On the Performance Evaluation of a Novel Offloading-Based Energy Conservation Mechanism for Wireless Devices

Constandinos X. Mavromoustakis[1](✉), Andreas Andreou[2],
George Mastorakis[3], Athina Bourdena[3],
Jordi Mongay Batalla[4], and Ciprian Dobre[5]

[1] Department of Computer Science, University of Nicosia,
46 Makedonitissas Avenue, P.O. Box 24005
1700 Nicosia, Cyprus
mavromoustakis.c@unic.ac.cy

[2] Faculty of Computer Science and Technology, University of Cambridge,
William Gates Building, 15 JJ Thomson Ave,
Cambridge CB3 0FD, UK
aa773@cam.ac.uk

[3] Department of Informatics Engineering, Technological Educational Institute
of Crete, Estavromenos, 71500 Heraklion, Crete, Greece
gmastorakis@staff.teicrete.gr, bourdena@pasiphae.eu

[4] National Institute of Telecommunications,
Szachowa Str. 1, 04-894 Warsaw, Poland
jordim@interfree.it

[5] Faculty of Automatic Control and Computers, University Politehnica
of Bucharest, 313, Splaiul Independentei, 060042 Bucharest, Romania
ciprian.dobre@cs.pub.ro

Abstract. Mobile Cloud computing paradigm includes plenty of critical challenges that have to be addressed for allowing application execution on remote terminals/servers. An integral part of mobile cloud computing reliable service provision is the establishment of a methodology that will guarantee the efficient execution of applications in an energy-efficient way. This work elaborates on the evaluation of a framework, which utilizes a cooperative partial process execution offloading scheme, aiming at offering energy conservation. The scheme uses a dynamic scheduling methodology in order to guarantee that no intermittent execution will occur on mobile devices. In addition, this work proposes a partial scheduling offloading algorithm for failure-aware resource allocation in an energy-efficient manner, associating temporal execution-oriented metrics. The proposed framework is thoroughly evaluated through event driven simulation experiments, towards defining the efficiency of the proposed offloading policy in contrast to the energy consumption of wireless devices, as well as for the reliability degree offered.

Keywords: Mobile cloud · Offloading methodology · Reliability · Dynamic resource migration · Dependable mobile computing · Temporal execution-oriented metrics

R. Agüero et al. (Eds.): MONAMI 2014, LNICST 141, pp. 179–191, 2015.
DOI: 10.1007/978-3-319-16292-8_13

1 Introduction

The recent popularity of smartphones and tablets created a fertile ground of new application paradigms for mobile wireless communications. This growth has also been fueled by the numerous applications that run per wireless device, creating the need for a reliable and high performance mobile computing application environment. All wireless devices are prone to energy constraints that very often impair the reliability of the correct execution for the running application on the device. In this context, this paper proposes a mechanism, which takes into consideration the energy consumption of the wireless devices, while running an application. This mechanism exploits an offloading technique, in case where the resources need to be partially or in totally outsourced. This mechanism is utilized by end-terminals (server rack) and/or run on mobile devices, where resources are redundant. The offloading methodology is applied as a part of the application initiation (run start-up), in order to minimize the GPU/CPU efforts and the energy consumption of the mobile device that is running out of resources.

On the contrary with the utility computing, mobile cloud services should only be offered in a synchronous mode [1]. To this end, different parameterized metrics of both wireless devices and availability of offloading by other terminals and/or servers [1] should be taken into consideration. Traditional cloud computing models are considered as ‘low offered’ throughput models [2] and [3], more expensive, significantly offering low Quality of Service (QoS) or Quality of Experience (QoE) to the end-recipients (i.e. wireless end users). The limited processing power and bounded capacity availability of wireless devices, aggravate the execution and negatively affect the reliability offered to the user’s mobile device, by causing capacity-oriented failures and intermittent execution. When there is lack of available resources (processing and/or memory-oriented), the wireless device may refer to a mobile cloud infrastructure as in [1] to enable precise execution through the resource/task migration mechanism. Within this context, a mechanism for ensuring that there is adequate processing power for executing the application of a wireless device and at the same time allowing the evaluation of the power consumption through the consideration of an energy-efficient application offloading, has not yet been extensively examined [4].

In this direction, a dynamic scheduling scheme for offloading resources from a wireless device to another mobile device is investigated in this work, to enable the improvement of the response quality and system’s throughput, according to the partial or in-total execution, as well as the request deadline. The ultimate target of this work is to prevent long application execution requests, which will result in greater energy consumption for each wireless device and enable efficient manipulation of local (device), as well as cloud resources. Within this context, the proposed scheme minimizes the utilization of local (device’s) resources to complete (GPU, CPU, RAM, battery consumption) and offers at the same time extensibility in the wireless devices lifetimes. To this end, this work presents an attempt to reduce the computational load of each wireless device so as to extend the lifetime of the battery. In addition, this study considers a partitionable parallel processing wireless system, where resources are partitioned and handled by a subsystem [1] that estimates and handles the resource offloading process. A certain algorithm is being proposed for the offloading process, in order to dynamically define the optimal resource manipulation in an energy-efficient manner.

The structure flow of this paper is as follows: related work is described in Sect. 2, by focusing in particular on the existing research approaches and the resource offloading/migration scheduling policies. The proposed offloading scheme and the associated mechanisms for minimizing the energy consumption and maximizing the lifetime of each wireless device is presented in Sect. 3. The proposed scheme is based on the available resources of each mobile device, as well as on temporal device characteristics and server-based parameters along with communication-oriented diversities for establishing and maintaining the efficient resource manipulation onto each mobile device in an energy-efficient manner. In turn, Sect. 4 presents the results obtained, by conducting simulation experiments for the performance evaluation, focusing on the behavioral characteristics of the scheme along with the system's response and the energy consumption achieved. Finally, Sect. 5 elaborates on the research findings and discusses the potential future directions.

2 Related Work and Research Motivation

It is undoubtedly true that over the past few decades, several research efforts have been devoted to device-to-device or Machine-to-Machine communication networks, ranging from physical layer communications to communication-level networking challenges. Wireless devices can exchange resources on the move and can become data "Prosumers", by producing a great amount of content [2], while at the same time as content providers devices can consume the content. The research efforts for achieving energy efficiency on-the-move for wireless devices, trades-off the QoS [3] offered, by significantly reducing the performance with energy-hungry applications such as video, interactive gaming, etc. While energy-hungry applications are widely utilized by wireless devices, the explicit lifetime of devices should be extended, towards hosting and running the application in device's entire lifetime. In order to achieve resource management in wireless devices within the context of the cloud paradigm, efficient allocation of processor power, memory capacity resources and network bandwidth should be considered. To this end, resource management should allocate resources of the users and their respected applications, on a cloud-based infrastructure, in order to migrate some of their resources on the cloud [5]. Wireless devices are expected to operate under the predefined QoS requirements as set by the users and/or the applications' requirements. Resource management at cloud scale requires a rich set of resource and task management schemes that are capable to efficiently manage the provision of QoS requirements, whilst maintaining total system efficiency. However, the energy-efficiency is the greatest challenge for this optimization problem [3], along with the offered scalability in the context of performance evaluation and measurement. Different dynamic resource allocation policies targeting the improvement of the application execution performance and the efficient utilization of resources have been explored in [6]. Other research approaches related to the performance of dynamic resource allocation policies, had led to the development of a computing framework [7], which considers the countable and measureable parameters that will affect task allocation. Authors in [8] address this problem, by using the CloneCloud approach [9] of a smart and efficient architecture for the seamless use of ambient computation to augment mobile device applications, off-loading the right portion of their execution onto device clones, operating in a computational cloud. Authors in [9] statically partition service tasks and resources between client and server portions, whereas in a later stage the service is reassembled on the mobile device. The spine of the proposal in [9] is based on a cloud-augmented execution, using a cloned VM image as a powerful virtual device. This approach has many vulnerabilities as it has to take into consideration the resources of each cloud rack, depending on the expected workload and execution conditions (CPU speed, network performance). In addition, a computation offloading scheme is proposed in [10] to be used in cloud computing environments, towards minimizing the energy consumption of a mobile device, in order to be able to run certain/specified and under constrains application. Energy consumption has also been studied in [11], in order to enable computation offloading, by using a combination of 3G and Wi-Fi infrastructures. However, these evaluations do not maximize the benefits of offloading, as they are considered as high latency offloading processes and require low amount of information to be offloaded. Cloud computing is currently impaired by the latency experienced during the data offloading through a Wide Area Network (WAN). Authors in [1] and [11], elaborate on issues, where the devices carried by human beings are always considered as delay sensitive. The variability of this delay in turn impairs the QoS/QoE of the end-users.

Authors in [12] address the resource processing poverty for ‘hungry’ applications that require processing resources in order to run on a handheld device, while authors in [13] provide a resource manipulation scheme as a solution based on the failure rates of cloud servers in a large-scales datacenters. However, these criteria do not include servers’ communications diversities in the communication process with mobile users’ claims, as well as the available processing resources, the utilization of the device’s memory, the remaining energy and the available capacity with the communication of each of the device with the closest –in terms of latency- cloud terminal. Research approaches in [14] and [15] have proposed different analytical models to address offloading computation and elaborate on offloading to offer energy conservation.

Within this context, this paper is making progress beyond the current state-of-the-art, by proposing an offloading resource mechanism, which is used in collaboration with an energy-efficient model proposed. The scheme uses an offloading methodology in order to guarantee that no intermittent execution will occur on mobile devices, whereas the application explicit runtime will meet the required deadlines to fulfil the QoS requirements. This paper also elaborates on the development of an offloading scenario, in which the scheduling policy for guaranteeing the efficiency in the execution of mobile users’ tasks/applications can be achieved in an energy-efficient manner. The proposed framework is thoroughly evaluated through event driven simulation experiments in order to define the efficiency of the proposed offloading policy, in contrast to the energy consumption of the wireless devices, as well as for the reliability degree offered.

3 Offloading-Based Energy Conservation Mechanism for Wireless Devices

3.1 Device 'Live' Offloading Mechanism

As explored in [14], the offloading may be beneficial if large amounts of computation resources are required with the minimization of the communication delays in the end-to-end relay path. However, there are numerous applications and interactive actions of the mobile users that are draining the energy resources due to the continuous and full utilization of all computational resources of mobile devices. This work considers the scenario of the mobile user case, who uses high processing and computational power on the mobile application. The proposed scheme allows the computational processes that cannot be performed (executed/run) on the mobile device, to be offloaded onto cloud, or onto other neighboring mobile devices with redundancy in availability of resources (as opportunistically formed cloud). Applications that do not consume significant amounts of energy due to high processing power requirements (i.e. voice recognition, on-device gaming etc.) can be hosted and run on the mobile device of the end-user.

Authors in [16] performed a detailed analysis in real-time of the energy consumption of a smartphone, based on measurements of a physical device for different applications. It was shown that there are specific energy consuming components within the smartphone device that are contributing to the overall "short-run" of the device's power, with primary factor the high utilization of the processing power simultaneously with high transmission powers, exposed by the wireless interfaces of each device. Having considered the issues above, the proposed scheme allows low-consumption applications that run on the mobile device to be offloaded, if this may positively affect the energy consumption, according to the proposed methodology of the scheme and the algorithm designed that is shown in the next section. Due to the heterogeneity in the hardware of both mobile devices and the servers on the cloud that the resources will be potentially (based on the proposed scheme) offloaded, the proposed framework encompasses the execution environment volatility and considers the cloud servers' response time, in order to a-priori compare them and select the appropriate server, according to the best fit-case.

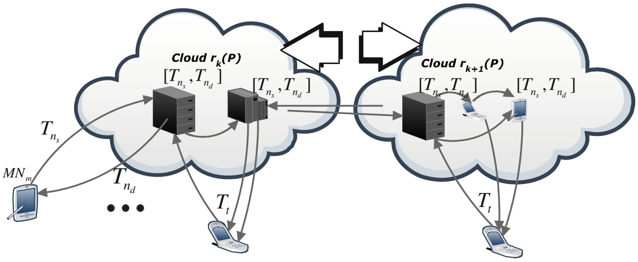

Fig. 1. Cloud configuration when a mobile device has no remaining resources to run an application, offloading resources on the cloud to achieve the best effort processing on-device power.

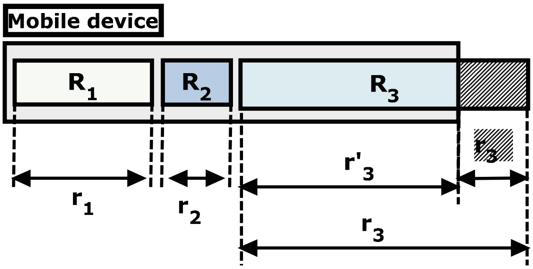

Fig. 2. Resource partitioning onto mobile device.

More specifically, this work considers the network-oriented parameters for bandwidth provisioning to achieve an acceptable resource offloading downtime (e.g., $\delta \leq 1.6\,\mathrm{s}$ as the experimental process validates in [1]). To this end, from the network perspective, the modelled parameters can be expressed, for an offloading process *O for an* executable resource task O_{a_j}, as a 5-tuple given by:

$O_{a_j}(MN) = <n_s, T_{n_s}, T_{n_d}, BW, T_t>$ where $\mathrm{n_s}$ is the devices or cloud terminals that the a_j from *MN* device will be offloaded, T_{n_s} is the source location best effort access time, T_{n_d} is the destination's device or cloud's location best effort access time (time to access the resource) from the source, BW is the required connection bandwidth, T_t is the connection holding duration for the a_j executable resource task. In essence the work done in [1] considers the resource transfer time, by taking into account the volume of traffic that should be transferred to the destination node. The total data volume that will be transferred, if the request meets the BW criteria can be provided by BW × T_t. In this work, the typical values ranges that were utilized in our experimental processes were $1\,\mathrm{MB} \leq BW \leq 15\,\mathrm{MB}$ and T_t= 2 s + $t_x, s = T_{n_s} + T_{n_d}$, where t_x is the time to process x-partitionable parts that are processed during the offloading process. Every executable resource task may have *x-partitions*, which in this work are considered as t_x partitioning parts/tasks where $1 \leq t_x \leq z^*P$, where z is the number of different devices that the resource can be offloaded. Therefore, the number of tasks per executable resource task is limited to the number of terminals in the system. An executable resource can be shared and partitioned to x_1, x_2, … x_n and can be simultaneously processed with r sequential partitions, where $0 \leq r < z^*P$, if and only if the following relation holds:

$$r + \sum_{i=1}^{n} p(x_i) \leq z * P \tag{1}$$

where $p(x)$ represents the number of cloud terminals (mobile and statically located) that are needed to host the a_j, and P is the number of terminals on the cloud that will be hosting the offloaded resource. The scheduling strategy that was used is based on the Largest First Served/ Shortest Sequential Resource First Served and the service notations of [1] with a-priori knowledge of the $[T_n, T_{n_d}]$ service durations, as shown in Fig. 1.

3.2 Energy-Consumption Model Using Temporal Capacity Measurements

As the consumed power varies with traffic and depends on the variations of the signal characteristics, as well as on the traffic-aware measurements [17], it is desirable to minimize the amount of power consumption according to the resources that cannot ‘run’ on the mobile device or devices. To this end, the proposed scheme in this paper makes a progress beyond the current state-of-the-art, by elaborating on the association of the measurements of the partitionable tasks for two distinct cases: when resources can run on the device, but in order to achieve energy conservation they may be offloaded to a cloud or any other peer-neighboring device (so that the device that needs to run may potentially conserve energy); and the case that the device or devices cannot run the resources (as in Fig. 2) as the processing and memory requirements cannot support this execution. Thus, the measurable energy consumption can be evaluated according to the:

$$E_{r(a_j)} = E_c(a_j) \cdot \frac{C}{S_{a_j}} \forall C \in O_{a_j}, T_{n_s} < T_t, T_{n_d} < T_t \tag{2}$$

In Eq. 2, parameter C is the number of instructions that can be processed within T_t, S_{a_j} is the processing time of the terminal that can host the offloaded executable task, provided that $T_{n_s} < T_t, T_{n_d} < T_t$ is satisfied, and $E_c(a_j)$ is the relative energy consumption in mW and it is expressed by:

$$E_c(r_i) = \frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c, \forall C \in O_{a_j} \tag{3}$$

where S_c is the server’s processing instruction speed for the computation resources, $Cost_c$ the resources’ processing instruction cost for the computation resources, and W_c energy consumption of the device or server in mW.

Each mobile device examines if all neighboring 2-hops devices (via lookup table) can provide information about their offloading capabilities without affecting their energy status (thus without draining their energy to run other devices’ resources). In addition, the closest cloud rack is considered if the relations exposed in (4) and (5) are not satisfied. Hence, for the neighboring devices within *2-hops vicinity coverage* (based on the maximum signal strength and data rate model [1]) should stand:

$$\frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c|^{r_i} > \frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c|^{1,2..N} \tag{4}$$

$$W_{r_i} > W_c \forall 1, 2, 3, \ldots N \tag{5}$$

The energy consumption of each device should satisfy the Eqs. (4)–(5) for each of the resources (executable processes) running onto the device MN_{m-1} hosting the r_i, where $m-1$ represents the remaining devices from total m devices. Otherwise, the r_i with the maximum energy consumption is running in a partitionable manner to minimize the energy consumed by other peer-devices. These actions are shown in the steps of the proposed algorithm in table I.

The resource allocation will take place, towards responding to the performance requirements as in [1]. A significant measure in the system is the availability of memory and the processing power of the mobile-cloud devices, as well as the server-based terminals. The processing power metric is designed and used to measure the processing losses for the terminals that the r_i will be offloaded, as in Eq. 4, where is an application and T_k^j is the number of terminals in forming the cloud (mobile and static) rack that are hosting application, and $T_{a_j}(r)$ is the number of mobile terminals hosting process of the application across all different cloud-terminals (racks).

$$C_{a_j} = \frac{T_k^j}{\sum_k T_{a_j}(r)} \forall \min(E_c(r_i)) \in 1, 2, ..N \tag{6}$$

Equation 4 shows that if there is minimal loss in the capacity utilization i.e. $C_{a_j} \cong 1$ then the sequence of racks $T_{a_j}(r)$ are optimally utilized. The latter is shown through the conducted simulation experiments in the next section. The dynamic resource migration algorithm is shown in Table 1 with the basic steps for obtaining an efficient execution for a partitionable resource that cannot be handled by the existing cloud rack and therefore the migration policy is used to ensure that it will be continuing the execution. The continuation is based on the migrated policy of the partitionable processes that are split, in order to be handled by other cloud rack terminals and thus omit any potential failures. The entire scheme is shown in Table 1, with all the primary steps for offloading the resources onto either MN_{m-1} neighbouring nodes or to server racks (as in [1] and [15]) based on the delay and resources temporal criteria.

Table 1 Dynamic Resource-based offloading scheme.

```
Inputs: $MN_m$, Location($[T_{n_s}, T_{n_d}]$), resources $r_1, r_2, r_3, ..r_i \forall MN_m$
for all Cloud devices that stands $r_1, r_2, r_3, ..r_i$ find the $r_i$ that can be offloaded to run
    onto another device
for all $MN_{m-1}$ do{
  while ($T_t$== TRUE) {
      while ($1 \le t_x \le z*P$)
          search for $MN_{m-1}$ device that satisfies
          $\frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c\Big|^{r_i} > \frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c\Big|^{1,2..N}, W_{r_i} > W_c \forall 1,2,3,...N$
    offload ($r_i, MN_{k(i)}$) //to $MN_{k(i)}$ to execute resource (i)
      end while
     end while ($C_{a_j} \forall \min(E_c(r_i)) \in 1,2,..N$)&&($E_c(r_i) == satisfied, \forall C \in O_{a_j}$))
   end for
end for
```

4 Performance Evaluation Analysis, Experimental Results and Discussion

Performance evaluation results encompass comparisons with other existing schemes for offered throughput and reliability degree, in contrast to the energy conservation efficiency. The mobility model used in this work is based on the probabilistic Fraction Brownian Motion (FBM) adopted in [17], where nodes are moving, according to certain probabilities, location and time. Towards implementing such scenario, a common look-up application service for resource execution offloading is set onto each one of the mobile nodes MN_m. Topology of a 'grid' based network [1] was modeled, where each node can directly communicate with other nodes, if the area situated is in the same (3 × 3 center) rectangular area of the node. For the simulation of the proposed scenario, the varying parameters described in previous section were used, by using a two-dimensional network, consisting of nodes that vary between 10–180 (i.e. terminal mobile nodes) located in measured area, as well as five cloud terminals statically located on a rack. All measurements were performed using WLAN varying with different 802.11X specifications. During simulation the transfer durations are pre-estimated or estimated, according to the relay path between the source (node to offload resources) and the destination (node to host the executable resources).

The dynamic offloading scheme and the instant that takes place, is an important measure to estimate as well as the effectiveness of the proposed framework and the impact on the system. To this end, the total failed requests among nodes with the number of requests and with the number of mobile devices participating in the evaluated area, are shown in Fig. 3. Towards examining the impact of the different capacities, several sets of experiments were conducted using the presented resource

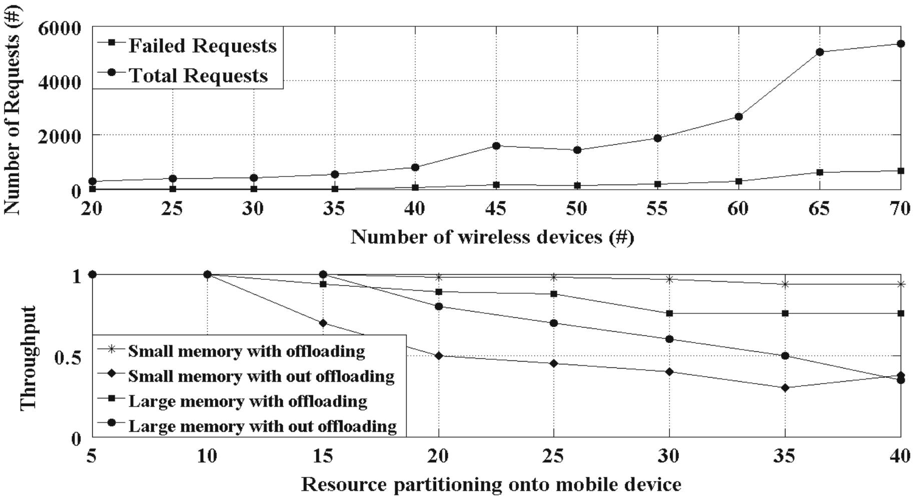

Fig. 3. Number of requests with the number of mobile devices participating in the evaluated area and throughput response with the mean number of executable resources that are partitioned per mobile device.

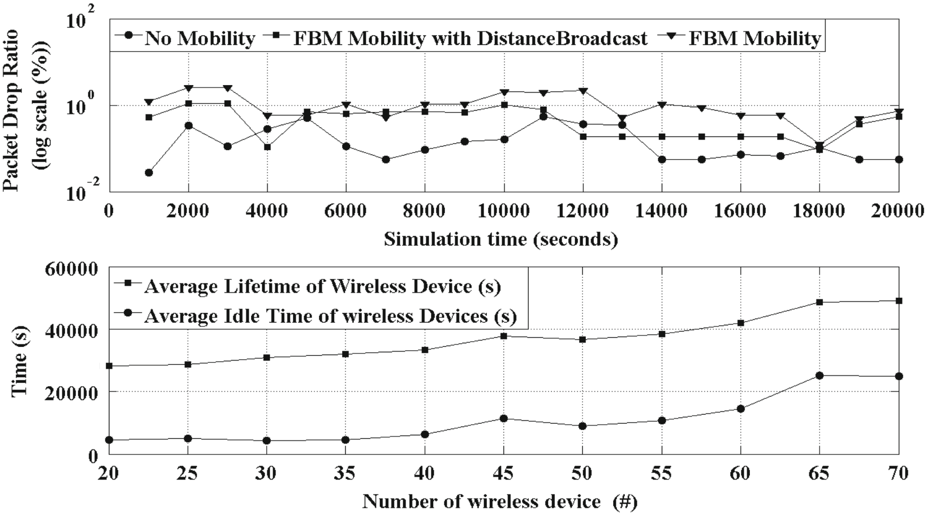

Fig. 4. Packet drop ratio of the proposed scheme for different mobility variations and no-mobility model over time and average lifetime for both active and idle time with the number of mobile devices.

offloading scheme. Large memory resources are executable resources/processes that are between 500 MB–1 GB, whereas small memory resources are executable processes that are hosting capacities between the range of 10–400 MB. The throughput response in contrast to the mean number of executable resources that are partitioned per mobile device is also shown in Fig. 3. The throughput response offered by the proposed scenario is greater for large files that are offloaded in partitionable parts onto other terminals on the cloud. Moreover, when utilizing the proposed framework, the small memory capacity requirements offer almost greater than 90 % for throughput response measurements. The packet drop ratio of the proposed scheme for different mobility variations and without mobility over time is shown in Fig. 4. It is important to emphasize that the proposed scheme scales well in the presence of FBM and even better when the FBM with distance broadcasting is applied. In addition, Fig. 4 presents the average lifetime for both active and idle time with the number of mobile devices.

Assuming that game-playing users are participating through their mobile devices utilizing game playing actions. These game playing actions require resources in GPU/CPU-level. These resource-constraints can be used as a measure to evaluate the efficiency offered by the proposed scheme under ‘heavy limitations’ and ‘strict latencies’. Hence, the lifetime of each of the mobile device is an important metric for the evaluation of the overall performance of the scheme and the impact on nodes lifetime. Measurements in Fig. 5 were extracted for the total number of 180 mobile terminals that are configured to host interactive gaming applications, using Wi-Fi/WLAN access technology. The overall energy consumption for each mobile device for three different schemes in the evaluated area (for the interactive game playing draining resources) is shown in Fig. 5. The proposed scheme shows that it outperforms the scheme proposed in [1], as well as the scheme in [8] for the Wi-Fi/WLAN connectivity configuration.

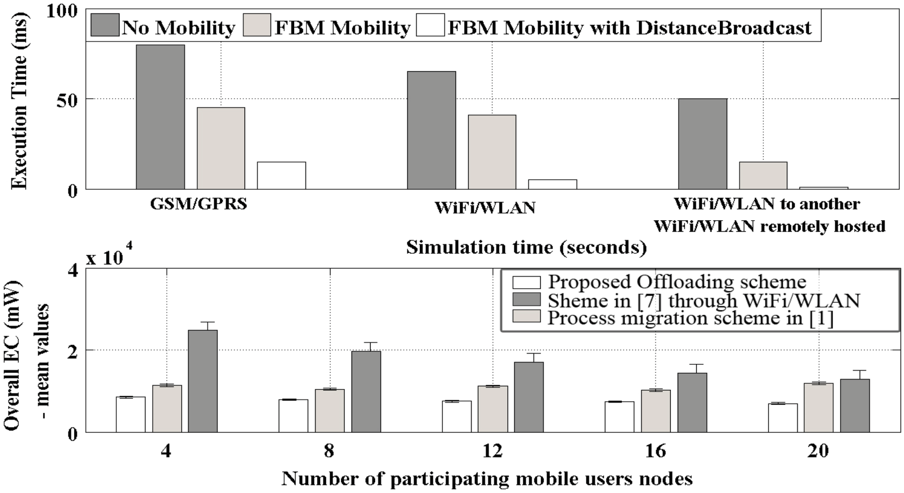

Fig. 5. Overall energy consumption for each mobile device for three different schemes in the evaluated area and execution time during simulation for nodes with different mobility patterns for three different schemes.

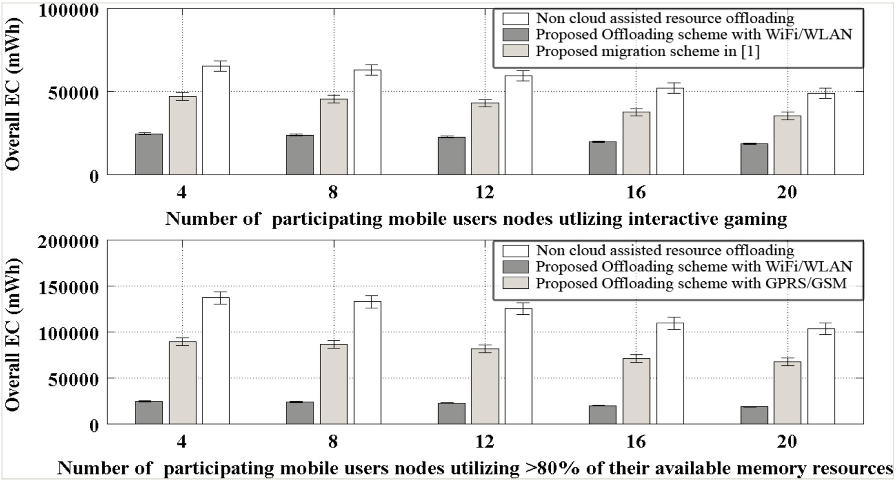

Fig. 6. Energy Consumption (EC) with the number of mobile users participating during an interactive game and Energy Consumption (EC) with the number of mobile users utilizing greater than 80 % of their resources.

When resources are offloaded, a critical parameter is the execution time, while nodes are moving from one location to another. In Fig. 5 the execution time during simulation for mobile nodes with different mobility patterns is also evaluated, for GSM/GPRS, Wi-Fi/WLAN and for communication within a certain Wi-Fi/WLAN to another Wi-Fi/WLAN remotely hosted. The latter scenario - from a Wi-Fi/WLAN to another Wi-Fi/WLAN- shows to exhibit significant reduction in terms of the execution time, whereas it hosts the minimum execution time through the FBM with distance broadcast mobility pattern.

The Energy Consumption (EC) with the number of mobile users participating during an interactive game (requirements in GPU/CPU) is shown in Fig. 6. During the interactive game-playing process, the processing requirements of each device dramatically increase. This results to the need for some devices to offload processing power into cloud terminals. In this regards, Fig. 6 presents the evaluation for the energy consumed (EC) for three schemes including a non-assisted cloud. Measurements were extracted for 150 mobile terminals that are configured to host interactive gaming applications. The proposed scheme outperforms the other compared schemes, with the associated EC to be kept in relatively low levels. In turn, Fig. 6 shows the EC with the number of mobile users utilizing greater than 80 % of their memory available resources for three different schemes. It is important to denote that the proposed scheme with the Wi-Fi/WLAN configuration enables lower EC than the other evaluated schemes, including the absence of any assistance through cloud. Devices that are utilizing greater than the 80 % of their computational resources are the best candidates to offload. Figure 6 shows that the EC is significantly minimized, by using the proposed scheme that utilizes the Wi-Fi/WLAN configuration, whereas it behaves almost the same with small number of mobile nodes, as well as with greater number of mobile nodes with lack of resources in the described scenario.

5 Conclusions

In this work, a novel task outsourcing mechanism using the Mobile Cloud paradigm is presented in contrast to the Energy consumption of wireless devices. The proposed scheme encompasses a cooperative partial process offloading execution scheme, aiming at offering energy conservation. In order to allow Energy conservation, partitionable resources can be offloaded using a latency-based scheduling scheme as well as by utilizing the state characteristics of each device (i.e. allowed execution duration). The offloading mechanism provides efficient cloud-oriented resources' exploitation and reliable task execution offered to the mobile end-recipients. The proposed offloading scheme is thoroughly evaluated through simulation experiments, in order to validate the efficiency of the offloading policy in contrast to the energy consumption of wireless devices, as well as for the reliability degree offered by the scheme. Future streams in our on-going research include the enhancement of an opportunistically formed federated mobile cloud, which will allow interactive game playing and exchanging of resources with strict resource constraints and streaming characteristics (delay-sensitive resource sharing) in a MP2P manner, on-the-move.

Acknowledgment. The work presented in this paper is co-funded by the European Union, Eurostars Programme, under the project 8111, DELTA "Network-Aware Delivery Clouds for User Centric Media Events".

References

1. Mousicou, P., Mavromoustakis, C.X., Bourdena, A., Mastorakis, G., Pallis, E.: Performance evaluation of Dynamic Cloud Resource Migration based on Temporal and Capacity-aware policy for Efficient Resource sharing. In: 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2013), Barcelona, Spain (2013)
2. Abolfazli, S., Sanaei, Z., Ahmed, E., Gani, A., Buyya, R.: Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges. IEEE Commun. Surv. Tutorials **16**(1), 337–368 (2013)
3. Dimitriou, C., Mavromoustakis, C.X., Mastorakis, G., Pallis, E.: On the performance response of delay-bounded energy-aware bandwidth allocation scheme in wireless networks. In: IEEE 2013 International Workshop on Immersive & Interactive Multimedia Communications over the Future Internet, organized in conjunction with IEEE International Communications Conference (ICC 2013), pp. 641–646, Budapest, Hungary (2013)
4. Miettinen, A.P., Nurminen, J.K.: Energy efficiency of mobile clients in cloud computing. In: 2nd USENIX Conference on Hot Topics in Cloud Computing. USENIX Association (2010)
5. Salehi, M.A., Javadi, B., Buyya, R.: Resource provisioning based on preempting virtual machines in distributed systems. Concurrency Comput. Pract. Exp. **26**(2), 412–433 (2013)
6. Slegers, J., Mitriani, I., Thomas, N.: Evaluating the optimal server allocation policy for clusters with on/off sources. J. Perform. Eval. **66**(8), 453–467 (2009)
7. Warneke, D., Kao, O.: Nephele: Efficient parallel data processing in the cloud. In: 2nd Workshop Many-Task Computing on Grids and Supercomputers, ACM, Portland, OR, USA (2009)
8. Chun, B., Ihm, S., Maniatis, P., Naik, M., Patti, A.: Clonecloud: Elastic execution between mobile device and cloud. In: 6th Conference on Computer systems of EuroSys (2011)
9. Chun, B.G., Maniatis, P.: Augmented smartphone applications through clone cloud execution. In: HotOS (2009)
10. Barbera, M.V., Kosta, S., Mei, A., Stefa, J.: To offload or not to offload? The bandwidth and energy costs of mobile cloud computing. Im: INFOCOM 2013, Turin, Italy (2013)
11. Cuervo, E., Balasubramanian, A., Cho, D., Wolman, A., Saroiu, S., Chandra, R., Bahl, P.: MAUI: making smartphones last longer with code offload. In: ACM International Conference on Mobile Systems, Applications, and Services, pp. 49–62, San Francisco, CA, USA (2010)
12. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for vm-based cloudlets in mobile computing. Pervasive Computing **8**(4), 14–23 (2009)
13. Vishwanath, V.K., Nagappan, N.: Characterizing cloud computing hardware reliability. In: 1st ACM Symposium on Cloud Computing, pp. 193–204 (2010)
14. Kumar, K., Lu, Y.H.: Cloud computing for mobile users: can offloading computation save energy? Computer **43**(4), 51–56 (2010)
15. Papanikolaou, K., Mavromoustakis, C.X.: Resource and scheduling management in Cloud Computing Application Paradigm. In: Mahmood, Z. (eds.) Cloud Computing: Methods and Practical Approaches, pp. 107–132. Springer International Publishing, Heidelberg (2013)
16. Carroll, A., Heiser, G.: An analysis of power consumption in a smartphone. In: 2010 USENIX, Boston, MA, USA (2010)
17. Mavromoustakis, C.X., Dimitriou, C.D., Mastorakis, G.: On the real-time evaluation of two-level BTD scheme for energy conservation in the presence of delay sensitive transmissions and intermittent connectivity in wireless devices. Int. J. Adv. Netw. Serv. **6**(3&4), 148–162 (2013)