# Self-Organizing Relays in LTE networks: Queuing analysis and algorithms

Richard Combes*,Zwi Altman* and Eitan Altman†
*France Telecom Research and Development
38/40 rue du Général Leclerc,92794 Issy-les-Moulineaux
Email:{richard.combes,zwi.altman}@orange-ftgroup.com
†INRIA Sophia Antipolis
06902 Sophia Antipolis, France
Email:Eitan.Altman@sophia.inria.fr

***Abstract*—Relay stations are an important component of heterogeneous networks (HetNets) introduced in the LTE-Advanced technology as a means to provide very high capacity and QoS all over the cell area. This paper develops a self-organizing network (SON) feature to optimally allocate resources between backhaul and station to mobile links. Static and dynamic resource sharing mechanisms are investigated. In the static case we provide a queuing model to calculate the optimal resource sharing strategy and the maximal capacity of the network analytically. The influence of relay planning and number of deployed relays is investigated, and the gains resulting from good planning are evaluated analytically. Self-optimizing dynamic resource allocation is tackled using a Markov Decision Process (MDP) model. Both stability in the infinite buffer case and blocking rate and file transfer time in the finite buffer case are considered. To achieve a scalable solution with a large number of relays, a well-chosen parametrized family of policies is considered, to be used as expert knowledge. Finally, a model-free approach is shown in which the network can derive the optimal parametrized policy, and the convergence to a local optimum is proven.** [1]

***Index Terms*—Relay, Queuing Theory, Stability, OFDMA, Load Balancing, Self configuration, Self Optimization**

## I. Introduction

Self-organizing networks (SON) mechanisms have been introduced in the Long Term Evolution (LTE) standard in order to empower the network by embedding autonomic mechanisms, namely self-configuration, self-optimization and self-healing ([1], [2]). These mechanisms aim at simplifying the network management, at reducing its cost of operation and at increasing its performance. Within release 10 of 3GPP, enhancement of SON features have been introduced into the LTE-Advanced technology, such as the enhancement of mobility robustness and load balancing self-optimization.

Dynamic self-optimization targets on-line network implementation of SON mechanisms with short time resolution (e.g. seconds to minutes) for adapting the network to new operation conditions such as traffic variations. The requirements for SON solutions to be adopted in radio access networks are the classical goodness criteria in optimization and control: existence of optimal solutions, convergence to an optimal solution, speed of convergence, monotonic improvement of the goodness of the solution, stability and robustness to noise. Previous work on on-line network optimization include the popular utility-based approach used in [3], [4] and [5]. Reinforcement learning has been investigated for example in [6].

LTE-Advanced introduces the concept of Heterogeneous Network (HetNet) as a mean to increase network capacity. HetNets comprise low power nodes deployed in high traffic areas to increase capacity, namely picocells, femtocells and Relay Stations (RSs). Autonomous resource management in HetNets is among the important and challenging research avenues in SON for next generation radio access networks, encompassing load balancing, Inter-Cell Interference Coordination (ICIC), mobility management, and other self-optimizing resource allocation mechanisms.

This paper focuses on self-optimizing RSs. RSs are linked to the macrocell by a wireless link which replaces the wired backhaul. We will use the term "station" to refer to a Base Station (BS) or a RS indifferently. Radio resources have to be shared between the BS to RSs links and the stations to users links. The resource allocation which maximizes the system capacity depends on system parameters such as traffic and RSs placement. Both static and dynamic mechanisms are investigated in this work.

We first derive the static resource allocation which maximizes the system capacity. We then show a dynamic resource allocation as an optimal control problem. We give a systematic method for the controller design, in three steps:

1) The problem is modelled as a Markov Decision Process (MDP), and the optimal controller is found. This optimal controller is to be used as expert knowledge during the next phase.
2) Based on the previous controller and a queuing theory result, we introduce a set of parametrized policies (the expert knowledge). A method to find the optimal parametrized controller is derived and its performance is compared with the optimal controller.
3) Finally, we show a model-free (reinforcement learning) approach to derive the optimal parametrized policy by observation and interaction with the network. We use the policy-gradient method featured in [7], [8], [9].

The contributions of the present paper are:

[1]This work has been partially carried out in the framework of the FP7 UniverSelf project under EC Grant agreement 257513

1) A queuing analysis to derive the optimal static resource allocation in closed form, and the impact of the major system parameters such as RS placement, number of deployed RS and RS size on the system performance.
2) A systematic step-by-step framework for controller design, with rigorous proofs of convergence and optimality of the methods used.
3) A model-free approach with monotonic improvement of the solution during the learning phase. This is fundamental for on-line implementation in an operational network.

The paper is organized as follows: Section II states the system model and the optimal static resource allocation strategy is derived in closed form based on a queuing analysis. The impact of RS placement, number of deployed RS and RS size is analysed. Section III models the problem as a MDP, and a parametrized set of policies is derived based on the optimal policy. Section IV presents a model-free approach to derive the optimal parametrized policy by interaction with the network, without degradation during the learning phase. Section V concludes the paper.

## II. Optimal static resource allocation

### A. System model

We consider the downlink scenario of a wireless network where users arrive randomly according to a spatial Poisson process of intensity $\lambda$, to receive a file of random size $\sigma$, with $\mathbb{E}[\sigma] < +\infty$. We assume independence between the arrival process and file sizes. We assume that there is no user mobility and that users leave the network upon service completion. We denote by $\mathbb{A} \subset \mathbb{R}^2$ the network area which we assume to be bounded. $\mathbb{A}$ contains a BS (alternatively called macro-cell) and several RSs. We denote by $N_R$ the number of RSs, and we use the convention that station $0$ is the BS and station $s$ , $1 \leq s \leq N_R$ is the $s$-th RS. Let $\mathbb{A}_s \subset \mathbb{A}$ denote the area covered by station $s$, $A_s = \int_{\mathbb{A}_s} dr$ its size and $A = \sum_{s=0}^{N_R} A_s$ the network size. As mentioned earlier, RSs have no direct link to the backhaul, and are connected to the BS by a wireless link. This wireless link uses the same radio resources as the station to users links and we are interested in finding an appropriate resource sharing method. This mechanism is often called in-band relaying. Depending on the multi-access radio technology, the radio resources can refer to codes in Code Division Multiple Access (CDMA), to time slots in Time Division Multiple Access (TDMA) or to time-frequency blocks in Orthogonal Frequency-Division Multiple Access (OFDMA). We ignore the granularity of resources and we denote by $x \in [0,1]$ the proportion of resources allocated to the link between the BS and RSs. We further assume that Round Robin (RR) scheduling applies in all links: the link between the BS and RSs is shared in a Processor Sharing (PS) way among the RSs, and that each link between a station and the users it serves is shared in a PS way among those users.

### B. System capacity

For a given $x \in [0,1]$ we now calculate the capacity of the system, and the optimal resource sharing strategy $x^*$ which ensures stability whenever it is possible. Namely, we denote by $C$ the capacity of the system defined as the maximal value of $\lambda\mathbb{E}[\sigma]$ that keeps the system stable i.e the number of users in the system does not grow to infinity. We write $R_{rel,s}$ , $1 \leq s \leq N_R$ the data rate of the link between BS and RS $s$ when it is the only active link, and $R_s(r)$ , $r \in \mathbb{A}_s$ the data rate between station $s$ and a user located at $r$ when he is alone in the system. The effect of inter-cell interference is incorporated in $R_{rel,s}$ and $R_s(r)$, hence the results given here hold regardless of the amount of inter-cell interference.

**Theorem 1.** *The capacity $C$ of the system is:*

$$C(x) = \min\left(C_{rel}(x), \min_{0\leq s\leq N_R}(C_s(x))\right) \tag{1}$$

*with:*

$$C_{rel}(x) = x\left(\sum_{s=1}^{N_R}\frac{A_s}{R_{rel,s}}\right)^{-1} \tag{2}$$

$$C_s(x) = (1-x)\left(\int_{\mathbb{A}_s}\frac{1}{R_s(r)}dr\right)^{-1} \tag{3}$$

*Furthermore, there exists a unique $x^* \in [0,1]$ which maximizes the capacity, with $C^*$ the corresponding maximal capacity:*

$$x^* = \frac{\left(\max\limits_{0\leq s\leq N_R}\int_{\mathbb{A}_s}\frac{1}{R_s(r)}dr\right)^{-1}}{\left(\max\limits_{0\leq s\leq N_R}\int_{\mathbb{A}_s}\frac{1}{R_s(r)}dr\right)^{-1} + \left(\sum_{s=1}^{N_R}\frac{A_s}{R_{rel,s}}\right)^{-1}} \tag{4}$$

$$C^* = \frac{\left(\sum_{s=1}^{N_R}\frac{A_s}{R_{rel,s}}\right)^{-1}\left(\max\limits_{0\leq s\leq N_R}\int_{\mathbb{A}_s}\frac{1}{R_s(r)}dr\right)^{-1}}{\left(\max\limits_{0\leq s\leq N_R}\int_{\mathbb{A}_s}\frac{1}{R_s(r)}dr\right)^{-1} + \left(\sum_{s=1}^{N_R}\frac{A_s}{R_{rel,s}}\right)^{-1}} \tag{5}$$

*Proof:* See appendix. ∎

It is noted that this result applies regardless of the underlying packet dynamics. More precisely, consider two scenarios:

1) Small files: When a user served by a RS arrives in the network, the file he wants to receive enters the BS to RSs link and once the whole file has gone through that link, it enters the corresponding RS to user link and is transmitted. This model is reasonable for small files.
2) Larger Files: In a more realistic setting, when a user served by a RS arrives in the network, the file he wants to receive arrives as small packets which enter the BS to RSs link, possibly with delays between packets. Once a packet has gone through the BS to RSs link it immediately enters the RS to user link. Here the file can be "split" between the two successive links.

In both scenarios the input process is stationary ergodic, and the value of $\lambda\mathbb{E}^0_A[\sigma_0]$ is the same. Namely $\lambda$ and $\mathbb{E}^0_A[\sigma_0]$ are different but their product remains the same. Hence the system capacity does not depend on the scenario chosen.

### C. Relay gain

We now introduce the concept of RS placement gain, and give a method to evaluate the resulting capacity improvement. We assume that the signal attenuation per distance unit is smaller for the useful signal between the BS and RSs than for interfering signals. This can be achieved by placing RSs high enough so that the propagation between the BS and RSs is close to the line-of-sight case, while taking advantage of buildings to increase the attenuation of interfering signals. Assume that the propagation loss at distance $\|r\|$ is $\frac{A}{\|r\|^{\eta_r}}$ with $2 \leq \eta_r \leq \eta$ for the useful signal between the BS and RSs, and $\frac{A}{\|r\|^{\eta}}$ for all other signals. The case $\eta_r = 2$ corresponds to line-of-sight propagation between BS and RSs. We call $\eta - \eta_r$ the relaying gain, and $\eta_r = 2$ gives gives an upper bound on the achievable capacity by intelligent relay placement.

### D. Numerical experiments

We now evaluate the influence of the system parameters on the performance using a classical model. The model parameters are given in Table I, and Figure 1 represents the network layout. Interference from neighbouring cells is taken into account. We now state the ergodic throughput $R_s(r)$ calculation method in the OFDMA case. Assuming that the fast-fading is a multiplicative random variable of mean 1, we have that:

$$R_s(r) = N_{RB} \int_{\mathbb{R}^+} \phi(SINR_s(r)x)p(x)dx \qquad (6)$$

with $N_{RB}$ the number of resource blocks, $\phi$ - a link-level curve mapping instantaneous Signal to Interference plus Noise Ratio (SINR) into data rate on a resource block, $SINR_s(r)$ - the mean SINR at $r \in \mathbb{A}_s$ and $p(x)$ the probability density function (p.d.f) of the fast-fading. In the Rayleigh case, $p(x) = e^{-x}$. Similar models apply in the TDMA and CDMA case (see for example [10], [11]). It is noted that we choose a large cell radius since [12] had shown that relays are only beneficial in such a setting.

| Model parameters | |
|---|---|
| Cell layout | Hexagonal |
| Antenna type | Omnidirectional |
| Cell Radius | $2km$ |
| Access technology | OFDMA |
| Fast-fading model | Rayleigh |
| $N_{RB}$ | 10 |
| Resource block size | $180kHz$ |
| BS transmit power | $46dBm$ |
| RS maximum transmit power | $30dBm$ |
| Thermal noise | $-174dBm/Hz$ |
| Path loss model | $128 + 37.6\log_{10}(d)$ dB, $d$ in km |
| File size | $10Mbytes$ |

TABLE I
MODEL PARAMETERS

Figure 2 and 3 show the capacity of the system and the optimal relay transmit power respectively as the number of relays grows, with and without relaying gain. The case without relaying gain is denoted "bad planning" and with relaying gain

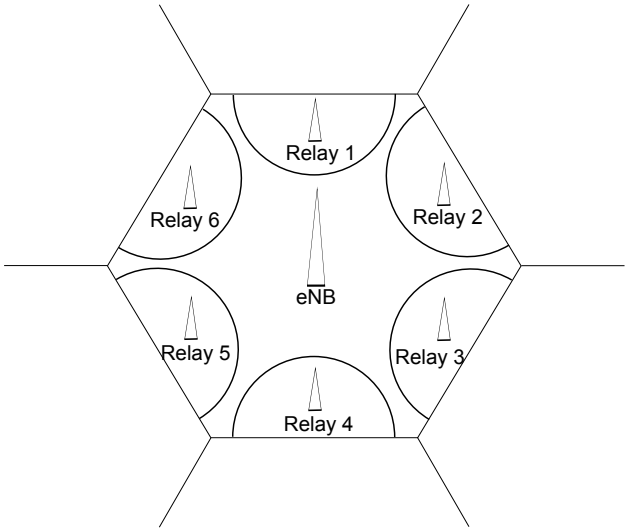


Fig. 1. Relay placement

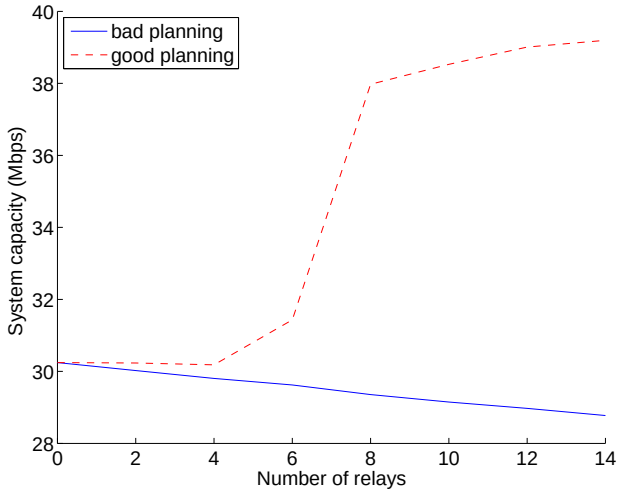


Fig. 2. System capacity as a function of the number of relays, for different planning strategies

"good planning". It is noted that the value of the optimal relay transmit power in the "bad planning" case is $0mW$ for all number of relays (below the x-axis). It demonstrates that the impact of relaying gain is fundamental since without relaying gain it is actually detrimental to deploy relays. With relaying gain however, the system capacity increases sharply.

Figure 4 shows the impact of the relaying gain on the system capacity for a fixed number of relays (15 in this case), and we can see that the capacity increases almost linearly in the relaying gain. This can be explained by the fact that $\log_2(1 + S\|r\|^{\eta-\eta_r})$ is close to $\log_2(S) - (\eta - \eta_r)\log_2(\|r\|)$ when $S\|r\|^{\eta-\eta_r}$ is large. It shows that if one is able to evaluate the relaying gain prior to deployment (by measuring the value of the path loss exponent in candidate sites for relay placement), one can actually determine if relay deployment is beneficial and the expected benefit. Furthermore the point where the two curves intersect represents the minimal relaying

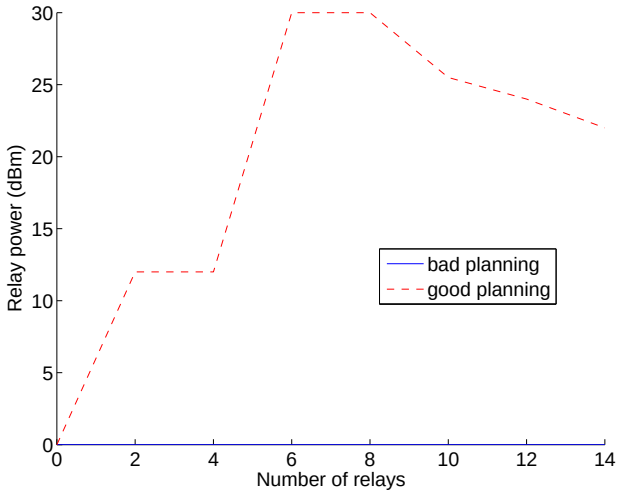


Fig. 3. Optimal relay transmit power as a function of the number of relays, for different planning strategies

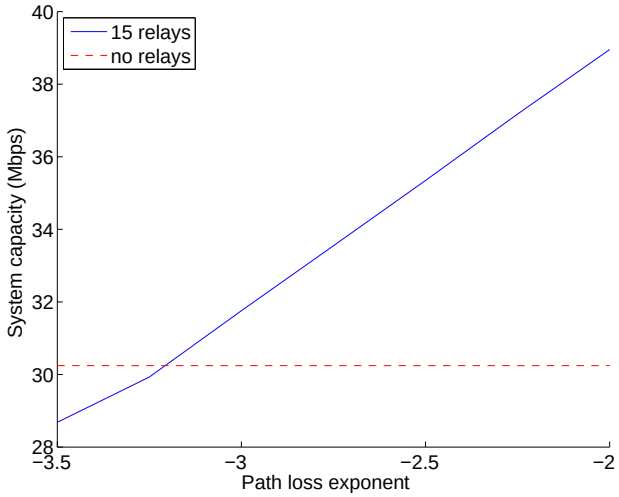


Fig. 4. Impact of the relaying gain on the system capacity

gain needed for any benefit from relay deployment to appear.

## III. Optimal dynamic resource allocation strategy

We now turn to the dynamic case. The BS observes the current state of the network and decides whether to activate the BS to RSs links or the stations to users links.

### A. Infinite buffer case: stabilizing policy

We partition each $\mathbb{A}_s$ into $N$ regions $\mathbb{A}_{s,i}$ , $1 \leq i \leq N$, each associated with a different radio condition. We call $i$-th traffic class in station $s$ the users who arrive in $\mathbb{A}_{s,i}$. The state of the system can then be described by a vector $\mathbf{S} \in \mathbb{N}^{(2N_R+1)N}$, $\mathbf{S} = ((S_{s,i})_{0\leq s\leq N_R, 1\leq i\leq N}, (S_{rel,s,i})_{1\leq s\leq N_R, 1\leq i\leq N})$. In the small files framework we count the number of users present in the links, otherwise we count the number of packets. Hence $S_{s,i}$ is the number of users (packets respectively) of class $i$ served by the station to user link in station $s$ , and $S_{rel,s,i}$ , $s \geq 1$ the number of users (packets respectively) of class $i$ served by the BS to RS $s$ link. We write $R_{s,i}$ the data rate of a user of class $i$ served by station $s$.

We first assume infinite buffer lengths and we want to find the policy that keeps the system stable whenever that is possible. The problem is in fact a particular case of the constrained queuing systems considered by [13]. It has been proven that such a policy exists and that it is a max-weight policy. We define the weights:

$$D_s = \max_{1\leq i\leq N}(S_{s,i}R_{s,i}) \, , \, 0 \leq s \leq N_R \quad (7)$$

$$D_{s,rel} = \max_{1\leq i\leq N}((S_{rel,s,i} - S_{s,i})R_{rel,s}) \, , \, 1 \leq s \leq N_R \quad (8)$$

The max-weight policy is then:

- *If* $\sum_{1\leq s\leq N_R} D_{s,rel} \geq \sum_{0\leq s\leq N_R} D_s$ : activate the BS to RS $s^*$ link with $s^* = \arg\max_{1\leq s\leq R_S} D_{s,rel}$,
- *Else:* activate the stations to users links, and in each station $s$ serve the class of users $i_s^* = \arg\max_i n_{s,i}R_{s,i}$

### B. Finite buffer case: MDP formulation

We now assume that the system state $\mathbf{S}$ is restrained to $\mathcal{S} \subset \mathbb{N}^{(2N_R+1)N}$ with $\mathcal{S}$ finite due to admission control mechanisms. We formulate the problem as a Continuous Time Markov Decision Process (CTMDP) and optimize Quality of Service (QoS) metrics such as blocking rate or file transfer time. We formulate the problem in the small files framework since we want to solve the MDP iteratively, in order to keep the state space relatively small. The learning approach of the next section however can handle large state spaces as will be demonstrated.

*1) State and action spaces:* We assume that each link has a maximal number of simultaneous active users.

$$\mathcal{S} = \big\{\mathbf{S} | S_{rel,s,i} \leq \overline{S_{rel,s,i}} \, , \, 1 \leq s \leq N_R \, , \, 1 \leq i \leq N \\ \text{and } S_{s,i} \leq \overline{S_{s,i}} \, , \, 0 \leq s \leq N_R \, , \, 1 \leq i \leq N\big\}$$

We define $\mathcal{A} = \{0, 1\}$ the action space, with the convention:

- $a = 0$ : activate BS to RSs links and share them in a PS sharing manner
- $a = 1$ : activate stations to users links and share them in a PS sharing manner

*2) Transition probabilities:* Assuming that file size $\sigma$ is exponentially distributed, the system is a CTMDP. Transitions from $\mathbf{S}$ to $\mathbf{S}'$ given action $a$ have the following intensities:

- *Arrival of a user from class $i$ in the BS:* $\mathbb{1}_{\mathcal{S}}(\mathbf{s}') \int_{\mathbb{A}_{0,i}} \lambda dr$
- *Arrival of a user from class $i$ in the BS to RS $s$ link:* $\mathbb{1}_{\mathcal{S}}(\mathbf{s}') \int_{\mathbb{A}_{s,i}} \lambda dr$
- *Departure of a user from class $i$ in station $s$:* $\mathbb{1}_{\{1\}}(a)\mathbb{1}_{\mathcal{S}}(\mathbf{s}') \frac{R_{s,i}S_{s,i}}{\mathbb{E}[\sigma]\sum_{i=1}^{N} S_{s,i}}$
- *Movement of a user of class $i$ from BS to RS $s$ link to RS $s$ to users link:* $\mathbb{1}_{\{0\}}(a)\mathbb{1}_{\mathcal{S}}(\mathbf{s}') \frac{S_{rel,s,i}R_{rel,s}}{\mathbb{E}[\sigma]\sum_{i=1}^{N}\sum_{s=1}^{N_R} S_{rel,s,i}}$

*3) Average reward:* We call policy a mapping $\mathcal{S} \to \mathcal{D}(\mathcal{A})$, with $\mathcal{D}(\mathcal{A})$ the set of probability distributions on $\mathcal{A}$. We write $(\mathbf{S}(t), a(t), r(t))_{t\in\mathbb{R}^+}$ a realisation of the CTMDP with $\mathbf{S}(t)$ the state, $a(t)$ the action, and $r(t)$ the reward at time $t$ respectively. We are interested in the average reward criterion of a policy $P$:

$$J_{\mathbf{S}_0}(P) = \lim_{T\to+\infty} \frac{1}{T} \mathbb{E}_{P,\mathbf{S}_0} \left[ \int_0^T r(t) \right] \tag{9}$$

with $\mathbb{E}_{P,\mathbf{S}_0}$ the expectation with respect to the probability generated by $P$, starting at $\mathbf{S}_0$, which does not depend on $\mathbf{S}_0$ if the system is ergodic under policy $P$.

*4) Performance criteria:* We consider two performance criteria: mean file transfer time and blocking rate (considering admission control). For each performance criteria we can define a corresponding instantaneous reward for each state-action pair, and finding the optimal policy for the resulting MDP will yield the best policy with respect to the considered performance criteria.

To optimize the mean file transfer time, we define the reward in state $\mathbf{S}$ as the number of users divided by the arrival rate $\frac{\sum_{i=1}^{N}(S_{0,i}+\sum_{s=1}^{N_R}(S_{s,i}+S_{rel,s,i}))}{\int_{\mathbb{A}}\lambda dr}$ , and for any policy $P$ that renders the system ergodic, $J_{\mathbf{S}_0}(P)$ is the mean file transfer time in the system using Little's law ([14]).

We define the blocking rate as the ratio between the mean number of blocked users and the mean number of users accessing the system, once again assuming ergodicity. Given action $a$, let $\beta(\mathbf{S}, a)$ the sum of transition intensities out of state $\mathbf{S}$ and $b(\mathbf{S}, a)$ the sum of the intensities of arrival or movements which would be blocked, then the reward is defined as $\frac{b(\mathbf{S},a)}{\beta(\mathbf{S},a)}$.

*5) Optimal control and parametrization:* Given the previous description, we associate a Discrete Time Markov Decision Process (DTMDP) by uniformization and we derive the optimal policy using an iterative method, by the method described in [15]. It is noted that the complexity of finding the optimal policy is exponential in the number of relays, limiting the approach to small problems. In order to preserve scalability, we introduce a well-chosen family of policies. For commodity of notation we will use the following indexing of $\mathbf{S}$ : $(S_1, \cdots, S_k, \cdots, S_{(2N_R+1)N}) = ((S_{s,i})_{0\le s\le N_R, 1\le i\le N}, (S_{rel,s,i})_{1\le s\le N_R, 1\le i\le N})$. For $\theta \in \mathbb{R}^{(2N_R+1)N}$ we write $< \mathbf{S}, \theta >= \sum_{k=1}^{(2N_R+1)N} \theta_k S_k$. To $\theta$ we associate the deterministic weighted policy $P_{d,\theta}$:

$$P_{d,\theta}(\mathbf{S}, 1) = \begin{cases} 1 & , < \mathbf{S}, \theta > \ \geq 0 \\ 0 & , < \mathbf{S}, \theta > \ < 0 \end{cases} \tag{10}$$

$$P_{d,\theta}(\mathbf{S}, 0) = 1 - P_{d,\theta}(\mathbf{S}, 1) \tag{11}$$

It is noted that a deterministic weighted policy is essentially an hyperplane separating the state space in two regions, each half-space corresponding to an action of $\mathcal{A}$.

It is also noted that the max-weight policy is a deterministic weighted policy. We then compare the performance of three policies: the optimal policy, the max-weight policy and the optimal deterministic weighted policy. The optimal deterministic weighted policy is well defined since the set of deterministic policies is finite.

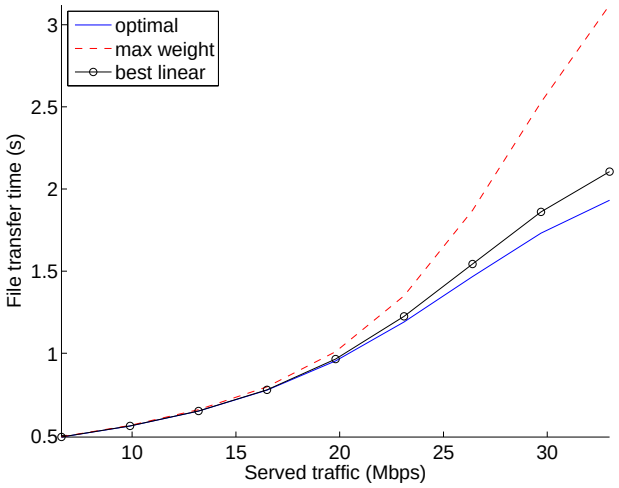


Fig. 5. File transfer time as a function of the traffic for different control strategies

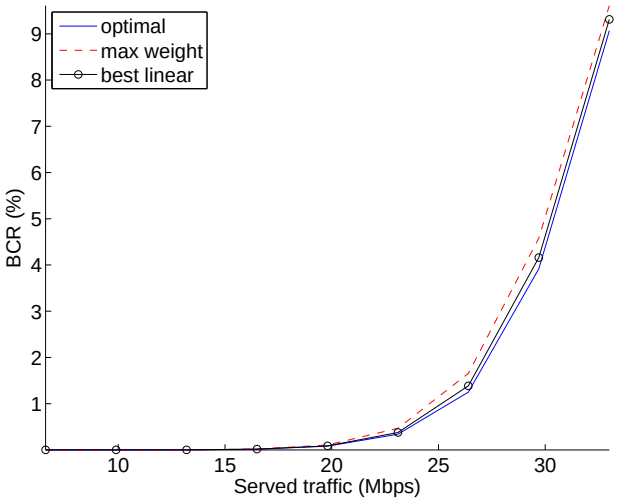


Fig. 6. Block call rate as a function of the traffic for different control strategies

Figure 5 and 6 show the file transfer time and the block call rate for the three policies, for one relay, one traffic class and a maximum of 10 users for all links. We can see that the max-weight policy is very close to the optimal policy when we are concerned with the block call rate, which is natural since it attempts to ensure stability. In the file transfer time case however, the optimal deterministic weighted policy is noticeably closer to the optimal policy than the max-weight. The fact the max-weight scheduling possibly incurs long delays has been reported in the literature. Hence based on those two results we can conclude that the set of deterministic weighted policies is rich enough to restrain the search to this

set, since with a high number of relays and/or traffic classes, finding the optimal policy becomes prohibitively expensive.

## IV. Learning

We have demonstrated that the set of weighted policies is rich enough to represent a good trade-off between performance and search complexity. We now move on to a model-free approach, and we assume no knowledge of the transition intensities and rewards. We are interested in learning the best weighted policy, simply by observing realisations of the Partially Observable Markov Decision Process (POMDP) $(\mathbf{S}(t), a(t), r(t))_{t\in\mathbb{N}}$. The model can be partially observed for various reasons. For example if user arrivals are correlated in time, the evolution of the system after $t$ depends on the user arrivals before $t$, and this information is not present in $\mathbf{S}(t)$. The method presented here is valid without assuming Poisson arrivals or exponentially distributed file sizes.

### A. Policy gradient approach

We use the approach introduced by [7] and extended to the average cost criteria in [8], [9]. It is noted that such algorithms work with stochastic policies, for the cost to be differentiable with respect to the policy parameter.

We introduce stochastic weighted policy $P_{s,\theta}$:

$$P_{s,\theta}(\mathbf{S},0) = 1 - f(<\mathbf{S},\theta>) \tag{12}$$

$$P_{s,\theta}(\mathbf{S},1) = f(<\mathbf{S},\theta>) \tag{13}$$

$$\text{with } f(x) = \frac{1}{1-e^{-x}} \tag{14}$$

we are interested in finding the $\theta$ which minimizes the average cost $J_{\mathbf{S}_0}(P_{s,\theta})$. The link with the policies introduced in the previous section is that any deterministic weighted policy $P_{d,\theta}$ can be approximated arbitrarily well by a stochastic weighted policy $P_{s,K\frac{\theta}{\|\theta\|}}$, with $K \in \mathbb{R}^+$ arbitrarily large.

### B. Convergence to a local optimum

We now show how to converge to a local optimum of the average cost. We differentiate the action probabilities:

$$\frac{\partial \log(P_{s,\theta}(\mathbf{S},0))}{\partial\theta_k} = -f(<\mathbf{S},\theta>)S_k = -P_{s,\theta}(\mathbf{S},1)S_k \tag{15}$$

$$\frac{\partial \log(P_{s,\theta}(\mathbf{S},1))}{\partial\theta_k} = (1-f(<\mathbf{S},\theta>))S_k = P_{s,\theta}(\mathbf{S},0)S_k \tag{16}$$

Using finiteness of $\mathcal{S}$, and the fact that $0 < P_{s,\theta}(\mathbf{S},a) < 1$, $a \in \{0,1\}, \mathbf{S} \in \mathcal{S}$ we have that:

- For every $\theta$, the Markov chain generated by policy $P_{s,\theta}$ is ergodic, implying that $J_{\mathbf{S}_0}(P_{s,\theta})$ is well-defined and does not depend on $\mathbf{S}_0$
- $\max\limits_{a\in\{0,1\}}\max\limits_{\mathbf{S}\in\mathcal{S}}\left|\frac{\partial \log(P_{s,\theta}(\mathbf{S},0))}{\partial\theta_k}\right| < +\infty$, $1 \le k \le (2N_R+1)N$
- $\max\limits_{a\in\{0,1\}}\max\limits_{\mathbf{S}\in\mathcal{S}} r(\mathbf{S},a) < +\infty$ , with $r(\mathbf{S},a)$ the reward given state $\mathbf{S}$ and action $a$

Given $\beta \in (0,1)$, and a realization of the POMDP $(\mathbf{S}(t), a(t), r(t))_{t\in\mathbb{N}}$ , we define the sequence of gradient estimates and the eligibility traces $(\Delta(t), z(t))_{t\in\mathbb{N}}$ by the following recursive equation:

$$z(0) = 0\ ,\ \Delta(0) = 0 \tag{17}$$

$$z(t+1) = \beta z(t) + \nabla_\theta \log(P_{s,\theta}(\mathbf{S}(t), a(t))) \tag{18}$$

$$\Delta(t+1) = \Delta(t) + \frac{1}{t+1}[r(t+1)z(t+1) - \Delta(t)] \tag{19}$$

Furthermore [8][Theorem 4] states that: $\Delta(t) \underset{t\to+\infty}{\to} \Delta_\infty(\theta)$ almost surely and that the dot product between $\Delta_\infty(\theta)$ and $\nabla_\theta J(\theta)$ is positive. In other words, for a given $\theta$, the limit of $-\Delta(t)$ is a descent direction. We consider $\Theta \subset \mathbb{R}^{(2N_R+1)N}$ a compact and convex set, $[.]^+_\Theta$ the projection on $\Theta$, $(\epsilon_n)_{n\in\mathbb{N}}$ a sequence of positive step sizes (satisfying the Wolfe conditions) and we define $\theta_n$ by:

$$\theta_0 \in \Theta \tag{20}$$

$$\theta_{n+1} = [\theta_n - \epsilon_n \Delta_\infty(\theta)]^+_\Theta \tag{21}$$

then we have that $\theta_n \underset{n\to+\infty}{\to} \theta_\infty$ with $\theta_\infty$ a local minimum of $J$ in $\Theta$ by a simple descent argument. $\theta_\infty$ is not necessarily unique if $J$ or $\Theta$ are not convex.

Furthermore, since $-\Delta_\infty(\theta)$ is a descent direction, we have that the performance of the system improves monotonically, which is a very interesting property for system implementation. This is in sharp contrast with the traditional "learning phase" of learning algorithms such as Q-learning ([16]) when the average reward changes rapidly.

The learning method converges to a locally optimum policy if $\{\theta_n\}$ converges to $\theta_\infty$ a local optimum of the cost. It is noted that convergence of the controller parameter $\theta$ implies convergence of policies.

### C. Implementation issues: assumptions on traffic and scalability

It is noted that the learning method is valid regardless of the statistical assumptions on traffic. Namely the validity of the policy gradient approach was shown by [8] even in the partially observable case.

It is noted that the algorithm is fully scalable when the number of relays increase since all the components of the descent direction $\Delta_\infty(\theta)$ are estimated from the same realization of the POMDP, incurring no additional costs when $N_R$ or $N$ increases. This is fundamental since some deployment scenarios include 30 RSs per BS.

### D. Numerical experiments

We now evaluate the performance of the learning algorithm in the same setting as Section III. Figures 7 and 8 represent the evolution of the mean file transfer time and the controller parameters $(\theta_1, \theta_2, \theta_3)$ respectively during the learning period. One update of $\theta$ corresponds to $10^3$ iterations of the underlying POMDP. As stated above, the mean file transfer time decreases in an almost monotonic fashion. The small variations are a

numerical artefact due to the fact that the average reward is calculated on a finite number of iterations of the POMDP.

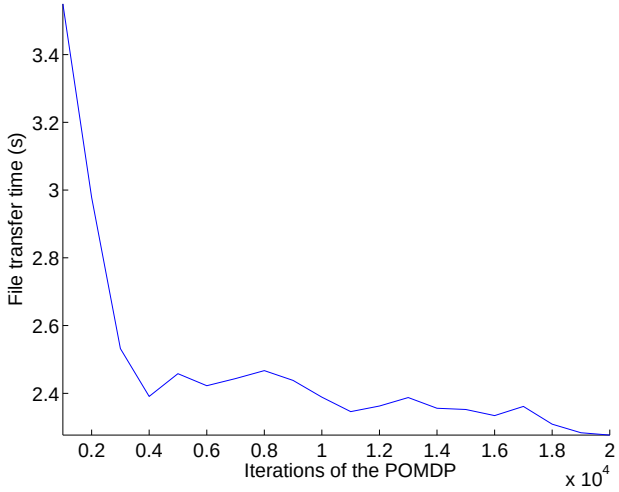


Fig. 7. File transfer time during the learning process

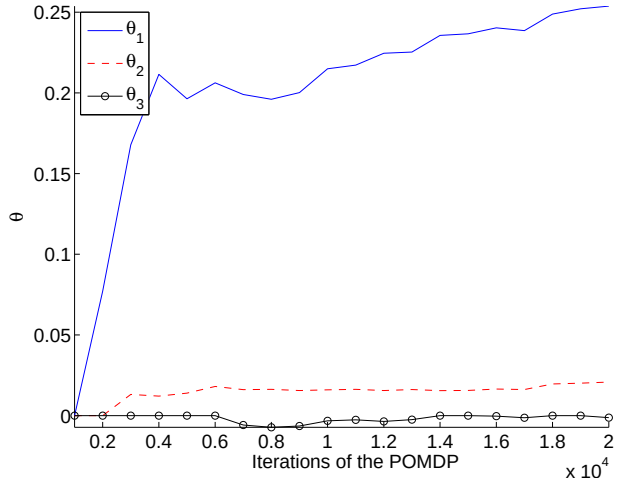


Fig. 8. Controller parameters $(\theta_1, \theta_2, \theta_3)$ during the learning process

## V. Conclusion

We have considered the problem of self-organized relays in a cellular network. The optimal static resource sharing between BS to RSs links and stations to users links has been derived in closed form using a queuing model. The influence of key system parameters has been investigated, showing the importance of relaying gain. Dynamic resource sharing has been considered using two approaches: stability for infinite buffers and blocking rate and file transfer time in the presence of admission control. The optimal policy has been derived using a MDP approach, which allowed us to introduce a well-chosen subset of the policy space as a form of expert knowledge. This expert knowledge has then been used in a model-free approach in which the optimal parametrized controller is found by observation and interaction with the system. Convergence to a local optimum has been demonstrated, and the fact that the performance of the system improves monotonically, which is a key property for system implementation.

## Appendix A
## Proof of Theorem 1

*Proof:* We first recall Loynes lemma for a G/G/1/First Come First Served (FCFS) queue: if $(A_n, \sigma_n)_{n\in\mathbb{Z}}$ is the stationary ergodic marked point process of arrival times and service requirements at a single server with service rate 1, then the stability condition is:

$$\lambda \mathbb{E}_A^0[\sigma_0] < 1 \tag{22}$$

with $\lambda$ the intensity of $A$ and $\mathbb{E}_A^0$ the Palm expectation with respect to $A$. The reader can refer to [17] for the proof. Furthermore, this remains valid for a G/G/1/PS queue since the workload process in the PS case is the same as in the FCFS case.

This allows to write the capacity of the link between the BS and users:

$$C_0(x) = (1-x)\left(\int_{\mathbb{A}_0} \frac{1}{R_0(r)} dr\right)^{-1} \tag{23}$$

and the capacity of the link between the BS and RSs:

$$C_{rel}(x) = x\left(\sum_{s=1}^{N_R} \frac{A_s}{R_{rel,s}}\right)^{-1} \tag{24}$$

Now assuming that the link between the BS and RSs is stable, its output process is stationary ergodic, and using a flow conservation argument it has the same intensity as the input. The capacity of the link between RS $s$ and its users is then:

$$C_s(x) = (1-x)\left(\int_{\mathbb{A}_s} \frac{1}{R_s(r)} dr\right)^{-1} \tag{25}$$

The stability of the system is equivalent to the stability of all queues, hence $C(x) = \min\left(C_{rel}(x), \min_{0\le s\le N_R}(C_s(x))\right)$. Furthermore $x \to C_{rel}(x)$ is strictly increasing and $x \to \min_{0\le s\le N_R}(C_s(x))$ is strictly decreasing, hence the unique optimal point $x^*$ is:

$$x^* = \frac{\left(\max_{0\le s\le N_R} \int_{\mathbb{A}_s} \frac{1}{R_s(r)} dr\right)^{-1}}{\left(\max_{0\le s\le N_R} \int_{\mathbb{A}_s} \frac{1}{R_s(r)} dr\right)^{-1} + \left(\sum_{s=1}^{N_R} \frac{A_s}{R_{rel,s}}\right)^{-1}} \tag{26}$$

Substitution of $x^*$ in the capacity formula yields $C^*$ which concludes the demonstration. ∎

## References

[1] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN); Overall description; Stage 2," 3rd Generation Partnership Project (3GPP), TS 36.300, Sep. 2008.

[2] ——, “Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions,” 3rd Generation Partnership Project (3GPP), TR 36.902, Sep. 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36902.htm

[3] A. Stolyar and H. Viswanathan, “Self-organizing dynamic fractional frequency reuse for best-effort traffic through distributed inter-cell coordination,” in *INFOCOM 2009, IEEE*, apr. 2009, pp. 1287 –1295.

[4] R. Combes, Z. Altman, and E. Altman, “Self-organizing fractional power control for interference coordination in ofdma networks,” in *IEEE ICC 2011*, 2011.

[5] R. Combes, M. Haddad, Z. Altman, and E. Altman, “Self-optimizing strategies for interference coordination in ofdma networks,” in *IEEE ICC 2011 PlaNet Workshop*, 2011.

[6] M. Dirani and Z. Altman, “A cooperative reinforcement learning approach for inter-cell interference coordination in ofdma cellular networks,” in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, may. 2010, pp. 170 –176.

[7] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine Learning*, vol. 8, pp. 229–256, 1992, 10.1007/BF00992696. [Online]. Available: http://dx.doi.org/10.1007/BF00992696

[8] J. Baxter and P. L. Bartlett, “Infinite-Horizon Policy-Gradient Estimation,” *Journal of Artificial Intelligence Research*, vol. 15, pp. 319–350, 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.8723

[9] J. Baxter, P. L. Bartlett, and L. Weaver, “Experiments with Infinite-Horizon, Policy-Gradient Estimation,” *Journal of Artificial Intelligence Research*, vol. 15, pp. 351–381, 2001. [Online]. Available: http://www.jair.org/papers/paper807.html

[10] R. Combes, Z. Altman, and E. Altman, “Scheduling gain for frequency-selective rayleigh-fading channels with application to self-organizing packet scheduling,” *Performance Evaluation*, Feb. 2011.

[11] R. Combes, S. E. Elayoubi, and Z. Altman, “Cross-layer analysis of scheduling gains: Application to lmmse receivers in frequency-selective rayleigh-fading channels,” in *WiOpt 2011*, 2011.

[12] L. Rong, S. Elayoubi, and O. Haddada, “Impact of relays on lte-advanced performance,” in *Communications (ICC), 2010 IEEE International Conference on*, May 2010, pp. 1 –6.

[13] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936 –1948, Dec. 1992.

[14] J. D. C. Little, “A Proof for the Queuing Formula: L= $\lambda$ W,” *Operations Research*, vol. 9, no. 3, pp. 383–387, 1961. [Online]. Available: http://dx.doi.org/10.2307/167570

[15] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 2005.

[16] R. Sutton and A. Barto, *Reinforcement Learning, an Introduction*. MIT Press, 1998.

[17] F. Baccelli and P. Bremaud, *Elements of Queueing Theory. Palm Martingale Calculus and Stochastic Recurrences*. Springer, 2nd ed, 2003.