The Survey of Wireless Transfer Energy Efficiency in Mobile Cloud Computing

[1]C.T.K.Amarnath, [2]Dr.S.Muruganatham
[1]Research Scholar, [2]Professor
[1]Department of Computer Science, [2]Department of Computer Applications
[1] Bharathiar University, Coimbatore, Tamilnadu, [2] S.T. Hindu College, Nagercoil, Tamilnadu.

ABSTRACT

The Mobile cloud computing is the New Focus research zone in the area of latest transmission paradigm. To complete energy optimization in mobile systems, power consumption concerned with each component or application needs to be expected subsequent to implementation. In This paper reviews the existing research contributions on minimizing energy consumption in mobile cloud computing. These efforts have allowed us to provide a comprehensive summary of recent work on transmission energy savings in mobile cloud computing. It is deduced that maximum of 90% of energy can be conserved by making use of effective task scheduling method. Finally, we conclude the analysis and discuss the upcoming research.

KEYWORDS
Mobile cloud computing, smartphone, power-saving mode, wireless, energy efficiency

1 INTRODUCTION

Mobile cloud computing has developed to utilize powerful computing resources easily and ubiquitously. In the meantime, mobile systems hardware and wireless communication technology have developed rapidly in recent years. Wireless mobile devices have maximum utility when they can be used "anywhere at any time"; additionally, some of the smartphone already have the ability to provide a more abundant user experience by running heavy PC-oriented applications.

With the explosion of mobile applications and user demand, mobile cloud computing is born to integrate cloud computing into the mobile environment. In addition to take full advantages of such typical cloud computing characteristics as no up-front investment, lower operating cost, highly scalable and easy access, mobile cloud computing also brings new types of services and facilities to mobile users and utilizes cloud computing to provide ubiquitous mobile service access.

Apart from the advantages mentioned above, mobile cloud computing also brings unprecedented opportunities and challenges associated with developing mobile user-oriented specific applications. To apply cloud computing to mobile environment, there are still many obstacles involving service availability, mobility, security, privacy, energy efficiency, etc. Among these various constraints placed on smartphone, the greatest one is limited power supplies. Battery lifetime is non-reversible, and has been found to directly influence user experience. Additionally, Robinson et al. show that battery capacity grows by only 5% annually, which cannot accommodate itself to the trend of the explosive development of mobile applications and services. Therefore, one of the most important concerns in mobile cloud computing relates to the power management of smartphone.

To reduce the power in a typical mobile device includes cpu, screen, disk, system memory, wnic, and other sensors. In order to reduce energy to consumers, an energy estimation model is needed to reduce energy and manage battery power by users and operating systems in a less energy. Additionally, due to the increase in mobile data traffic and use of mobile internet applications, wireless transfer energy mechanism becomes increasingly significant. We provide a summary of wireless transfer energy in mobile cloud computing mechanism in this paper.

The rest of the paper is organized as follows. Section 2 provides a brief summary of, mobile cloud computing including its definition and architecture. Section 3 presents several construction schemes of power estimation model of mobile device's general components. Additionally, we provide a summary on wireless transfer energy in mobile cloud computing mechanisms in section 4. Finally, to address future trends, we summarize and conclude the survey in section 5.

2 SUMMARY OF MOBILE CLOUD COMPUTING

The concept of mobile cloud computing was suggested in 2007, not long after the concept "cloud computing" began to gain popularity in 2006. Over the past five years, there have been a number of studies focusing on applying cloud computing to smartphone. These studies have involved new computing offloading system structures, task partitioning schemes, virtual machine based process/runtime migration, the reduction of server response latency by using intelligent surrogate discovery protocol, and the scheduling of specific protocol-level energy saving schemes (e.g., Wi-Fi Power-Saving Mode (PSM) or cellular Radio Resource Control (RRC)) to lower wireless communication energy. In summary, the efforts detailed in these studies could be divided into two classes.

The first class or structure refers to carrying out data processing and storage outside smartphone. In this situation, smartphone are to be viewed as simple clients, and all the cloud services are deployed on a cloud computing platform. The advantages of this scheme allow the computation and storage limitations of smartphone to be avoided, and the centralized cloud service provider ensures the simple deployment of middleware.

The second mobile cloud computing class or structure mainly considers ubiquitous cloud service providing and fast service discovery. Data storage and processing could be carried out on other nearby smartphone or resource-rich PCs. Similar concepts have emerged in recent years to include "Crowd-sourced Computing" [7, 19]. Crowd sourced computing spreads opportunistically through a network, using ad-hoc wireless connections which form as other devices come into proximity with an existing device. The connected devices have the ability to exchange input data and intermediate results. The advantages are obvious; primarily, cloud services and data can be easily accessed, and the workload of backbone internet is reduced. Moreover, it could be used as a means of distributing human interaction tasks to smartphone. However, there are problems with this proposed structure: the centralized cloud service provider does not currently exist in this structure and it does not involve a "pay-as-you-use" scheme. Additionally, several obstacles to include security/privacy/authentication control, service availability and surrogate incentive mechanism are extremely difficult to design.

Since most of the related research studies and implemented systems mainly focus on solving problems in the first environment, and it is more in line with the design concept of cloud computing, in this paper, mobile cloud computing is defined as cloud computing extended with mobility and energy efficiency support. Mobile cloud applications move the computing power and data storage into the cloud away from mobile phones, bringing applications and mobile computing to not just smartphone users but a much broader range of mobile subscribers [1], which follows the first definition above.

Here we present a general architecture of mobile cloud computing, as shown in Fig. 1. Smartphone typically connect to the access network via various wireless access entrances (e.g., WLAN access point, cellular base station or satellite). There exists a mobile network service layer between the wireless access network (consisting of mobile core network and Internet) and mobile cloud platform. The main functions of this layer vary from device's mobility control and computation task offloading management to security control (including authentication and private encryption). The core cloud platform is partly overlapped with the traditional data center network. The application service providers may either lease basic mobile cloud infrastructure as a deployment platform, or directly provide their services through the Internet.

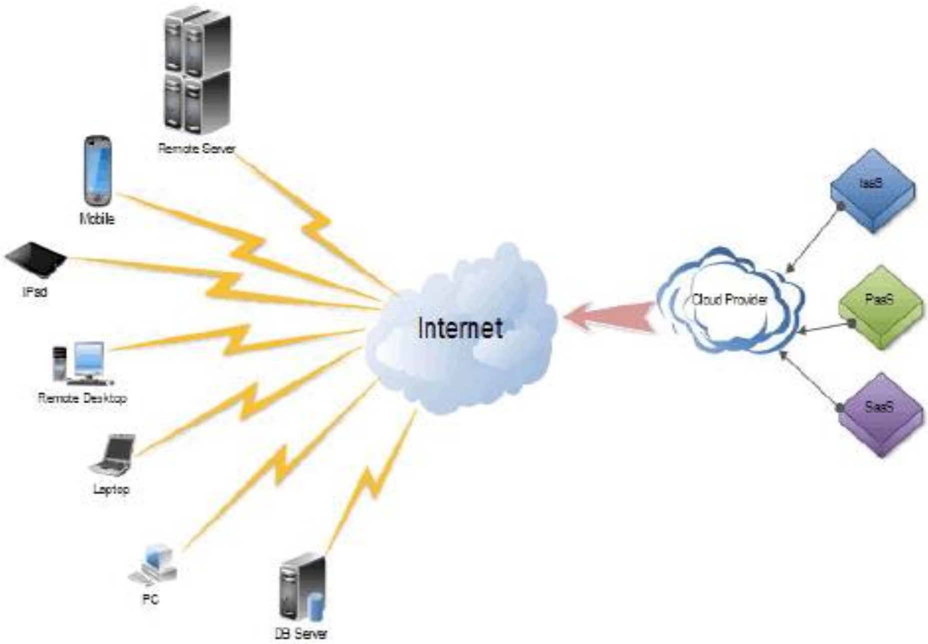

Fig. 1 Structural design of Mobile cloud computing

3 MOBILE CLOUD COMPUTING ENERGY MODEL

In order to achieve energy optimization in mobile systems, we need to be aware of the power consumption involved with each application prior to execution. Energy model estimates the energy consumption corresponding to different power states of the smartphone's components. With an energy model of high accuracy, less energy applications can be easily designed, and battery power can be managed by users/operating systems in an efficient way. However, the diversity of components and user's behaviors [11, 23], as generated by various sensors and mobile applications, presents a great challenge to the estimation accuracy. In this section, we present a survey of the universal energy estimation model for smartphone, first elaborating on the modelling methods step by step, and then introducing the state-of-the-art research as proposed in the past few years.

To build an energy model, we need to first select several target components and list the factors (such as CPU clock rate, cellular radio state, called predictors) which may influence their power consumption characteristics, we then must divide these predictors into different power states (e.g., idle state and high clock rate state for CPU). After this initial stage, we run the designed training applications to measure the power consumption corresponding to different power states. Finally, based on the experimental results, the estimation model is constructed with different modelling ideas, finite state machines (FSM) [20], etc.).

The component's energy consumption characteristics are highly associated with user activity. For example, an off-line video game player mainly consumes battery power by LCD and CPU, while a web surfer mainly by WNIC. Shye et al. [26] studied the impact of user behavior and presented several considerable advices to help save energy. First, they utilized linear regression to build the energy model based on high-level measurement results collected from a set of components. The input for the model comes from the current utilization state of the components, while the output consists of the total energy cost. To find the relationship between user activity and energy cost, they designed a special application to gather data on user behavior and then send the tracing data to the server. Through analysing the data, they proposed two user aware optimizations: (1) slowly changing the brightness of the screen and (2) on-demand dynamic frequency scaling (DFS) algorithm for the CPU.

Since the measurement based on power meter is time-consuming and prevents the automatic construction of the power model, some construction schemes attempt to eliminate the dependence on external equipment. Power Booter [27] is an automated power model construction technique that uses built-in battery voltage sensors and the knowledge of battery discharge behaviour to monitor power consumption while explicitly controlling the power management and activity states of individual components. The authors also developed PowerTutor to inform smartphone developers and users of the power consumption implications of decisions about application design and use. The advantage of this construction is that it does not rely on an external power meter, hence reducing the burden on developers. Additionally, the results indicate that the power model built with PowerBooter is accurate to within 4.1% of values measured over 10-second intervals. However, the length of this time interval (10 s) is still too long in some specific, and the accuracy is not high enough due to the changing of energy capacity and the related discharge curve.

The previous approaches are generally intended for energy estimation over time intervals of one second or longer, hence cannot provide the required rate for fine-grained use. Sesame [10] proposed in 2011 is a self-modeling paradigm that devotes itself to a self-constructive and high rate of estimation. The model is also based on linear regression, which can be automatically generated with the battery interface. The OS can get the values stored in the registers through ACPI which can also help us obtain the power status of most of the system components. Compared with [27], Sesame does not require the discharge curve to be obtained first. The accuracy of the model is improved by transforming the original predictors to the principal component analysis (PCA). The main idea of PCA is to utilize the top two principle components as predictors instead of handling too many components; PCA reduces the workload of the construction and guarantees equally high accuracy.

Regardless of reliance on an external power meter or battery interface, these modeling schemes are both based on a single implicit assumption each occurrence of an event implies that some energy expenditure has taken place. This assumption breaks down if the marginal energy cost of an event is low when compared to the active power draw from a subcomponent. Additionally, these counter-based models require sampling, and at the same time, subsequently suffering from the inherent tradeoff between overhead and agility when selecting a sampling rate.

Different from the previously discussed counterbased energy models, Pathak et al. [20] utilized a finite state machine to build the model (FSM) that does not rely on the assumption. FSM uses events to trigger transitions. Since the model arguably model components as what they really are (state machines), it can better accommodate domain knowledge such as notions of batching and timeouts, and account for events or conditions which cause state transitions. Three steps are required to automatically construct the FSM model for a phone. First, the power consumption of the selected components under a single system call is measured using a power meter, starting from the beginning of the system call and covering the transitions of the whole power states; Based on the power consumption traces, FSM for the modeled power states is generated under single system call. Second, they integrate the FSMs generated in the first step to multiple concurrent system calls. If the later system call arrives after the previous one is out of its productive power state, the power behavior can be modeled simply by superimposing the second system call on top of the first one; otherwise, if the later system call is initialization-based, the FSM of the later system call will be superimposed on the tail state of the previous one. Third, the FSM model is generated for the whole system by driving the CTester application suite which is designed to measure all possible combinations of the sets of conditions.

There are also some limitations involved with our study of the energy estimation model. First, there may exist power-related events in which the hardware itself does not expose to the driver due to efficiency or lack of perceived need; thus, if the device driver cannot capture such behavior, the model construction procedure will be impacted. Second, our discussion is based on one assumption—the drivers should expose power states to the operating system. However, as this is obviously not necessary for the drivers, the design of the incentive mechanism becomes a considerable question. Additionally, are there any extra opportunities brought to energy modelling with the emergence of mobile cloud environment. For example, the complex modeling computation could be offloaded to the cloud through wireless interface for the purpose of reducing construction overhead. This research direction is also considerable.

4 ENERGY EFFICIENCY:

Due to the increase in mobile data traffic caused associated with the growing popularity and use of mobile Internet applications, energy consumption of wireless data communications on smartphone is growing rapidly. With the energy-efficient wireless data transmission mechanism, the total energy cost of mobile systems could decrease. Smartphone are able to take full advantage of the processing, storage and networking resources in the cloud to provide secure, low-cost and efficient power management. Cellular and Wi-Fi are the most widely used wireless transmission technologies. The analysis [15] indicates that both technologies are likely to succeed in the marketplace and they can complement each other for a better service. Here we provide a summary of energy-efficient transmission mechanisms for both of them.

4.1 Cellular power management

Caused by the lack of visibility into the resource constrained mobile execution environment, inefficient resource usage may incur a mass of energy consumption [22]. A considerable amount of energy (nearly 60%), referred to tail energy, is consumed after each transfer procedure is completed [4]. Compared with tail energy, the ramp energy consumed in switching to high power states before the transfer is small. Additionally, signal strength is also considered to be a non-negligible factor. According to Bartendr [25], when the signal is weak, the energy consumed per bit is as much as 6× higher than when it is strong.

Radio Resource Control (RRC) RRC protocol refers to the transmission power management in cellular network, Fig. 2 shows RRC state machine and an instantaneous power measurement result of an example transmission through 3G [4]. The DCH state is a high power state that occurs during periods of high performance. In contrast, the FACH state consumes about half of the power compared with the DCH state, while the IDLE state consumes about one percent of the power compared with the DCH state. The inactivity timers control the transition between different power states. When the DCHstate starts, the protocol initiates a timer (T1); when transforming to the FACH state, the cellular radio changes after the timeout and initiates another timer (T2).When T2 expires, the radio changes back to IDLE state.

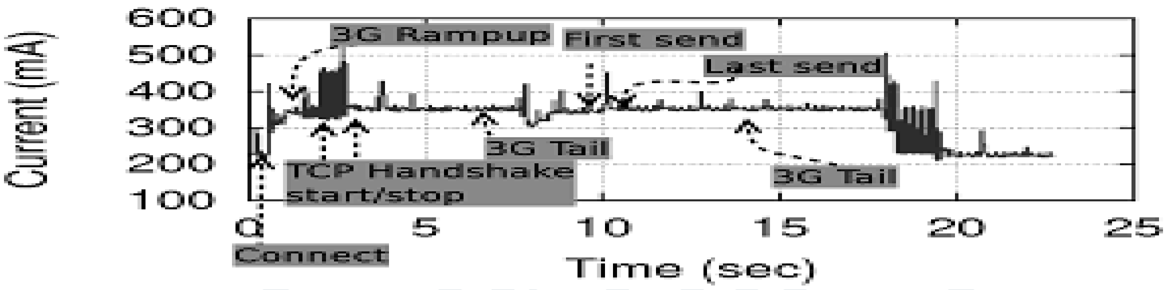

Fig. 2 Instantaneous power measurements [4] for an example transfer over 3G showing the radio resource control state transition and tail energy state.

Several studies have made efforts to achieve energy efficiency by reducing the time spent in high power state. Balasubramanian et al. [4] reduce the time spent in FACH by scheduling both delay-tolerant and prefetching-benefit applications respectively. The successive transmissions are merged to reduce the time spent in FACH. Additionally, since Web browsing is one of the most important and commonly used applications in smartphones, Zhao et al. [29] studied the computation sequence of web browsing, combined data transmissions together and switched to FACH state once after the transmission is complete. To reduce the time spent in FACH state, they propose a prediction algorithm based on Gradient Boosted Regression Tree (GBRT) to predict the user's reading time after downloading the web pages. If the predicted reading time is longer than a threshold, the state will switch from FACH to IDLE.

In cellular network, there exists a unique interface usage mode called "tail time". It is designed to mitigate the delay and overhead associated with state transition. Eliminating the tail time is not appropriate since it may incur huge delay (about 300%) caused by state transfer [21]. However, the energy consumed in tail time is considerable. Many works conducted to reduce tail energy could be classified into two categories: (1) changing the timeout value to reduce the frequency of meeting the tail time, and (2) scheduling the transmissions to eliminate the tail time.

For permanently online applications, unwanted sporadic data traffic incurs the radio always active. Labiod et al. [14] analysed the distribution of the incoming packet interval, and measured the power consumption in different RRC states under different static fast dormancy timeout values. Based on the experimental data obtained, they were able to obtain the optimal static fast dormancy timeout value. However, this work is not applicative for the tail energy incurred by the needed traffic.

Several research studies also make a contribution to eliminate tail time. Liu et al. [16] leveraged Virtual Tail mechanism and a Dual Queue Scheduling algorithm to schedule other transmissions to the tail. The transmissions that can be scheduled are also classified into two categories: delay-tolerant and prefetching enable. Compared with approaches that attempt to adjust the optimal timeout value, this method can reduce the delay and transition overhead occurring between different states due to incorrect predictions. However, this method is limited based on the scenarios of small transmissions.

Additionally, the energy cost in low signal strength is also considerable. Bartendr [25] used the signal tracks to predict the signal strength and designed a scheduling algorithm for scheduling the sync and streaming applications. The main idea is to delay the sync traffic when the signal is weak, and to prefetch the streaming traffic when the signal is strong. Ma et al. [17] predicted the signal strength through its spatial-temporal periodicity while on high speed trains. They find that large scale variations of signal strength only depend on the distance from the base station, and the base stations are usually deployed at the same distance from each other. The prediction is based on the past records. However, the application scenario may be too unique.

4.2 Wi-Fi power management

The energy consumption associated with Wi-Fi is considerable. Its inherent CSMA mechanism leads to energy inefficiency, the energy consumption of idle listening is comparable to that of active transmission reception [2, 5]. In addition, Wi-Fi also has background traffic issue. The primary source of power consumption in Wi-Fi comes from the idle listening (IL), which is aroused by its intrinsic CSMA mechanism. Because 802.11 power saving mode (PSM) has been widely supported by wireless interface card, most recent works of Wi-Fi power optimization are based on PSM mode.

Power Saving Mode (PSM) PSM, including both static and adaptive PSM, achieves energy efficiency through the reduction of time in idle listening (IL). In static PSM, the access point (AP) shapes the traffic by buffering the packets, thus informing PSM clients of the presence of the packets outstanding at AP [6]. The PSM clients wake up to check the presence of the packets periodically. Different from the static PSM, the adaptive PSM wakes up based on some heuristics (as opposed to a fixed value which is complicated to set [13]). The interactive performance of the adaptive PSM is better than the static PSM, while for the background applications, static PSM is more efficient than adaptive PSM.

According to Xinyu et al. [28]'s work, PSM reduces the unnecessary IL time caused by the network-level latency through aggregating downlink packets. Nevertheless, PSM cannot reduce the IL time associated with carrier sensing and contention. As a result, they find that IL still consumes the majority of energy even with PSM enabled: 80% while in a busy network and 60% while in a relatively idle network. Different from previous studies that focused on reducing the time spent in IL, Xinyu et al. attempt to lower the power during IL. Flautner et al. [12] and Dieter et al. [8] show that the power consumption of digital devices is known to be proportional to their voltage-square and clock-rate. According to this phenomenon, EmiLi [28] reduces the power consumption by reducing the clock-rate without compromising packet-detection capability. They also find that by changing the voltage along with clock-rate, the system can be more energy-efficient.

PSM achieves energy efficiency by reducing the waiting time spent in IL; however, PSM is also assumed to be useless during data transfers when applications are constantly sending data based on the overhead caused by constantly entering into sleep mode. The works in [3, 9, 13] attempt to achieve energy efficiency towards data-oriented applications by reducing the level of power consumption during the data transfer. These studies design a proxy that decouples wired and wireless segments and develop a scheduler which operates on the ADU to resolve the bandwidth bottleneck between wireless and wired links.

The work in NAPman [24] shows that PSM reduces the power consumption without considering the competing background traffic which could result in a significant increase in a client's energy consumption (potentially up to 300%). To resolve this problem, NAPman utilizes AP virtualization mechanism to make each PSM client believe that it is associated to a different AP. In this way, each client can monopolize the channel to transmit data. In addition, for adaptive PSM clients, the inability to receive expected data due to background traffic delays will cause them to go to sleep. NAPman presents an energy-aware fair scheduling algorithm designed to reduce overall energy consumption while assuring the fairness between the PSM clients associated with the same AP, as well as with CAM clients.

NAPman only improves PSM in the cases where several clients are associated to an isolated AP; however, network contention among different APs can increase the PSM client's power consumption in quantity. In [18], each AP dynamically re-schedules its own period to minimally overlap with others while regulating the client's slept and wake-up schedules. This allows PSM to deal with the contention caused by the multiple clients connecting to the same AP by leveraging TSFadjustment.

The emergence of mobile cloud computing provides an always on-line, resourceful remote platform to the device, which can be interpreted as a powerful extension of the device's ability. In order to interact with remote application service process in real time, smartphone could not avoid using a wireless network frequently. The main idea of reducing transmission energy is to keep the WNIC in low power state (e.g., sleep state for Wi-Fi or idle state for cellular) as long as possible. At the same time, the traffic needs to be shaped to fit the energy consumption characteristics (e.g., burst is better than scattered data flow).

5 CONCLUSIONS

The latest technology updated in smartphones such as the iPhone and android phones, as a witness of the potential of Mobile cloud computing, have developed at an alarming pace. In this paper, we present a survey of the energy-efficient technologies in mobile cloud computing, provide the definitions and architectural designs of mobile cloud computing. In addition, we present a general construction scheme of an energy estimation model step-by-step. We summarize related works in energy-efficient wireless transmission.

Since the current technologies are not mature enough to tap the full potential of mobile cloud computing, we believe that there are still tremendous opportunities for researchers to make ground-breaking contributions in this field, thus bringing significant impacts to the development in the industry. We hope our work will provide a better understanding of design challenges surrounding energy-efficient mobile cloud computing.

REFERENCES

1. http://www.mobilecloudcomputingforum.com.

2. Agarwal Y, Chandra R,Wolman A, Bahl P, Chin K, Gupta R (2007) Wireless wakeups revisited: energy management for voip over wi-fi smartphones. In: Proceedings of the 5th international conference on mobile systems, applications and services.

3. Anand M, Nightingale E, Flinn J (2005) Self-tuning wireless network power management. Wirel Netw 11(4).

4. Balasubramanian N, Balasubramanian A, Venkataramani A (2009) Energy consumption in mobile phones: a measurement study and implications for network applications. In: Proceedings of the 9th ACM SIGCOMM conference on internet measurement conference.

5. Chandra R, Mahajan R, Moscibroda T, Raghavendra R, Bahl P (2008) A case for adapting channel width in wireless networks. In: ACM SIGCOMM computer communication review, vol 38.

6. IEEE Computer Society LAN MAN Standards Committee (1997) Wireless lan medium access control (mac) and physical layer (phy) specifications. In: Inst. Electr. Electron. Eng., New York, Tech.

7. Demirbas M, Bayir M, Akcora C, Yilmaz Y, Ferhatosmanoglu H (2010) Crowd-sourced sensing and collaboration using twitter. In: 2010 IEEE international symposium on a world of wireless mobile and multimedia networks (WoWMoM). IEEE.

8. Dieter W, Datta S, Kai W (2005) Power reduction by varying sampling rate. In: Proceedings of the 2005 international symposium on low power electronics and design.

9. Dogar F, Steenkiste P, Papagiannaki K (2010) Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices. In: Proceedings of the 8th international conference on mobile systems, applications, and services.

10. Dong M, Zhong L (2011) Self-constructive high-rate system energy modeling for battery-powered mobile systems. In: Proceedings of the 9th international conference on mobile systems, applications, and services.

11. Falaki H, Mahajan R, Kandula S, Lymberopoulos D, Govindan R, Estrin D (2010) Diversity in smartphone usage. In: Proceedings of the 8th international conference on mobile systems, applications, and services.

12. Flautner K, Reinhardt S, Mudge T (2002) Automatic performance setting for dynamic voltage scaling. Wirel Netw 8(5).

13. Krashinsky R, Balakrishnan H (2002) Minimizing energy for wireless web access with bounded slowdown. In: Proceedings of the 8th annual international conference on Mobile computing and networking.

14. Labiod H, Badra M (2007) new technologies, mobility and security. Springer-Verlag.

15. Lehr W, McKnight L (2003) Wireless internet access:3g vs. wifi? Telecommun Policy 27(5–6).

16. Liu H, Zhang Y, Zhou Y (2011) Tailtheft: leveraging the wasted time for saving energy in cellular communications. In: Proceedings of the 6th international workshop on MobiArch.

17. Ma X, Liu J, Jiang H (2011) Energy-efficient mobile data uploading from high-speed trains. Mobile Netw Appl 17(1).

18. Manweiler J, Choudhury R (2011) Avoiding the rush hours: Wifi energy management via traffic isolation.

19. Murray D, Yoneki E, Crowcroft J, Hand S (2010) the case for crowd computing. In: Proceedings of the 2nd ACM SIGCOMM workshop on networking, systems, and applications on mobile handhelds.

20. Pathak A, Hu Y, Zhang M, Bahl P, Wang Y (2011) Finegrained power modeling for smartphones using system call tracing. In: Proceedings of the 6th conference on computer systems.

21. Qian F, Wang Z, Gerber A, Mao Z, Sen S, Spatscheck O (2010) Top: tail optimization protocol for cellular radio resource allocation. In: 2010 18th IEEE international conference on network protocols (ICNP). IEEE.

22. Qian F, Wang Z, Gerber A, Mao Z, Sen S, Spatscheck O(2011) Profiling resource usage for mobile applications: a cross-layer approach. In: Proceedings of Mobisys.

23. Rahmati A, Zhong L (2010) A longitudinal study of nonvoice mobile phone usage by teens from an underserved urban community.

24. Rozner E, Navda V, Ramjee R, Rayanchu S (2010) Napman: network-assisted power management for wifi devices. In: Proceedings of the 8th international conference on mobile systems, applications, and services.

25. Schulman A, Navda V, Ramjee R, Spring N, Deshpande P, Grunewald C, Jain K, Padmanabhan V (2010) Bartendr: a practical approach to energy-aware cellular data scheduling. In: Proceedings of the 16th annual international conference on mobile computing and networking.

26. Shye A, Scholbrock B, Memik G (2009) into the wild: studying real user activity patterns to guide power optimizations for mobile architectures. In: Proceedings of the 42nd annual IEEE/ACM international symposium on microarchitecture.

27. Zhang L, Tiwana B, Qian Z, Wang Z, Dick R, Mao Z, Yang L (2010) accurate online power estimation and automatic battery behavior based power model generation for smartphones. In: Proceedings of the 8th IEEE/ACM/IFIP international conference on hardware/software codesign and system synthesis.

28. Zhang X, Shin K (2011) E-mili: energy-minimizing idle listening in wireless networks. In: Proceedings of the 17th annual international conference on mobile computing and networking.

29. Zhao B, Zheng Q, Cao G(2011)Energy-aware web browsing in 3g based smartphones.