

REVIEW

Open Access



Self organization based energy management techniques in mobile complex networks: a review

Bahjat Fatima* and Munam Ali Shah

* Correspondence:
Bahjatfatima.se14@gmail.com
COMSATS Institute of Information
technology, Islamabad, Pakistan

Abstract

Today's mobile devices offer a variety of computational, memory, storage, communication and sensing resources. In addition, mobile communication technologies are continuously evolving and mobile networks are becoming more and more complex. Modern mobile devices are capable of supporting a wide range of new innovative applications from real-time location-based tracking to mobile gaming. However, the usage of power-hungry applications, sensors and their demand for 24/7 Internet connectivity requires an efficient energy management mechanism in mobile devices. With the increasing energy limitations, there has been a corresponding rise of energy management solutions proposed by researchers. However, this research area is still immature and existing literature lacks the critical review of recent self organization based energy management techniques. This paper aims to provide a structured overview of the research developments on self organization based energy management techniques used in mobile complex networks. This review paper surveys the state-of-the-art self organization based energy management techniques that have been proposed over the period of 2010–2015. Based on the proposed optimization, we have grouped the existing approaches in different categories, which are further classified at different levels, from energy-efficient operating systems to computation off-loading. With this classification we aim to provide an easy and summarized view of the latest self organization based energy management techniques that can be implemented in mobile devices.

Keywords: Energy management; Operating system; Self organization; Complex networks; Mobile devices

Introduction

Mobile industry has experienced a tremendous evolution and growth in recent years, beginning with simple talk and type devices to modern smartphones. All the manufacturers want to provide best features in their mobile devices. As a result, in today's market a wide variety of mobile devices are available under various brands, with different operating systems and features on a competitive edge. Today's modern mobile devices are equipped with more powerful processors, operating systems and have a range of computational, memory, sensing, communication, and storage resources. They are as powerful as the personal computers and are capable to support a wide range of new innovative applications from real-time location-based tracking to mobile gaming. Many users and businesses heavily rely on mobile devices for emails, communication, sensitive data and business, transactions etc.

Moreover, due to the continued evolution of mobile communication technologies, cellular and other wireless networks are becoming more complex over time. In modern large-scale networks, interaction of components often involves complex communications, which can have unpredictable effects on different aspects of networks such as communication costs, traffic congestion and so on. Due to the intrinsic nonlinearity, modern networks can be modeled and simulated in a better way by treating them as artificial Complex Adaptive Systems (CAS), or generalizing as Complex Adaptive COmmunicatiOn Networks and environmentS (CACOONS) (Niazi and Hussain 2013a) (Niazi and Hussain 2013b). Agent-based modeling (ABM) and complex networks-based modeling (CN) are the two complexity-based multidisciplinary techniques that can be used for simulation and modeling modern large-scale, mobile, social, and communication networks (Niazi 2013) (Niazi and Laghari 2012) (Batool K. 2015).

Mobile devices are battery driven to provide users with the maximum freedom and it should empower all the exciting and new features available. Some mobile devices come with power-saving features which often employ self organization based energy policies to adjust the power consumption of hardware components as per the required functionality. But such improvements at hardware level are very slow as compared to the rapidly growing energy demands. Moreover, with increasing availability of wireless interfaces and sensors, use of rich mobile applications such as social networking applications, data transfer applications, and location-aware services etc. has also increased. Unfortunately, these applications consume battery of mobile devices rapidly and reduce it to few hours. Hence, to get the maximum productivity of mobile devices a sound understanding of their energy requirements is required.

Nowadays, energy efficiency is considered an essential feature for any mobile device, particularly for smartphones. Excessive power consumption is holding back the evolution of mobile devices because the improvements at the battery levels are not significant in comparison to the rapidly growing energy demands for wireless interfaces, sensors, applications and features. Thus, the power management in mobile devices has become a major concern for everyone including software engineers, operating system designers, hardware manufacturers, and researchers. Several solutions have been proposed to reduce the power consumption in mobile devices at hardware, operating system, and application level (Vallina-Rodriguez and Crowcroft 2013).

In this paper, we will critically review the recent self organization based energy management techniques used in mobile complex networks. We will discuss and analyze different software and hardware solutions suggested for energy efficiency in mobile devices. The study intends to classify and provide a summary of the recent research work done in the domain of energy management for reducing power consumption in mobile devices.

The rest of the paper is organized as follows: Section II reviews the background, Section III presents self organization based energy management techniques, Section IV includes performance evaluation of different techniques, Section V contains summary & findings and Section VI concludes the paper.

Background

Before we analyze the solutions for efficient energy management, we must understand the energy consumption in today's modern mobile devices. A study published in 2011 (Perrucci et al. 2011), measured the power consumption by different smartphone

components using the smartphone Nokia N95 (Symbian OS 9.2). Each component was measured separately, in more detail than shown in Table 1. For example, they measured wireless interfaces, such as Bluetooth, Wi-Fi etc., all under different power modes and at different states. Table 1 only includes summarized average power consumption by different smartphone components.

The overall energy consumption in smartphones heavily depends on their usage. Fig. 1 shows the summarized results of a recent report regarding the typical mobile devices usage (Perez 2014). The study measured time spent in various activities while using mobile devices. Gaming dominates with 32% of time spent, followed by the facebook with 17% of time consumed. Facebook, plus social messaging applications (9.5%) and twitter (1.5%) shows 28% of time. Meanwhile, entertainment and utility applications accounted for 16% of time spent. The above findings suggest that in mobile devices energy efficiency can be achieved at different levels particularly through optimization of operating systems and wireless interfaces.

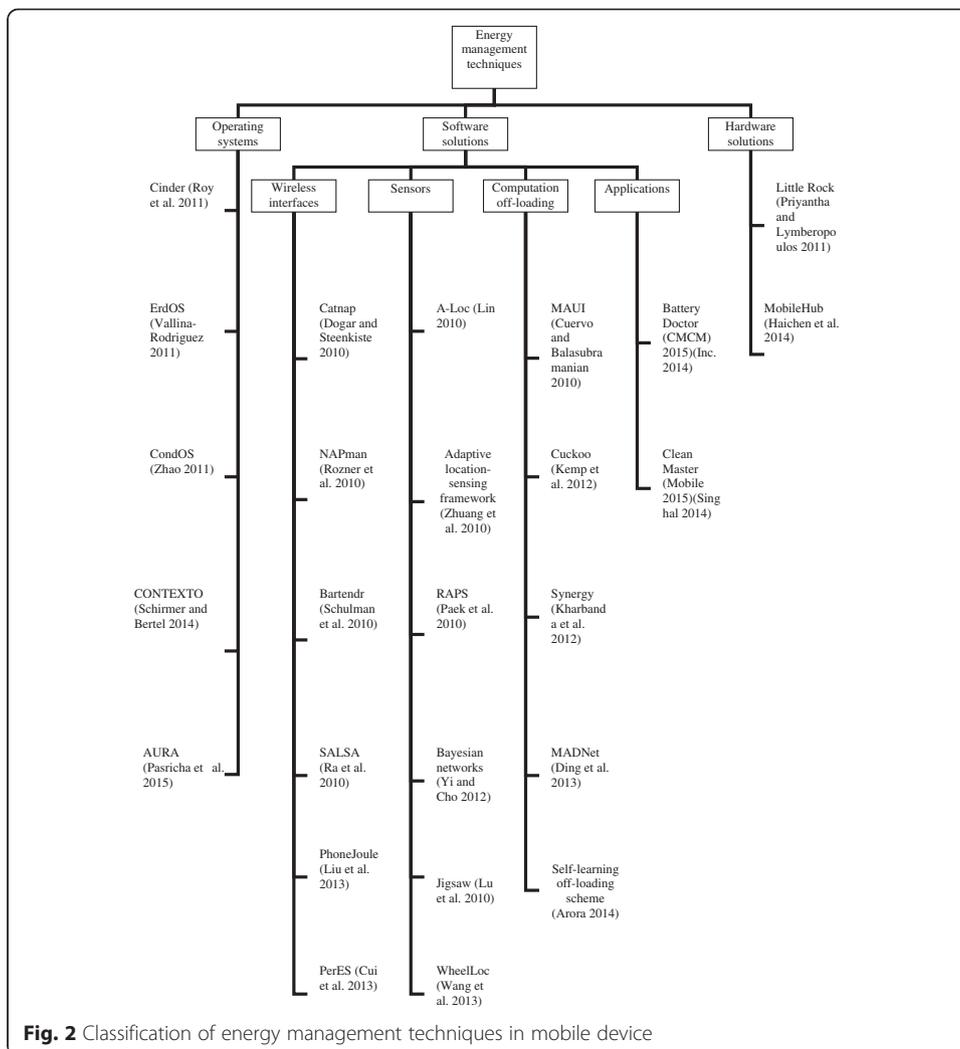
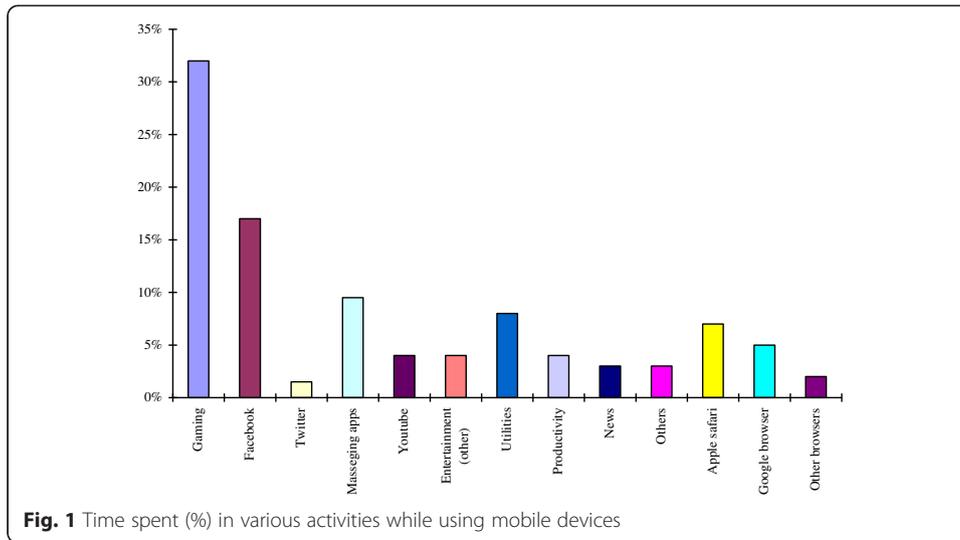
The previous research lacks the detailed analysis of the latest approaches used to reduce energy consumption in mobile devices. This lack in existing literature prompted to review and provide a comprehensive analysis of the latest self organization based energy management techniques that can be implemented in mobile device.

Review

In this section, we will discuss and analyze different solutions for energy efficiency in mobile devices. Based on the proposed optimization, the existing energy management techniques have been grouped in three categories namely Operating systems, Software solutions and Hardware solutions. Software solutions are further classified at four major axis: energy- efficient communication via wireless interfaces optimization, energy-efficient sensing via sensors optimization, energy- efficient computation off-loading via cloud and battery management via mobile applications. Fig. 2 shows the classification of energy management techniques.

Table 1 Power consumption by different smartphone components(Perrucci et al. 2011)

Technology	State/Action	Power (mW)
Wi-Fi	In connection	868
	Idle	58
3G	Downloading@1Mbps	1400
2G	Downloading@44 kbps	500
Bluetooth	On	15
	Connected and idle	67
	Receiving/sending	432
CPU usage	50%	462
	75%	561
	100%	612
Display	Black light background - 60%	98.65
	White light background - 60%	254.65
	Black light background - 100%	259.65
	White light background - 100%	527.05



Operating systems

An important concern about energy efficiency is “How and by whom the energy-aware policies should be performed?” Some researchers believe that applications must dynamically conform to energy requirements. However, other researchers suggest that entire energy management should be done within the operating system. At operating system level, energy consumption can be reduced by unifying resources and leveraging cooperation between applications and operating system. The essential part of managing resources efficiently is to have a sound understanding of how applications and users demand system resources.

The idea of an energy-efficient operating system was first proposed in the 90s. Prominent works on energy-efficient operating systems include Quanto (Fonseca et al. 2008), ECOSystem (Zeng et al. 2002), and Odyssey (Noble et al. 1995). In Odyssey and ECOSystem both operating system and applications mutually strive for energy efficiency. Nevertheless, they were developed for embedded systems and laptops and their performance is compromised when used for mobile devices with complex user interaction patterns. In 2000, researchers pointed out that energy efficiency should also be considered while maximizing operating system performance. Unfortunately, not much research was carried out on this issue till mid 2000s. However, in recent years due to the battery limitations of mobile devices in which most applications consume battery of mobile devices so rapidly to reduce it to few hours, the problem has gained researchers attention (Kundu and Paul 2010) (Jindal 2012).

Cinder (Roy et al. 2011) is one of the few energy-efficient mobile operating system. Cinder is designed on top of the HiStar (Zeldovich et al. 2011) secure exokernel using the HTC Dream as prototype mobile platform. Cinder aims to track entities responsible for resource consumption and to provide efficient power distribution through isolation, subdivision and delegation. For resource management and control, Cinder complements the work of ECOSystem (Zeng et al. 2002) & Quanto (Fonseca et al. 2008) and takes advantage of device-level accounting and modelling. Unlike previous approaches, Cinder traces the services and applications responsible for resource consumption and allows assigning their resources in terms of rates or quantities.

Cinder uses a resource consumption graph and two notions (taps and reserve) to allocate energy to applications. The device battery is represented as the root reserve in the resource consumption graph. When an application use a resource, meanwhile the values in reserve reduces and more threads are allowed to run only if there are enough reserves. Taps control the reserves consumption rate and once all the reserves are consumed no more threads are allowed to run. Experimental results showed that in total, 12.5% less energy is used for an equivalent amount of work within the same time interval. Moreover, the authors claim that Cinder operating system is designed around fine-grained resource accounting and control as well as security. To provide strong security, Cinder exploits the information flow control facilities of the HiStar (Zeldovich et al. 2011) secure exokernel.

Unlike previous cases, ErdOS (Vallina-Rodriguez 2011) presented a different approach for energy-efficient operating system. ErdOS is designed as an extension of Android operating system. The main idea behind ErdOS was the fact that the typical energy management scheme used by mobile operating systems is not efficient, where operating system strives to save the battery of mobile device by using the resources

selectively. Decisions are mostly based on static algorithms and policies. They do not consider the diversity of interactions, habits of mobile users and the importance of context on resource availability. Moreover, the usage patterns can cause complex interdependencies between software and hardware components.

ErdOS proposes proactive resource management aided by contextual information gathered by monitoring resources state, user's interaction patterns, and applications resource's demands and by learning from user's behaviour and habits. In addition, ErdOS also supports the idea that accessing remote resources in co-located devices opportunistically is beneficial in terms of usability, energy efficiency, and quality of service. To achieve this ErdOS exploits social networks information and low energy wireless interfaces to provide access control to nearby devices in a secure way. However, the paper lacks performance evaluation.

CondOS (Zhao 2011) is a context-aware operating system that efficiently manages available sensing resources to reduce power consumption in mobile devices. CondOS work is based on the argument that sensing resources should be managed at operating system level rather than application level. Although, allowing applications to manage the sensing resources independently appears appealing as they are most aware of their own context needs but applications cannot maintain context when outside the execution scope. CondOS operating system converts raw sensor data to Contextual Data Units (CDUs) and allows both applications and operating system services to query CDUs.

CondOS maintains CDU context vocabulary and allows applications to expand the CDU vocabulary by adding their own CDUs. CondOS's centralization of context generation results in energy reduction, like principled flow degradation and shared dataflow processing. The authors claim that making operating system responsible for managing sensing resources is beneficial not only in terms of energy savings but it also provides other benefits as well like better scheduling, I/O, memory management, and security. However, the paper lacks clear performance evaluation.

With the growing market for context-aware applications, energy requirements of such applications have become one of the major concerns for the mobile application developers. CONTEXTO (Schirmer and Bertel 2014) is a context classification and recognition framework that specifically assists mobile developers for developing energy-efficient context-aware applications. Concept and implementation of CONTEXTO highly benefit from the previous research work including Context Toolkit (Dey et al. 2001) and Hydrogen Context-Framework (Hofer et al. 2003). Prototype is currently available as an iOS framework, but it can easily be extended, modified, or customized to other mobile operating systems.

CONTEXTO framework is complemented with an Energy Demand Calculator that provides information about the energy demands for all used components (sensors and features) on a specific device. Within the framework, this information is made available to developers early in the design process and at runtime. The calculator also roughly estimates the time a component would need to completely drain a charged battery. Moreover, the calculator allows developers to optimize their application by trying various combinations of information sources and visualizing their impact on energy demand.

Recently Pasricha *et al.* (Pasricha et al. 2015) proposed an application-aware and user-interaction aware energy management middleware framework (AURA) for mobile

devices. The main idea behind AURA is to optimize CPU and backlight energy consumption without diminishing user experience with the device. To do so, AURA makes use of the user-specific patterns of interactions with applications and Bayesian application classifier to classify the applications at runtime. After the application is classified, AURA utilizes the management strategies based on Markov Decision Processes or Q-Learning to set the CPU frequency and backlight level. The prototype for proposed framework was implemented as a middleware on two android based smart phones, HTC Dream and Nexus One. Evaluation studies showed that AURA can achieve up to 29% energy savings as compared to the baseline device manager and it is 5 times more energy efficient than previous approaches in this domain.

Software solutions

Energy-efficient wireless interfaces

Due to the continued evolution of mobile communication technologies, cellular and other wireless networks are becoming more complex overtime. Moreover, with increasing availability of wireless interfaces like cellular networks (GPRS/EDGE/3G/4G) and Wi-Fi, use of rich mobile applications such as social networking applications, data transfer applications, location-aware services etc. has also increased. These wireless interfaces are major power drainers in mobile devices and continuous network activity consumes a significant portion of the device battery. Cellular network consumes less amount of energy as compared to Wi-Fi to keep it continuously awake. However, Wi-Fi requires very less energy to transfer the data as compared to cellular networks. In addition, cellular network is available almost everywhere whereas Wi-Fi is not and the strength of Wi-Fi varies from location to location (Ljung 2011).

With these increasing energy limitations, it has become essential to improve the energy efficiency of wireless interfaces. To make efficient communication possible, several mechanisms are proposed in research literature for the optimization of wireless interfaces. Some of them save energy during Wi-Fi communication by delaying communication or rescheduling data transfer until low energy network is available like Bartender (Schulman et al. 2010) and SALSA (Ra et al. 2010). In addition, there are various energy saving mechanisms that exploit idle opportunities during data transfers present at different levels, targeting different applications and network scenarios like Catnap (Dogar and Steenkiste 2010).

Catnap (Dogar and Steenkiste 2010) is based on the fact that IEEE 802.11 standards support higher data rates than wired networks. Catnap strives to extend the battery by allowing power-hungry wireless interfaces to sleep during packets transfers. To do so, Catnap exploits high bandwidth wireless interfaces to combine small gaps between data transfer into bigger sleep slots. Catnap works in collaboration with a middle-box, which differentiates between the wired segment and the wireless one. Catnap buffers the data at middle-box and sends it in a burst on high bandwidth wireless link so that device can sleep for rest of the time. Catnap extends TailEnder's (Balasubramanian et al. 2009) batching mechanism to improve energy efficiency of transmissions but it offers broader applicability by allowing the batching of both outgoing and incoming data. In contrast to the previous work, Catnap focuses on peak network activity periods and it leverages a range of sleep modes. Results revealed that Catnap allows a significant long sleep time for mobile wireless interface especially for larger data transfers and improve the

battery life up to 2-5x. However, it only targets data oriented applications that can afford delay in packets transfer and its performance can be compromised due to congestion on wireless side as the transfer time may increase.

NAPman (Rozner et al. 2010) work is based on the argument that implementation strategies for energy-conserving Power Save Mode (PSM) used by the clients/Access Points (APs) are not efficient. The author also highlighted the negative consequences of this approach in case of competing background traffic e.g. decreases in network capacity, a notable increase in client's power consumption, and unfairness. NAPman is a network assisted power management system for Wi-Fi devices that provides solution for all the above problems. To avoid unnecessary retransmissions and reduce power consumption NAPman relies on an energy-aware fair scheduling algorithm and AP virtualization. Results revealed that NAPman can reduce energy consumption up to 70% on a smartphone under different settings of background traffic. NAPman is easy to implement via software updates and does not require any changes to the mobile clients or the 802.11 protocol. However, there are also some limitations such as it is not evaluated for applications that require QoS such as gaming or VoIP. In NAPman latency increases with the higher background traffic as compared to the High Priority 802.11 PSM.

Two publications regarding energy management during Wi-Fi communication are Bartender (Schulman et al. 2010) and SALSA (Ra et al. 2010). Both papers exploit the fact that rescheduling data transfer or delaying communication can significantly reduce the energy consumption. Bartendr (Schulman et al. 2010) is an energy-aware cellular data scheduling system that strives for energy-efficient application communication by scheduling non-urgent communication with periods of strong signal. To accomplish this, Bartender performs track-based prediction of future signal strength at little energy cost. An energy-aware scheduling algorithms use the anticipated signals strength as input to schedule syncs and streaming applications. Next sync time is determined from the current location. While, for the scheduling of streaming applications time is divided into slots and data is divided into frames, taking into account the playback deadline and tail energy. Results showed significant reduction in energy consumption for both sync and streaming applications with the minimum energy overhead. However, it does not take into account wireless link quality and delay impact that makes it difficult to apply a prediction based scheduling algorithm.

Likewise, SALSA (Stable and Adaptive Link Selection Algorithm) (Ra et al. 2010) presents an online algorithm for the energy-delay trade-off. It is focused on delay-tolerant applications that can delay the upload until a low-energy Wi-Fi connection becomes available. To overcome the limitations of Bartendr (Schulman et al. 2010), it employs a non prediction-based Lyapunov optimization framework (Wen et al. 2012) (Neely 2006) to chose the most suitable link for a communication without increasing delay. Its transmission decision is based on several factors like data backlog, energy expenditure, fairness, channel quality, and throughput maximization. Evaluation results showed that SALSA can be used to achieve significant energy-delay tradeoffs and it can save up to 40% of device battery. Nevertheless, being application specific is the limitation; SALSA can only be used for video uploading applications.

PhoneJoule (Liu et al. 2013) is another energy management system based on Android operating system, which intelligently monitors CPU usage and conservers device energy

by using different saving modes and switching between the wireless networks like GPRS and Wi-Fi. It can work on all handsets that support Android OS 2.2 or later versions. Evaluation results proved effectiveness of PhoneJoule for energy savings in smartphones. Moreover, PhoneJoule makes it very easy for users to monitor battery usage of their mobile devices.

Finally, PerES (Cui et al. 2013) is the recent work that addresses the limitations of the previous peer schemes. The motivation behind PerES is the fact that existing prediction based algorithms are not suitable for a practical wireless system, considering the dynamic channel conditions and user traffic. PerES present an efficient online scheduling algorithm that reduces power consumption due to both transmissions and tail energy, while taking in account the user expectations regarding device performance. The evaluation results revealed that the performance of PerES is better than previous approaches like TailEnder (Balasubramanian et al. 2009) and SALSA (Ra et al. 2010).

Energy-efficient sensors

With the increasing availability of diverse range of built-in sensors in mobile devices, the usage of rich mobile applications especially location-aware applications has also increased. It requires aggressive use of sensors with different resolutions and energy requirements. Consequently, there is a trade-off between accuracy and power consumption. Hence, energy-efficient continuous sensing has become a hot concern for both researchers and developers of mobile applications. Many previous works explored the accuracy-energy trade-off using a variety of available location-sensing mechanisms. Several schemes were proposed for energy-efficient sensing, which combine different sensors to minimize the energy consumption as well as the error. Most of them are based on probabilistic models of user's history and patterns to deduce future locations. The main idea is to minimize the sensing reads and using less expensive sensors more as in A-Loc (Lin 2010), RAPS (Paek et al. 2010) and Jigsaw (Lu et al. 2010).

A-Loc (Lin 2010) proposes an adaptive location-sensing service for mobile devices that dynamically trade-off location accuracy and energy. This is achieved by incorporating probabilistic models of user location and sensor errors. A-Loc automatically determines the application accuracy requirements, if not explicitly specified by the application and based on these requirements selects the most suitable location sensor available. The proposed solution was prototyped as a middleware for Android devices. Results showed that this approach provides significant energy savings and higher accuracy than existing techniques.

Zhuang *et al.* (Zhuang et al. 2010) combined four design principles from the previous works and presented an adaptive location-sensing framework that notably reduces power consumption in mobile handsets running location-based applications. The elementary design principles include, substitution of alternative location-sensing mechanisms, accelerometer-based suppression, piggybacking the location-sensing requests and finally the alteration of sensing parameters based on the remaining battery life. Similar to A-Loc (Lin 2010), this system was also implemented as a middleware layer on the Android operating system. Evaluation results showed significant improvements in battery life up to 75 % by reducing the energy-consuming GPS usage by up to 98%.

Concurrent with Zhuang's work (Zhuang et al. 2010) , Paek *et al.* (Paek et al. 2010) proposed the similar optimization to reduce power consumption in smartphones by

minimizing the number of GPS reads. RAPS (rate-adaptive positioning system) (Paek et al. 2010) leverage the observation that in urban areas GPS is generally less accurate and it uses a collection of techniques for selective activation of GPS, only when it is necessary to achieve the accuracy. The system uses accelerometer data as input and estimate user velocity. To improve the accuracy, it utilizes Bluetooth to synchronize GPS readings among nearby mobile devices. Finally, it exploits GSM cell tower information to blacklist the GPS unavailable areas, which further reduces the GPS activation. Results showed that combining these features altogether can reduce the energy consumption around 40% as compared to the approach where GPS is always on.

Yi *et al.* (Yi and Cho 2012) presented an energy management system for GPS sensor using Bayesian networks. The proposed system employs a probabilistic model and offers user to control the GPS based on the user's context information. It utilizes various built-in low power sensors to gather context information, which is then pass through pre-processing in order to obtain proper variables. The variables are then inputted to Bayesian network, which deduce whether user's location is indoor or outdoor. The prototype solution was implemented as an application for Android devices and results showed energy saving up to 5% per hour. However, the suggested scheme was appropriate for active users only.

In contrast to previous cases, Jigsaw (Lu et al. 2010) looks at the challenges associated with efficient continuous sensing on mobile devices. Motivation behind this was the fact that one of the most power-consuming tasks for continuous sensing is to process raw sensor data from multiple sensors. Jigsaw presents a middleware solution that refines the resolution of three most common sensors, (i.e. GPS accelerometer, and microphone), adaptively adjusts the sensing process based on nature of input data and selectively activates power-hungry sensors like GPS only as often as required. The authors claim that the system can be extended for new sensors and is capable of performing long-term energy-efficient GPS tracking. However, the paper lacks clear performance evaluation.

WheelLoc (Wang et al. 2013) is another continuous location-sensing service that deviates significantly from the accuracy-energy trade-off path. Unlike previous approaches, WheelLoc strives to obtain the location using low-power sensors only while completely avoiding the energy-expensive sensors. To do so, WheelLoc first obtain the user mobility trace by magnetometer and then obtain user location by accelerometer. It exploits the cell tower location information and road maps to maintain required accuracy. Evaluation results revealed that WheelLoc gives an improved accuracy-energy trade-off than GPS.

Energy-efficient computation off-loading

Demand for mobile rich applications is growing but unfortunately, mobile devices have limited resources such as processor, memory, bandwidth, and storage. To overcome these limitations the concept of computation off-loading came forward. Capabilities of mobile devices can be augmented by migrating computation intensive tasks to more powerful servers. Computation off-loading has shown to be useful for improving battery capacity of mobile devices but it depends on different parameters such as the amount of data exchanged and the network state. Over the past decade, extensive research has been performed on computation off-loading and the related issues (Kumar

et al. 2013)(Lee et al. 2013). Several schemes have been proposed to assist off-loading decisions, as computation off-loading is not always energy-efficient until it is energy-aware.

Many researchers have utilized remote execution in their work to improve the performance and resource availability for mobile applications. However, unlike previous work, Cuervo *et al.* (Cuervo and Balasubramanian 2010) analyzed remote execution to improve the energy conservation in mobile devices. MAUI is architecture for process migration, which dynamically selects the methods for remote executions. To do so, MAUI uses a profiler to monitor resources and based on their states predict the process behaviour. Based on the profiler data, an optimization engine then suggests the most suitable energy saving scheme under the current connectivity constraints. In this way, code is off-loaded to a remote server only if predictions suggest that such execution will save energy. MAUI supports full process migration as well as it gives users the option to explicitly specify program states for remote execution. It does not require any separate interface or component for off-loading code and it is a less obtrusive approach for application developers, as it does not need strict separation for off-loadable code parts. Evaluation results showed that by using this approach energy conservation of up to one order of magnitude can be achieved.

Cuckoo (Kemp et al. 2012) is another computation off-loading framework that speeds up the computation while reducing energy consumption for Android applications. It provides a runtime system that dynamically decides where the execution of application will take place, locally or remotely. The decisions taken are based on the application proxy. Cuckoo framework requires an application developer to define interfaces and is highly dependent on the domain expertise of the developer. However, to simplify the development process it offers a development model for application developers, which integrates with the Eclipse built system. Moreover, mobile devices are not always connected to the network which makes cloud resources sometimes unavailable. To overcome this limitation, Cuckoo's programming model supports both remote as well as local methods implementation. The authors claim that with little effort computation off-loading can be enabled for object recognition and gaming application, using the Cuckoo framework. However, there are some limitations; it does not consider the security measures for communication and the additional energy overhead introduced by off-loading.

Kharbanda *et al.* (Kharbanda et al. 2012) proposed a middleware solution that can be used as an alternative, in cases where cloud is unavailable. Synergy (Kharbanda et al. 2012) extends the battery of a peer-to-peer ad-hoc network of mobile devices. It treats the mobile devices connected in network as a single multi core device interconnected by Wi-Fi. Energy conservation is achieved by scaling the frequencies and wisely distributing the processing workload among devices connected in a peer-to-peer fashion. Synergy carefully chooses the number of devices to distribute the processing, based on the estimated amount of energy consumed by data transmission between the devices. Results revealed that Synergy prototype for Android operating system can save up to 30.6% of the system battery with less than 5% latency penalty. However, Synergy has few limitations such as it has data privacy issue and it only targets CPU intensive applications. Other applications with low CPU utilization or significant amount of communication cannot gain from Synergy.

In 2013, Han *et al.* (Ding et al. 2013) proposed an architecture for off-loading mobile traffic from cellular networks. Motivation behind MADNet (Ding et al. 2013) was the argument that transferring the same amount of data on a low throughput Wi-Fi network may consume more energy than transferring over a high speed 3G. MADNet enables smartphones to select the most suitable energy-efficient Wi-Fi AP to off-load mobile data traffic while avoiding the off-loading to low throughput Wi-Fi networks. MADNet provides an energy-aware algorithm to assist the off-loading decision. Off-loading is performed only if the Wi-Fi APs saves more energy than extra energy consumption overhead. It employs Wi-Fi interfaces to prefetch predictable data that further improves energy efficiency of data transfer. Moreover, MADNet leverages cellular operators, Wi-Fi providers, and end-users to increase the off-loading opportunities. Results revealed that using this approach up to 80% energy saving can be achieved.

Recently Arora *et al.* (Arora 2014) proposed a new off-loading scheme based on self learning. The key idea was to reduce the overhead associated with decision-making process by allowing system to self learn from its off-loading decisions. This can be achieved by maintaining a database of previously taken off-loading decisions along with the input parameters. Future off-loading decisions can be made by first looking in the database and then off-loading decision block runs only if the entry is not available in the record. Authors claim that using this scheme not only makes system more energy-efficient but also more reliable and fast. However, the paper lacks comprehensive evaluation of proposed scheme.

Battery management mobile applications

With the increasing energy limitations in mobile devices, battery management mobile applications have become a must-have. Some mobile devices has built-in battery management features, however, several battery monitoring and power saving applications are also available which can be used to achieve significant energy savings (Corpuz 2014) (Geier 2011).

Battery Doctor (CMCM) 2015)(Inc. 2014) is a battery saving application that can monitor and regulate power consumption. It allows users to quickly look up battery status as well as track down what applications are draining battery life. It has a very useful power optimization feature that allows users to define power saving profiles and automatically schedule them. It also helps to charge the device healthily with 3 Stage Charging System. Battery Doctor is available for both android and iOS platform and it can extend the battery up to 50%.

While using internet on mobile devices, users unintentionally collect a lot of extra data like residual files, histories etc. It can take up valuable space and subsequently slow down the device. Perhaps, Clean Master (Mobile 2015)(Singhal 2014) is one of the best applications that can help users manage the cleaning up of their device. This app is multifunctional which works like a cleaner and as an antivirus at the same time. Clean Master gives mobile device a good spring clean by removing junk files and optimizing memory, which in turn makes it speedier and more efficient. It also protects the device from unwanted malware and spyware that can be transferred on the device from some unsecure applications. Clean Master is currently available for android devices and it can boosts mobile applications by almost 32%.

Hardware solutions

Apart from the optimization at software level, some researchers also demonstrated the need to re-design the system architecture at the hardware level for energy-efficiency (Samreen 2015) (Priyantha and Lymberopoulos 2011) (Haichen et al. 2014).

Similar to Jigsaw (Lu et al. 2010) and WheelLoc (Wang et al. 2013), Little Rock (Priyantha and Lymberopoulos 2011) addresses the same power overhead issues associated with continuous sensing on mobile devices. However, it uses a different approach of re-designing the mobile phone architecture at hardware level. Little Rock sensing architecture incorporates a small energy-efficient co-processor that is connected to all available sensors, forming a sensor hub. It strives to conserve energy by off-loading the sampling and processing of raw sensor data, if possible, to the dedicated processor. While the co-processor is acquiring and processing sensor data, Little Rock allows the phone to sleep. Moreover, it facilitates the fast on demand data exchange between two processors. The architecture was evaluated using a pedometer application and results showed it three times more efficient as compared to the existing solutions.

Priyantha's work (Priyantha and Lymberopoulos 2011) proved the effectiveness of sensor hub and motivated researches to expand his work. Extensive research is being carried out to realize the potential of sensor hub in the area of energy-efficient sensing. Recently, Haichen *et al.* (Haichen et al. 2014) proposed a translation-based approach that allows existing Android applications to seamlessly leverage low power sensor hub hardware. MobileHub rewrites the byte code of sensing application without accessing the source code and with zero developer effort. Experimental results showed that MobileHub could improve power consumption by up to 83%, with no effort from the developer.

Performance evaluation

It is not possible to make a quantitative comparison of all the energy management techniques at different levels due to the variation of evaluation methodologies/setup and mobile platforms used in the research papers we studied. In this section, we will sum up all the energy measurement techniques we have reviewed above, in the form of a table. Table 2 highlights the classification of techniques based on whether it's related to operating system/software/hardware level, the prototype implementation, whether or not the proposed scheme is evaluated, if yes then what experimental setup/methodology is used for evaluation and to what extent the proposed solution is energy-efficient.

Summary and findings

Although today's mobile devices offers a variety of power-consuming resources such as location sensors, touch screen displays, and wireless interfaces but they do not consider energy efficiency as an essential property. For efficient energy management it is important to have a sound understanding of when, where and how users interact with mobile devices and what are their energy requirements. Consequently, this information can be translated into significant energy saving patterns. Table 3 highlights the summary and findings of our research work.

Table 2 Performance evaluation of energy management techniques

Energy management technique	Implementation	Evaluated YES/NO	Evaluation setup/ methodology	Energy consumption	
OPERATING SYSTEMS					
Cinder (Roy et al. 2011)	Abstractions are implemented in the Cinder kernel, which runs on AMD64, i386, SPARC, and ARM architectures. It is freely available on Internet	Yes	HTC Dream (Google G1), based on the Qualcomm MSM7201A chipset	12.5% total system power reduction over the 20min experiment	
ErdOS (Vallina-Rodriguez 2011)	Prototype is implemented as an extension of the Android OS	No	N/A	Authors claim that it improves the battery capacity of smart phones by managing resources proactively	
CondOS (Zhao 2011)	N/A	No	N/A	Authors claim that it provides several opportunities for energy reduction, such as shared dataflow processing, dataflow-to-hardware mapping, and principled flow degradation	
CONTEXT0 (Schirmer and Bertel 2014)	Prototype is currently available as an iOS framework	Yes	Apple iPhone 4, 4S, and 5	Provides energy-awareness to developers of context-aware applications	
AURA (Pasricha et al. 2015)	Prototype is implemented as a middleware on two android based smart phones	Yes	HTC Dream and Nexus One	Can achieve up to 29% energy savings as compared to the baseline device manager & it's 5 times more energy efficient than previous approaches	
SOFTWARE SOLUTIONS					
Energy-efficient Wireless Interfaces	Catnap (Dogar and Steenkiste 2010)	Prototyped in C for the Linux environment	Yes	Nokia N810 and IBM Thinkpad T60 both supporting 802.11 PSM	Allows the NIC to sleep for around 40% of the time for a 10MB transfer while 70% of the time for a 5MB transfer. Improves battery capacity up to 2-5x for real devices like Thinkpad T60 & Nokia N810
	NAPman (Rozner et al., 2010)	Prototyped using the MadWifi v0.9.4 driver for Atheros-based WiFi cards on the Linux platform	Yes	HP iPAQ hw6945, iPhone 3GS, gPhone HTC Magic and HTC Tilt 8900	Under varied settings of background traffic, it improves the energy savings on a smartphone by up to 70% while ensuring fairness

Table 2 Performance evaluation of energy management techniques (*Continued*)

	Bartendr (Schulman et al. 2010)	N/A	Yes	4 cellular networks across 2 metropolitan areas, one in US & the other in India, and spans 3G networks based on both EVDO & HSDPA	Significant energy savings of up to 10% for email sync and up to 60% for on-demand streaming
	SALSA (Ra et al. 2010)	Implemented SALSA algorithm in Urban Tomography system which runs on the Nokia N95 smartphone, having 802.11b/g WiFi interface, 3G/EDGE, a 2GB micro-SD card, & supports 640x480-resolution video recording capability	Yes	Nokia N95 and Android G1	Closer to an empirically determined optimal than any other alternatives compared with it, and, can save 10-40% of battery for some workloads
	PhoneJoule (Liu et al. 2013)	Prototype is implemented using java and Eclipse integrated with Android SDK & ADT. It can work on all smartphones which support Android OS 2.2 or later versions	Yes	ZTE v880 smartphone which supports Android OS 2.2 and SEMO to measure power consumption	Very effective for energy saving in smartphones and makes it very convenient for users to manage battery usage of their smartphones
	PerES (Cui et al., 2013)	Implemented as a traffic management application by utilizing IPTABLES (a system tool in Android)	Yes	Google Nexus S and Monsoon Power Monitor device	Better than peer schemes, TailEnder & SALSA. Using 821 million traffic flows collected from commercial cellular carrier, it can achieve on average 32% to 56% energy savings with different levels of user experience
Energy-efficient Sensors	A-loc (Lin 2010)	Prototype is implemented on an Android G1 phone	Yes	Android G1 and AT&T Tilt phones, on paths that include indoor and outdoor locations, using war driving data from Microsoft & Google	Saves significant amount of energy and also improves the accuracy
	Adaptive location-sensing framework (Zhuang et al., 2010)	Design principles are implemented as a middleware on G1 Android Phone with OS version 1.5 Cupcake, by modifying the Application Framework	Yes	G1 Android Developer Phone (ADP)	Minimize the usage of the energy-consuming GPS up to 98% and improve battery life by up to 75%
	RAPS (Paek et al., 2010)	Prototype is implemented in Symbian C++ for the Symbian S60 3rd FP1 devices	Yes	Nokia N95-3 smartphone, with GPS, accelerometer, Bluetooth, WiFi and 3G/EDGE interfaces, & 2GB micro-SD card	Can increase phone battery by more than a factor of 3.8 as compared to the approach where GPS is always on
	Bayesian Networks (Yi and Cho 2012)	Proposed context-aware system for GPS has prototyped as an application in Android platform	Yes	LG SU-660 with Android OS 2.2 version	Active person and inactive person can save energy of about 5% and 3% per hour, respectively

Table 2 Performance evaluation of energy management techniques (*Continued*)

	Jigsaw (Lu et al. 2010)	Proposed continues sensing engine has implemented on two smartphone platforms, Nokia N95 & Apple iPhone, as background service and library, respectively	Yes	Nokia N95 and Apple iPhone	Authors claim that Jigsaw is capable of performing long-term energy efficient GPS tracking without sacrificing the accuracy. However, the paper lacks clear performance evaluation results
	WheelLoc (Wang et al. 2013)	Implemented as a continuous background system service on NexusOne phones running Android 2.3	Yes	NexusOne phones with Android OS 2.3 version	Can return a location estimate within 40ms with an accuracy about 40 meters, consumes only 240mW energy, & effectively strikes a better energy-accuracy tradeoff than GPS duty-cycling
Energy-efficient Computation-offloading	MAUI (Cuervo and Balasubramanian 2010)	Prototype is implemented on HTC Fuze smartphone running Windows Mobile 6.5 with .Net Compact Framework v3.5	Yes	HTC Fuze smartphone running Windows Mobile 6.5 with .Net Compact Framework v3.5 and for MAUI server, dual-core desktop with 3GHZ & 4GB RAM running Win 7	For 4 applications running on Windows Mobile phones, it can achieve energy conservation of up to one order of magnitude
	Cuckoo (Kemp et al., 2012)	Integrates with the popular open source Android framework and the Eclipse development tool	Yes	2 real world apps that contain heavy weight computation, eyeDentify and PhotoShoot	With little effort computation off-loading can be enabled for object recognition and gaming app, using the Cuckoo framework
	Synergy (Kharb et al., Kharbanda et al. 2012)	Prototype implementation is developed for the Android operating system	Yes	2 compute intensive apps – image smoothing & video processing running on Google Nexus S with Android OS 2.2 version and PowerTutor	Can save up to 30.6% of the system battery with less than 5% latency penalty
	MADNet (Ding et al., 2013)	Prototype is implemented on Nokia N900 smartphones	Yes	Nokia N900, Nokia E71 and Samsung Nexus S. Energy Profiler application and Monsoon Power Monitor	Can achieve more than 80% energy saving
	Self learning off-loading scheme (Arora 2014)	N/A	No	N/A	Authors claim that enabling the off-loading system to self learn makes it more reliable, fast and energy efficient
Battery Management Mobile Applications	Batter Doctor (CMCM) 2015) (Inc.2014)	Android and iOS platform	Yes	Android and iOS	It allows users to quickly look up battery status as well as track down what applications are draining battery life. It also helps to charge the device healthily with 3 Stage Charging system and it can extend the battery life up to 50%

Table 2 Performance evaluation of energy management techniques (*Continued*)

Clean Master (Mobile 2015)(Singhal 2014)	Android	Yes	Android and AV-TEST	Multifunctional application which works like a cleaner and as an antivirus at the same time. It can boost mobile applications by almost 32% and it also protects the device from unwanted malware & spyware
HARDWARE SOLUTIONS				
Little Rock (Priyantha and LyMBERopoulos 2011)	Integrated Little Rock into an actual prototyping phone	Yes	Pedometer app while running on the phone, on Little Rock as well as on a hybrid architecture that includes the phone with an embedded Little Rock board	For a pedometer application, the energy savings by running with Little Rock is three orders of magnitude compared to the normal approaches
MobileHub (Haichen et al., 2014)	Prototyped with a sensor hub comprised of an 8-bit AVR micro-controller attached to sensors, and by extending the Android OS to use this sensor hub	Yes	Galaxy Nexus phones with Android OS 4.2.2 version	For three applications downloaded from the Android marketplace, it can improve power consumption by up to 83%, with no effort from the developer

Table 3 Summary and findings

Operating systems	<ul style="list-style-type: none"> • An energy-efficient operating system must be user-centric and context-aware in order to anticipate future battery limitations. • To prioritize the tasks at runtime operating system should employ self organization based energy management techniques.
Wireless interfaces	<ul style="list-style-type: none"> • Modern networks can be modeled and simulated in a better way by treating them as artificial Complex Adaptive Systems (CAS), or generalizing as Complex Adaptive COmmunicatiOn Networks and environmentS (CACOONS). • Energy-efficient communication via wireless interfaces optimization usually requires some collaborative mechanism between operating system, applications, and network infrastructure.
Sensors	<ul style="list-style-type: none"> • Energy-efficient continuous sensing has become a hot concern for both researchers and developers of mobile applications. • Several schemes have been proposed for energy-efficient sensing, which combine different sensors to minimize the energy consumption as well as the error. • Most of the techniques are based on probabilistic models of user's history and patterns to deduce future locations.
Computation off-loading	<ul style="list-style-type: none"> • Extensive research is being carried out on using cloud services for energy efficiency in mobile devices. However, it depends on different factors (e.g. the amount of data exchanged, the network state, etc.) which can affect the efficiency of computation off-loading. • Self organization based collaborative mechanism for resource sharing with co-located devices using low-energy wireless connectivity has shown to be very effective for energy saving and improving the user experience.
Battery management mobile applications	<ul style="list-style-type: none"> • Recently, increased energy-awareness at application level has also received interest and is likely to gain more attention in the future.
Hardware solutions	<ul style="list-style-type: none"> • Some researchers have suggested that re-designing the mobile device's hardware architecture is beneficial for energy-efficient sensing. • Experimental results revealed that the idea of additional low energy micro-controller or core for managing sensors is actually very effective in terms of saving energy.

Conclusions

Researchers have been emphasizing that energy efficiency should also be considered while maximizing the performance of mobile devices, since late 90s. In this review paper, we have critically reviewed the state-of-the-art self organization based energy management techniques that have been proposed over the period of 2010–2015. Based on the proposed optimization, we grouped the existing approaches in different categories, which are further classified at different levels, from energy-efficient operating systems to computation off-loading.

Self organization based energy management techniques often needs to change the power states of the different components at runtime and some tasks to be delayed to extend the battery life. However, it can affect the user experience with mobile devices. Nowadays, managing device resources from energy efficiency perspective without affecting the user experience is one of the most difficult tasks. Some mobile devices have power-saving features to dynamically adjust the power consumption of hardware components as per the required functionality but such improvements at hardware level are very slow as compared to the rapidly growing energy demands.

Despite of the enormous attempts by operating system designers, hardware manufacturers, and researchers, mobile devices still remain energy-consuming devices. Therefore, operating systems must exploit all the chances they have to reduce power consumption without diminishing the user experience. We strongly believe that this

review paper would accelerate the research efforts for self organization based energy-aware support in mobile devices, at both hardware and software level.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MS designed the study. BF carried out the studies and performed the analysis. BF drafted the manuscript and MS wrote the conclusion. Both authors edited the paper individually and together. Both authors read and approved the final manuscript.

Acknowledgement

None.

Received: 5 March 2015 Accepted: 29 April 2015

Published online: 04 June 2015

References

- Arora SSSM (2014) Self Learning Offloading scheme for Android Phones. http://www.ijarcse.com/docs/papers/Volume_4/6_June2014/V4I5-0806.pdf. Accessed 3 Nov 2014.
- Balasubramanian N, Balasubramanian A, Venkataramani A (2009) Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. Proceedings of the 9th {ACM} {SIGCOMM} Conference on Internet Measurement Conference. pp 280–293.
- Batool K, NMA (2015) Self-Organized Power Consumption Approximation in the Internet of Things. 2015 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, 2015.
- CMM (NYSE: (2015) Battery Doctor (Battery Saver) - Android Apps on Google Play. https://play.google.com/store/apps/details?id=com.ijinshan.kbatterydoctor_en&hl=en. Accessed 21 Jan 2015.
- Corpus J (2014) Best Smartphone Battery Monitoring Apps to Improve Endurance. <http://www.tomsguide.com/us/best-battery-apps-improve-endurance,review-2335.html>.
- Cuervo E, Balasubramanian A (2010) MAUI: making smartphones last longer with code offload. Proc 8th. 17:49–62. doi:10.1145/1814433.1814441.
- Cui Y, Xiao S, Wang X, et al. (2013) Performance-aware energy optimization on mobile devices in cellular network.
- Dey A, Abowd G, Salber D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interact.* 2001;16:97–166.
- Ding AY, Srinivasan A, Kojo M, Tarkoma S (2013) Enabling energy-aware collaborative mobile data offloading for smartphones. 2013 IEEE International Conference on Sensing, Communications and Networking (SECON). IEEE, pp 487–495.
- Dogar FR, Steenkiste P (2010) Catnap: Exploiting High Bandwidth Wireless Interfaces to Save Energy for Mobile Devices. *Energy* 107–122. doi: 10.1145/1814433.1814446.
- Fonseca R, Dutta P, Levis P, Stoica I (2008) Quanto: Tracking energy in networked embedded systems. *Symp A Q J Mod Foreign Lit* 323–338.
- Geier E (2011) 11 Android apps to boost speed, battery life and more | Computerworld. <http://www.computerworld.com/article/2511206/mobile-apps/11-android-apps-to-boost-speed-battery-life-and-more.html>. Accessed 21 Jan 2015.
- Haichen Shen, Aruna Balasubramanian, Anthony LaMarca DW (2014) Improving Power Efficiency Using Sensor Hub Without Re-Coding Mobile Apps.
- Hofer T, Schwinger W, Pichler M, et al. (2003) Context-awareness on mobile devices - the hydrogen approach. 36th Annu Hawaii Int Conf Syst Sci 2003 Proc. doi:10.1109/HICSS.2003.1174831.
- Inc. KM (2014) Battery Doctor - Must-have Battery Management App on the App Store on iTunes. <https://itunes.apple.com/sg/app/battery-doctor-must-have-battery/id615987910?mt=8>. Accessed 21 Jan 2015.
- Jindal G. A Comparative Study of Mobile Phone's Operating Systems. *IJCAIT.* 2012;1:10–5.
- Kemp R, Palmer N, Kielmann T, Bal H (2012) Cuckoo: a computation offloading framework for smartphones. *Mob Comput Appl.* 59–79. doi:10.1007/978-3-642-29336-8_4.
- Kharbanda H, Krishnan M, Campbell RH (2012) Synergy: A middleware for energy conservation in mobile devices. Proceedings - 2012 IEEE International Conference on Cluster Computing, CLUSTER 2012. pp 54–62.
- Kumar K, Liu J, Lu YH, Bhargava B. A survey of computation offloading for mobile systems. *Mob Networks Appl.* 2013;18:129–40. doi:10.1007/s11036-012-0368-0.
- Kundu TK, Paul K (2010) Android on mobile devices: An energy perspective. Proceedings - 10th IEEE International Conference on Computer and Information Technology, CIT-2010, 7th IEEE International Conference on Embedded Software and Systems, ICESS-2010, ScalCom-2010. pp 2421–2426.
- Lee K, Lee J, Yi Y, et al. Mobile data offloading: How much can wifi deliver? *IEEE/ACM Trans Netw.* 2013;21:536–50. doi:10.1109/TNET.2012.2218122.
- Lin K (2010) Energy-Accuracy Aware Localization for Mobile Devices. Proc 8th Int Conf Mob Syst Appl Serv MobiSys 10 In *Mobisys*:285.
- Liu X, Ding F, Li J, et al. (2013) PhoneJoule: An energy management system for android-based smartphones. Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom 2013. pp 1996–2001.
- Ljung P. Opportunities for energy savings in mobile devices. Indoor and Mobile Radio Communications, PIMRC: IEEE International Symposium on Personal; 2011. p. 2394–401.
- Lu H, Yang J, Liu Z, et al. The Jigsaw continuous sensing engine for mobile phone applications. Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10. New York, New York, USA: ACM Press; 2010. p. 71.

- Mobile C (2015) Clean Master (Speed Booster) - Android Apps on Google Play. https://play.google.com/store/apps/details?id=com.cleanmaster.mguard&hl=en_GB. Accessed 21 Jan 2015.
- Neely MJ. Energy optimal control for time-varying wireless networks. *IEEE Trans Inf Theory*. 2006;52:2915–34. doi:10.1109/TIT.2006.876219.
- Niazi MA. Complex Adaptive Systems Modeling: A multidisciplinary Roadmap. *Complex Adapt Syst Model*. 2013;1:1. doi:10.1186/2194-3206-1-1.
- Niazi MA, Hussain A. Complex adaptive communication networks and environments: Part 1. *Simulation*. 2013a;89:559–61. doi:10.1177/0037549713492122.
- Niazi MA, Hussain A. Complex adaptive communication networks and environments: Part 2. *Simulation*. 2013b;89:787–9. doi:10.1177/0037549713497714.
- Niazi MA, Laghari S (2012) An Intelligent Self-Organizing Power-Saving Architecture: An Agent-Based Approach. 2012 Fourth International Conference on Computational Intelligence, Modelling and Simulation. IEEE, pp 70–75.
- Noble BD, Price M, Satyanarayanan M, SCIENCE. C-MUPPDO (1995) A programming interface for application-aware adaptation in mobile computing. 2nd USENIX Symp Mob Locat Comput 8:345–363.
- Paek J, Kim J, Govindan R. Energy-efficient rate-adaptive GPS-based positioning for smartphones. *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*. New York, New York, USA: ACM Press; 2010. p. 299.
- Pasricha S, Donohoo BK, Ohlsen C (2015) A middleware framework for application-aware and user-specific energy optimization in smart mobile devices. *Pervasive Mob Comput*. doi: 10.1016/j.pmcj.2015.01.004.
- Perez S (2014) Mobile App Usage Increases In 2014, As Mobile Web Surfing Declines | TechCrunch. <http://techcrunch.com/2014/04/01/mobile-app-usage-increases-in-2014-as-mobile-web-surfing-declines/>. Accessed 3 Nov 2014.
- Perrucci GP, Fitzek FHP, Widmer J (2011) Survey on Energy Consumption Entities on the Smartphone Platform. 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring). IEEE, pp 1–6.
- Priyantha B, Lymberopoulos D. LittleRock: Enabling Energy-Efficient Continuous Sensing on Mobile Phones. *IEEE Pervasive Comput*. 2011;10:12–5. doi:10.1109/MPRV.2011.28.
- Ra M-R, Paek J, Sharma AB, et al. (2010) Energy-delay tradeoffs in smartphone applications. *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*. p 255.
- Roy A, Rumble SM, Stutsman R, et al. (2011) Energy management in mobile devices with the cinder operating system. *Proceedings of the sixth conference on Computer systems - EuroSys '11*. ACM Press, p 139.
- Rozner E, Navda V, Ramjee R, Rayanchu S (2010) NAPman: network-assisted power management for wifi devices. *Proc 8th Int Conf Mob Syst Appl Serv - MobiSys '10* 91. doi:10.1145/1814433.1814445.
- Samreen NMA; L (2015) Highly Modular Electronics: Self-Organized Power Hotspot Management in Components.
- Schirmer M, Bertel SJP. Contexto: Leveraging Energy Awareness in the Development of Context-Aware Applications. ICT-ENSURE consortium: EnviroInfo; 2014.
- Schulman A, Navda V, Ramjee R, et al. (2010) Bartendr: a practical approach to energy-aware cellular data scheduling. *Proceedings of the sixteenth annual international conference on Mobile computing and networking - MobiCom '10*. p 85.
- Singhal D (2014) Clean Master App review – Keep your phone optimized and secure by using only one app. <http://atechpoint.com/android/clean-master-app-review-keep-your-phone-optimized-and-secure-by-using-only-one-app>. Accessed 21 Jan 2015.
- Vallina-Rodriguez N (2011) ErdOS: achieving energy savings in mobile OS. *Proc ACM MobiArch* 37–42.
- Vallina-Rodriguez N, Crowcroft J. Energy Management Techniques in Modern Mobile Handsets. *IEEE Commun Surv Tutorials*. 2013;15:179–98. doi:10.1109/SURV.2012.021312.00045.
- Wang H, Wang Z, Shen G, et al. (2013) WheelLoc: Enabling continuous location service on mobile phone for outdoor scenarios. 2013 Proceedings IEEE INFOCOM. IEEE, pp 2733–2741.
- Wen Y, Zhang W, Luo H (2012) Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones. *Proceedings - IEEE INFOCOM*. pp 2716–2720.
- Yi S-H, Cho S-B (2012) A Battery-Aware Energy-Efficient Android Phone with Bayesian Networks. 2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing. IEEE, pp 204–209.
- Zeldovich N, Boyd-Wickizer S, Kohler E, Mazières D. Making information flow explicit in HiStar. *Commun ACM*. 2011;54:93.
- Zeng H, Ellis CS, Lebeck AR, Vahdat A. ECOSystem: managing energy as a first class operating system resource. *SIGOPS Oper Syst Rev*. 2002;36:123–32. doi:10.1145/635508.605411.
- Zhao DCAKJL and F (2011) Mobile Apps: It's Time to Move Up to CondOS - Microsoft Research. <http://research.microsoft.com/apps/pubs/default.aspx?id=147238>. Accessed 3 Nov 2014.
- Zhuang Z, Kim K-H, Singh JP. Improving energy efficiency of location sensing on smartphones. *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*. New York, New York, USA: ACM Press; 2010. p. 315.