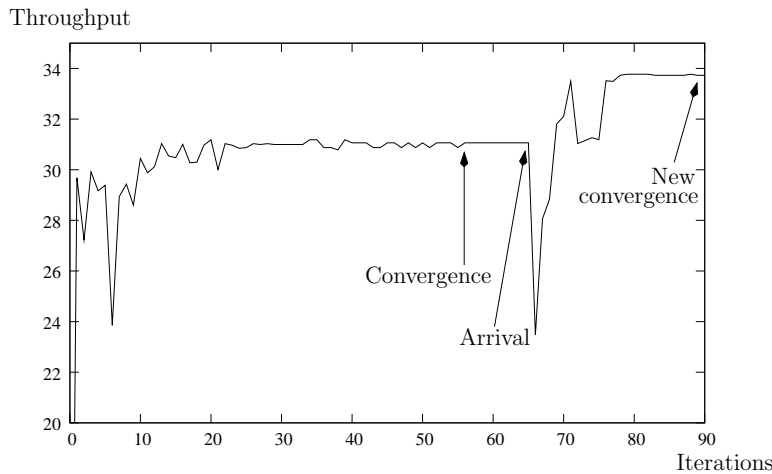
Hence, while purely efficient schemes produce lightly loaded WiMAX cells (with only the users in zone 0), the fair scheme leads to more balanced loads (here, 4 users in the WiMAX cell and about 2 users in each WiFi cell).

## 5.6 Seamless Adaptation to User Arrivals and Departures

As mentioned at the beginning of Section 5, the association algorithm has to be run at every arrival or departure of a user in a cell. In this section, we discuss the impact of such events. First, typical time scales compare nicely: while arrivals or departures of users in WiMAX or WiFi cells occur every minute or so, the association algorithm converges in less than a second in most cases, when the best heuristic (CUS) for the step sizes is used.

Second, the convergence of the algorithm is much faster when the initial state is chosen close to the former optimal solution for all old users and one change occurs (an arrival of a departure). This is illustrated in Figure 6. In the first phase, 30 users with 3 choices over 10 cells run the algorithm using CUS step updates. Starting with uniform probabilities (1/3, 1/3, 1/3), convergence occurs after 60 iterations. Upon the arrival of a new user, the algorithm is run again but the starting points is (1/3, 1/3, 1/3) for the newcomer and (1/4, 1/4, 1/2) for all other users, with probability 1/2 given to

<sup>4</sup>Note that they cannot be strictly equal due to the discrete nature of the problem.



**Figure 6:** The convergence of the CUS heuristic starting with an equi-probability distribution takes 60 iterations. When one new user becomes active, starting with probabilities close to the former optimal point, convergence takes 27 iterations.

the previously chosen cell. Here, the algorithm converges after 27 iterations only.

## 6 Conclusion and Future Works

In this paper, we have designed a distributed algorithm that converges to an optimal (in terms of fairness or efficiency) network association in heterogeneous wireless networks. This opens the way to several interesting future works, such as the implementation of such methods in modern mobile devices in collaboration with Alcatel-Lucent.

## 7 Acknowledgement

We wish to thank Laurent Thomas, Sabine Randriamasy and Erick Bizouarn from Alcatel-Lucent, who helped us with the design of several scenarios using realistic parameters.

## References

- [1] D. Barth, O. Bournez, O. Boussaton, and J. Cohen, “A dynamic approach for load balancing,” LORIA Research Report, Tech. Rep., 2008, <http://www.loria.fr/bournez/load/Soumis-Octave-Fev-2008.pdf>.

- 
- [2] P. Sastry, V. Phansalkar, and A. Thathachar, “Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information,” *IEEE Transactions on System, man, and cybernetics*, vol. 24, no. 5, pp. 769–777, 1994.
  - [3] D. Kumar, E. Altman, and J.-M. Kelif, “User-network association in a WLAN-UMTS hybrid cell: Global & individual optimality,” *Research Report, INRIA, Sophia Antipolis, France*, 2006.
  - [4] M. Coupechoux, J.-M. Kelif, and P. Godlewski, “Network controlled joint radio resource management for heterogeneous networks,” in *Proc. of IEEE VTC Spring*, 2008.
  - [5] S. Shakkottai, E. Altman, and A. Kumar, “Multihoming of users to access points in wlans: A population game perspective,” *IEEE Journal on Selected Areas in Communication, Special Issue on Non-Cooperative Behavior in Networking*, vol. 25, no. 6, pp. 1207–1215, 2007.
  - [6] J. W. Weibull, *Evolutionary Game Theory*. MIT Press, 1995.
  - [7] J. Hofbauer and K. Sigmund, “Evolutionary game dynamics,” *Bull. Amer. Math. Soc.*, vol. 40, pp. 479–519, 2003.
  - [8] E. Koutsoupias and C. Papadimitriou, “Worst-case equilibria,” in *Proc. of STACS*, 1998.
  - [9] A. Legrand and C. Touati, “Non-cooperative scheduling of multiple bag-of-task applications,” in *Proc. of the 25th Conference on Computer Communications (INFOCOM’07)*, Alaska, USA, May 2007.
  - [10] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” in *Proc. of SPIE, International Symposium on Voice, Video and Data Communications*, 1998.
  - [11] C. Touati, E. Altman, and J. Galtier, “Generalised Nash bargaining solution for bandwidth allocation,” *Computer Networks*, vol. 50, no. 17, pp. 3242–3263, Dec. 2006.
  - [12] S. Shakkottai, E. Altman, and A. Kumar, “The case for non-cooperative multihoming of users to access points in IEEE 802.11 WLANs,” in *Proc. of INFOCOM*, 2005.
  - [13] J. F. Nash, “Equilibrium points in n-person games,” *Proceeding of the National Academy of Sciences*, vol. 36, pp. 48–49, 1950.
  - [14] R. Cole, Y. Dodis, and T. Roughgarden, “Pricing network edges for heterogeneous selfish users,” in *Proc. of STOC*, 2003.

- 
- [15] H. J. Kushner, *Approximation and Weak Convergence Methods for Random Processes with Applications to Stochastic Systems Theory*. Cambridge, MIT Press, 1984.
  - [16] D. Monderer and L. S. Shapley, “Potential games,” *Games and Economic Behavior*, vol. 14, pp. 124–143, 1996.
  - [17] W. H. Sandholm, “Potential games with continuous player sets,” *Journal of Economic Theory*, vol. 24, pp. 81–108, 2001.
  - [18] D. Miorandi, A. A. Kherani, and E. Altman, “A queueing model for HTTP traffic over IEEE 802.11 WLANs,” *IEEE Computer Networks*, vol. 50, pp. 63–79, 2006.
  - [19] C. Tarhini and T. Chahed, “System capacity in ofdma-based wimax,” in *Proc. of the International Conference on Systems and Networks Communications*, 2006.
  - [20] F. Yousaf, K. Daniel, and C. Wietfeld, “Performance evaluation of iee 802.16 wimax link with respect to higher layer protocols,” in *Proc. of the International Symposium on Wireless Xommunication Systems*, 2007, pp. 180–184.





---

Centre de recherche INRIA Grenoble – Rhône-Alpes  
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex  
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq  
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex  
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex  
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex  
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex  
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399