Joint Scheduling of Communication and Computation Resources in Multiuser Wireless Application Offloading

Marc Molina, Olga Muñoz, Antonio Pascual-Iserte, Josep Vidal

Dept. Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
Email: olga.munoz@upc.edu

***Abstract*—We consider a system where multiple users are connected to a small cell base station enhanced with computational capabilities. Instead of doing the computation locally at the handset, the users offload the computation of full applications or pieces of code to the small cell base station. In this scenario, this paper provides a strategy to allocate the uplink, downlink, and remote computational resources. The goal is to improve the quality of experience of the users, while achieving energy savings with respect to the case in which the applications run locally at the mobile terminals. More specifically, we focus on minimizing a cost function that depends on the latencies experienced by the users and provide an algorithm to minimize the latency experienced by the worst case user, under a target energy saving constraint per user.**

***Index Terms*—Multiuser systems, small cell networks, application offloading, scheduling, energy efficiency, adaptive rate.**

I. Introduction

Cloud computing is a flexible and cost-effective concept that allows mobile terminals (MTs) to have access to larger computational and storage resources than those available in typical user equipment [1]. On the other hand, small cells deployments can be seen as an opportunity to offer low-cost solutions for cloud services if the small cell base stations (BSs) are enhanced with computational and storage capabilities [2,3]. In addition to the economic advantages, bringing computational power closer to the end user is expected to provide a lower latency along with an energy saving. As a result, an improvement on the user experience and a prolonged battery lifetime may be obtained. This, however, requires effective resource allocation and power control mechanisms so that the small cell BSs can properly perform computations from multiple users.

A description of the challenges for supporting mobile cloud computing applications in heterogeneous networks is provided in [4], including the offloading decision, admission control, cell association, power control, and resource allocation. Most of the work done so far corresponds to the offloading decision based on energy consumption criteria and experimental evaluation of the offloading performance [5-11]. The resource allocation problem in a multiuser set up is considered in [12] where, to avoid the instability of the queues, a mechanism is proposed that assigns more resources to those users with more bits to send in the uplink (UL)/downlink (DL) or instructions to execute. Different from [12], in this paper, we focus directly on the optimization of the quality of service (QoS) perceived by the different users in terms of the average latency. In addition to that, our scheme allows to trade easily between such QoS and the energy saving obtained with respect to the case of executing all the code locally at the MTs.

At this point it is important to remark that, in general terms, the offloading procedure depends on the kind of application that the users want to run. In this paper we are considering continuous-execution applications where there is a single execution stream per application (i.e., the different modules of the same application are not executed in parallel). More concretely, we consider that each user in the system has already decided to offload its corresponding application. For this scenario, different from [1], the paper focuses on how to deal with the allocation of the communication and remote computational resources among the users to achieve a low latency in the execution of the applications. Of course, other kinds of applications and scenarios could also be considered, but they are out of the scope of this paper.

The rest of the paper is organized as follows. Section II describes the system. Section III provides a model for the energy consumption of the MT and explains how the target energy saving impacts on the selection of the communication data rates, which in turns impact on the average latency of the users. The solution to the resource allocation problem is given in Section IV. Finally, Sections V and VI provide simulations results and conclusions, respectively.

II. System Description

We consider a system with K active users connected to a small cell BS endowed with some computational capabilities. In such scenario, each MT offloads a continuous-execution application to the BS. Each application or piece of code offloaded requires that the MT sends some input bits through the UL and that the BS executes a set of instructions in order to process (P) such bits. Then, the output bits are sent back from the BS to the user through the DL. See Fig. 1 as an illustrative example of the scenario.

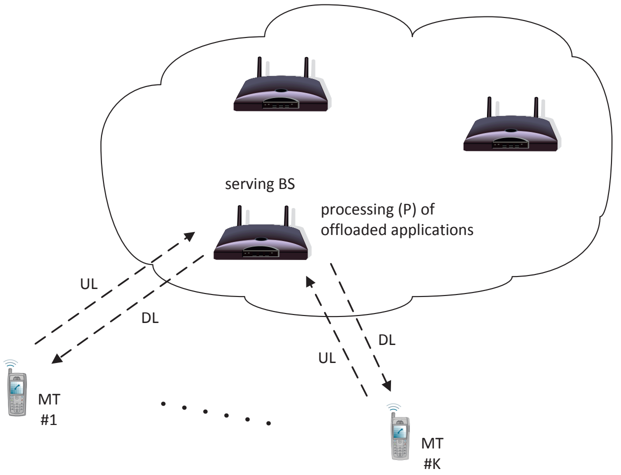

Fig. 1. Example of the scenario considered in this paper.

In order to accommodate the bursty nature of the packets to be processed, each stage (UL, P, and DL) implements one queue per user, which are served under TDMA.

Fig. 2 shows the UL, P, and DL queues for this system, where each queue is fed by the outputs coming out from the previous server.

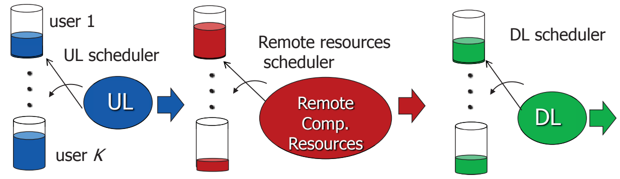

Fig. 2. System model with UL, P, DL queues and associated servers.

We will denote by $t_{UL,k}[n]$, $t_{P,k}[n]$, and $t_{DL,k}[n]$ the UL, P, and DL times allocated to the *k*-th user in the *n*-th scheduling period. Note that the *n*-th scheduling period of the three queues may be misaligned in time as a result of the different duration of the scheduling periods on each server, denoted by T_{UL}, T_P, and T_{DL} respectively (see Fig. 3). As it is shown in Fig. 3, the *n*-th scheduling period of the DL phase must follow the *n*-th scheduling period of the P stage that must, in turn, follow the *n*-th scheduling period of the UL phase.

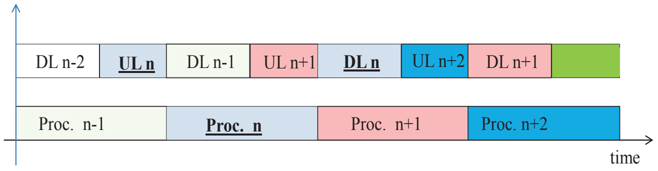

Fig. 3. UL, P, DL scheduling periods.

According to Little's theorem [13] for stable queues, the average latency experienced at each stage (UL, P, DL) measured in scheduling periods is given by

$$\bar{d}_{UL,k} = \frac{\bar{q}_{UL,k}}{\bar{\alpha}_{UL,k}}, \quad \bar{d}_{P,k} = \frac{\bar{q}_{P,k}}{\bar{\alpha}_{P,k}}, \quad \text{and} \quad \bar{d}_{DL,k} = \frac{\bar{q}_{DL,k}}{\bar{\alpha}_{DL,k}}, \tag{1}$$

with $\bar{q}_{UL,k}$, $\bar{q}_{P,k}$, and $\bar{q}_{DL,k}$ being the average number of elements waiting in the *k*-th user's UL, P, and DL queues, respectively, and $\bar{\alpha}_{UL,k}$, $\bar{\alpha}_{P,k}$, $\bar{\alpha}_{DL,k}$ the average number of input bits, instructions, and output bits arriving at the *k*-th user's UL, P, and DL queue per scheduling period (i.e., every T_{UL}, T_P, and T_{DL} seconds, respectively).

Note that if the system queues are stable, the values $\bar{\alpha}_{UL,k}$, $\bar{\alpha}_{P,k}$, and $\bar{\alpha}_{DL,k}$ will depend only on user behavior and application characteristics.

In the following, we focus on the average latency as QoS metric. In the case that absolute latency (instead of average) requirements have to be fit, Markov's inequality establishes that the latency outage probability (i.e. percentage of elements exceeding the absolute latency requirement) is upper bounded by the average latency divided by the absolute delay requirement [14]. Therefore, the minimization of the average latency is still meaningful in this case.

An estimation of the time-varying average latency can be obtained using an exponential weight window:

$$\hat{\bar{d}}_{i,k}[n+1] = (1-\beta)\hat{\bar{d}}_{i,k}[n] + \beta d_{i,k}[n+1], \tag{2}$$

where $d_{i,k}[n+1]$ is the estimated instantaneous latency experienced by the *k*-th user in the *i*-th queue (*i* ϵ {UL, P, DL}) at the beginning of the (*n*+1)-th scheduling period.

The instantaneous latency, $d_{i,k}[n+1]$, can be estimated [14] as

$$d_{i,k}[n+1] = q_{i,k}[n+1] / \bar{\alpha}_{i,k}, \tag{3}$$

with $q_{i,k}[n+1]$ being the number of elements in the *i*-th queue of the *k*-th user at the beginning of the (*n*+1)-th scheduling period.

We may replace $q_{i,k}[n+1]$ by

$$q_{i,k}[n+1] = q_{i,k}[n] + \alpha_{i,k}[n] - s_{i,k}[n], \tag{4}$$

with $\alpha_{i,k}[n]$ is the number of elements arriving at the *i*-th queue of the *k*-th user during the *n*-th scheduling period and $s_{i,k}[n]$ is the number of bits sent (in UL and DL) or instructions processed (in P) in the allocated time $t_{i,k}[n]$.

Now combining(2), (3), and (4) we obtain:

$$\hat{\bar{d}}_{i,k}[n+1] = (1-\beta)\hat{\bar{d}}_{i,k}[n] + \beta \frac{q_{i,k}[n] + \alpha_{i,k}[n] - s_{i,k}[n]}{\bar{\alpha}_{i,k}}. \tag{5}$$

Note that while $\alpha_{UL,k}[n]$ is the rate of bits generated directly by the MT, $\alpha_{P,k}[n]$ and $\alpha_{DL,k}[n]$ depends on the decisions of the UL and P schedulers during previous scheduling periods (see Fig. 2).

It is worthy to remark that expression (5) provides the estimation of the average latency per bit in the UL or DL queues, or per instruction in the P queue. In case that the processing of a set of instructions cannot start until the packet of data bits to be processed by such instructions has been sent completely through the UL, we should add a term to (5) accounting for this. The same applies for the DL, if we need to wait until a set of instructions have been executed before piling information to the DL queues.

The problem that needs to be tackled within this framework is how to split the time resources at the beginning of each scheduling period (see Fig. 3) to reduce the latency experienced by the users. Note that we may always reduce this latency by increasing the communication rate in UL and DL, provided that the channel and the maximum transmission power of the transmitter support it. This increment, however, will have an impact on the energy consumption of the MTs. We will address this issue in the next section, while in section IV we will formalize and propose a method that solves the scheduling problem and allocates the time resources of the UL, P, and DL.

For the sake of clarity in the presentation of the scheduling technique in the following sections, we collect the notation used in the paper in the table II, where the subindex i denotes the UL, P, and DL stages/queues, i.e. (i ϵ {UL, P, DL}).

TABLE I
NOTATION USED

$t_{i,k}[n]$	Times allocated to k-th user in the n-th scheduling period of the i-th stage
T_i	Duration of the scheduling period associated to the i-th stage
$q_{i,k}[n]$	Number of elements in the k-th user's i-th queue at the beginning of the n-th scheduling period
$\bar{q}_{i,k}$	Average number of elements waiting in the k-th user's i-th queue
$\alpha_{i,k}[n]$	Number of elements arriving at the k-th user's i-th queue during the n-th scheduling period
$\bar{\alpha}_{i,k}$	Average number of input bits, instructions, and output bits arriving at the k-th user's i-th queue per scheduling period
$s_{i,k}[n]$	Number of bits sent (in UL and DL) or instructions processed (in P) in the allocated time corresponding to the k-th user's i-th stage during the n-th scheduling period
$\bar{d}_{i,k}$	Averaged latency experienced by the k-th user at the i-th stage
$\hat{\bar{d}}_{i,k}[n]$	Exponential temporal average of the latency experienced by the k-th user associated to the i-th queue (UL, P, DL) up to the n-th scheduling period
$d_{i,k}[n]$	Instantaneous estimate of the latency experienced by the k-th user associated to the i-th queue (UL, P, DL) at the n-th scheduling period

III. ENERGY CONSUMPTION

In addition to the expected reduced latency, the offloading will be worthy if the energy spent in average by the MT when offloading an application (which requires wireless transmission of data) is lower than the energy required for doing the processing locally. To compare both quantities, we need a model for the energy consumption of the terminal when the terminal offloads an application and also when the terminal runs the application locally.

The measurements provided in [15] for a LTE-MT dongle show that the UL transmit power, $p_{tx,k}$, and the DL data rate, $r_{DL,k}$ (because of the increase on decoding complexity), greatly affect the power consumption, while the UL encoding rate and the DL received power has little effect. In addition to that, a baseline power is also consumed just for having the transmitter and receiver chains switched on [15]. However, for conventional user terminals without microsleep capabilities [16,17], the baseline energy consumption due to having the transmission circuitry on is present even if the computation is performed locally. For this reason we will not considered this baseline power consumption when comparing the energy consumption with and without offloading. Following the assumptions mentioned in previous works, we adopt the following approximated power consumption models associated to the UL and DL transmissions to compute the extra power required for performing the remote processing with respect to the case of doing the processing locally:

$$p_{UL,k} = k_{tx,k} p_{tx,k}, \qquad p_{DL,k} = k_{rx,k} r_{DL,k}, \tag{6}$$

with $k_{tx,k}$ and $k_{rx,k}$ being model dependent constants.

An upper bound on the transmission rate that depends on the transmission power and the quality of the channel is given by Shannon's formula [18]. Despite it is a theoretical bound, it is widely used to predict the supported rate. A constant Γ can account for the gap between the theoretical and real performance [19]:

$$r_{UL,k} = \Gamma W \log_2\left(1+\gamma_{UL,k} p_{tx,k}\right). \tag{7}$$

In eq. (7), W represents bandwidth measured in Hz, and $\gamma_{UL,k}$ and $\gamma_{DL,k}$ are the channel gains normalized by the noise power in UL and DL, respectively.

From expressions (6)-(7), the energy spent by the MT in the offloading process at each UL and DL scheduling period can be computed as follows:

$$e_{UL,k}[n] = k_{tx,k} t_{UL,k}[n] \frac{2^{\frac{r_{UL,k}[n]}{\Gamma W}}-1}{\gamma_{UL,k}}, \tag{8}$$

$$e_{DL,k}[n] = k_{rx,k} s_{DL,k}[n]. \tag{9}$$

The average energy consumption per second for the k-th user can be written as follows, assuming that $r_{UL,k}$ remains constant over the averaging period:

$$\bar{e}_k\left(r_{UL,k}\right) = k_{tx,k} \frac{\bar{\alpha}_{UL,k}}{T_{UL} r_{UL,k}} \frac{2^{\frac{r_{UL,k}}{\Gamma W}}-1}{\gamma_{UL,k}} + k_{rx,k} \frac{\bar{\alpha}_{DL,k}}{T_{DL}}. \tag{10}$$

Note that in (10) we do not include the energy consumption for the processing stage since, when offloading is performed, the computation and execution of the instructions imply an energy spenditure outside the MT. The scheduling strategy that we present in this paper could be directly adapted to include the energy consumption of the BS. However, in this paper, we focus only on the MT energy consumption as it is the MT the terminal that is battery limited.

An important observation is that $\bar{e}_k\left(r_{UL,k}\right)$ (see eq. (10)) does not change with the DL data rate, but it is an increasing function of the UL date rate.

We will consider that the offloading is worthy for the MT from an energy point of view if $\bar{e}_k\left(r_{UL,k}\right)$ in (10) is lower than a percentage of the average energy per second spent by the MT when performing all the processing locally i.e.

$$\bar{e}_k\left(r_{UL,k}\right) \le (1-\rho)\bar{\alpha}_{P,k}\varepsilon_0 / T_P. \tag{11}$$

In (11), ε_0 is the energy required to process an instruction locally (i.e. the energy consumed per CPU cycle for computation at the MT), and ρ the energy saving that we want to achieve.

Combining (10) and (11) results in an upper bound for the UL data rate equal to inverse function of (10), $\bar{e}_k^{-1}$, evaluated at $(1-\rho)\bar{\alpha}_{P,k}\varepsilon_0 / T_P$:

$$r_{UL,k} \le \bar{e}_k^{-1}\left(\frac{(1-\rho)\bar{\alpha}_{P,k}\varepsilon_0}{T_P}\right). \tag{12}$$

Increasing the target energy saving, ρ, will result in a tighter bound for the UL data rate. This means that we can exchange latency in the UL by energy consumption or viceversa. The reason is because we can increase the UL data rate (and therefore reduce latency) at the expense of increasing the energy consumption of the MT. On the other hand, as far as the DL data rate is concerned, the best strategy from the MT point of view is to increase the DL data rate as much as possible, since this reduces the latency experienced by the MT, without affecting the energy consumption of the MT.

IV. Problem Formulation and Solution

While the UL and DL schedulers may work together, it may be unpractical or too complex that the processor scheduler and the radio schedulers take decisions jointly because of the high associated complexity and the huge amount of messages that would need to be exchanged among different layers of the protocol stack. Due to these considerations, we assume in this paper a decoupled resource allocation problem with three schedulers, each one managing the queues of the UL, P, and DL independently. Although the schedulers operate separately, the global performance of the system implies an inherent coupling since each queue is fed by the outputs coming out from the previous server.

The problem to be solved at each queue during the *n*-th scheduling period has the same structure and can be formulated as (for $i \in$ {UL, P, DL}):

$$\begin{aligned} \min_{\{t_{i,k}[n], s_{i,k}[n]\}_{k=1}^{K}} \quad & c\left(\left\{\hat{\bar{d}}_{i,k}[n+1]\right\}_{k=1}^{K}\right) \\ \text{s.t.} \quad & \sum_{k=1}^{K} t_{i,k}[n] \le T_i, \\ & s_{i,k}[n] \le q_{i,k}[n], \\ & s_{i,k}[n] \le t_{i,k}[n] \cdot R_{i,k}. \end{aligned} \tag{13}$$

In problem (13), $c(\cdot)$ is the cost function to be minimized (it depends on the estimated average latencies of the *K* users), and $R_{i,k}$ is a rate constraint. Note that:

- In the P stage, $R_{P,k}$ is the number of instructions per second that the remote processor can execute, and hence is independent of the user.
- As far as $R_{UL,k}$ is concerned, this value given by the right-hand side of (12) and depends on the target energy saving, ρ, of the *k*-th MT. For higher target energy savings, $R_{UL,k}$ will be lower, and therefore the latency will increase.
- $R_{DL,k}$ is the maximum DL data rate that it is only determined by the DL channel and the physical layer.

In the following we adopt as cost function the latency for the worst case user, i.e.,

$$c\left(\left\{\hat{\bar{d}}_{i,k}[n+1]\right\}_{k=1}^{K}\right) = \max_k \left\{\hat{\bar{d}}_{i,k}[n+1]\right\}_{k=1}^{K}. \tag{14}$$

The previous problem can be rewritten in a simplified way by using a dummy variable *d* that refers to the latency of the worst case user. In addition to that, it can be proved easily that the optimum solution implies that the rate constraint is fulfilled with equality (i.e., $s_{i,k}[n] = t_{i,k}[n] \cdot R_{i,k}$), otherwise, the scheduled time for the users for which the rate constraint is fulfilled with strict inequality could be lowered until equality is fit without affecting the value of the cost function and the fulfillment of the other constraints. Thanks to this, we can consider the scheduling times $\{t_{i,k}[n]\}$ as the only optimization variables in the new reformulated problem (where the expression of the average latency given in eq. (5) has been included):

$$\begin{aligned} \min_{d, \{t_{i,k}[n]\}_{k=1}^{K}} \quad & d \\ \text{s.t.} \quad & (1-\beta)\hat{\bar{d}}_{i,k}[n] + \beta \frac{q_{i,k}[n] + \bar{\alpha}_{i,k} - t_{i,k}[n] \cdot R_{i,k}}{\bar{\alpha}_{i,k}} \le d, \\ & \sum_{k=1}^{K} t_{i,k}[n] \le T_i, \\ & t_{i,k}[n] \cdot R_{i,k} \le q_{i,k}[n]. \end{aligned} \tag{15}$$

Note that in the estimation of the latency appearing in the first constraint of problem (15), we have substituted $\alpha_{i,k}[n]$ (see eq. (4) with its average value $\bar{\alpha}_{i,k}$ becuase, in practice, the scheduler will not be aware of $\alpha_{i,k}[n]$ when taking the resource allocation decision (this is the approach followed also in [14]). From the first and last constraints in (15), it can be proved that the time to be allocated to the *k*-th user has to fulfill the following conditions:

$$t_{i,k}[n] \ge \frac{1}{R_{i,k}} \left(q_{i,k}[n] + \bar{\alpha}_{i,k} - \frac{\bar{\alpha}_{i,k}}{\beta} \left(d - (1-\beta)\hat{\bar{d}}_{i,k}[n] \right) \right)^{+}, \tag{16}$$

$$t_{i,k}[n] \le \frac{q_{i,k}[n]}{R_{i,k}}, \tag{17}$$

where $(x)^{+} = \max\{0, x\}$. Considering jointly constraints 1 and 3 for all the users in problem (15), we can derive the following condition:

$$d \ge d_{\min_i} = \max_k \left\{ (1-\beta)\hat{\bar{d}}_{i,k}[n] + \beta \right\}. \tag{18}$$

For a given value of *d*, the optimum allocated time for each user are given by the lower bound in (16). Note that any other solution would spend more time of the total scheduling period T_i without reporting any improvement in the cost function, i.e., in the latency of the worst-case user. Thus, for a given value of *d*, the optimum scheduled time for the *k*-th user is given by (16) with equality.

The optimum solution consists, then, in solving the following problem:

$$\begin{aligned} \text{find} \quad & d \ge d_{\min_i} = \max_k \left\{ (1-\beta)\hat{\bar{d}}_{i,k}[n] + \beta \right\} \\ \text{s.t.} \quad & \sum_k \frac{1}{R_{i,k}} \left(q_{i,k}[n] + \bar{\alpha}_{i,k} - \frac{\bar{\alpha}_{i,k}}{\beta} \left(d - (1-\beta)\hat{\bar{d}}_{i,k}[n] \right) \right)^{+} = T_i. \end{aligned} \tag{19}$$

The previous problem can be solved by applying any standard algorithm for solving non-linear equations, taking into account that the restriction decreases monotonically in *d*. In this case, for example, the nested intervals algorithm is a good choice since it assures convergence with exponential speed [20].

If $\sum_k t^*_{i,k}[n] < T_i$, with $t^*_{i,k}[n]$ the optimal value of the time allocated to k-th user in the n-th scheduling period of the i-th stage, then problem (15) has a non-unique optimum solution. In this case, we propose a concrete optimum solution that consists in identifying the user producing the worst latency, i.e., the user k_1 such that $d_{\min_i} = (1-\beta)\hat{\bar{d}}_{i,k_1}[n]+\beta$. Then we allocate the time needed for that user to achieve the minimum latency, which is $t^*_{i,k_1}[n] = q_{i,k_1}[n]/R_{i,k_1}$, and finally, we subtract from the scheduling time this quantity: $T_i \leftarrow T_i - q_{i,k_1}[n]/R_{i,k_1}$. Once this is done, the same procedure is applied to the rest of the users iteratively. The complete algorithm is described in detail in Table II.

TABLE II

SCHEDULING PROCEDURE

```
1: define $\bar{K} = \{1,2,\ldots,K\}$, $T_i$ is the scheduling time
2: calculate: $d_{\min_{i,k}} = (1-\beta)\hat{\bar{d}}_{i,k}[n] + \beta$
3: calculate: $d_{\min_i} = \max_{k\in\bar{K}}\{d_{\min_{i,k}}\}$
4: calculate $x = \sum_{k\in\bar{K}} \frac{1}{R_{i,k}}\left(q_{i,k}[n] - \frac{\bar{\alpha}_{i,k}}{\beta}\left(d_{\min_i} - d_{\min_{i,k}}\right)\right)^+$
5: if $x \ge T_i$
6: find $d$ such that $\sum_{k\in\bar{K}} \frac{1}{R_{i,k}}\left(q_{i,k}[n] - \frac{\bar{\alpha}_{i,k}}{\beta}\left(d - d_{\min_{i,k}}\right)\right)^+ = T_i$
7: set $t^*_{i,k}[n] = \frac{1}{R_{i,k}}\left(q_{i,k}[n] - \frac{\bar{\alpha}_{i,k}}{\beta}\left(d - d_{\min_{i,k}}\right)\right)^+ \quad \forall k \in \bar{K}$
8: go to step 15
9: else
10:    find $k_1 \in \bar{K}$ such that $d_{\min_{i,k_1}} = d_{\min_i}$
11:    set $t^*_{i,k_1}[n] = q_{i,k_1}[n] / R_{i,k_1}$
12:    set $\bar{K} \leftarrow \bar{K} - \{k_1\}$ and $T_i \leftarrow T_i - q_{i,k_1}[n] / R_{i,k_1}$
13:    go to step 3
14: end if
15: end algorithm
```

V. SIMULATION RESULTS

In order to evaluate the performance of the scheduler, we have considered a detection application that compares an audio signal with size 262.144 kBytes captured by the terminal with N=20 patterns by performing cross-correlations. Such application requires around 337 cycles per byte [21].

We have considered that the same number of bits is sent in the UL and the DL. The maximum transmission power for both the MT and the BS is 100 mW. The MT processor corresponds to a commercial model (Nokia N900) that performs 650 Mcycles per Joule when operating at 600 MHz [5]. It is assumed that the remote processor is twice faster than the local one (i.e., 1.2 GHz). The values of the constants in the energy model are $k_{tx,k} = 18$ and $k_{rx,k} = 2.86$ mW/Mbps. Finally, the value of Γ in (7) is 1, and the scheduling times are $T_{UL} = T_{DL} = 1$ ms as in LTE [22], and $T_P = 20$ ms [23].

We have considered 5 MT's that offload this application to the same small cell BS. The request arrival to the UL queue follows a Poisson distribution with a mean rate of one audio signal per second. The channels gains are assumed to be $\gamma_{UL,1} = \gamma_{DL,1} = 17.5$ dB for the first user and 19 dB for the other users.

Fig. 4 shows the estimated average latency per user at each one the three queues for a specific energy saving (40%).

Fig. 5 shows the actual average latency experienced by the users to process completely a signal as a function of the energy saving factor ρ (see eq. (12)) with respect to the local computation. It is important to remark that the latency is the sum of the latencies experienced through the three queues. The figure compares the performance of the algorithm proposed in Table II (i) with two different algorithms: all the scheduling period is allocated to the user with more bits/instructions waiting in the queue (ii), or with more signals waiting to be transmitted/processed (iii). As expected, we observe that in all cases the average latency to process a signal is higher as the target energy saving increases. For a particular energy saving, we can see that the average latency obtained with the proposed algorithm (i) is significantly lower than the obtained with the other ones, especially if we compare it with (ii), where the improvement is around 35%. Note also that for a given target latency (for instance 350 ms), the energy saving that we can achieve with our algorithm (i) is much higher than with the algorithms (ii) and (iii).

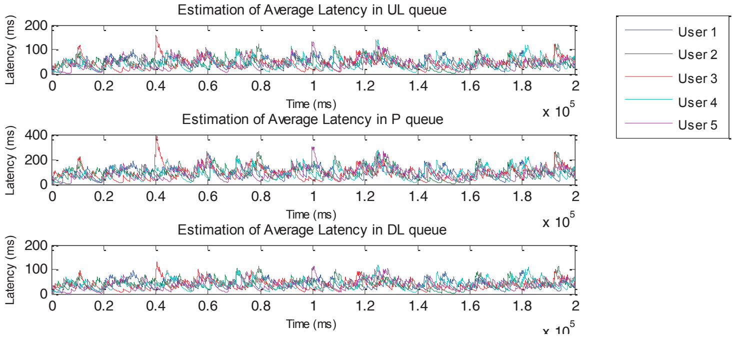

Fig. 4. Estimated average latency for an energy saving of 40%.

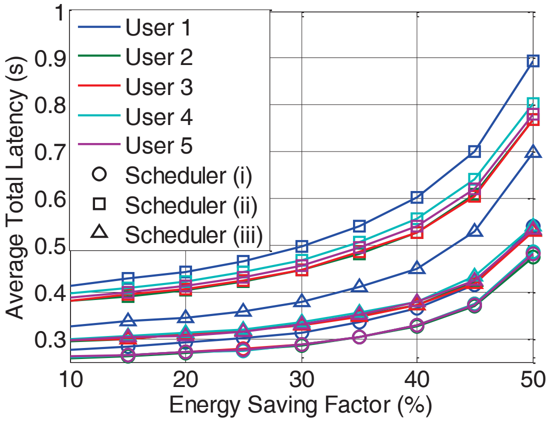

Fig. 5. Actual average total latency per signal versus the target energy saving factor ρ.

Fig. 6 shows the empirical cumulative density function (CDF) of the latency per signal obtained with the three algorithms for a specific energy saving (40%). We observe that the proposed algorithm shows a better performance not only in terms of average latency, but also in terms of jitter. Note also that, although user 1 has the worst channel, its experienced latency is closer to the other users' latencies with our algorithm than with the other two strategies.

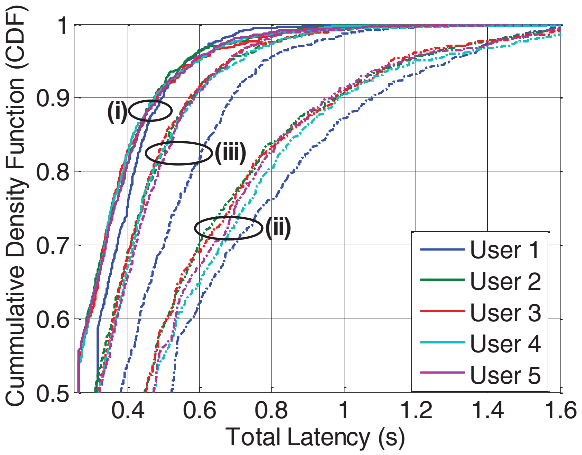

Fig. 6. CDF of the actual average total latency per signal for an energy saving of 40%.

VI. Conclusions

We have developed a scheduling strategy for multiuser offloading systems where the resources are allocated under the objective of minimizing the average experienced latency of the worst case user. We have seen that our algorithm obtain good results even if we force a high reduction in the energy spending of the offloading process with respect to the case of performing all the computations locally.

We leave as open problem for future research the generalization of the proposed approach by including admission control policies to avoid, for instance, that one user degrades severely the performance of the others; and the formulation and solution of the single scheduler managing jointly the queues of the three stages for benchmark purposes. Another issue to be considered as a future work is the consideration of applications that can be split so that some subprocesses of the applications can run at the MT whereas other subprocesses can run in parallel at the BS.

Acknowledgments

This work was supported in part by the European Commission (FP7) through TROPIC INFSO-ICT-318784, by the Spanish "Ministerio de Economía y Competitividad" through projects TEC2011-29006-C03-02 (GRE3N-LINK-MAC) and TEC2013-41315-R (DISNET), and by grant 2014 SGR 60 of the Catalan Administration (AGAUR).

References

[1] S. Ren, and M. Schaar, "Efficient Resource Provisioning and Rate Selection for Stream Mining in a Community Cloud," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 723-734, June 2013.

[2] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia Cloud Computing," *IEEE Signal Processing Magazine,* vol. 28, no. 3, pp. 59-69, May 2011.

[3] TROPIC: Distributed computing, storage and radio resource allocation over cooperative femtocells, http://www.ict-tropic.eu/

[4] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on Wireless Heterogeneous Networks for Mobile Cloud Computing," *IEEE Wireless Communications,* vol. 20, no. 3, p. 34-44, June 2013.

[5] A.P. Miettinen and J.K. Nurminen, "Energy Efficiency of Mobile Clients in Cloud Computing," in *Proc. 2nd USENIX Conference on Hot Topics in Cloud Computing 2010 (HotClout'10)*, Boston (USA), June 2010.

[6] E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making Smartphones Last Longer with Code Offload," in *Proc. International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, San Francisco, California (USA), pp. 49-62, June 2010.

[7] K. Kumar and Y-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?," *IEEE Computer,* vol. 43, pp. 51-56, April 2010.

[8] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A Survey of Computation Offloading for Mobile Systems," *Mobile Networks and Applications,* Springer Science, pp. 129–140, April 2012.

[9] E. Lagerspetz, S. Tarkoma "Mobile search and the cloud: The benefits of offloading," in *Proc. Ninth Annual IEEE International Conference on Pervasive Computing and Communication (PerCom'11)*, pp. 21-25, Seattle (USA), March 2011.

[10] D. Kovachev, and R. Klamma , "Framework for Computation Offloading in Mobile Cloud Computing", *International Jorunal of Interactive Multimedia and Artificial Intelligence,* vol. 1, no. 7, pp. 6-15, December 2012.

[11] O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Joint Allocation of Radio and Computational Resources in Wireless Application Offloading," in *Proc. Future Network & Mobile Summit (FUNEMS'13)*, July 2013.

[12] S. Barbarossa, S. Sardellitti, P. Di Lorenzo, "Joint Allocation of Computation and Communication Resources in Multiuser Mobile Cloud Computing," in *Proc. 14th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Darmstadt (Germany), pp. 16-19, June 2013.

[13] J. Little, "A Proof of the Queueing Formula $L=\lambda W$", *Oper. Res. J.*, 18:172-174, 1961.

[14] X. Wang, G. B. Giannakis, and A. G. Marques, "A Unified Approach to QoS-Guaranteed Scheduling for Channel-Adaptive Wireless Networks", *Proc.* IEEE, vol. 95, no. 12, December 2007.

[15] A. Jensen, M. Lauridsen, P. E. Mogensen, T. Sørensen, and P. Jensen, "LTE UE Power Consumption Model: For System Level Energy and Performance Optimization," in *Proc. IEEE Vehicular Technology Conference Fall (VTC'12)*, pp. 1-5, Sept. 2012.

[16] J. Wigard, T. Kolding, L. Dalsgaard, and C. Coletti, "On the User Performance of LTE UE Power Savings Schemes with Discontinuous Reception in LTE," in *Proc. IEEE International Conference on Communications Workshops (ICC Workshops)*, June 2009.

[17] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A Close Examination of Performance and Power Characteristics of 4G LTE Networks", in *Proc. International Conference on Mobile Systems, Applications, and Services (MobiSys'12)*, Low Wood Bay, Lake District (UK), June 25–29, 2012.

[18] T. M. Cover, J. A. Thomas, *Elements Of Information Theory*, John Wiley & Sons, 1991.

[19] O. Muñoz-Medina, A. Agustín, and J. Vidal, "MCS and Sub-band Selection for Downlink Interference Coordination in LTE-A Femtocells," in *Proc. of IEEE Vehicular Technology Conference (VTC Fall),* Quebec City, 3-6 September 2012.

[20] A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics, Texts in Applied Mathmatics*, chapter 6.2, Springer-Verlag, New York, Inc. (2000), Second edition, 2007.

[21] S. G. Johnson and M. Frigo, "A modified split-radix FFT with fewer arithmetic operations", *IEEE Transactions on Signal Processing*, vol. 55, pp. 111-119, 2007.

[22] S. Sesia, I. Toufik, and M. Baker, *LTE-The UMTS Long Term Evolution: From Theory to Practice,* John Wiley & Sons, 2009.

[23] A. Noon, A. Kalakech, S. Kadry, "A New Round Robin Based Scheduling Algorithm for Operation Systems: Dynamic Quantum Using the Mean Average", *Internation Journal of Computer Science Issues (IJCSI),* vol. 8, issue 3, no. 1, May 2011.