A Context-aware Collaborative Model for Smartphone Energy Efficiency over 5G Wireless Networks

Radu-Corneliu Marin[a], Radu-Ioan Ciobanu[a], Ciprian Dobre[a,b,*], Constandinos X. Mavromoustakis[c], George Mastorakis[d]

[a]*Faculty of Automatic Control and Computers, University Politehnica of Bucharest 313 Splaiul Independentei, Bucharest, Romania*
[b]*National Institute for Research and Development in Informatics 8-10 Maresal Averescu, Bucharest, Romania*
[c]*University of Nicosia, Department of Computer Science 46 Makedonitissas Avenue, 1700 Nicosia, Cyprus*
[d]*Technological Educational Institute of Crete Estavromenos 71500, Heraklion, Crete, Greece*

Abstract

The staggering progress of mobile computing has brought forth exciting opportunities in the research community which are currently stretching beyond the limits of modern battery technologies. Although energy efficiency is of utmost importance in mobile systems, current solutions fail to take into consideration the intrinsic mobility of handhelds and are based on overusing power-hungry cellular networks for offloading into the cloud. We propose a novel collaboration model based on context-awareness and opportunistic networking in the context of 5G wireless networks which offers the possibility of offloading tasks in an opportunistic cloud based on mobile communities. We apply our solution to a real-life use-case, namely preventive patient monitoring, and show through experimental analysis based on real user traces that it maximizes power saving and minimizes overall execution time of tasks.

Keywords: context-aware, opportunistic networking, energy efficiency, collaborative model, eHealth.

*Corresponding author

Email addresses: radu.marin@cti.pub.ro (Radu-Corneliu Marin), radu.ciobanu@cti.pub.ro (Radu-Ioan Ciobanu), ciprian.dobre@cs.pub.ro (Ciprian Dobre), mavromoustakis.c@unic.ac.cy (Constandinos X. Mavromoustakis), gmastorakis@staff.teicrete.gr (George Mastorakis)

1. Introduction

Applications normally found in traditional computing environments are recently migrating towards handheld devices, which, based on recent advancements in mobile technologies, are providing excelling computing power while augmenting the user experience with contextual information [1, 2]. Unfortunately, such computationally demanding applications lead to an important issue which seems to have eluded mobile Original Equipment Manufacturers (OEMs) and software engineers alike, namely extreme energy consumption. Although CPU and memory technologies are constantly improving, battery manufacturers are struggling with the physical limit of current technologies and have yet to provide a proper solution.

In an attempt to overcome such physical limitations, current solutions related to energy efficiency can be grouped into two categories: (1) bounding resource utilization, and (2) offloading power-hungry tasks into the cloud. The solutions from the first category are based on limiting the usage of both hardware and software resources, and, although they have provided undeniable power reduction, their applicability proves to be insufficient for the ever-increasing requirements of smartphone applications. The latter category takes on a different approach for dealing with mobile energy efficiency and is represented by moving computation away from mobile devices and into the cloud while still benefiting from context-awareness. In the mobile cloud computing (MCC) paradigm, power-hungry tasks are offloaded from mobile devices for execution in the cloud. However, the advantages of mobile cloud computing must be weighed by their shortcomings. First of all, most MCC applications rely heavily on the availability and steadfastness of large bandwidth mobile networks. However, due to the inherent mobility of users, there is no guarantee that such communication media is always available and, in its absence, mobile cloud applications are unreliable or even rendered useless. Furthermore, mobile networks tend to be more costly, both financially and energetically.

1.1. Towards the 5G era

As the mobile industry continues to scale rapidly, the network infrastructure has to adapt to cope with the increasing communication demands of these subscribers. Current mobile devices connect to the Internet through multifarious broadband technologies, as we are slowly reaching the 5G era of communication [3, 4]. With 5G, new concepts of connecting end-user devices are being investigated, such as wireless mesh networking and dynamic ad-hoc networking [5, 6]. Such systems allow users to simultaneously connect to multiple wireless access technologies and seamlessly roam between them. By making use of such varied access schemes, it will be possible to link to other nearby devices to provide ad-hoc wireless networks for faster data flows. New networking paradigms, such as User-Centric Networking (UCN), explore this concept to allow the creation of user-centric wireless networks, autonomously formed between wireless devices able to inter-connect using standard wireless protocols (such as Wi-Fi, Bluetooth, Wi-Fi Direct or NFC). The term user-centric, in this context, is meant to express a community model that extends the traditional multi-access broadband communication with a perspective where the end-user device is actually part of the network.

One type of such mobile networks that have been deeply researched in recent years is represented by opportunistic networks (ONs), which evolved naturally from mobile ad-hoc networks (MANETs). Opportunistic networks are dynamically built when mobile devices collaborate to form communication paths while users are in close proximity. They are based on a *storecarryandforward* paradigm[7], meaning that a device (i.e. a smartphone) that wants to relay a message begins by storing it, then carries it around the network until an opportunity occurs: a wireless link becomes available, and either the carrier encounters the destination or a node that is more likely to bring the data close to the destination, and then finally forwards it. ONs have gained popularity because they come as an alternative to using the existing wired infrastructures, which may lead to significant power reduction, as well as the decongestion of said infrastructures. Such mobile networks differ considerably from the classic wired networks, both in terms of structure, but also with regard to the protocols and algorithms used for routing and data dissemination. Since we assume no stable topology (users carrying smartphones are mobile, and they can connect opportunistically to any other device in proximity, using device-to-device direct communication over Wi-Fi, Wi-Fi Direct, etc.), nodes in mobile networks are not aware of a global structure and have no knowledge of their relationship with other nodes (like proximity, connection quality, etc.). Each node is only aware of information about nodes that it is in contact with at a certain moment of time, and may act as data provider, receiver, and transmitter, during the time it spends in the network. Thus, a node can produce data, carry it for other nodes and transmit it, or receive it for its own use.

It is foreseen that phone-to-phone communication will be the key to decongestion of the already congested broadband networks through new communication models. As of Release 12, the Third Generation Partnership Project (3GPP) focuses on utilizing device-to-device communication for providing proximity-based services on top of current cellular infrastructure. 3GPP introduces new technologies, to allow energy-efficient proximity-based service discovery and communication for users on-the-go, catering to the whole potential of ONs: unlicensed-spectrum Wi-Fi Aware and in-band LTE-Direct. LTE-Direct has already been implemented and tested, making it a radio technology designed specifically for opportunistic device discovery. Due to its synchronous duty cycling scheme, it will significantly reduce energy consumption in devices. This can be seen as a strong indication of the rise of ONs.

1.2. Contribution

In an attempt to solve the energy-constraint and advance the MCC model towards the 5G era, we propose a hybrid solution where mobile devices are both clients and resources in an ad-hoc community-centric opportunistic mobile cloud. By sharing their combined resources, handhelds are able to reduce their global energy footprint by intelligently collaborating over ONs. Our solution is not intended as a replacement for MCC, but instead it attempts to complement mobile clouds in order to fill the presented gaps. We introduce the concept of mobile-to-mobile contextual offloading, in which handhelds make use of a contextual search algorithm to schedule the remote execution of tasks in trusted smartphone communities, based on predicting the availability and mobility of nearby devices. We present the Hybrid Contextual Cloud for Ubiquitous Platforms comprised of Smartphones (HYCCUPS), a framework that implements the proposed contextual offloading model.

HYCCUPS is focused on energy efficiency and, as such, its aim is bringing cloud resources closer to the mobile devices in order to reduce network transfers. Moreover, it makes use of opportunistic communication channels, as they are less power consuming in comparison to mobile networks. By doing so, it also solves the connectivity problems of MCC applications in the lack of support for the latter. Furthermore, the burden on mobile application developers is eased, as offloading is hidden from them behind inter-process communication constructs and also because the offloaded tasks are executed on other mobile devices using the same (local) implementation.

Preliminary versions of our work were published in [8]. This paper presents more extensive work by providing the context of augmenting 5G cellular networks based on context-aware collaboration over opportunistic networks. Furthermore, we expand on the methodology for assessing the feasibility of our solution, and we show how it can be applied in a real-world scenario of adamant importance, to eHealth systems, namely preventive patient monitoring.

2. Task Offloading

The corner stone of mobile cloud computing is represented by task offloading which proposes transferring the execution of power hungry tasks and moving their data from mobile devices onto clouds. This approach takes advantage of moving computation away from mobile devices for extending the battery lifetime, and also makes better use of the data storage capacity and of available processing power of cloud platforms [9]. Furthermore, MCC also benefits from the inherited strengths of cloud computing as well: dynamic provisioning, scalability, multi-tenancy, and ease of integration. These features of MCC enable mobile application developers to create a consistent user experience, regardless of various mobile platforms, thus offering fairness and equality to all users.

2.1. Application Code Partitioning

In the *mobile-to-cloud offloading* model introduced by MCC, most challenges arise from partitioning the mobile application code into tasks that can be executed remotely, and tasks that are bound to the device, based on the dependencies of each task. As such, MCC solutions can be categorized by partitioning technique and by offloading semantics.

The partitioning technique is an indicator of the adaptability of offloading, as it describes how tasks are specified by developers and also how they are viewed by the offloading system. Thus, MCC solutions can be classified into:

1. static: tasks are clearly defined at the design of the mobile application; this approach is less adaptive, but it has the advantage of directly specifying the data dependencies of tasks

2. dynamic: tasks are determined at runtime and data dependencies are inferred by the offloading system; these solutions are more error-prone,

but also more adaptable, as the granularity of tasks can be modified to suit the current context of offloading.

On the other hand, offloading semantics have a great influence on the mobile application development process, as they define the level of knowledge required to integrate with cloud infrastructures. MCC solutions are classified by offloading semantics into:

1. implicit: cloud bindings are hidden from developers behind programming language constructs typical to the target mobile platform; however, such transparency usually leads to less control over the offloading process

2. explicit: the development process is dramatically changed, as the link to cloud computing is implemented and maintained by the developers of the mobile application; Although this solution offers full control over offloading, the cost of development increases.

2.2. Offloading Decision

The naive approach towards offloading is to always choose to delegate cloud infrastructures to run computational-intensive tasks. Unfortunately, such an approach does not take into consideration the intrinsic mobility of users and, when disconnections occur, MCC applications are rendered useless.

Therefore, contextual information is used to enhance the offloading decision in order to dynamically determine at runtime whether a task should be transferred or executed locally. Such optimization engines collect performance measures in order to increase energy efficiency, reduce network congestion and preserve user experience. By monitoring the user and device context, as well as the network state, MCC solutions can better schedule the execution of tasks.

2.3. Towards Multi-Tier MCC Architectures

The main concern of MCC is the lack of privacy in storing data on clouds. Furthermore, most MCC solutions rely on the ubiquity of fast cellular networks which provide large bandwidth dependable links towards cloud computing. Unfortunately, such communication media has been proven to be power-hungry and can actually lead to increased power consumption [10].

To solve the above issues, novel solutions have been proposed which tend to characterize MCC as having a three-tier architecture by offering multiple levels of offloading support:

1. into the cloud: the classical approach of offloading

2. onto nearby servers (cloudlets): such commodity servers in the vicinity of mobile users can dramatically reduce the overhead and power consumption of network transfers; furthermore, privacy issues are appeased as data travels in internal or hybrid clouds

3. onto mobile devices in the vicinity which act as cloud service providers.

The final tier in the MCC architecture raises most interest, as it is not only oriented at energy efficiency and data privacy, but also paves the way towards novel mobile applications by fully tapping into contextual data. In this sense, Bourdena et al[11], make use of both social ties of mobile users and network-centric parameters exchanged through opportunistic interactions to augment the decision to offload into the cloud. Hyrax [12] proposes the use of smartphones as resources in the cloud, instead of mere clients. This solution offers cloud computing for Android devices, allowing client applications to access data and offload execution on heterogeneous networks composed of smartphones. Moreover, Huerta-Canepa and Lee [13] introduce a framework that allows creating ad-hoc cloud computing providers based on devices in the nearby vicinity, in order to treat disconnections from the cloud in a more elegant fashion. Murray et al. [14] take one further step towards opportunistic computing by introducing *crowd computing*, which combines mobile devices and social interactions with the goal of achieving large-scale distributed computing. By deploying a task farming computing model similar to that of Hyrax onto real-world traces, they place an upper-bound on the performance of opportunistically executing computationally-intensive tasks and obtain a 40% level of useful parallelism.

Unfortunately, none of the above third-tier solutions provides a proper programming model to support developers in creating MCC applications: authors in [12, 14] propose using the task-farming model borrowed from cloud computing, which is unsuitable for the type of workloads in mobile computing, whereas [13] rules in favor of enforcing cloud APIs for mobile cloud providers, thus encumbering the mobile application development process. We believe that such solutions must make use of static partitioning to enforce the modularity of mobile applications and of implicit offloading semantics.

3. Opportunistic Networking

Due to the drastic increase in the number of mobile devices all around us, new network paradigms have appeared in recent years, one such example being opportunistic networks. ONs are networks generally formed of devices with a high degree of mobility. Unlike other types of mobile networks, the disconnections between devices in ONs are considered as the norm, instead of issues that need addressing. Thus, opportunistic nodes communicate through a *store-carry-and-forward* paradigm: nodes store data items, carry them around until a suitable node is found, and then forward them. A suitable node might be the destination of a data item, or a device that has a higher chance than the carrier of delivering the item to its destination. However, unlike regular networks, ONs are fully decentralized and thus the nodes can only access local information. There is no global view of the network containing paths between nodes, because these paths might not even exist.

Because nodes only have a local view of the network, they have to find other ways to estimate whether an encountered peer is a more suitable forwarder for a given data item. This is where context information (such as encounter history, social network, interests, battery level, etc.) comes into play, since it can aid in these decisions. Based on context, opportunistic nodes use heuristics for assigning utilities to data items, in order to decide which of the two nodes in contact is a better carrier. For example, if two nodes are socially connected (e.g., the device carriers are friends on Facebook), there is a high chance of them meeting, since they belong to friends that might see each other. The same is true for nodes that have common interests, as shown in [15]. The more context information a node has available, the better its chance is of performing an informed decision with regard to carrying specific data items. Selecting the correct forwarder for an item may not only increase its chance of delivery, but also reduce the network congestion (since nodes will receive fewer irrelevant messages).

Opportunistic networks are not employed only for communication, but they may also act as the foundation of opportunistic computing [16]. Thus, the mobile devices of people in a region can act as an ad-hoc cluster or a cloud, each node having the potential of performing computations for the benefit of others. Altruism is the basis of opportunistic computing, since the purpose is not necessarily to help an individual, but to help the group.

4. Contextual Collaboration Model

In order to promote energy efficiency, we propose *mobile-to-mobile contextual offloading*, a novel mobile collaboration solution based on a contextual search algorithm which schedules the remote execution of tasks in an opportunistic community by predicting the availability and mobility of nearby devices.

The first step in our contextual search algorithm is represented by sensing contextual data by gathering and tracing raw data from environment sensors. The context retrieval is provided by the HYCCUPS Tracer [17], while in [18] we study the predictability of mobility and opportunistic interactions and we prove the feasibility of the proposed model. The features that we consider interesting for recording are: CPU load, CPU frequency, CPU idle time, available memory, network usage, Wi-Fi interactions, battery level, battery charging state, screen usage, and application activity events. The tracing data is aggregated into the trained availability records, which behave similarly to circular buffers. Based on the aggregation method, there are two types of records:

1. averaging records: the data stored in these records is a continuous real value and it is aggregated by using the average (examples: CPU load and frequency, available memory, battery level)

2. probability records: these records contain boolean events and the aggregation method is the probability that the event is true (examples: battery charging state, peer interactions, user activity).

As long as the contextual search is running on a device, the records are continuously updated so that the offloading algorithm is able to adapt to environmental changes (e.g., user changes her patterns). However, these averaged momentary values simply represent an instance in a user's history. In order to make good use of these values, we need to extract temporal patterns over intervals of execution, so that the system can predict future values for them. In order to predict the values for features on any given interval of time, the offloading model uses univariate linear regression to be able to approximate values for features for any given time-frame. Given that such an offloading model is constantly adapting to environmental changes, it is only natural to believe that predictors may output erroneous values. In this sense, the prediction output is weighed against the degree of informedness and their own accuracy.

The univariate linear regression determines a hypothesis h which is able to approximate values for features for any given time frame:

$$h(x) = \theta_1 + \theta_2 x \tag{1}$$

The predictor for the average of a feature is defined as $predictFeature(\Delta t)$ over an interval starting from the current time and lasting as much as Δt and it computes this future value as follows:

1. construct the statistical analyzed interval by tripling the predicted interval as to provide sufficient information for the actual prediction:

$$interval = [currentTime - \Delta t, currentTime + 2 * \Delta t] \tag{2}$$

2. extract the n samples from the feature's record in the statistical analyzed interval. The records are considered to be circular (the weeks wrap around).

3. construct the X matrix, where x_i is the i-th sampling point in the analyzed interval:

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ ... & ... \\ 1 & x_n \end{pmatrix} \tag{3}$$

4. construct the y column vector, where y_i is the i-th feature value:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{pmatrix} \tag{4}$$

5. compute the Θ column vector, by solving the following equation:

$$X * \Theta = y \tag{5}$$

The Θ column vector contains the parameters of our heuristic h which is able to predict values of a feature:

$$\Theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \tag{6}$$

In order to find Θ, we use the normal function:

$$\Theta = (X^T \times X)^{-1} \times X^T \times y \tag{7}$$

6. compute the average of the feature in the prediction interval:

$$predictFeature(\Delta t) = h(currentTime + \Delta t/2) \tag{8}$$

Given that such a system is constantly adapting to environmental changes, it is only natural to assume that predictors may output erroneous values. In this sense, the prediction output is weighed against degree of informedness and to their own accuracy:

$$\begin{split} predictFeature(\Delta t) = w \times h(currentTime + \frac{\Delta t}{2}) + \\ (1 - w) \times momentary\_value \end{split} \tag{9}$$

where w is the feedback weight:

$$w = \frac{informedness}{2} + \frac{accuracy}{2} \tag{10}$$

The degree of informedness expresses how many samples have been recorded in the statistical analysis interval:

$$informedness = \frac{no\_filled\_samples}{no\_total\_samples} \tag{11}$$

In order to measure the efectiveness of the predictor, we use the balanced accuracy:

$$accuracy = \frac{1}{2} \times \frac{positives}{positives + false\_positives} + \frac{1}{2} \times \frac{negatives}{negatives + false\_negatives} \tag{12}$$

The heart of the *mobile-to-mobile contextual offloading* model is the contextual search algorithm based on the feature prediction model, which is responsible for deciding whether to offload a task on a nearby device or to execute it locally. To be able to schedule tasks in opportunistic networks, the contextual search is split into two phases:

1. workload profiling: ascertaining the amount of resources needed to actually run a task

2. offloading decision: determining if a task is better run locally or on a different device.

Given that any task can run on multiple devices with different traits, workload profiling needs to determine which resources are required by the task, compute a workload score which uniquely describes the computational requirements of the task, and normalize the workload score to be able to compare the executional needs of a task for various mobile platforms. In this sense, we introduce the computational potential, defined as the time required by the processor to run a workload, considering current and predicted values for CPU load and frequency in the nearby future.

In essence, the potential is a measure of the time a task requires CPU time on a specific device. At a first glance, this measure may seem inequitable, as devices with better performing CPUs might seem disadvantaged. But, as presented further on, time turns out to be a good invariant in our problem. Let us consider an example: two devices - a low end device (A) and a high end device (B) with a 5 times faster CPU are part of the hybrid cloud; device (A) shortly becomes overwhelmed by the tasks it's executing so it offloads 5 tasks with $potential_A = 1s$ to (B) which evaluates them at $potential_B = 0.2s$ so, instead of, (A) executing them in 5s, (B) runs all of them in 1s; not much later, the user of device (B) becomes active and launches a task which

(B) evaluates to $potential_B = 1$ and offloads it to (A) which evaluates it to $potential_A = 5$. Now both devices have cleared their debt and were able to run more efficiently (while the remote device was executing the offloaded task, the local device was able to execute other tasks). One should keep in mind that the workload of a task is usually directly proportional to the computational possibilities of the device that it's running on (because users tend to use applications that are suited for their device). As such, tasks from low end devices such as (A) are usually much smaller than tasks running on high end devices (B).

Furthermore, in order to reduce the number of leechers in the system, a cost model was put in place. As opposed to [19], our solution cannot use a reputation-based approach, as, although a node might correctly offload all tasks, there is no guarantee that it can handle any tasks from its community – if a device is, by far, less-performing computationally than other devices, it will not be able to offload for any other device due to time constraints that would lead to poor quality of experience. As such, we rule in favour of a credit-based approach: each time a node offloads a task for another, it remembers the debt it is owed by that node as the sum of negative potentials; also the node which is requesting the offload stores the debt it owes the other node as the sum of positive potentials. When a node responds to an offload request, it will advertise that its potential is: $potential = true\_potential - debt\_for\_requestor$. This ensures altruism as it discourages nodes to take advantage of other nodes, and it encourages nodes with debt to offload in order to reduce their debt.

The following steps are required to determine the potential, given the number of CPU cycles estimated for a task through workload profiling:

1. determine the maximum acceptable time (MAT) to run a task (the time needed to run a task at lowest frequency with 50% CPU load):

$$MAT = num\_cycles \times \frac{1}{0.5 \times min\_frequency} \tag{13}$$

2. determine the potential of a task (the predicted time to run a task in

current conditions):

$$potential = \frac{num\_cycles}{predictCPULoad(MAT) \times predictCPUFreq(MAT)} \tag{14}$$

Although it is not the absolute maximum time of running a task on a device, the MAT represents the maximum time that we would allow an offloaded task to run on a device, and it should be considered as the lower acceptable limit for offloading. It's more of a quantitative approach to determine the interval of time that a task should be executed in. As such, we compute the actual potential by predicting the CPU performance over the MAT interval. Therefore, the maximum acceptable time is computed only to determine the interval of prediction for the potential.

A major concern raised by using linear regression for predicting features in our context-aware solution is the stability of the data we are analyzing. Equation 7 is based on computing the pseudo-inverse of X, which may provide numerical instability for large matrices. The size of X is influenced mostly by the duration of the inter-contact time of opportunistic interactions, as a offloaded tasks are single-hop opportunistic messages to guarantee that the results of an offload reach the node that is interested in them. As such, the duration of the opportunistic interaction directly influences the size of the statistical interval for predicting features. By analyzing the tracing sets, we have discovered that such interactions do not last longer that several minutes, with few exceptions leading up to at most one hour. Furthermore, the largest matrixes are obtained for the contextual features collected with higher frequencies. As such, CPU samples are collected and aggregated on a per-minute basis which leads to the conclusion that the X matrix should never be greater than $60 * 60$. Last, but not least, mobile tasks seldom last more than a few minutes as users do not generally interact with their devices on a long-term basis.

Based on a task's estimated potential, the offloading model is able to correctly schedule a task in an opportunistic network. When a task is generated by an application, it passes through the workload profiler, which determines its computing potential. This potential will further augment the contextual search algorithm that actually decides whether to execute the task locally or offload it. Given that the task is executed in an opportunistic network, the

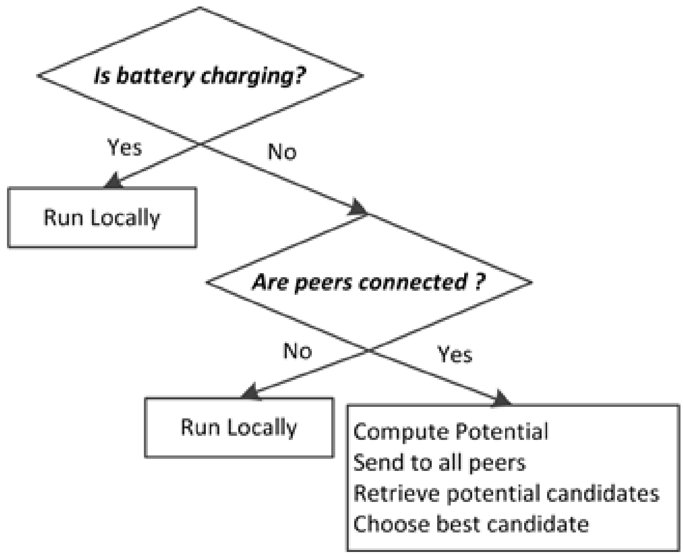


Figure 1: Local decision for offloading.

decision is distributed. Therefore, there are actually two types of decisions that need to be taken:

1. local decision: the current device decides whether it is best to execute the task locally or whether it should be offloaded

2. remote decision: a remote node receives the offloading task request and needs to determine if it can handle the execution of the remote task.

4.1. The local decision

Before offloading a task onto nearby devices, a node first decides, through contextual search, if the task actually needs to be offloaded and executed remotely, or if is better run locally. Figure 1 depicts the decision tree for local task resolution. Each decision node in the tree is actually a feature predictor.

If the offloading model decides to remotely execute the task, it sends a task offload request to all connected peers, and waits for their potentials to run the task. After gathering all of the results, it adds the local potential to the list as well, and chooses the best candidate that will use minimal potential to execute the task.

If a node decides to offload a task, it creates an opportunistic message containing the data needed for completing the computation and forwards it to the desired peer. All such messages are single-hop messages as we are currently unable to determine if the results will be able to return in due time to the original node, if the offloading node forwards it to other nodes instead of executing it itself.

4.2. The remote decision

When a node receives an offload request, the offloading model needs to decide whether it will accept the request to run the task, or if it will reject it because it is unable to spare the resources. The contextual search in the remote decision is illustrated in Figure 2.

The remote decision is much more intricate than the local one, because each node wants to maximize the number of offloaded tasks, while minimizing the offloading it's actually doing. However, the cost model presented above is refereeing this process so that no node will be taken advantage of by others.

Also, similar to the local decision, under certain conditions (e.g., when a node is charging), it will attempt to act altruistically and it will advertise only a tenth of its actual potential to execute the task, in order to encourage other nodes to offload to it while it is still charging.

5. Pilot case studies: applying the context-aware opportunistic collaborative model to smart eHealth systems

As the number of people aged 65 and above is estimated to double over the next 5 years, the demand for healthcare in Europe [20] alone will face increasing pressure. The surge in ageing population worldwide impose a substantial increase in expenditures with public social and health services. Consequently, there is a need for innovative ICT health applications to relieve some healthcare resources and automate as much as possible in medical processes.

Fortunately, senior citizens are becoming more computer-savvy. Although they may not be as quick on the keyboard as young people, they are embracing technology and its many benefits. Currently, smartphones and other handheld communication devices are no longer just gadgets. Not surprisingly, the over-65 age group has downloaded approximately half as many applications as people half their age, but if the trend toward tablet computers continues at its current rate, the number is bound to increase as well.

In our increasingly tech-savvy society, it is of little surprise that the most useful products to aid seniors and caregivers are innovative technologies that

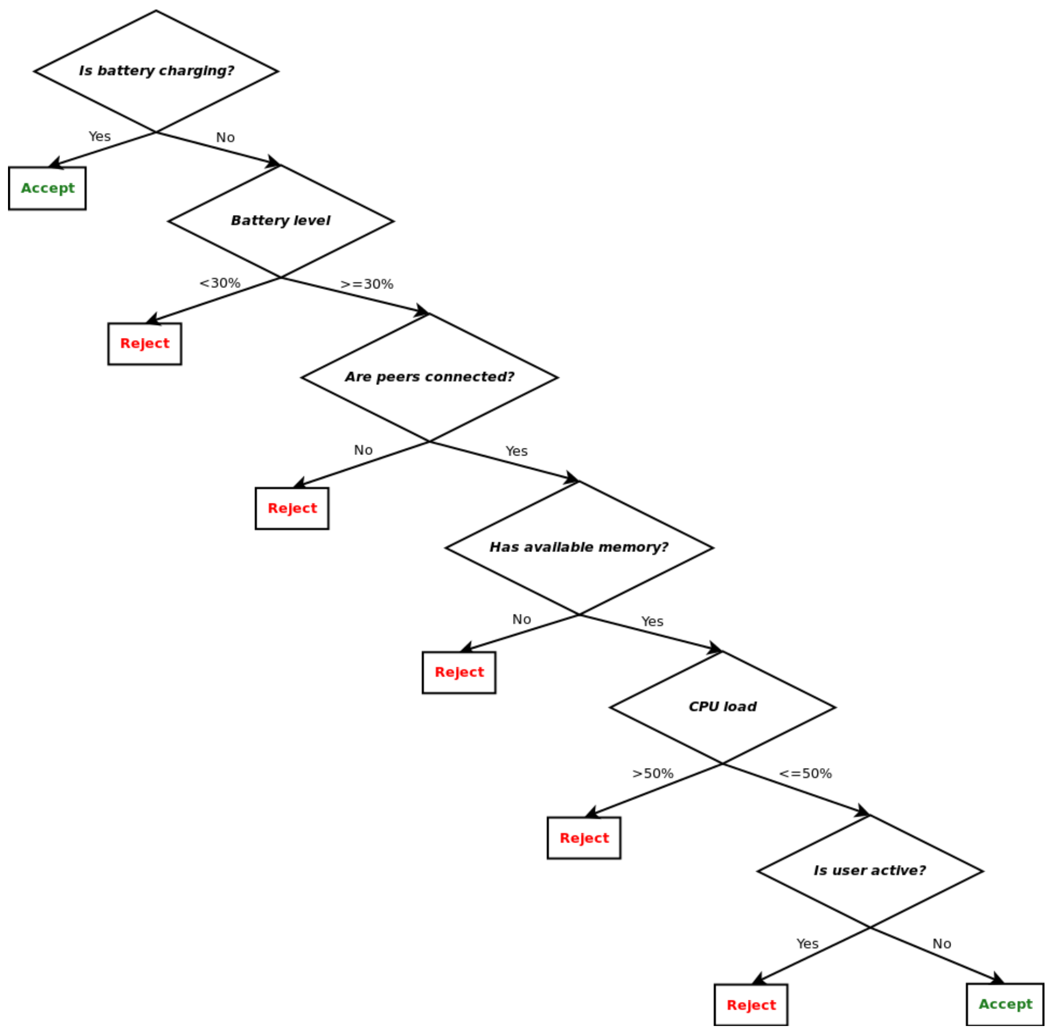


Figure 2: Remote decision for offloading.

promise to make day-to-day life easier by providing health monitoring [21]. There are many examples today of such applications and technologies for elderly people to use on their mobile devices.

Preventive Patient Monitoring is a technology designed to monitor at-risk patients with wearable sensors capable to collect real-time health data [22]. A wearable device called BodyGuardian Remote Monitoring System is used to monitor cardiac health using a lightweight wearable sensor. The data is further sent to a smartphone and, from there, using broadband communication, to a cloud-based platform accessed by physicians at any time.

9Solutions Integrated Positioning and Communication System (IPCS) provides a system to monitor residents in nursing facilities and care homes [9]. Emergency and assistance calls can be made using everything from wearable tags to call units to mobile phones. The same tags can also track movement and help prevent wandering in dementia patients.

The Metria's Wearable Sensor is designed for medical providers to monitor the health of seniors and others [23]. It sticks to the body with skin-friendly adhesive, collecting data like heart rate, blood pressure, and amount of sleep and sending it wirelessly to be interpreted by sophisticated computer algorithms. A connected mobile app allows both patients and caregivers to examine the data.

Such applications rely on portable/wearable devices equipped with various sensors. Smart bracelets will be able, for example, to read an elderly person's heart rhythm through an electrocardiogram (ECG), or his current temperature and location. Devices such as Samsung Gear, Simband, etc., already include adequate sensors for this. The captured data will probably be sent through short- and medium-range wireless protocols (Wi-Fi, Bluetooth) to a forwarder (either a background service running on the user's smartphone, if he carries one, or a wireless Access Point available). From the smartphone, the data will be forwarded further through broadband communication (3G/4G), where it will be analyzed on a Cloud premises. However, in this picture, MCC can provide a potential support for advanced health mobile services.

For example, cellular radio communication is a significant contributor to battery energy drain on smartphones, in some cases inflating the energy cost by a factor of 5 or more compared to the energy cost of the base device [24]. Through the proposed contextual collaboration model, we aim to provide the technology for collecting the sensed data opportunistically: if an elderly is not carrying his smartphone, and has no connection to the Internet (e.g., he is in a park with no Wi-Fi coverage), a simple encounter with another person carrying a smartphone could transform into an opportunity for transferring the data over to this device, to carry it further. The same stands for deploying part of the heavy computation for recognizing health-related events or conditions into the mobile cloud formed at the edge of the actual Cloud platform.

In order to evaluate the optimization brought forward, we have developed the HYCCUPS Emulator, which provides valuable insight into the innerworkings and performances of our proposed solution. It makes use of traced contextual data collected in a tracing experiment developed at the Faculty of Automatic Control and Computer Science, University Politehnica of Bucharest. A total of 66 volunteers participated. They were selected in the experiment such as to have a wide range of study years and specializations covered: one first-year Bachelor student, one third-year Bachelor student, 53 fourth-year Bachelor students, three Master students, two faculty members and six external participants (participants selected from office environments) [25].

For evaluating our solution, we considered the following application: a monitoring application that automates the recognition of hazardous situations (i.e., loss of memory, increased stress) for the elderly, delivers alarms to caregivers, and finds potential caregivers that are in the vicinity of an elderly when such an event is generated. The application could use a registry of volunteers that have signed to receive such requests for help when they are in the physical proximity of a potential victim (elderly) – for privacy reasons, the application will not continuously track the volunteers, but will leverage the intrinsic locality of Wi-Fi to be able to reach only those volunteers that are close enough without any a-priori knowledge of their position. Using Wi-Fi network mappings, the system could identify the network it needs to broadcast the request onto for guaranteeing that help is discovered quickly. The request for help will be protected from malicious eavesdropping by a broadcast encryption scheme. Once a volunteer has replied to the request (and confirmed his position to the system) an even stronger encryption scheme could be used to provide the precise detail of the help to be provided.

The realization of a high quality MCI monitoring system requires a model of the assisted people. A psycho-cognitive model might look similar to the one presented in Figure 3.

The figure shows several intervals over the model: the first defines the normal parameters for good living conditions for a monitored patient, the second one defines boundaries of living conditions, considering the particular person's health status, and the last one defines the borderline between normal and critical health conditions. A monitoring system related to catastrophic events (e.g., for monitoring memory-impaired elderly patients, an event would notify that the person is lost in a park) needs to recognize boundary conditions in order to prevent misinterpretation of situations. This can be used to cope with situations where a person deviates from the usual route because of a municipality-operated scheduled detour (e.g., maintenance

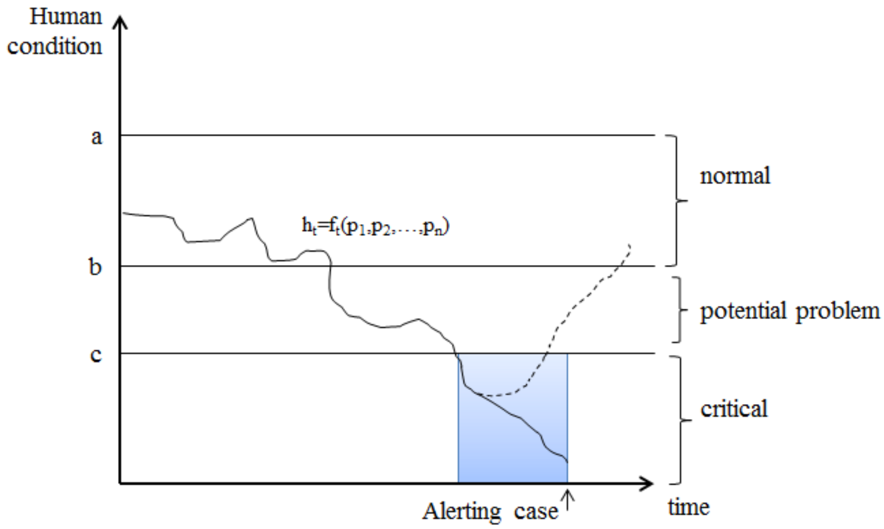


Figure 3: A condition-monitoring patient model.

work to replace an underground pipe in the park), which otherwise might be wrongly recognized as a catastrophic event (i.e., alerting emergency units of potentially lost persons). Also, the context parameters will be used to formulate user-centric boundaries, considering personalized health conditions. For instance, a person with a locomotor disturbance will move slower than a person who does not suffer from this disability. Slow movement of such a person must again not be misinterpreted as a critical situation resulting in a false emergency alert. The fictive curve in the figure defines the overall condition of a monitored person as a function over time. The ultimate goal would be to detect the entering of the shadowed area in the figure. In such a case, the monitoring system would further undertake a series of possible corrective actions to prevent the emergency (e.g., if the person is detected as having a heart stroke, the system will use animations to remind the patient of medication, drinking, etc. for a service provided on the patient's smartphone, and alert the caregiver), thereby bringing the health and status condition of the person back to normal (i.e., the dotted line). The monitoring application would use data mining and machine learning techniques over the set of monitored data for the patient (cognitive and behavioural data), namely reasoning algorithms (such as Decision Trees). Of course, for such heavy-computing techniques, the local capacity of a single mobile device would probably not suffice. Yet, moving the computation into the cloud might unacceptably delay the received result. This is where HYCCUPS could balance things. Furthermore, based on the identified events/situations, the application would operate proactively and do its job automatically with minimal human intervention. It could interact with humans by speech, gestures, and other forms of natural communication. The caregivers will be able to interact and create personalized alerts. They will receive detailed information about the event on a Web-based interface, but will also be able to specify that they need to be alerted on his smartphone, for the case they are not in vicinity of their computers. From there, they will be able to initiate the appropriate corrective actions, such as establishing the danger level for the elderly's current situation, and possibly alerting rescuers.

We have attempted to cover multiple types of workloads, pertaining to the kind of application presented above, while analyzing our proposed solution. We have generated three scenarios using a total of 726 tasks over 2 months of emulator execution which are sent in a pseudo-random fashion to all active users in the system, with the following computational requirements:

1. 100Mcycle tasks: small tasks which are easily executed by any node. Such tasks can be associated to computing the probability of an event correlated against the psycho-cognitive model in Figure 3.

2. 1Gcycle tasks: regular sized tasks which should be executed by most devices. This scenario could be associated with finding nearest devices.

3. 10Gcycle tasks: large tasks which should be a burden for all devices. These tasks could relate to the calibration of the condition-monitoring patient model above.

The emulation process is quite straightforward: the previously collected tracing data sets are replayed for re-enacting the state of all of the volunteers that participated in the tracing experiment. As such, the emulator instantiates the same number of virtual devices as were present in the experiment, deploys the offloading decision algorithm on all such devices, and acts as a source of time for synchronizing them. Furthermore, it feeds in all contextual events into the algorithm similar to an application reacting to the data collected by the HYCCUPS Tracer. This process guarantees that the virtual devices are as informed as applications running on the real devices at the

Scenario	Model	Offloaded	FMT offloads	Time saved (s)
100Mcycle	w/ cost model	57	16	48
	w/o cost model	57	16	45
1Gcycle	w/ cost model	57	15	472
	w/o cost model	57	16	448
10Gcycle	w/ cost model	55	15	4723
	w/o cost model	57	16	4481

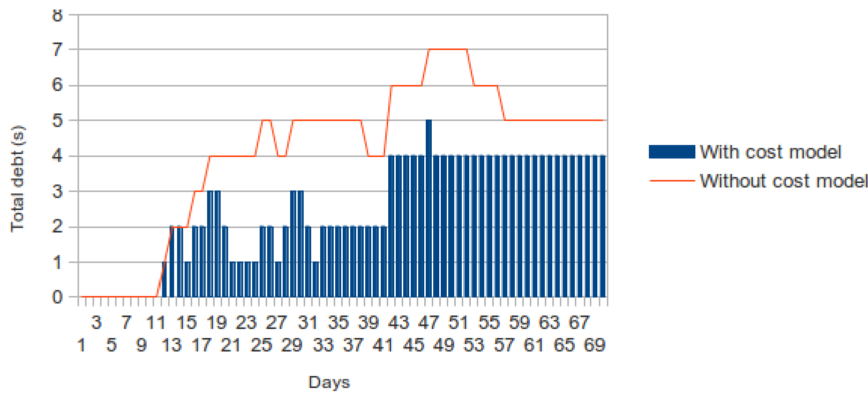
Table 1: Offloading statistics for all three scenarios.

time the data was collected. When a node decides to offload a task for another, the emulator will update the CPU and memory usage of said device, to mimic the influence of running such a task on the physical processor. Furthermore, tasks are generated similar to real-life situations, as reactions to user input or as a consequence of devices interacting opportunistically. This process allows us to fully measure the performance of our offloading decision, by re-iterating the process and doing parameter sweeps based on the types of tasks of interest.

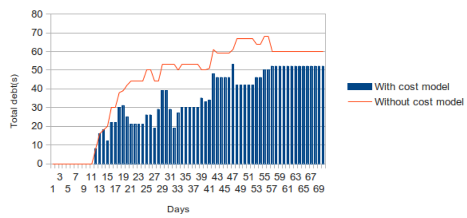
Figure 4 presents the total debt accumulated by all users on a daily basis for all three scenarios. As can be seen in all three cases, the cost model gives better performance, as it seems that altruism serves the community better than leeching off of other nodes. Furthermore, Table 1 shows that even though working without a cost model can actually offload more, the overall benefit of using such a model is much greater as it actually saves more computing time.

As can be seen in Table 1, HYCCUPS manages to save valuable computing which is actually proportional to the size of the task. Although at a first glance the amount of saved time does not seem to be large, only three devices were actively offloading tasks. This should generally change the perspective over the results. Also, adding that more than a quarter of the tasks were executed remotely on devices that are charging, the actual computing power saved increases. Moreover, out of the 57(55) offloaded tasks, none of them failed to complete, which proves that the adaptive and predictive models used in collaboration with the contextual search algorithm are working correctly.

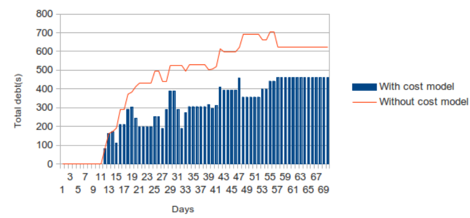
Last, but not least, Figure 5 illustrates the total execution time as tasks are sequentially run. Moreover, they compare the HYCCUPS execution model with the current traditional Android model. As can be observed,



(a) 100Mcycle tasks.



(b) 1Gcycle tasks.

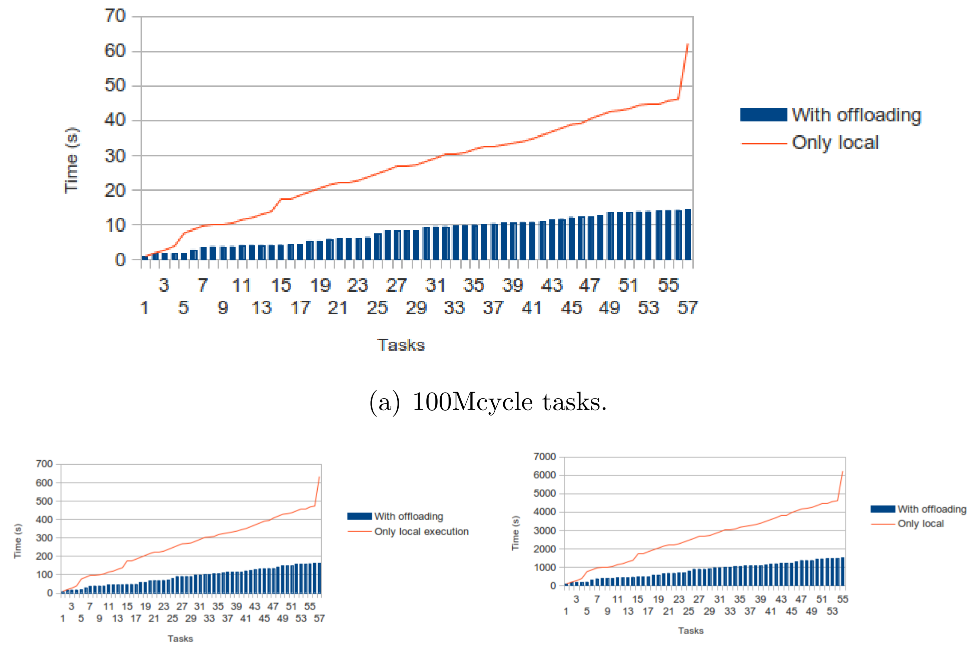


(c) 10Gcycle tasks.

Figure 4: Total accumulated debt.

the HYCCUPS framework speeds up mobile applications by means of offloading onto remote idle nodes. However, these figures do not contain the execution for all of the 726 tasks, but only for the 57(55) offloaded tasks. We are not interested in the tasks that are always run locally as HYCCUPS can only optimize workloads that actually have the possibility to be offloaded (although the predictors might have missed some offloading opportunities).

Although the solution was applied to preventive patient monitoring, the contextual collaboration model can be easily adapted to multiple types of applications: computing alpha-beta pruning in determining the next move in a chess game; determining the shortest path in a graph-based game; applying Haar classifiers on satellite images. The general scope of the contextual collaboration model is that of isolating computational tasks from the rest of the application, encapsulating it in a separate container that can be either executed locally or remotely while hiding the complexity of offloading from the application developers.



(a) 100Mcycle tasks.

(b) 1Gcycle tasks.

(c) 10Gcycle tasks.

Figure 5: Total accumulated execution time for offloaded tasks.

6. Conclusion

We believe that current models in the design, development and deployment of mobile applications are becoming inadequate, as they do not take sustainability into account. Furthermore, we consider the endeavours currently taken in mobile cloud computing to be incomplete and limited. As such, the need for novel computational models that focus on power saving is growing exponentially.

HYCCUPS introduces intelligent collaboration, a new computational model which offers feasible synergy between interacting peers by correlating mobility with availability information by means of a contextual search algorithm. Moreover, the algorithm is adaptive to environmental changes so as to improve battery health, maximize saved power, minimize overall execution time of mobile applications, and preserve or even enhance user experience.

We proved the correctness, feasibility and performance of our solution by emulating the solution based on traced contextual data, and the results show that not only does the overall execution time of tasks decrease, but also the utilization of resources in our ad-hoc cloud is optimized, thus reducing the power consumption of active peers. We showed results of how such technology could create the premises for advances in healthcare domain, showing a feasible way to integrate emerging technologies such as cloud computing and/or M2M in future healthcare systems and applications.

Acknowledgment

The research presented in this paper is supported by national project MobiWay, Project PN-II-PT-PCCA-2013-4-0321. We would like to thank reviewers for their constructive comments and valuable insights.

References

[1] A. Castiglione, M. Gribaudo, M. Iacono, F. Palmieri, Modeling performances of concurrent big data applications, Software: Practice and Experience 45 (8) (2015) 1127–1144. doi:10.1002/spe.2269.
URL http://dx.doi.org/10.1002/spe.2269

[2] M. D'Arienzo, M. Iacono, S. Marrone, R. Nardone, Estimation of the energy consumption of mobile sensors in wsn environmental monitoring applications, in: Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on, 2013, pp. 1588–1593. doi:10.1109/WAINA.2013.33.

[3] C.-W. Tsai, H.-H. Cho, T. K. Shih, J.-S. Pan, J. J. Rodrigues, Metaheuristics for the deployment of 5g, IEEE Wireless Communications 22 (6) (2015) 40–46.

[4] X. Ge, J. J. Rodrigues, B. Rong, Editorial for monet special issue on networking in 5g mobile communications systems: Key technologies and challenges, Mobile Networks and Applications 20 (6) (2015) 701–703.

[5] X. Wang, G. Han, X. Du, J. J. Rodrigues, Mobile cloud computing in 5g: Emerging trends, issues, and challenges [guest editorial], IEEE Network 29 (2) (2015) 4–5.

[6] J. M. Batalla, Advanced multimedia service provisioning based on efficient interoperability of adaptive streaming protocol and high efficient video coding, Journal of Real-Time Image Processing 12 (2) (2016) 443–454.

[7] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, IEEE Communications Magazine 44 (11) (2006) 134–141.

[8] R.-C. Marin, C. Dobre, Reaching for the clouds: Contextually enhancing smartphones for energy efficiency, in: Proceedings of the 2Nd ACM Workshop on High Performance Mobile Opportunistic Systems, HP-MOSys '13, ACM, New York, NY, USA, 2013, pp. 31–38. doi:10.1145/2507908.2507912.
URL http://doi.acm.org/10.1145/2507908.2507912

[9] 9Solutions, The integrated positioning and communicating system (ipcs), http://www.9solutions.com/en/technology/ipcs-technology-breakdown, accessed: January 11th, 2015 (2014).
URL http://www.9solutions.com/en/technology/ipcs-technology-breakdown

[10] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, P. Bahl, Maui: making smartphones last longer with code offload, in: Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM, 2010, pp. 49–62.

[11] A. Bourdena, C. Mavromoustakis, G. Mastorakis, J. Rodrigues, C. Dobre, Using socio-spatial context in mobile cloud offload process for energy conservation in wireless devices, IEEE Transactions on Cloud Computing PP (99) (2016) 1–1. doi:10.1109/TCC.2015.2511736.

[12] E. E. Marinelli, Hyrax: cloud computing on mobile devices using mapreduce, Tech. rep., DTIC Document (2009).

[13] G. Huerta-Canepa, D. Lee, A virtual cloud computing provider for mobile devices, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, ACM, 2010, p. 6.

[14] D. G. Murray, E. Yoneki, J. Crowcroft, S. Hand, The case for crowd computing, in: Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds, ACM, 2010, pp. 39–44.

[15] R.-I. Ciobanu, R.-C. Marin, C. Dobre, V. Cristea, C. X. Mavromoustakis, Onside: Socially-aware and interest-based dissemination in opportunistic networks (2014) 1–6.

[16] M. Conti, S. Giordano, M. May, A. Passarella, From opportunistic networks to opportunistic computing, Communications Magazine, IEEE 48 (9) (2010) 126–139.

[17] R.-C. Marin, C. Dobre, F. Xhafa, Exploring predictability in mobile interaction, in: Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on, IEEE, 2012, pp. 133–139.

[18] R.-I. Ciobanu, R.-C. Marin, C. Dobre, Interaction predictability of opportunistic networks in academic environments, Transactions on Emerging Telecommunications Technologies.

[19] J. A. F. F. D. Dias, J. R. Rodrigues, F. X. Xia, C. M. Mavromoustakis, A cooperative watchdog system to detect misbehavior nodes in vehicular delay-tolerant networks, IEEE Trans. on Industrial Electronics 62 (12) (2015) 7929–7937. doi:10.1109/TIE.2015.2425357.

[20] E. Commission, et al., The 2012 ageing report, economic and budgetary projections for the 27 eu member states (2010-2060), European Economy. Economical and financial affairs.

[21] S. Stevenson, Top products to make your life easier, http://www.aplaceformom.com/blog/2014-6-1-cutting-edge-products-for-seniors/, accessed: January 11th, 2015 (2014).
URL http://www.aplaceformom.com/blog/2014-6-1-cutting-edge-products-for-seniors/

[22] Preventice, Bodyguardian remote monitoring system, http://www.preventice.com/products/bodyguardian/, accessed: January 11th, 2015 (2014).
URL http://www.preventice.com/products/bodyguardian/

[23] A. Dennison, Metria informed health, http://www.averydennison.com/en/home/technologies/creative-showcase/metria-wearable-sensor.html, accessed: January 11th,

2015 (2014).
URL http://www.averydennison.com/en/home/technologies/creative-showcase/metria-wearable-sensor.html

[24] B. Aggarwal, P. Chitnis, A. Dey, K. Jain, V. Navda, V. N. Padmanabhan, R. Ramjee, A. Schulman, N. Spring, Stratus: energy-efficient mobile communication using cloud support, ACM SIGCOMM Computer Communication Review 40 (4) (2010) 477–478.

[25] R.-C. Marin, C. Dobre, F. Xhafa, A methodology for assessing the predictable behaviour of mobile users in wireless networks, Concurrency and Computation: Practice and Experience 26 (5) (2014) 1215–1230.