Energy-Efficient Task Offloading for Multiuser Mobile Cloud Computing

Yun Zhao, Sheng Zhou, Tianchu Zhao and Zhisheng Niu
Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
Email: zhaoyun12@mails.tsinghua.edu.cn

***Abstract*—Mobile cloud computing enables the resource-poor mobile terminals to deal with the resource-hungry applications thanks to the extra storage and computational resources at the cloud side. However, the advantages of mobile cloud computing cannot be fully exploited without proper collaboration of radio transmission and computing, which is challenging under multiuser scenarios due to the coupling of radio and computational resources. In this paper, targeting at reducing the terminal energy consumption, we study the joint optimization of radio and computational resources for multiple users in mobile cloud computing and propose a heuristic strategy, based on the latency constraint and the application type of each MT, for resource allocation of low computational complexity. Numerical results show that our proposed task offloading strategy can significantly reduce the total energy at the mobile terminal side by 40% with 3 mobile terminals in the system, compared with the non-offloading mobile computing scheme, while at the same time satisfying the delay constraints. Moreover, it performs fairly close to the optimum.**

***Index Terms*—mobile cloud computing, multiuser, radio and computational resources, energy-efficient offloading.**

I. Introduction

Mobile cloud computing emerges to deal with the tension between resource-hungry applications and resource-poor mobile terminals (MTs). It is a flexible technology to allow MTs to obtain more compute and storage resources from the so-called clouds through wireless links. However, mobile cloud computing also encounters several technical bottlenecks despite its rapid development, among which the unsatisfactory MT battery duration and delay constraints of various applications greatly restrict the system performance [1]. In recent years, numerous media plays and video calls run on the MTs, which attracts more and more users and stimulates new mobile applications. However, this also causes the exponential data growth and the MTs suffer desperately from the limited battery capacity due to the limited physical size [2]. On the other hand, the next generation wireless communication systems aim to achieve millisecond-level delay and 100 times data rate and researchers put more emphasis on delay requirements of the users [3]. Therefore, proper resource allocation is essential to solve the challenge of terminal energy consumption and delay requirements in mobile cloud computing.

Due to the flexible resource allocation from the resource pool and the capability of assisting MTs with energy reduction [4], various mobile cloud platforms, such as Cloudlet [5], Clone Cloud [6] and CONCERT [7] have been proposed, among which the distributed cloud architecture [8] stands out. Concretely, the small-cell base stations are endowed with additional while limited cloud functionalities called femto-cloud. Whenever the MTs' request can be met by the local femto-cloud, everything is performed locally. Otherwise, the base station may ask the remote cloud server for help. In this way, both the radio and computational resources are brought closer to the MTs, which is quite similar with the framework in CONCERT.

It is important to note that the offloading and partitioning procedure varies according to different kinds of applications the MTs run. In this paper, we consider that the program supports partitioning when the program is developed. We assume that both the computation at the MTs and the femto-cloud can be processed in parallel [9] in order to show the fundamental trade-offs in offloading and will not go into the details of partitioning algorithms.

Resource allocation in mobile cloud computing has been an important area of research. In [10], the problem of allocating jobs to data centers has been formulated as a Markov Decision Process and an index policy is proposed. However, it only consists of task allocation at the cloud side. Reference [9] analyzes the optimization of the computational and radio resources usage in single mobile user scenario. References [11]–[13] apply multiuser scenario and assume that all the mobile users in the system have already decided to offload all their tasks to the mobile cloud. Different from the previous references, we consider the scenario with multiple MTs in the system and assume each task at the MT can be partially offloaded. We try to optimize the scheme for resource allocation to reduce the energy consumption at the MT side while satisfying delay constraints.

The main contributions of the paper include:

1)We provide a framework for the multiuser joint optimization of radio and computational resources in the mobile cloud computing scenario.

2)We propose a task and delay based resource allocation scheme that can be easily implemented. Numerical results show that the proposed scheme performs fairly close to the optimum.

The rest of this paper is organized as follows. Section II describes the delay and energy model for processing the application in the femto-cloud scenario. Then we formulate the optimization problem and make some simplifications in

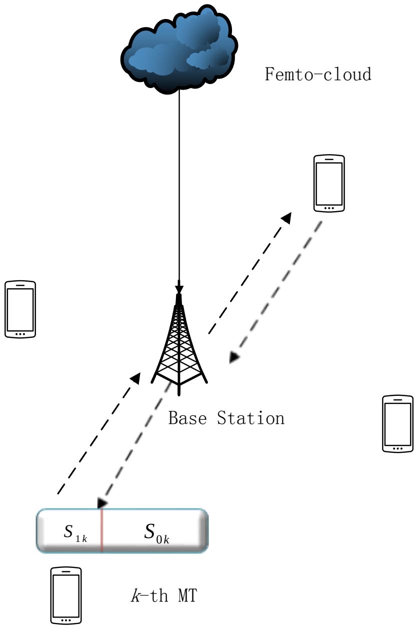

Fig. 1. Femto-cloud scenario.

Section III. A heuristic resource allocation scheme is proposed in Section IV. Finally, numerical results and conclusions are provided in Sections V and VI, respectively.

II. System Model

We consider a set of base station and femto-cloud with some storage and computational resources, as is shown in Fig. 1 Each of the MTs has one application to execute. The MTs can decide whether or not to ask the femto-cloud for help. In case of offloading, the applications support data partitioning which means that they can all be divided into two parts of any size. One part will be processed locally and the other one at the femto-cloud. We do not consider any overhead brought by such division for the sake of simplicity. Our partitioning model is a upper bound to the real application to some degree. Thus the partitioning model can be formulated as follows [9]:

$$S_{0k} + S_{1k} = S_k, \tag{1}$$

where S_{0k} bits are executed at the MT side, while S_{1k} bits are offloaded to the femto-cloud through radio access point and S_k is the total amount of the application.

The time of processing a task at the k-th MT is modeled to be proportional to the amount of task. We denote the time for the k-th MT to finish a single digit to be τ_k, which depends jointly on the CPU rate and the complexity of the application. Similarly, the energy used to deal with the not offloaded part is modeled as $\xi_k S_{0k}$, in which ξ_k accounts for the energy consumption of one bit processed locally at the MT side. Note that ξ_k also depends both on the CPU rate of the MT processor and the type of the application.

When offloaded, the overall latency of the k-th mobile terminal offloading task consists of the following four parts,

$$\Delta_k = \Delta_k^t + \Delta_k^e + \Delta_k^r + T_B, \tag{2}$$

where Δ_k^t is the time duration of uplink transmission to send the state and input necessary for cloud execution. We assume the bits for the uplink transmission to be $\mu_{uk} S_{1k}$, where the coefficient μ_{uk} accounts for the compression or overhead of MT k due to the uplink communication. Using Shannon Formula to get the uplink rate, the uplink transmission time can be formulate as

$$\Delta_k^t = \frac{\mu_{uk} S_{1k}}{B_k \log_2(1 + \dfrac{P_k g_k}{N_0 B_k})}, \tag{3}$$

where B_k is the bandwidth allocated to the k-th MT. P_k denotes the uplink transmission power. g_k represents the channel gain between the k-th MT and the base station. N_0 is the power spectrum density of the thermal noise.

Δ_k^r is the latency to send the results of the computation back to the k-th MT. Usually the downlink transmission time is much smaller compared with the uplink time because the transmit power of the base station is much larger.

Δ_k^e denotes the duration for the femto-cloud to process S_{1k} CPU cycles. We assume the number of CPU cycles/ second of the virtual machine running the application of MT k over the femto-cloud to be f_k. Then

$$\Delta_k^e = \frac{S_{1k}}{f_k}. \tag{4}$$

The backhaul transmission time T_B is supposed to be a constant, which depends on the distance between the base station and the femto-cloud as well as the link connection (e.g., optical fiber).

III. Problem Formulation

Our ultimate goal is to find the optimal resource allocation strategy for the K MTs in order to minimize the overall energy spent by the MTs under latency constraints. Such terminal energy consumption consists of the energy spent in the radio transmission and the local processing. The degrees of freedom are radio and computational parameters and the amount of tasks offloaded to the femto-cloud. Concretely, we use FDM in our formulation and divide the total bandwidth into several sub-bands for multiple MTs to avoid unnecessary interference. The server at the femto-cloud side support multi-task processing, thus the MTs can share the overall computational capabilities. The MTs can determine whether or not to partition the application and how much for offloading. It is considered that the compute processes at the MTs and the femto-cloud can be performed in parallel, and the division and integration do not introduce any overhead for simplicity.

A. Optimization Problem

Our objective is to minimize a weighted sum of the energy consumption of each MT: $E = \sum\limits_{k=1}^{K} \beta_k E_k$, where coefficients $\beta_k > 0$ are weighting factors which can be adjusted to satisfy different battery capacities of the MTs. $E_k = P_k \Delta_k^t + \xi_k S_{0k}$ is the energy consumption of the k-th MT, which includes the energy for the radio transmission and the local terminal compute. P_k is the transmission power of the k-th MT. Note that the delay and energy consumption during the downlink transmission is negligible due to the fact that the transmit power of the base station is much larger than that of the MTs. So we focus on the uplink transmission.

Based on the previous analysis, we are able to formulate the multiuser resource allocation problem as follows:

$$\min_{S_{1k}, S_{0k}, P_k, B_k, f_k} \quad E = \sum_{k=1}^{K} \beta_k E_k \tag{5}$$

$$s.t. \quad i) S_{ok} + S_{1k} = S_k, \forall k \tag{6}$$

$$ii) \max\{\tau_k S_{0k}, \Delta_k\} \leq L_k, \forall k \tag{7}$$

$$iii) P_k \leq P_{kt}, \forall k \tag{8}$$

$$iv) \sum_{k=1}^{K} f_k \leq F, \tag{9}$$

$$v) \sum_{k=1}^{K} B_k \leq B. \tag{10}$$

The meaning of the 5 constraints are listed as follows:
i) The application of the k-th MT can be partitioned into two arbitrary parts: one processed at the local side, the other at the femto-cloud side;
ii) The computational processes at the MTs and the femto-cloud side can be performed in parallel. The executions of both sides have to be finished within the delay constraints of L_k;
iii) The transmit power of MT k has to meet the power budget limit P_{kt};
iv) The sum of the compute rate f_k allocated to each application can not exceed the total serving computational capacity F;
v) The sum of the sub-band B_k allocated to each MT can not exceed the total bandwidth B;

B. Problem Simplification

Unfortunately, the optimization problem is nonlinear programming, which is nonconvex in both the objective function and the constraints. However, there are some good properties in the problem, based on which we can simplify the overall resource allocation problem:

Thanks to constraint i), we can eliminate S_{0k} from the optimization variables by expressing it in terms of S_{1k} as $S_{0k} = S_k - S_{1k}$.

The problem can be treated as linear programming in accordance with the amount of offloading by each MT k S_{1k}, and S_{1k} appears only in the objective function and the first two constraints. Solving the linear programming problem, S_{1k} can be represented by P_k and B_k as a piecewise function (The detailed deduction is omitted because of the space limitation):

$$S_{1k} = \begin{cases} \max\{0, S_k - \frac{L_k}{\tau_k}\} & \text{if} \frac{P_k \mu_{uk}}{B_k \log_2(1+\dfrac{P_k g_k}{N_0 B_k})} > \xi_k \\ \dfrac{L_k}{\dfrac{\mu_{uk}}{B_k \log_2(1+\dfrac{P_k g_k}{N_0 B_k})} + \dfrac{1}{f_k}} & \text{else} \end{cases} \tag{11}$$

IV. TASK AND DELAY BASED RESOURCE ALLOCATION SCHEME

Although we make full use of the properties according to the single optimization variable to simplify the problem, it is still difficult to obtain a general closed-form solution because of the fraction form in the delay expression. We may use the exhaustive search to obtain the global optimal solution. However, due to the fact that exhaustive search is much too computational complex, especially when the number of the MTs is large, we turn to look for a heuristic algorithm to find sub-optimal solutions. Actually, as mentioned in the previous simplification, both S_{1k} and S_{0k} can be expressed by the resource allocated to the k-th MT. Thus we aim to find a resource allocation scheme to solve the problem. Note that we do not consider power control in this section and just use bandwidth as the radio resource for allocation to show the corresponding relationship between radio and compute resources. Based on the experience that energy sensitive device (with large β_k) should have access to more resources and that applications, which have the right delay requirement, should get the most resources, we assume the resource should be allocated proportionally to the ratio of $\beta_k|\tau_k S_k - |\tau_k S_k - L_k||$.

Algorithm 1 Task & Delay based Resource Allocation Scheme

```
for $k = 1$ to $K$ do
    $B_k = \frac{\beta_k|\tau_k S_k - |\tau_k S_k - L_k||}{\sum_{i=1}^{K} \beta_i|\tau_i S_i - |\tau_i S_i - L_i||} B$
    $f_k = \frac{\beta_k|\tau_k S_k - |\tau_k S_k - L_k||}{\sum_{i=1}^{K} \beta_i|\tau_i S_i - |\tau_i S_i - L_i||} F$
end for
if $\frac{P_k \mu_{uk}}{B_k \log_2(1+\frac{P_k g_k}{N_0 B_k})} > \xi_k$ then
    $S_{1k} = \max\{0, S_k - \frac{L_k}{\tau_k}\}$
else
    $S_{1k} = \frac{L_k}{\frac{\mu_{uk}}{B_k \log_2(1+\frac{P_k g_k}{N_0 B_k})} + \frac{1}{f_k}}$
end if
return $E = \sum_{k=1}^{K} \beta_k E_k$
```

TABLE I
SYSTEM PARAMETERS

Parameters	Value
Transmission Power ($P_k, \forall k$)	0.2 W
Total Bandwidth (B)	20 MHz
Cloud Compute Capacity (F)	5×10^7 cycles/second
$\xi_k, \forall k$	8.6×10^{-8} J/bit
$\tau_k, \forall k$	10^{-7} s/bit
$\mu_{uk}, \forall k$	1
$\beta_k, \forall k$	1
$\frac{g_k}{N_0}.\forall k$	8 Hz/W

According to the above resource allocation method and the previous conclusion that S_{1k} and S_{0k} can be expressed by P_k and B_k, we propose a heuristic scheme in Algorithm 1 to obtain the overall energy consumption at the MT side in the system.

V. NUMERICAL RESULTS

In this section, we apply our proposed resource allocation scheme and discuss numerical results. The parameters used are listed in Table I. Among them the MT energy consumption and delay parameters are from [14]. We set all the MTs to have the same channel conditions ($g_k, \forall k$ to be the same) and the same processing capabilities (the time to process 1 bit ($\tau_k, \forall k$) to be the same, and the energy consumed in the processing of 1 bit ($\xi_k, \forall k$) to be the same). Using the quantities of the N810 device in [14], we obtain ξ_k and τ_k of each MT in our system. Note that we assume the speed of CPU at the femto-cloud F to be 4 times faster than that of MTs ($\frac{1}{\tau_k}$) (This can be achieved by allocating 5 processors at the same time). Without loss of generality, we fix the parameters of MT 1, and observe the trends of the overall energy consumption by varying the delay requirements and the application size of the other MTs. We set the application size of MT 1 to be $S_1 = 5\,\mathrm{Mb}$ as in [9] and $S_k (k \neq 1)$ to be integral multiples of S_1.

Fig. 2 and Fig. 3 show the bandwidth resource and the computational resources allocated to MT 2 with 2 MTs in all in the system, respectively. We apply both our task and delay based resource allocation scheme and exhaustive search to analyze the relationship between the delay constraints and resource allocation. We vary the application amount of the MT 2 to get 2 groups of curves. Here we set $L_1 = 0.4s$ at the beginning. Generally, the allocated computational resources for MT 2 in Fig. 3 fluctuate similarly with the allocated bandwidth resource in Fig. 2. This can be intuitively explained as more offloaded tasks through the wireless link typically require more compute resources to process at the femto-cloud. Concretely, when the delay constraint of MT 2 L_2 is smaller than the time required to finish all the task at the mobile terminal side ($\tau_2 S_2$), then the allocated radio and computational resources would increase in accordance with the relaxation of delay requirements. In this case, the MT has to ask the cloud for help to meet the delay requirements. Looser constraints in delay may allow the MT to offload more tasks to the cloud to save energy at the MT side, and more tasks offloaded under such circumstance typically requires more resources for transmission and computation. However, when L_2 is larger than $\tau_2 S_2$, addition on L_2 will lead to a decrease in the demand of both radio and compute resources. This is due to relatively loose delay constraint of MT 2. MT 2 can meet the delay requirement without using too much resources thus can leave more resources for other MTs in the system for the minimization of the total energy consumption. Moreover, the trends of our proposed scheme and exhaustive algorithm fits really well in Fig. 2 and Fig. 3, which validates our previous deduction.

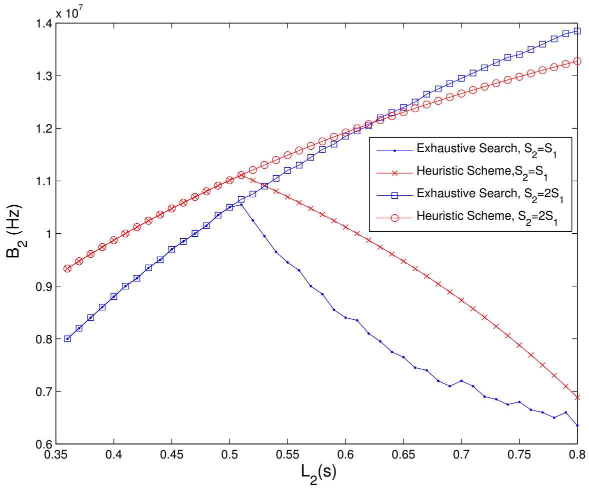

Fig. 2. Bandwidth Allocation for MT 2 with two MTs in the system.

TABLE II
2 ALGORITHMS FOR 3 MTS IN THE SYSTEM

E/J L_2/s L_1/s	0.55	0.65	0.75
0.55	(0.5947,0.5973)	(0.5572,0.5581)	(0.5272,0.5460)
0.65	(0.5275,0.5509)	(0.5060,0.5134)	(0.4731,0.5021)
0.75	(0.4731,0.4967)	(0.4464,0.4558)	(0.4125,0.4405)

The impact of delay requirements to the terminal energy consumption is illustrated in Fig. 4. The total energy consumption reduces significantly with the increase of the delay constraints, which shows that mobile cloud computing can achieve respectable energy saving for MTs in sacrifice of the processing delay. Processing two applications with the amount of S_{1k} at the MT side requires 0.86 W energy dissipation. However, through partial mobile cloud offloading the required energy is less than 0.2 W, when L_2 is 0.8 s, which could lead to a over 75% reduction of energy required to process the same job. Also, from Fig. 4, we can see the performance of our heuristic resource allocation scheme performs really well, especially when the delay constraint is relatively tight compared with the duration to process the task at the MT side.

Table II compares the performance of our proposed resource allocation scheme with that of exhaustive algorithm for 3 MTs in the system. We let the total application amount of the 3 MTs S_k all be 5 MHz. With respect to the arrays in the

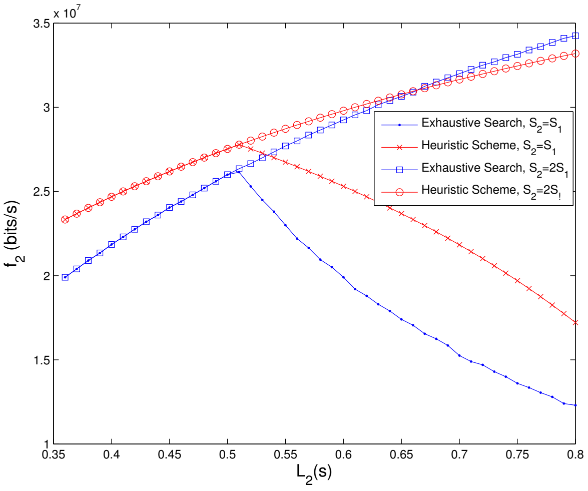

Fig. 3. Computational Resource Allocation for MT 2 with two MTs in the system.

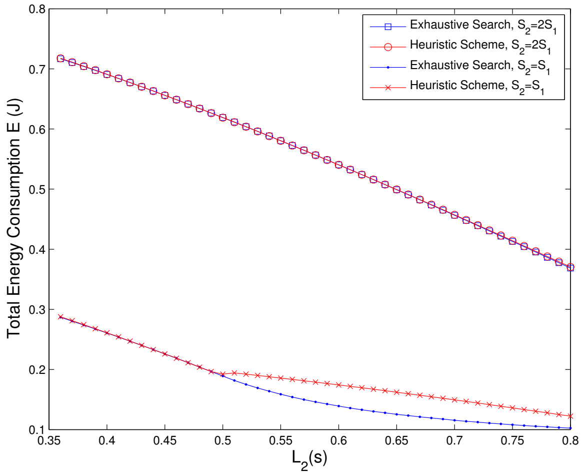

Fig. 4. Total Energy Consumption with 2 MTs in the system.

table, the first number in each array is the energy consumption result by exhaustive search, while the second is the result of our heuristic resource allocation scheme. We fail to list too many data due to the limited space. Typically, our proposed scheme achieves 40% average total energy reduction compared with no offloading and 10% worse than the results of the exhaustive search for the particular case of 3 MTs. However, the proposed scheme only requires $O(K)$ complexity and possesses practical feasibility while the raw optimization problem is non-convex and the exhaustive algorithm is much too computationally complex.

VI. Conclusion

In this paper, we have addressed the issue of resource allocation in mobile cloud computing targeting at minimizing the terminal energy consumption, while satisfying the delay requirements. We formulate the scenario as a non-linear constraint optimization problem which is intractable. The task and delay based resource allocation scheme is proposed in which the k-th MT having delay constraint L_k closer to $\tau_k S_k$ obtains more resources. The proposed resource allocation scheme is easily implementable, which requires only $O(K)$ complexity, and numerical results show it has good performance. As future work, one can further consider the resource allocation problem in the multi-BS and multi-femto-cloud scenarios.

Acknowledgment

This work is sponsored in part by the National Basic Research Program of China (973 Program: 2012CB316001), the National Science Foundation of China (NSFC) under grant No. 61201191, No. 61401250 and No. 61321061, and Intel Collaborative Research Institute for Mobile Networking and Computing.

References

[1] D. Dev and K. Baishnab, “A review and research towards mobile cloud computing,” in *2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, Apr 2014, pp. 252–256.

[2] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, “Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 337–368, Jan 2014.

[3] “IMT-2020 (5G) promotion group 5G vision and requirements,” IMT-2020 (5G) Promotion Group, Tech. Rep., May 2014.

[4] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep 2013.

[5] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct 2009.

[6] B. Chun and P. Maniatis, “Augmented smartphone applications through clone cloud execution,” in *Conference on Hot Topics in Operating Systems*, 2009.

[7] J. Liu, T. Zhao, S. Zhou, Y. Cheng, and Z. Niu, “CONCERT: a cloud-based architecture for next-generation cellular systems,” *IEEE Wireless Commun. Mag.*, vol. 21, no. 6, pp. 14–22, Dec 2014.

[8] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, “Communicating while computing: Distributed mobile cloud computing over 5g heterogeneous networks,” *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov 2014.

[9] O. Munoz-Medina, A. Pascual-Iserte, and J. Vidal, “Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading,” *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, pp. 1–1, 2014.

[10] X. Guo, R. Singh, Z. Niu, and P. R. Kumar, “Joint delay-efficiency optimal task assignment in cloud computing networks via index policies,” submitted to Globecom 2015.

[11] S. Sardellitti, S. Barbarossa, and G. Scutari, “Distributed mobile cloud computing: Joint optimization of radio and computational resources,” in *2014 Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 1505–1510.

[12] S. Sardellitti, G. Scutari, and S. Barbarossa, “Distributed joint optimization of radio and computational resources for mobile cloud computing,” in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, Oct 2014, pp. 211–216.

[13] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, “Joint allocation of computation and communication resources in multiuser mobile cloud computing,” in *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jun 2013, pp. 26–30.

[14] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 4–4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863107