Department of Computer Science and Engineering

Modeling and Managing Energy Consumption of Mobile Devices

Yu Xiao

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 139/2011

Modeling and Managing Energy Consumption of Mobile Devices

Yu Xiao

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the School of Science for public examination and debate in Lecture Hall T2 at the Aalto University School of Science (Espoo, Finland) on the 20th of December 2011 at 12 noon.

Aalto University
School of Science
Department of Computer Science and Engineering
Data Communications Software

Supervisor
Professor Antti Ylä-Jääski

Instructor
Dr. Matti Siekkinen

Preliminary examiners
Professor Thomas Plagemann University of Oslo, Norway
Professor Liviu Iftode Rutgers University, USA

Opponent
Professor Jon Crowcroft University of Cambridge, UK

Aalto University publication series
DOCTORAL DISSERTATIONS 139/2011

ISBN 978-952-60-4429-3 (printed)
ISBN 978-952-60-4430-9 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Unigrafia Oy
Helsinki 2011

Finland

The dissertation can be read at http://lib.tkk.fi/Diss/

NORDIC ECOLABEL
441 697
Printed matter

Aalto University

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Yu Xiao

Name of the doctoral dissertation
Modeling and Managing Energy Consumption of Mobile Devices

Publisher School of Science

Unit Department of Computer Science and Engineering

Series Aalto University publication series DOCTORAL DISSERTATIONS 139/2011

Field of research Telecommunications Software

Manuscript submitted 23 August 2011 **Manuscript revised** 22 November 2011

Date of the defence 20 December 2011 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

Thanks to the significant improvement in the processing and networking capabilities of mobile devices, mobile devices today can run applications that require complex computation and high network bandwidth. As these applications become ever more popular, a rise is seen in the energy demand that is generated by a typical usage of mobile devices, with the result that existing battery technology is not able to satisfy the growing demand. Improving the energy efficiency of mobile devices and applications has, therefore, become essential. In this thesis, we investigate the energy consumption of mobile devices and propose practical solutions for improving the energy efficiency of wireless data transmission.

We propose power models of wireless data transmission over Wi-Fi and show how the power consumption is related to power-saving mechanisms, to Internet traffic characteristics, and to the network throughput. We utilize the linear dependency of transmission costs on network throughput in order to extend the linear regression power models from microprocessor level to system level. These power models provide us with an insight into developing software with energy-efficient wireless data transmission.

In this thesis, we present three strategies for reducing transmission cost: applying lossless data compression to network traffic data, scheduling the transmission based on the prediction of network conditions, and power management of the wireless network interface based on the predicted traffic intervals. Our strategies consider the trade-offs between computational and transmission costs, and between energy consumption and transmission performance. In addition, we apply statistical methods for implementing prediction utilities. Finally, considering the complexity in the context collection and processing, we propose an event-driven framework that can be used for implementing, deploying and managing various energy-efficient strategies on mobile platforms.

Keywords Power management, power modeling, mobile devices

ISBN (printed) 978-952-60-4429-3 **ISBN (pdf)** 978-952-60-4430-9

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Espoo **Location of printing** Helsinki **Year** 2011

Pages 143 **The dissertation can be read at** http://lib.tkk.fi/Diss/

Preface

I owe my deepest gratitude to my supervisor, Prof. Antti Ylä-Jääski, for letting me to freely explore the paths of science under his guidance. He was always there to provide support for every step of the way towards the completion of this thesis.

Thanks to my instructor, Dr. Matti Siekkinen, for guiding my research work. I was fortunate to have an instructor who was always there to listen to me and to give me advice. His patience and support helped me to overcome many challenges and finish this thesis.

Thanks to Prof. Jussi Kangasharju for his sound words of wisdom and encouragement when I was struggling with self doubt. I am grateful to him for the long discussions that helped me sort out my work. Thanks to Prof. Jukka Nurminen for sharing his expertise, and for his help during the writing of this thesis.

Thanks to Prof. Thomas Plagemann and Prof. Liviu Iftode for their rigorous pre-examination. Their comments helped me improve the quality of this thesis.

Thanks to Ramya Sri Kalyanaraman with whom I made my first publications. I am grateful to her for sharing ideas with me and for reviewing my work. Thanks to Petri Savolainen for his company, support and encouragement. He always did his best to help me solve the problems I would face along the way. I am also indebted to Arto Karppanen, Zhirong Yang, Rijubrata Bhaumik and Pan Hui for co-authoring the publications included in this thesis. Thanks to Pan Hui for hosting my visit to Deutsche Telekom Laboratories.

I also would like to extend my thanks to Prof. Vera Goebel and Dr. Jarle Søberg from University of Oslo for their highly valuable comments and advice. Thanks to the former and current staff of Data Communications Software group for their help in my thesis work. Particularly, thanks to

Vilen Looga, Khaled Chowdhury, Chengyu Liu, Francisco Casasus, Aldara De Castro, Dr. Jaakko Kangasharju, Dr. Andres Arjona, Laura Takkinen, Prof. Sasu Tarkoma, Mohammad Hoque and Dr. Andrey Lukyanenko.

I am also thankful to the secretaries and system administrators at our department for creating a great working environment. I would especially like to thank Soili Adolfsson for her friendly support in taking care of all the important day-to-day matters during my work.

I appreciate the financial support from Helsinki Graduate School in Computer Science and Engineering (Hecse), China Scholarship Council, ICT SHOK Future Internet programme of Finland, UbiLife project, and Nokia Scholarship Foundation.

Thanks to all the teachers who cultivated my interest in learning and encouraged me to create.

Thanks to all my friends. Without them life would have been much more difficult during the dark winters.

Lastly, and most importantly, I wish to thank my parents for their endless love. To them I dedicate this thesis.

Helsinki, November 21, 2011,

Yu Xiao

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Yu Xiao, Petri Savolainen, Arto Karppanen, Matti Siekkinen, and Antti Ylä-Jääski. Practical Power Modeling of Data Transmission Over 802.11g for Wireless Applications. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, Passau, Germany, 75-84, April 2010.

II Yu Xiao, Rijubrata Bhaumik, Zhirong Yang, Matti Siekkinen, Petri Savolainen, and Antti Ylä-Jääski. A System-Level Model for Runtime Power Estimation on Mobile Devices. In *Proceedings of 2010 IEEE/ACM International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing* , Hangzhou, China, 27-34, December 2010.

III Yu Xiao, Matti Siekkinen, Antti Ylä-Jääski. Framework for Energy-Aware Lossless Compression in Mobile Services: The Case of E-Mail. In *Proceedings of 2010 IEEE International Conference on Communications*, Cape Town, South Africa, 1-6, May 2010.

IV Ramya Sri Kalyanaraman, Yu Xiao, Antti Ylä-Jääski. Network Prediction for Energy-aware Transmission in Mobile Applications. *International Journal on Advances in Telecommunications, issn 1942-2601*, Vol.3, no.1& 2, 72-82, September 2010.

V Yu Xiao, Wei Li, Matti Siekkinen, Petri Savolainen, Antti Ylä-Jääski, Pan Hui. Power Management for Mobile Devices Using Complex Event Processing. *Aalto University publication series SCIENCE+TECHNOLOGY Aalto-ST 26/2011*, 1-27, 2011.

Author's Contribution

Publication I: "Practical Power Modeling of Data Transmission Over 802.11g for Wireless Applications"

The author of this thesis was the lead architect of this project. She proposed the original idea for the power modeling, guided the power measurement, and did the data analysis for model evaluation.

Publication II: "A System-Level Model for Runtime Power Estimation on Mobile Devices"

The author of this thesis was the lead architect of this project. She proposed the original idea for the problem settings, guided the data collection, and did the model building and evaluation.The second author did parts of data collection from mobile devices under the author's close guidance.

Publication III: "Framework for Energy-Aware Lossless Compression in Mobile Services: The Case of E-Mail"

The author of this thesis proposed the original idea for the problem settings, implemented parts of this system presented, and did the system evaluation.

Publication IV: " Network Prediction for Energy-aware Transmission in Mobile Applications"

The author of this thesis shared similar idea of adaptive network transmission with the first author of this article. The author of this thesis implemented parts of the system, guided the power measurement for system evaluation, and did the result analysis. The first author of this article guided the collection and modeling of the signal-to-noise ratio traces used in the experiments.

Publication V: "Power Management for Mobile Devices Using Complex Event Processing"

The author of this thesis was the lead architect of this project. She proposed the original idea of the problem settings, implemented the complex event processing part of the system, and did parts of the measurement experiments. The second author implemented the other parts of the system under the author's close guidance.

1. Introduction

During the previous decade, the wireless industry has moved into the Internet era and we have witnessed revolutionary changes in wireless networks, mobile devices and mobile applications.

Mobile broadband networks that provide high-speed Internet access have been widely deployed. In 2010, 3G cellular networks were available in 143 countries and had 694 million subscribers[1]. In addition, over 750,000 Wi-Fi hotspots had been installed and were used by 700 million people around the world [24]. Compared with 3G cellular networks, Wi-Fi can provide a much higher data rate, e.g. up to 54 Mbps for 802.11g Wireless Local Area Network (WLAN), although the coverage of a Wi-Fi hotspot is smaller than that of a 3G base station. Thus, it is popular to have both 3G and Wi-Fi interfaces in a single mobile device.

The increasing coverage of mobile broadband networks has been accompanied by a significant boost in the capabilities of mobile devices. Feature phones and Personal Data Assistants (PDAs) are being replaced by smartphones and mobile Internet devices such as Nokia Internet Tablets.[2] The new ones are equipped with high-performance processors, large-volume storage, multiple network interfaces, high-resolution displays and rich sensors. All these capabilities together make it possible for mobile devices to handle much more complex tasks. It opens a door to mobile applications that require heavy computation, high-speed data transmission, and rich context information.

In the previous five years, we have seen the explosion of the mobile application market. Today, thousands of new applications are published ev-

[1] http://www.itu.int/ITU-D/ict/newslog/14+Of+Wireless+Subs+Connected+To+3G +Networks.aspx

[2] Because the distinction between smartphones and mobile Internet devices is diminishing, we will not distinguish between these two categories in the rest of this thesis.

ery day through online shops like Android MarketPlace, Apple Store and Ovi Store. The most popular mobile applications are no longer telephone services, but Internet services, such as Facebook, YouTube, Google Maps, Twitter and Pandora[3]. This phenomenon also reflects the ongoing transformation of the wireless industry. Companies like Apple, Google and Microsoft, which previously were not wireless companies, are now playing more and more important roles in the mobile device and application markets. On the other hand, the change in the usage of mobile devices poses new challenges to the research on mobile computing. Energy efficiency in mobile computing, especially in the wireless data transmission involved in mobile applications, is one of these challenges that have attracted much attention from mobile device manufacturers, mobile application providers and network operators.

Compared with traditional telephone services like voice calls and short message service, executing modern mobile applications consumes a lot more computing and networking resources and therefore demands much more energy. However, battery technology has not developed as fast as mobile computing technology and has not been able to satisfy the increasing energy demand. This has directly resulted in a dramatic decrease in battery life. For example, the battery life of a mobile device may drop to between 3 to 6 hours, if the mobile user is using Internet services such as video streaming and web browsing. Hence, energy efficiency in mobile computing, although it is a research area that has been established for more than a decade, has once again become a hot topic.

A major target of this research area is to develop techniques for reducing the energy consumption of mobile devices while trying to maintain the device performance. About a decade ago when this research area was last popular, the research focus was on the energy efficiency of computation [5], such as the energy consumption of microprocessors, since mobile Internet services such as email were still in their early stages. Today, mobile devices, as well as application scenarios, have changed drastically. With mobile Internet services becoming popular, wireless data transmission is becoming a major cause of energy consumption on mobile Internet devices. Additionally, with more sensors like Global Positioning System (GPS) receivers available on the devices, the context monitoring and its energy consumption also becomes a challenge. Hence, now is the right time to revisit energy-efficient techniques and to develop techniques to

[3] http://blog.nielsen.com/nielsenwire/online_mobile/the-state-of-mobile-apps/

solve the existing and upcoming challenges.

To develop energy-efficient techniques, the first step is to understand how energy is consumed on a mobile device. A mobile device consists of hardware components, such as microprocessors, wireless network interfaces, storage, cameras and a touchscreen, and software running on top of these hardware components.

Hardware components are the actual energy consumers. Given a hardware component, its power consumption usually consists of two parts, static power consumption and dynamic power consumption. Static power consumption, also known as leakage, persists, regardless of the state of the hardware component. It depends on the physical characteristics of the hardware component. In contrast, dynamic power consumption is caused by the hardware activities, which are controlled by the software running on top of the hardware component. The workload generated by the software, in addition to the physical characteristics, determines the amount of the dynamic power consumption. Thus, it is important to understand how power consumption is related to hardware activities and software operations.

Power measurement can tell how much energy is being consumed by the mobile device, whereas it does not give us insight into the above relationships. To solve this problem, power modeling is a technique that is often proposed in the literature for describing the above relationships using mathematical models. A power model can be specified for a certain hardware component, a certain mobile device or a certain piece of software. The information used for defining the model variables can be provided by hardware, OS, and/or applications, while the coefficients of these variables can be derived from power measurement using deterministic and/or statistical methods.

Power models provide hints on improving the energy efficiency in mobile devices and applications. Regarding the software solutions, many power management mechanisms have been proposed and some of them have already been successfully commercialized. The commercial ones are usually implemented as part of hardware resource management in mobile OSs. They adapt the operating modes to system workload and try to gain energy savings from the difference in power consumption between the operating modes of hardware components like CPU and wireless network interfaces. In other words, they try to leave the hardware components in lower-power states as long as possible. For example, the power consumption of microprocessors increases with clock frequency. Dynamic Voltage Scaling (DVS) [51, 16] is this kind of power management mechanism that scales the clock frequency with the computational workload. Similarly, the Power Saving Mode (PSM) [1] for Wi-Fi forces the Wi-Fi interfaces to go to sleep if there is no data to transmit or to receive. Furthermore, the display will be dimmed and later turned off for energy savings, if it has been idle for a while, e.g. 1 minute.

These power management mechanisms have shown their potential in saving energy. However, there are also downsides to using them. First, they might cause performance degradation, such as increasing the round trip times when using PSM [66]. Second, the transition between operating modes takes time and costs energy. Sometimes, the overhead can even overtake the energy savings gained by the transition.

One way to reduce the negative side effect is to improve the design of the power management itself. Another way is to optimize the software design so that the effectiveness of power management based on hardware resource management can be fully utilized. For computation, typical examples include the proposals for taking DVS into account in the compilation of mobile applications [3] and in the task scheduling in mobile OSs [148]. For wireless data transmission, traffic shaping for streaming applications and web prefetching have been proposed for reducing the transition overhead and increasing the duration spent in the sleep mode. Note that these solutions focus on changing the patterns of the load, and do not necessarily reduce the workload of CPU processing and wireless data transmission.

As the demand for hardware resources comes from mobile applications, many energy-efficient mobile applications have been proposed in accordance with the criteria of trying to reduce as much processing and/or data transmission workload as possible. For example, the power consumption of video playback depends on the quality of the videos. Transcoding proxies were introduced into video streaming systems for compressing the videos into ones with lower quality before forwarding them to the mobile devices [81]. Another recent example of ways of reducing the workload of the mobile devices is offloading computation from mobile devices to the cloud [23, 28].

Energy savings that can be achieved from the above solutions depend on the trade-offs between power consumption and performance, and/or between computational cost and transmission cost. Predicting the near-

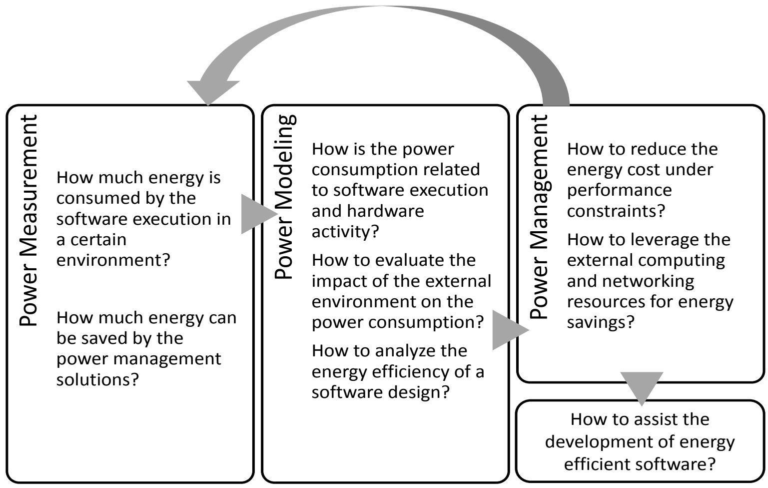

Figure 1.1. Research methodology used in our work.

future values of context variables can help in deciding whether executing the adaptations defined in the solutions would save energy or not. Accurate prediction requires knowledge of the system and its power consumption. A challenge comes from the complexity of wireless data transmission. Compared to computation, the execution of wireless data transmission includes many more uncertainties because it depends on the network protocols used for implementing the transmission, the network devices carrying the data through the network, and the network environment where the transmission happens. If these influential factors can be described using context, it follows that the power management software must be aware of the context and be able to adapt to their changes.

Nowadays, it is very common for mobile devices to handle multiple tasks concurrently. This requires the power management at system level to be able to handle more complex situations. For example, it is possible that multiple application-level solutions are applied to the same system, with one solution for one running application. In that situation, power management software must extend its functionality from hardware resource management to the management of these solutions, e.g. scheduling the context sharing among solutions and avoiding the conflicts in resource usage between them. Despite the increased complexity in power management, we are also seeing the opportunities of coordinating these solutions to further improve the energy savings.

1.1 Research Question, Scope and Methodology

Existing techniques have contributed a lot to the energy efficiency of mobile devices. However, new challenges are posed by the revolutionary development of mobile networks, devices and applications. Hence, in this thesis, we ask the following question: What is the power consumption of mobile devices after the changes and what would be potential solutions for the existing and upcoming challenges?

Due to the huge scale of the question, we limit our scope to the wireless data transmission that is involved in mobile applications, and more precisely, to the application scenarios of Wi-Fi-based data transmission. We follow the research methodology as illustrated in Figure 1.1 and look into the following three issues from an application point of view.

a) Power modeling of wireless data transmission. As power consumption of Wi-Fi-based data transmission is mainly caused by the operations of the Wi-Fi network interface (WNI), previous power models were built mainly based on the operating modes of the WNI, and on the PSM-induced transitions in the operating modes. The models did not look deeply into the dependency of power consumption on the characteristics of the network traffic generated by applications, and gave few hints for the development of mobile applications with energy-efficient wireless data transmission. Therefore, we seek to understand and to reveal the relationship between power consumption and application design, and to provide hints on improving existing power management mechanisms.

b) Energy-efficient wireless data transmission in mobile applications. Many application-specific solutions have been proposed based on traffic reduction and scheduling. In the power estimation made in these solutions, the influence of varying network conditions has usually been missing. However, under different network conditions, the network transmission performance, and further, the energy cost of data transmission varies a lot and could change the result of the energy savings. Therefore, we should take the impact of such external factors into account to figure out the trade-offs under different conditions.

c) Context-aware power management. As discussed above, wireless data transmission and its energy cost is context dependent. Many application-specific solutions have been proposed, each of which might be using the same or different contexts existing in the mobile system. They make adaptations to the changes in contexts individually. Because there is a variety of applications running on a mobile device nowadays, it is not clear how these application-specific solutions could work compatibly with each other, and collaboratively with the default power management software installed on the devices. Hence, we need utilities for handling such situations and easing the implementation and deployment of energy-efficient techniques on mobile devices.

Our methodology includes power measurement used in building power models, and in evaluating the software solutions for energy savings. This methodology is more practical since the power models and the software solutions are implemented and tested on real mobile devices. On the other hand, this poses a challenge in applying the models and the solutions to mobile devices other than the experimental devices we have been using, because the coefficients of the power models and the parameter settings of the software solutions may be based on the power measurement on the experimental devices. The power measurement needs to be repeated on the new mobile devices although the modeling method and the design of the software solutions can remain the same. In practice, this limitation cannot be removed but can be alleviated by evaluating the research results on multiple mobile platforms.

We note the focus of this thesis leaves out the following important aspects of energy-efficient wireless data transmission.

a) Network architecture and protocols for improving energy efficiency are out of our scope, since our focus is on the network applications, and not the underlying network protocols.

b) Solutions for reducing the computational cost caused by wireless data transmission are not addressed in this thesis, even though we look into ways for modeling that part of energy cost.

c) Energy efficiency of data transmission in ad-hoc networks is not studied in this thesis, since we are focusing on the data transmission via the WLAN operating in infrastructure mode.

d) Our example scenarios do not cover all kinds of network applications. For example, location-based services including data transmission with location servers are not discussed.

However, we point out that our study on power modeling could provide knowledge for the development of solutions for issues omitted above. In addition, much of the research conducted on these issues is orthogonal to our design of energy-efficient mobile applications and context-aware power management. Therefore, we expect our efforts to be largely complementary to the efforts focusing on the topics out of our scope.

1.2 Contributions

This thesis is a summary of five publications. The contributions of these publications are briefly described below. More detailed discussion can be found in Chapter III.

Publication I provides a practical model of Wi-Fi-based data transmission in which the model parameters can be easily obtained from applications and traffic statistics.

Publication II presents a system level power model of a mobile Internet tablet, covering the energy consumption of wireless data transmission, computation and display.

Publication III presents a framework of proxy-based lossless compression for reducing transmission cost involved in mobile applications.

Publication IV proposes to predict the network conditions, measured by wireless signal-to-noise ratio (SNR), using statistical methods and to adapt the network transmission to the predicted changes in SNR.

Publication V describes an event-driven framework for context-aware power management. The framework includes utilities for the event generation based on the monitored contexts, the complex event processing, and the scheduling of power management policies.

1.3 Structure of This Thesis

In chapter II, we go through the essential background that is necessary for understanding the issues we address. After that, we discuss our contributions in Chapter III. The original papers are presented after a conclusion in Chapter IV.

2. Background

This chapter presents the background of power modeling and management of mobile devices. We have limited ourselves to the work that is essential for understanding the field of this thesis. As this thesis focuses on the software solutions for improving energy efficiency in wireless data transmission, low-power hardware design, such as transistor sizing [34], transistor reordering [53], and energy-efficient on-chip communication [103], are out of our scope.

We start with an overview of energy consumption on mobile devices in Section 2.1 and move to the related techniques of power measurement and modeling in Section 2.2 and 2.3. We focus on the power models using the information that can be collected with software and leave the power analysis of hardware design [18] out of our scope. After that, we classify the proposed techniques for reducing the energy cost of wireless data transmission into three categories, and analyze the design criteria of each category in Section 2.4. What follows after that in Section 2.5 is a discussion of system-level power management for mobile devices.

2.1 Energy Consumption of Mobile Devices: An Overview

The first step towards energy-efficient mobile computing is to understand how the energy is consumed on mobile devices. In practice, there are two methods for achieving this goal: power measurement and power modeling. Power measurement focuses on providing accurate measurement of power consumption, while power modeling focuses on describing how the energy is consumed using mathematical models. These two methods complement each other.

Power measurement is only applicable to hardware. It can be conducted

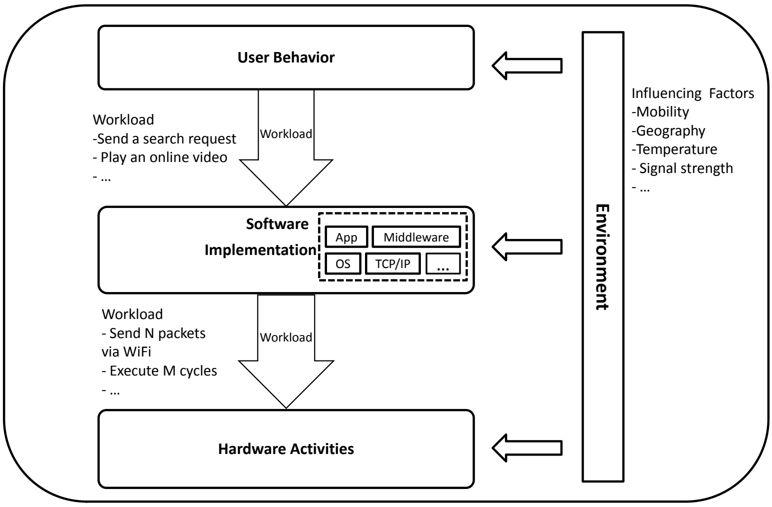

Figure 2.1. Interaction of hardware, software, mobile user and environment.

at system level or component level. The system-level power measurement only provides the overall power of the mobile device, whereas component-level power measurement can provide the power consumption of any specified hardware component. The techniques used in power measurement will be introduced in Section 2.2.

Besides hardware components, which are the actual energy consumers, a mobile device also includes software that controls the hardware operations. Power consumption of software, either mobile OSs or applications, means the total power consumed by all the hardware components involved in and during the software execution. In order to analyze the power consumption of this software, power models, with the relevant factors abstracted into the model variables, are needed for describing the relationship between software execution and hardware activities, and between software execution and power consumption.

As shown in Figure 2.1, software defines the workload of computing, I/O access, encoding/decoding, etc, including the amount of the work and the distribution of the work over time. The workload is transformed into a set of circuit activities on the corresponding hardware components, and the circuit activities consume energy. Therefore, through the workload the software execution can be linked to the activities on the actual energy consumers.

Figure 2.1 also shows that the workload can link the mobile device to its user and to the external environment. The inputs from the mobile device user and the influence of the external environment affect the workload of the software and hardware operations. For example, the network interference might cause data retransmission during wireless data transmission, which results in a change in the workload of I/O access. Hence, it is possible to investigate the external factors that influence the power consumption by applying workload analysis.

In Section 2.3, we will categorize the methodologies used in the power modeling techniques into the classes of deterministic and statistical modeling and their combinations. The information used in the techniques is a description of the workload at various granularities. In practice, which methodology to use depends on the information that is used for building the power model. We will go through our categories and introduce representative examples from the literature of the techniques in each category.

2.2 Power Measurement

Power measurement is a quantitative method of power analysis. It provides numerical data and statistics using metrics such as voltage, current, power and energy. The two key concepts, power and energy, need to be clearly defined when being used as metrics. "Energy is the total amount of work a system performs over a period of time, while power is the rate at which the system performs that work" [136]. In other words, power can be defined as the amount of the energy that is consumed in a unit of time. In practice, power is the product of voltage and current and is measured in *watts*, while energy is measured in *joules*. Battery life that is often listed in the product specification is determined by the amount of the energy stored in the battery and the rate at which the energy is drained.

Power measurement has been widely used as a black-box method for analyzing the power consumption of mobile applications, especially for comparative studies. For example, Balasubramanian et al. [11] compared the power consumption of downloading via Wi-Fi with that of downloading via 3G. Similar comparisons between Wi-Fi and Bluetooth [102], and between 2G and 3G [89] have also been presented in the literature. In addition, given a mobile application, its power consumption depends on its settings and operations. The studies of mobile applications, such as email [78], YouTube-like video sharing applications [140], cloud-based

BitTorrent content sharing service [63] and sensor-enabled physical activity monitoring applications [26], have been reported.

Most of the results reported in the literature are obtained from system-level power measurement but not component-level power measurement, although component-level power measurement is supposed to provide more details of power consumption. It is mainly because the information required for component-level power measurement [110, 19], such as the circuit design of the mobile device, is not publicly available except for very few products like the FreeRunner mobile phone [1], the one used by Carroll and Heiser in their work on component-level power modeling [19].

There are two methods of system-level power measurement, which are often used in the literature.

First, obtaining battery information including instantaneous voltage and current directly through the application programming interfaces (APIs) of the mobile OS [33], or from the energy profiling software such as Nokia Energy Profiler (NEP) [39] for Symbian phones [11, 140].

Second, using physical power meters to measure the voltage, current or directly the power [22, 146]. This method can also be applied to component-level power measurement, if the required information is available. The power meters can be directly attached to the battery to measure the working voltage over the battery and the current through the battery [113, 114]. Alternatively, the meters can be connected to a battery adapter, which is a circuit connecting the mobile device with an external DC power supply, or to a resistor that is connected in series with the battery adapter.

Comparing the above two methods, each one has its own strengths and weaknesses. The sampling frequency of physical power meters, e.g. 5KHz for the Monsoon Power Monitor [2], is much higher than that of the NEP-like software (at most 4Hz for NEP). However, the NEP-like software is more feasible for measurement in mobility scenarios and can be deployed in distributed mobile systems. As the APIs or the software that can be used for obtaining battery information are not available for all the mobile platforms, the first method is not always feasible. Hence, the choice of method depends on the availability of instruments and the requirements of power measurement. We have seen the hardware power instrumentation combined with a system activity monitor into energy profiling tools like PowerScope [40], and integrated into a wireless network testbed for

[1] http://www.openmoko.com/freerunner.html

[2] http://www.msoon.com/LabEquipment/PowerMonitor/

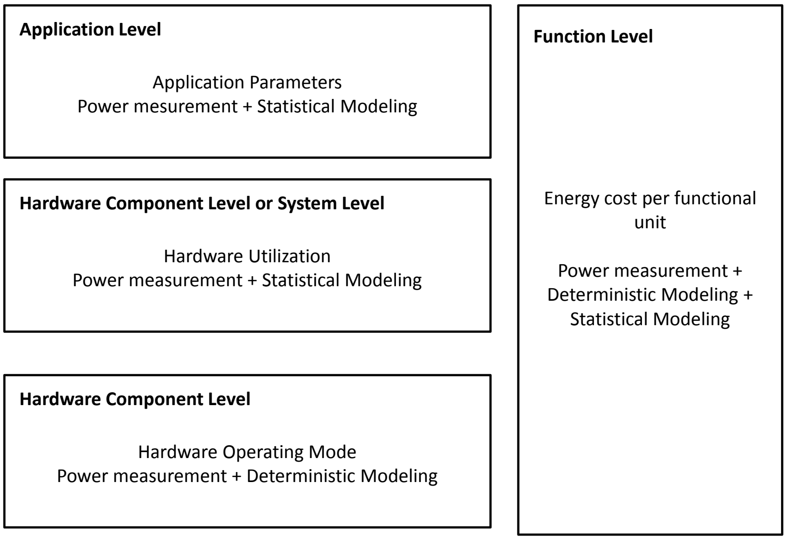

Figure 2.2. Power modeling techniques overview.

the power analysis of network protocols and applications [124].

2.3 Power Modeling

As shown in Figure 2.2, both deterministic and statistical methods can be used for building power models. The basic idea of the deterministic power modeling is to map software operations to hardware activities based on expert knowledge and to estimate the power consumed by the hardware components involved based on their activities. Differently from the deterministic power modeling, the statistical power modeling aims at finding out the relationship between power consumption and the model variables based on statistical models like linear regression. Examples of these two methods and their combination are given below.

2.3.1 Deterministic Power Modeling Based on Operating Mode

Many hardware components are able to work in several power states that correspond to different levels of power consumption. During runtime, the power state is determined by the activities that are carried out and the processing capacity of the hardware component at any particular moment. From a software viewpoint, each hardware component has several operating modes, corresponding to different activities and processing capacities. In other words, each of these operating modes can be mapped to a power state at physical level. Given an operating mode, it is possible to derive the power consumption of the hardware component.

In this subsection, we introduce the power modeling of a hardware component based on the analysis of its operating mode. We first give a general form of the power model. As shown in Equation 2.1, the energy consumption of a hardware component over a duration is composed of the energy spent in each operating mode and the overhead caused by the transitions between operating modes.

$$E(t) = \sum_{j} E_j(t_j) + \sum_{j} \sum_{k} E_{j,k} \times C_{j,k}(t), \tag{2.1}$$

where E(t) is the total energy consumed by the hardware component over the duration t, t_j is the duration spent in operating mode j and $E_j(t_j)$ is the energy cost during t_j. $E_{j,k}$ is defined as the overhead caused by the transition from operating mode j to k, while $C_{j,k}(t)$ shows how much of this transition has occurred during t. The energy wasted in waiting for the transition into a lower-power state is often called tail energy.

The operating mode can be tracked using two methods. One is to directly read it from mobile OS. For instance, Quanto [42], a network-wide energy profiler for embedded network devices, adopts this method. However, Quanto requires modifications to device drivers so the drivers can expose the power states of the underlying hardware components during runtime. The other method is to derive the operating mode from a workload description based on transition rules. Transition rules determine when to switch to another operating mode. Based on them, t_j and $C_{j,k}(t)$ can be estimated.

In the literature, the power of a hardware component is usually assumed to be approximately constant in each power state [42]. Based on that assumption, $E_j(t_j)$ is the product of the constant power and t_j. When the assumption fails, $E_j(t_j)$ can be a function of the energy consumption with the operating mode, workload description and duration as variables. $E_{j,k}$ depends on the physical characteristics and is usually assumed to be constant. The value of $E_{j,k}$ can be measured using the methods presented in Section 2.2. Sometimes, the transition overhead is not counted in the energy consumption, because the transition overhead is small enough to be safely ignored, or the monitoring of transition is not feasible.

Operating Modes of Wireless Network Interfaces

The activities of wireless network interfaces are under the control of the software running on top of them, including the hardware drivers, the network protocol stack and the network applications. The software uses a set of I/O access operations, such as sending/receiving a unit of data through a wireless network interface, in addition to computational operations like network protocol processing. These I/O access operations are transformed into the activities on the wireless network interfaces. Accordingly, the physical activities can be abstracted into operating modes that indicate the software operations being carried out while executing these physical activities.

A wireless network interface may have several operating modes, each of which corresponds to a software operation. Take the WNI as an example, it can work in TRANSMIT, RECEIVE or IDLE mode, while the network interface is transmitting, receiving or listening for traffic. These software operations determine the workload of the WNI, in terms of the direction, the interval and the size of the network traffic going through the WNI. Accordingly, the transitions caused by these software operations can be detected from the changes in the traffic.

With low-power states at physical level becoming increasingly available, some of them have been utilized by the power saving mechanisms implemented in mobile OSs or hardware management applications. As a result, new operating modes of the hardware components, corresponding to these low-power states, are defined by the power saving mechanisms. We use the power state machine, as proposed by Benini et al. [14], to describe the transitions between operating modes, with each operating mode of the hardware component defined as a state. We take WNI as an example again and show how the power saving mechanisms could have changed the power state machine of the hardware component.

The PSM is a power saving mechanism defined in the 802.11 standards [1] for WLANs. It introduces a SLEEP mode, which consumes much less power than the IDLE mode, into the power state machine of the WNI. When the WNI stays in the SLEEP mode, it only wakes up at a granularity of beacon intervals (e.g. 100ms) to check for incoming traffic. The traffic that arrives between beacons is either buffered at the access point or simply dropped if the buffer overflows. To reduce the potential performance degradation, an adaptive version of PSM, also known as PSM Adaptive, has been proposed and widely adopted in commercial products.

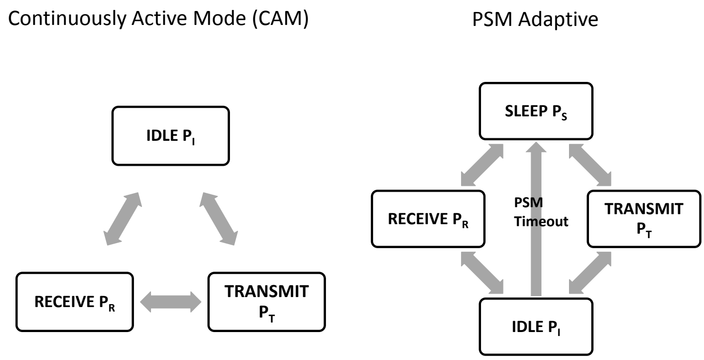

Figure 2.3. Power State machine of the WNI with the PSM Adaptive or the CAM enabled. CAM is enabled when PSM Adaptive is disabled, and vice versa. P_T, P_R, P_I, and P_S represent the power consumption at TRANSMIT, RECEIVE, IDLE and SLEEP mode, respectively.

In PSM Adaptive, the network interface will stay in the IDLE mode for a fixed period of time, such as 100 ms, before going to sleep. We call the length of this period the PSM timeout, whose default value varies from device to device. As shown in Figure 2.3, the transition from IDLE to SLEEP mode is defined using the PSM timeout.

Besides the SLEEP mode, which is designed for reducing the energy wasted in IDLE mode, some low-power states have been used for improving the energy efficiency in TRANSMIT and RECEIVE modes. For example, on Android G1, the TRANSMIT mode of the WNI is refined into two sub-states [87]. Each of these sub-states corresponds to a certain level of power consumption. Given a sub-state, the power consumption of the WNI is assumed to be constant. The WNI works in the sub-state with higher power consumption only if the packet rate is over a certain threshold. This is similar to the DVFS used in the microprocessor, which adapts the clock frequency to processing workload [136].

The above examples describe the transitions determined by the traffic interval and the packet rate. They show that the transitions caused by power saving mechanisms could also be derived from the traffic information. As traffic information can be described with various granularity, different methods will be applied for the model building, depending on the traffic description.

Packet-level Power Modeling of Wireless Data Transmission

On a mobile device, the energy consumption of wireless data transmission is mainly caused by the operations on wireless network interfaces. The workload on the wireless network interface depends on the traffic to be handled over a certain duration. Given the traffic is composed of packets, the workload can be modeled as a packet arrival process.

We take WNI as an example to explain how to obtain the values of the parameters listed in Equation 2.1 from packet-level traffic traces. The traces can be collected using packet analyzers like tcpdump [3], which have been ported to many mobile OSs, including Maemo, Android and iOS. Assuming that the WNI adopts PSM Adaptive as illustrated in Figure 2.3, the power model of the WNI can be built following the two steps below.

First, detecting the TRANSMIT and RECEIVE modes based on the transmission direction of the packets. The time spent in TRANSMIT and RECEIVE modes can be derived from the traffic size and the processing capacity of the WNI in the corresponding operating mode. The processing capacity is also known as the throughput of the WNI. Given an operating mode of the WNI, the processing capacity can be assumed to be fixed. In case the TRANSMIT and/or RECEIVE mode includes sub-states, each of which corresponds to a certain processing capacity, each sub-state is treated as an individual operating mode.

Second, detecting IDLE and SLEEP modes based on packet intervals and the PSM timeout, and counting the transitions that happen during the data transmission. Only if the packet interval is bigger than the PSM timeout, does the transition from IDLE to SLEEP mode occur and is the packet interval divided into two parts. The one equal to the PSM timeout is spent in the IDLE mode, while the rest of the interval is spent in the SLEEP mode. Depending on whether the network interface stays in IDLE or SLEEP mode at the end of each packet interval, the transition from IDLE/SLEEP mode to TRANSMIT/RECEIVE mode can also be derived.

The above method can also be applied for the power analysis of the 3G WCDMA network interface [96], whose power state is defined by the Radio Resource Control(RRC) protocol [2]. The inactivity timers defined in RRC control the state transitions in the same manner as the PSM timeout used in PSM Adaptive [48, 95].

[3] http://www.tcpdump.org

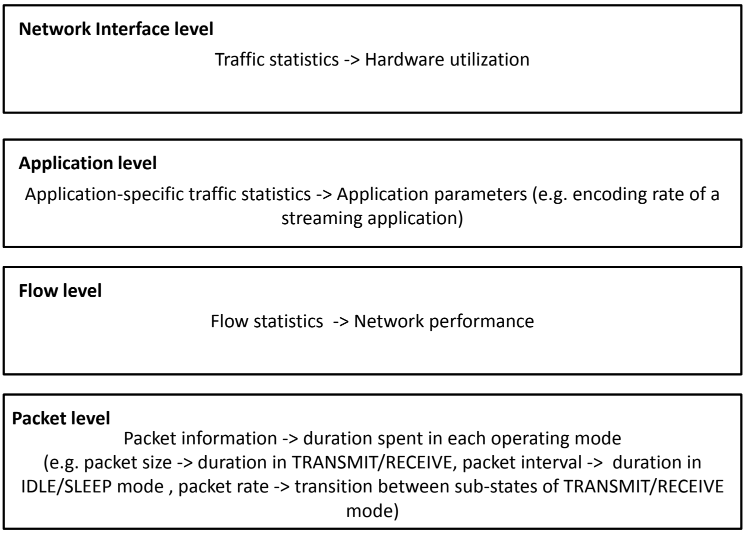

Figure 2.4. Workload description of data transmission.

Going Beyond Packet-level Power Models

Packets captured over a certain duration can be generated by different flows or even applications, as it is possible for a mobile device to run multiple network applications at the same time, and to have more than one flow involved in one network application. Each flow includes all the packets exchanged between the same source and destination IP addresses and having the same port pair. Accordingly, traffic statistics at the flow, application and network interface levels can be derived from the packet information. On the other hand, the packet-level power models can be transformed into the models with these traffic statistics. As illustrated in Figure 2.4, these traffic statistics are also related to transmission performance metrics, application parameters or the metrics that reflect the utilization of the WNI.

Concerning the power analysis of mobile applications and network protocols, the power models at flow and application levels can provide a more meaningful insight than the others, because the traffic statistics at the flow and application levels are more related to the parameters of the mobile applications and network protocols. On the other hand, it is not always feasible for the packet-level traffic profiling to be run on mobile platforms, because it usually requires root access and causes more overhead than the traffic profiling with finer-granularity. Hence, it is worth developing power models based on the statistics at flow or application level although their accuracy might be lower than that of the packet-level power models.

2.3.2 Statistical Power Modeling Based on Hardware Utilization

Statistical methods have been used for modeling the power consumption of hardware components. The model variables are defined using the performance metrics, which can reflect the utilization of the hardware components. Additionally, a linear regression model is often used as a base model, with the variable coefficients fitted to the collected data sets including the variable values and the corresponding power measurement. For example, Snowdon et al. [127] proposed performance-counter-based power models for microprocessors using the least square regression method.

In actual fact, the performance counters and the values derived from them have been widely used in the power models of microprocessors [56, 70, 25]. These counters are a set of special-purpose registers built into microprocessors. They store the counts of hardware-related activities, such as retired cycle count and L1 cache miss count. The values of these counters can be obtained using system profiling tools like Oprofile [4] for Linux.

The result models can be used for estimating the computational cost of network applications, such as the cost of reading/writing data from/to memory. The model-based power estimation can be utilized for online power management as well as the energy efficiency analysis of design choices at software design stage [61].

Equation 2.2 shows a general form of a linear regression model with p predictor variables [38].

$$f(y_i) = \beta_0 + \sum_{j=1..p} \beta_j g_j(x_{i,j}), \tag{2.2}$$

where $g_j(x_{i,j})$ is a preprocessing function of the original value of the predictor variable $x_{i,j}$. Given n observations including the values of p predictor variables and the values of the corresponding responses y_i (i = 1..n), the values of the intercept β_0 and each coefficient β_j (j = 1..p) are automatically adjusted during the model fitting towards a model in which the response can be the best predicted from the predictor variables. After the model is built, given the values of the predictor variables, the estimated power consumption will be outputted.

[4] http://oprofile.sourceforge.net/

Based on the linear dependency of the output on the values of the predictor variables, the values of the coefficients in linear regression models can reflect which variables have relatively more effect on the power consumption. The bigger the coefficient value is compared to others, the more effect the corresponding variable has on the power consumption. According to this, the power models presented in [127] showed that instruction and cache accesses have significant effect on computational cost.

The above method can be applied to other hardware components than microprocessors. For example, the power consumption of a H.323 video encoder was modeled as a function of the bit rate [75]. For wireless network interfaces, the power models can be built using network transmission performance metrics like throughput and packet rate as variables.

Statistical method is also feasible for the power modeling of software components or functional units. For example, it has been used for building a linear energy model of a software encryption module based on encryption parameters such as the number of encryption rounds [22].

2.3.3 Power Modeling at Function Level

Hardware Function level power estimation was earlier used for predicting the power dissipation of the microprocessor [98, 69] at design stage. This method assumes that the energy required to execute a functional unit is approximately constant, and calculates the total energy consumption as the aggregate of the per-functional-unit cost. As this assumption could also hold for some software functional units, this method has been extended to the power modeling of mobile OSs and applications.

The study in [130] identified the energy components of an embedded OS by studying its internal operations and classified them into system functions. It proposed to obtain the base energy of each system function from the power measurement, and to calculate the energy consumption of an embedded OS based on the base energy per system function. Similarly, Li et al. [71] profiled the execution of mobile OS as a set of kernel service routines, and calculated its energy consumption based on the energy per kernel service routine.

Mobile applications can also be decomposed into software functional units. The functional unit can be defined with various granularities, depending on the software structure. For instance, Feeney et al. [37] proposed a collection of linear equations for calculating the energy consumption of the WNI in ad-hoc mode. Each linear equation corresponds to a software activity, such as sending a point-to-point data packet. A similar method has been applied for the power modeling of data transmission through other network interfaces such as Bluetooth [85], and also for analyzing the processing overhead of protocols such as TCP [137] and Secure Sockets Layer (SSL) protocol [93].

2.3.4 Analyzing External Influencial Factors of Energy Consumption

User Behavior

User behavior has significant effect on the workload and energy consumption of a mobile device. For example, users can choose which applications to use and how to use those applications, such as when to send a search request and which video to watch. Moreover, to some extent, they can also decide which access point to connect to and where to store the data.

Some user studies [122, 135] have focused on the relationship between the user behavior and the power consumption of the mobile device. A typical methodology is to install a context monitoring application on users' mobile devices. The application tracks the mobile device settings, hardware resource consumption, network traffic and/or the user inputs. This information is then used for estimating the power consumption based on power models and for matching user activities with the power consumption. For example, Shye et al. [122] developed a logger application for Android G1 mobile phones and used it for collecting traces of real user activity. The log included the hardware utilization information, such as the CPU utilization at each CPU operating mode, the brightness of screen, and the count of bytes transferred with Wi-Fi during a given interval. The log was inputted into a linear regression power model to obtain the power consumption corresponding to user activities.

As shown in the user study presented in [104], most mobile users had no knowledge of the power characteristics of their devices and applications. Moreover, most mobile users underutilize the power saving settings of their mobile devices. Hence, in addition to improving the energy efficiency of hardware/software by taking user behavior into account, we also firmly believe that the tools that can show the predicted battery life [111] and can demonstrate the relationship between power consumption and user activities will help mobile users to extend their device battery lifetimes.

Environmental Factors
Environmental factors such as geography, temperature and network interference cannot be controlled by mobile users or software. Nevertheless, these factors can influence user behavior, the functioning of the software, and even the physical characteristics of the hardware components. For example, the performance of a GPS receiver on a mobile device can be affected by reflecting elements like vehicles, buildings, and trees [112]. The longer it takes for the GPS device to lock to the satellites and the longer it takes to receive data, the more energy will be consumed. Moreover, the SNR of wireless networks has been shown to have a significant impact on the performance [145], as well as the energy consumption of wireless data transmission [118].

2.4 Energy-efficient Wireless Data Transmission

Our summary of the solutions for energy-efficient wireless data transmission reveals three categories: workload-based adaptation of operating modes, workload scheduling, and workload reduction. These solutions are motivated by the results obtained from power measurement and modeling.

Power measurement of mobile applications shows that the default power management for WNIs on commercial mobile devices (the so-called PSM Adaptive introduced in Section 2.3.1) is inefficient in many application scenarios. To solve this problem, many revisions have been proposed (Section 2.4.1). The power models discussed in Section 2.3 reveal that the power consumption is dependent on the workload, including both its size and its pattern. These findings provide a theoretical support for the solutions of energy-efficient wireless data transmission based on workload scheduling (Section 2.4.2) and workload reduction (Section 2.4.3).

2.4.1 Adapting Operating Mode to Workload

The PSM Adaptive controls the transition from IDLE to SLEEP mode, based on the PSM timeout. Depending on the mobile device, the PSM timeout might be set to a different value. This value has a significant impact on the effectiveness of energy savings. Compared with CAM, which always keeps the WNI in idle mode during traffic intervals, the maximum energy savings that could be gained by the PSM Adaptive from each traffic interval can be estimated by using Equation 2.3.

$$E_{savings} = \begin{cases} 0, if t \leq T, \\ (t - T - T_{overhead}) \times (P_I - P_S) - E_{overhead}, if t > T, \end{cases} \tag{2.3}$$

where T is the PSM timeout, t is the traffic interval, and P_I and P_S are the power consumption at IDLE and SLEEP modes, respectively. In addition, $T_{overhead}$ denotes the time overhead caused by the transitions between IDLE and SLEEP modes, and $E_{overhead}$ denotes the corresponding energy overhead. In the case of the WNI being turned off instead of entering into SLEEP mode, (P_I-P_S) should be replaced by P_I, while $T_{overhead}$ should be the time overhead that is caused by turning off and waking up the WNI.

According to Equation 2.3, the energy savings depends on the distribution of traffic intervals. In the application scenarios where most packet intervals are smaller than the PSM timeout, such as web browsing and streaming [21, 73], the PSM Adaptive has proven to be very inefficient. Furthermore, recent studies on mobile traffic [76, 36] have shown that web browsing and streaming applications contribute a major part of today's mobile traffic. Thus, it is essential to improve the power management of the WNI.

Many proposals have focused on reducing the energy waste in IDLE mode through intelligent control over the transitions between operating modes. Intelligent control is based on both the adaptation to the traffic characteristics and on the performance requirements of the mobile applications. For example, STPM [8] proposed to switch between CAM and PSM based on two factors: the potential energy savings and the possible performance degradation. It estimated the energy savings following Equation 2.3, and compared the delay that could be tolerated by mobile applications with the maximum latency that might be generated by the usage of PSM. It enabled PSM only when the energy savings could be achieved while the latency was tolerable. The results showed that STPM was better suited for delay-tolerant applications than the delay-sensitive applications such as streaming. Similarly, Weissel et al. [138] classified applications based on their traffic characterization and adapted the switch between CAM and PSM based on the application profile.

As oppopsed to STPM, which focused on coarse-granularity adaptation, Liu and Zhong [73] proposed micro power management (μPM) for reducing the busy-time power consumption of WNI. Here, busy-time means that the idle intervals are shorter than a small value (e.g. 100ms). μPM tries to put the WNI into power saving mode during idle intervals, which can be as short as several microseconds. To control the frame delay and data loss, μPM determined when to wake up and how long to stay awake to receive any possible retransmitted data. The decisions were made based on the history-based prediction of the next incoming and outgoing traffic frames and the load of the access network. The evaluation of μPM through simulation showed that more than 30% of energy could be reduced without perceptible quality degradation for certain applications such as audio streaming.

A key issue in the above solutions is the prediction of the future incoming and outgoing traffic. There are different ways to implement the prediction. One is to use the hints disclosed by mobile applications [8]. The hints reveal when the applications will transfer data, how much data to transfer, and the maximum delay the applications could tolerate. Anand et al. [7] proposed a framework for enabling the applications to provide such hints to the power management modules. Another method is to predict statistically the next arrival time based on the history of the previous packet arrival information [73]. Regarding the traffic generated by interactive applications, the traffic prediction also requires the input from the prediction of user interaction [27, 68].

The above solutions try to save energy by switching the WNI to SLEEP mode. Other proposals have tried to push the energy savings further by turning off the WNI completely during traffic intervals. For instance, it was proposed to turn off the WNI during the bursty transfers in streaming applications until the amount of data remaining in the playback buffer was less than a predefined threshold [15]. The threshold value was chosen on the basis of the network bandwidth and the WNI characteristics.

An issue that exists in the above solutions with the WNI turned off for some time is how to awaken it. Different methods have been proposed for addressing this issue. For example, Wake-on-Wireless [121] used a second special-purpose radio to serve as a wake-up channel for the WNI, while Cell2Notify [4] used cellular radio to wake up the WNI when there was incoming traffic. In these systems, an intermediate proxy or server is needed for monitoring the incoming traffic and waking the WNI. Some solutions like Wake-on-Wireless [121] also require modification to the mobile devices in order to support the wake-up implementation.

As shown in Equation 2.3, the energy savings generated by the difference in power consumption between the different operating modes must overtake the transition overhead. Otherwise, the transition itself might waste more energy. To reduce the transition overheads, one way is to schedule the data transmission in such a way that the smaller intervals can be merged into bigger ones to reduce the number of transitions. The work related to this issue is introduced in Section 2.4.2.

Except for energy consumption during traffic intervals, some researchers have been trying to reduce the energy consumption in transmit mode. For example, transmit power control, which was earlier proposed for improving the throughput of wireless networks [83, 45], has also been used for reducing the cost in transmit mode [97, 108]. The basic idea is to use the minimum required power level to transmit data.

2.4.2 Workload Scheduling

Traffic Shaping

In the previous section, we discussed the solutions for configuring the operating mode of the WNI to make it more traffic-aware. In addition to traffic-awareness, another way of increasing the energy savings is to shape the traffic so that the traffic-aware adaptations of operating modes can be fully utilized. Since PSM Adaptive is inefficient for traffic with small intervals, traffic shaping based solutions have been focusing on merging the packets with small intervals into bursts. In practice, traffic shaping can be implemented as either a hardware or software utility on the sender, on the receiver or on intermediate proxies. In this subsection, we limit our scope to the software solutions of traffic shaping.

Buffering has been commonly used for shaping traffic on the senders or on the proxies located on the way [32, 52, 62]. This method is independent of the transport layer protocols and can be applied to different mobile applications. For example, Catnap [32] was designed for data-oriented applications, such as web browsing and file transfers. It is assumed that these kinds of applications do not mind small delays for individual packets as long as the overall transmission time does not increase. Assuming a wireless access link with a high bandwidth and a wired link with a lower bandwidth on the path to the Internet, Catnap proposed to separate the wireless and wired segments by using a proxy. The proxy, which can be deployed on an access point, was expected to buffer the data coming from the wired network before forwarding it to the mobile device in bursts, so that the small packet intervals can be merged to form a bigger interval during which the WNI can go to sleep.

Traffic shaping solutions like [52, 80, 20] focused on delay-sensitive applications such as audio streaming. In these solutions, delay constraints have been taken into account when configuring the traffic shapers. For example: for streaming applications, the buffer size on the proxy depends on the downloading rate from the streaming server, the bandwidth of the link between the proxy and the mobile client, the playback buffer size on the mobile client and the tolerable delay of the data. In situations where a proxy serves more than one mobile client [46], traffic shaping needs to be combined with the traffic scheduling under packet contraints among mobile clients. In addition, most traffic shaping solutions are designed for the scenario involving a single application. Considering those situations where multiple applications run concurrently, the scheduling between the application-specific traffic shapers can be achieved similarly to the buffering-based workload scheduling for processing tasks [109].

In addition to grouping the packet intervals, traffic shaping can also be utilized for making the traffic intervals more predictable. For example, Chandra and Vahdat [21] proposed to shape the streams into the traffic with predictable intervals on streaming servers, so that mobile clients are better informed about the future arrivals of the streaming data, and could, therefore, more accuractly adapt their operating modes to the traffic. Their experiment results showed that the combination of traffic shaping and power management for the WNI could improve the energy efficiency of receiving data on mobile clients by up to 83%.

Besides buffering, it is also possible to implement the traffic shaping on the receiver side for TCP-based data transmission. The idea is to utilize the flow control mechanism of TCP protocol to shape the traffic into periodic bursts, as proposed in [129, 106]. When the receiving window size is set to 0, the TCP server will stop sending and will start buffering the data until the window size is reset to a value greater than 0.

Regardless of how the traffic shaping is implemented, the efficiency of traffic shaping partly depends on whether the traffic shape could be maintained when the traffic arrives at the mobile device; this is because the shape might change on the way. The change might be caused by the interference in the access network [21], or by the fluctuation in the quality of the link between the traffic shaping proxy and the access point [52]. Generally speaking, it is better to have as little interference as possible and to locate the traffic shaping as close to the end device as possible [52].

Traffic shaping and sleeping mechanisms could also be utilized for reducing the energy consumption of other network elements such as routers and switches [84]. Since we are focusing on mobile devices in this thesis, we leave the readers to refer to [84] for details.

Traffic Scheduling at Access Point

The traditional first-come-first-serve policy of wireless access points is not optimal for energy efficiency, because the mobile devices might need to stay awake while the access point is serving other mobile devices. Hence, Zheng et al. [144] proposed an optimal scheduling algorithm for minimizing the awake duration while taking the fairness of energy consumption as well as performance requirements into account.

Because background network traffic has a significant impact on the power consumption of devices stations, He et al. [50] and Rozner et al. [117] focused on scheduling algorithms for reducing background network traffic, and further, energy consumption. Both focused on the background traffic generated by the mobile devices within the range of a single access point. Instead of merely looking into the contention inside the range of a single access point, SleepWell [77] focused on reducing the energy consumption of mobile clients by reducing contention between different APs.

Traffic scheduling at AP can also aim at shaping the traffic into longer intervals so that the mobile stations can remain in sleep mode for longer periods. For example, [125] proposed a scheduling algorithm based on time slots. This algorithm aimed at grouping packets into bursts within given QoS requirements.

Workload Scheduling Between Wireless Network Interfaces

Mobile devices are increasingly being equipped with multiple and heterogeneous wireless interfaces, such as 3G, Wi-Fi and Bluetooth. These wireless interfaces differ from each other in terms of communication range, throughput, and the energy efficiency of data transmission. According to the quantitative measurement of mobile devices [100, 43, 11, 88], Wi-Fi supports a relatively high data rate (e.g. 54Mbps for 802.11g) compared with 3G and Bluetooth, whereas Bluetooth could be an order of magnitude more energy-efficient than Wi-Fi. On the other hand, Wi-Fi is more energy-efficient than 3G for bulk transfer, although its communication range is typically less than 100 meters, much smaller than 3G, and Wi-Fi connectivity is not always available either.

With each wireless interface having their own strengths and weaknesses, many intelligent strategies for wireless interface selection have been proposed in an attempt to leverage the strengths of these interfaces to achieve energy savings. Examples include offloading data transfer from cellular networks to Wi-Fi whenever Wi-Fi connectivity is available [9, 134, 105], using Bluetooth instead of Wi-Fi in a personal area network [43, 88], and handover between Wi-Fi and other wireless networks like WiMax [141] and WiBro[57]. Techniques like multipath TCP have also been proposed to avoid the session breakage during handover [90].

A challenge to these strategies comes from the overheads caused by non-communicating modes such as searching for and associating with an available Wi-Fi access point, scanning and pairing with a Bluetooth device, and staying in idle mode. A detailed report on the power consumption of Wi-Fi and Bluetooth in non-communicating modes can be found in [43].

To reduce the overheads caused by the searching for Wi-Fi availability, Context-for-Wireless[105] proposed to use cell-tower information as location information to predict Wi-Fi availability because collecting cell-tower information requires much less energy than periodic scanning of Wi-Fi access points. Blue-Fi [9] proposed to combine cell-tower information with the Bluetooth contact-patterns to improve the prediction accuracy of Wi-Fi availability. In addition, Blue-Fi proposed a collaborative prediction framework in which information concerning visible Bluetooth devices, cell-tower and other available Wi-Fi connectivity can be collected by mobile users and shared with each other through peer-to-peer communications or centralized web servers.

Given the overheads, the potential energy savings to be gained from intelligent wireless network interface selection could be estimated based on the predicted workload of data transmission. Besides energy savings, performance requirements like delay and throughput should also be taken into account when making the selection [134, 100].

Similarly with the workload scheduling between wireless network interfaces, tasks can be offloaded from one hardware component to another for energy savings, provided both components can implement the software functionality although with different energy efficiency. For example, it has been proposed that compass and accelerometer can be used together with a GPS receiver for energy-efficient positioning and trajectory tracking [65, 132]. Cell-ID [86] and Wi-Fi fingerprint [64] have also been proposed to provide more energy-efficient localization. In addition, LittleRock [94] and Turducken [128] proposed to add additional low power processors to the mobile device. By offloading the sensing tasks to the additional processors, the rest of the mobile device can remain in sleep mode and therefore reduce energy consumption.

2.4.3 Workload Reduction

The power consumption of a hardware component depends on the amount of workload given to it. Hence, the power consumption can be decreased if the workload is reduced.

Data compression [123, 41, 120] has been widely used for reducing the data size. Compression is often executed on remote proxies before the data is delivered to the mobile devices [41]. The compression ratio that determines how much the traffic can be reduced depends on the compression algorithm [12] and on the type of the media to be compressed. The amount of energy saved is determined by the trade-off between the decrease in the communication cost and the potential increase in the computational cost for compressing and decompressing the data. If both sender and receiver are mobile devices, global power management (as discussed in [91]), can select which transcoder to use and where to implement the transcoding based on the predicted energy savings and the predicted effect on performance.

The trade-off between computational and communication cost here is similar to the trade-off faced in computation offloading from mobile devices to the cloud [23, 101]. In addition, the tradeoff exists in the collaborative computation among multiple processors [142], among a network of collaborative nodes [29] and among mobile device and the sensors connected to it [60].

The traffic scheduling that can reduce retransmission could also be considered as an energy-efficient solution based on traffic size reduction. Examples have been presented in the previous section. Besides traffic size reduction, it is also possible to reduce the cost by removing unnecessary functional components. For example, some fields in network protocols are not necessary and could be disabled to achieve energy savings.

2.5 System-level Power Management for Mobile Devices

Benini et al. [13] classified dynamic power management techniques into two categories: predictive schemes and stochastic optimum control schemes. Predictive schemes can be utilized for predicting the idle interval [67, 54] and the hardware utilization [47, 126]. The stochastic optimum control schemes model the system as stochastic processes [17, 55], such as Markov chains, and formulate power management as the optimization of decision making based on the tradeoff between power and performance. The input of the optimal algorithm is the future workload that can be provided by the predictive schemes or by the stochastic models of the workload [131]. We note that the above two kinds of schemes can be used individually, and also collaboratively if the input of the stochastic optimal algorithm is generated by the predictive schemes.

The above schemes have been mostly used in the power management of individual hardware components. However, minimizing the per-component energy does not necessarily minimize the overall energy of the mobile device [115, 72]. Since a mobile device may be thought of as a combination of hardware components, system-level power management, with its focus on the overall energy consumption of the mobile device, must take the interactions between the different hardware components into account. The above schemes have therefore been further worked into the workload-based joint adaptations of hardware components and into holistic optimization of power management.

Some mechanisms [79, 92] have been proposed for coordinating the power management strategies of the various hardware components in order to gain more energy savings. For example, Min et al. [79] proposed to use the information gathered from the WNI to control the CPU voltage and frequency. Furthermore, Poellabauer and Schwan [92] proposed to coordinate the power management of CPU with the power management of WNI and traffic shaping. In order to enhance the burstiness of the traffic, the CPU frequency was scaled down when the packet scheduler queue was empty in order to delay the generation of new packets, and it was scaled up when the first packet was generated.

Devadas and Aydin [30] proposed an optimal algorithm for minimizing the system-level energy consumption. The algorithm determined the optimal speed of the DVS-enabled CPU and the state transition of I/O devices. Similarly, Ashwini et al. [10] defined the system-level power state as the combination of the power states of the hardware components. They proposed an optimal algorithm for selecting the power state of each hardware component based on the application requirements. These so-called optimal algorithms utilize the difference in power consumption between operating modes or power states. Some optimal power management algorithms are also based on the nonlinear discharging process of the battery [107]. For example, Zhang et al.[147] presented an optimal algorithm, aiming at minimizing the charge loss through DVS-based job scheduling under job deadline constraints.

If we look at the system-level power management solutions from the viewpoint of software architecture, the proposed solutions can be implemented on OS, middleware or application levels.

For example, the coordination of DVS and PSM is usually implemented at OS level. In addition, as hardware resource management is one of the major functionalities of a mobile OS, some revised OSs, such as ECOsystem [143] and Cinder [116], have been proposed with improvement in the resource allocation among competing tasks. Take Cinder [116] as an example; it provides mechanisms for controlling how many resources to allocate to each application, and how fast the resources can be consumed. Cinder includes a cooperative and energy-restricted network stack. Before handling a network system call, the network stack first checks whether the allocated energy is enough to turn the radio on and to perform the transmission.

At middleware level, Ashwini et al. [10] provided interfaces for applications to reveal their application requirements. In the case of proxy-based power management, such as Parm [82] and Dynamo [80], the middleware provides interfaces for the contextual information exchange between mobile devices and the proxies. Recently, proxy-based power management has been developed into computation offloading [23] in the mobile cloud computing paradigm.

At application level, Liu et al. [74] proposed to allow each application to make local power management decisions based on CPU demand and availability, instead of globally optimizing the system-wide energy consumption. In order to implement this, OS or middleware are supposed to provide interfaces for applications for monitoring the hardware resource consumption [31] and for controlling the hardware operating modes.

2.6 Summary

In this section, we have introduced the techniques used for power measurement and power modeling. Our introduction to power modeling covered the deterministic method, statistical method and their combinations. Then, we presented the solutions for energy-efficient wireless data transmission focusing on the adaptation of hardware operating mode, workload scheduling and workload reduction, before we moved on to the discussion about system-level power management.

Our discussion about energy-efficient wireless data transmission only cover the network protocol layers that are related to our work. We leave readers to refer to the surveys for details about the solutions at physical layer [58, 49], such as dynamic modulation scaling [119], and the energy-efficient routing protocols at network layer [6, 35]. At MAC layer, we introduce the revisions to PSM and the traffic scheduling at access point. At transport layer, we present the traffic shaping based on TCP flow control mechanisms. At application layer, we discuss the solutions related to workload scheduling, workload reduction, and system-level power management.

3. Modeling and Managing Energy Consumption of Mobile Devices

This chapter discusses the published contribution of this thesis. The contribution consists of power models, the schemes of adaptive wireless data transmission for energy efficiency, and an event-driven framework for system-level power management. The future work is discussed at the end of this chapter.

3.1 Power Modeling

PSM[1] for Wi-Fi has proved to be inefficient for many application scenarios. As discussed in Section 2.4.1, the efficiency of PSM depends heavily on the traffic generated by mobile applications. To increase the energy savings, it is essential to optimize the design of the PSM, as well as to make mobile applications more PSM-friendly. This requires software developers to understand how the energy consumption is related to the traffic and to the application design. All of this motivates us to develop a practical power model that can reflect such relationships.

We follow the deterministic method that is described in Section 2.3.1 to build a power model of Wi-Fi-based wireless data transmission. We base our model on the operating mode of WNI estimated from traffic traces. Unlike the packet-level power models, our models estimate the operating mode from the detected burst information.

As illustrated in Figure 3.1, we define a burst as a group of packets flowing in the same transmission direction, and where each packet interval is smaller than a threshold. The operating mode of WNI is assumed to be fixed over the burst duration. As the packet intervals inside a burst are always set to be smaller than the PSM timeout, transitions between

[1] In this chapter, we use the abbreviation PSM to refer to PSM Adaptive as done in Publications I-V.

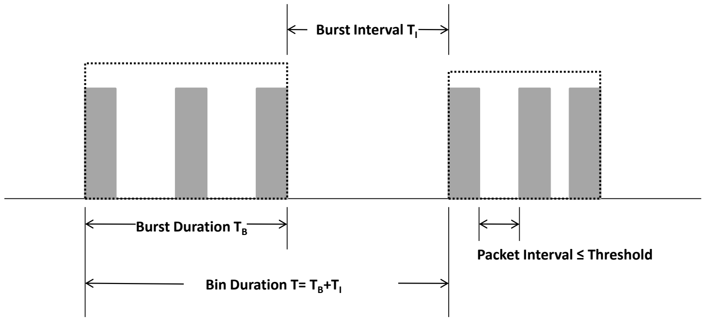

Figure 3.1. Burst definition.

IDLE and SLEEP modes only occur during the intervals between bursts. In Publication I, we compare the power consumption with and without using PSM and find that according to our power model, the PSM does not save energy if the data rate is high. This provides the theoretical support for the power management solutions that switch the WNI into CAM for bulk transfers [8].

Our power model is independent of transport layer protocols. Due to space limitations, we only show the results for TCP transmission, even though our model can also be applied to UDP transmission. We first present a power model using network data rate and burst size as variables. We notice that the burst size is related to certain application parameters, such as the encoding rate of a streaming application. Our model also provides hints on implementing energy-efficient strategies based on traffic burstiness. An example of such a strategy is the traffic shaping discussed in Section 2.4.2.

We also show a simplified power model using only network throughput as input. Obtaining a throughput reading is relatively easy for the applications if we compare it with the collection of the system-wide packet-level traffic trace which requires root privileges on the mobile devices. This simplified model is thus highly useful in estimating the power consumption of mobile applications such as YouTube during runtime.

In addition to power, we present another power metric: the energy utility. Energy utility quantifies how many bits can be transferred with a single unit of energy. The metric does not include information on the duration and is hence often used when describing proxy-based adaptations such as computation offloading.

We validate our model with physical power measurement on three mobile phones: Nokia N810, HTC G1 and Nokia 95. The experimental result shows that the power consumption of wireless data transmission increases with the data rate meaning that the general assumption of fixed power consumption during data transmission (which is often made in the relevant literature) does not hold in practice. This result also shows that energy efficiency does not conflict with transmission performance. It follows, therefore, that the techniques developed for improving network throughput might well be borrowed for the studies on energy efficiency.

The linear dependency on data rate is later leveraged in building the system-level power model, as presented in Publication II. Our power model is based on a linear regression model. The model variables are selected from the metrics that are able to reflect the hardware utilization of CPU, the WNI and the display. As proposed earlier in the literature, we use Hardware Performance Counter (HPC) based variables for describing the microprocessor utilization. Our main contribution is that we propose to use a linear regression model with non-negative coefficients. The features of non-negative coefficients help towards an efficient reduction in the number of variables in the model. This solves the problem of being able to monitor only a few HPCs simultaneously on mobile microprocessors. The test cases we use for collecting the data sets are also described in Publication II.

Extending from the HPC-based computational power model, we show how to integrate the computational power model with a transmission power model by using linear regression. Motivated by the power models described in Publication I, we use downlink and uplink data rates to describe the hardware utilization of the WNI. We also use another variable to indicate whether the CAM is enabled or not. In the final model, the coefficients for the downlink/uplink data rates are close to the per-unit data download/upload costs obtained from the power measurement. The coefficient for the CAM indicator is close to the difference between the IDLE and SLEEP modes in power consumption. This result is consistent with the model presented in Publication I, as we can see that the estimated power increases with the downlink/uplink data rates. Publication II also shows the feasibility of a statistical model of the transmission cost, based on an abstracted workload description at a higher level as discussed in Section 2.3.1.

At the end of Publication II, we discuss the process-level power analysis. In the literature, power consumption per mobile application or process is often estimated by assuming that the power consumption of each application can be directly summed up into the overall power consumption of the mobile device. We argue that this assumption cannot be directly applied to the power analysis of wireless data transmission because there is no common rule about how to divide the energy cost during traffic intervals, or the tail energy between applications or processes.

3.2 Adaptive Wireless Data Transmission for Energy Efficiency

Energy consumption of wireless data transmission depends on how much traffic is delivered and how the delivery is implemented. Publication III presents a proxy-based framework for energy-aware lossless compression in which the incoming data can be first compressed at a proxy and then forwarded to the mobile receiver. Our major contribution is a decision-making algorithm based on the trade-off between the computational and communication costs. Taking a different approach to previous work, we consider the potential impact of network conditions on the communication cost, in addition to the factors that affect computational costs.

We define the metric 'compression effectiveness' for evaluating the potential energy savings brought by the compression. Only when the compression effectiveness is greater than 1 is it more energy-efficient to use compression. We model the energy utility of decompression statistically by using the compression ratio as a parameter. Moreover, prompted by the linear relationship between power and network throughput, we define the energy utility of transmission as a function of the network throughput. We predict the network throughput based on its history, and derive its energy utility from that of the closest reference data rate based on the linear relationship.

We evaluate this framework using mobile Email as a case study. The results prove that the energy savings are highly dependent on the efficiency of data compression and network conditions. The efficiency of data compression depends on the type of data and on the available compression algorithms, which are usually fixed. Network conditions, however, introduce factors of uncertainty: when the network throughput is very high, the costs of decompression might overtake the energy savings, and when the network throughput is low enough, compression saves energy.

Our decision-making algorithm can be extended to other proxy-based workload reduction and offloading scenarios. For example, in computation offloading, there is a tradeoff between computational and communication costs. When estimating the communication costs, network conditions should be taken into account, since they have a significant impact on the results. In addition, our framework is scalable to other mobile applications such as web browsing. Due to space limitations, the evaluation of other applications is not presented in Publication III.

Publication III points out that the network conditions have a significant impact on transmission cost. In Publication IV, we make a deep study of this impact and investigate the adaptation of network transmission to the network conditions for energy savings. We chose SNR, an indicator of wireless link quality, as our measurement metric for network conditions. The value of the SNR depends on many factors, such as the distance from the access point, the transmit power of the WNI, and the interference near the mobile device.

Generally, better link quality, indicated by a higher SNR, can lead to a higher network throughput and therefore to a higher energy efficiency. Inspired by this, we propose to adapt the network transmission to the future trend of SNR in mobility scenarios. We set a threshold for the SNR. When the SNR is predicted to cross the threshold, the adaptation is triggered. In practice, network transmission is paused when the SNR falls below the threshold. Otherwise, network transmission continues.

We propose three prediction algorithms for predicting SNR, Autoregressive Integrated Moving Average (ARIMA) [99], Newton Forward Interpolation (NFI) [44], and Markov Chain [59]. Except for the NFI, the other two prediction algorithms require offline training. In Publication IV, we explain the procedure of model building, including data collection, model training and evaluation.

By using the prediction algorithms to compare the prediction accuracy and the energy savings of adaptive network transmission, we notice that a more accurate prediction does not necessarily lead to more energy savings in adaptive network transmission. In other words, we may ignore an error in prediction if it does not wrongly estimate the point when the threshold is crossed.

We measure the network throughput and transmission costs both with and without adaptations under different network conditions. To ease the analysis, we classify the network conditions into four scenarios using the mean and the standard deviation of SNR. Our results show that the effectiveness of our adaptation varies from scenario to scenario. If we look deeply into the reasons behind the variation, we find that the effectiveness depends on that fraction of time during which the SNR is lower than a certain value (e.g. 15). For example, in the scenarios where the mean SNR is lower than 15, or where the mean is higher than 20 but the standard deviation exceeds 5, it is profitable to conduct the adaptation. On the other hand, if the SNR is not high and the standard deviation is also very small so that the SNR itself rarely becomes very small, the adaptation could not bring much benefit. This finding gives more insight into the impact of SNR in data transmission performance and energy cost. Although we have only tested it in Wi-Fi, we believe our work can also provide valuable insight into the SNR-based adaptation in 3G networks [118].

3.3 System-Level Power Management for Mobile Devices

In the context of Publication V, power management refers to the software solutions that manage the energy consumption of wireless data transmission by controlling the behavior of mobile devices and applications. As discussed above, the energy consumption of wireless data transmission depends on both the size and type of the transferred data and on the conditions in which the transmission takes place. The effectiveness of power management depends greatly on how well it can adapt the operations of the mobile devices and applications to varying conditions.

Publication V presents an event-driven framework for power management in mobile devices. The changes in contexts are defined as events, and the changes in conditions are described as different combinations of events. As a result, context-aware power management applications can be described with event-driven adaptations. The main components of our framework are the event generator, the event processing agent, the scheduler, the context storage and the rule base. The most important contribution of this framework is to introduce complex event processing techniques into event-driven power management. To the best of our knowledge, our framework is the first event-driven power management framework that supports both complex event processing and simple event processing.

We argue that complex event processing is more suitable for power management than simple event processing, especially in wireless data transmission scenarios. In Publication III and IV, we have seen that prediction is very much needed in implementing adaptive network transmission. However, simple event processing can only generate events that are based on the contexts that can be directly measured. It cannot generate events based on changes in predictions. In contrast, however, complex event processing, as shown in Publication V, can create events based on statistics and prediction of contexts. In Publication V, we also discuss traffic scheduling scenarios where the detection of traffic patterns is the basis for adaptations. The detection of traffic patterns cannot be handled by simple event processing, and complex event processing is needed. We show how to apply the functions of complex event processing, such as instance partitioning, filtering, derivation and pattern matching, in traffic scheduling based power management in Publication V.

From the viewpoint of the corresponding policy management, complex event processing is often compared with simple event processing within the framework of the event processing itself. Complex event processing leads to simple policy management, whereas simple event processing is usually accompanied by complex policy management. We argue that complex event processing brings extra benefits for power management at OS and middleware levels, because it can provide more meaningful information about the conditions, which in turn makes it easier for application developers to define policies and to detect the potential conflicts between policies. Furthermore, the situational information might need to be shared between applications. By using complex event processing, the applications do not need to repeat the same processing for atomic events.

Our framework provides user-friendly interfaces for implementing and configuring power management applications. We propose to use event-condition-action (ECA) rules to describe which actions to invoke upon the occurrence of an event conforming to certain conditions. Developers only need to define the rules using structural XML and leave the rule-based adaptation scheduling to the framework. The rules used for processing atomic events into more meaningful ones can also be defined as ECA rules and can therefore be processed using the same processing engine.

We demonstrate the framework using two power management applications: traffic-aware WNI control, and the SNR-based transmission adaptations. The first one is an example of using complex event processing. It focuses on prediction-based traffic scheduling. As opposed to the work using application hints for traffic prediction, we propose to predict the no-data intervals based on the self-similar burstiness of network traffic and to learn the traffic patterns online based on statistics. The other scenario is based on the work shown in Publication IV. In Publication V, we focus on its implementation using an event-driven framework. The experimental results from these two case studies prove the functionality of our framework and show more than 10% energy savings in each scenario.

3.4 Open Questions

In this subsection, we discuss future work that can be based on the results of this thesis.

First, as discussed above, network conditions have a significant impact on power consumption. It remains an open question, though, how to show the impact explicitly in the power models that are designed for application developers. Our suggestion is to measure the impact as the change in workload such as the increase in traffic size and the increase in burst count and to apply the power models presented in Publication I for the traffic description, including both background and foreground traffic. The accuracy of such power models need to be evaluated under different network conditions. Moreover, the model we presented in Publication I did not consider the transmit power control, because transmit power control was not supported by our experimental devices. If transmit power control becomes available in the near future, our models can be updated by replacing the fixed power consumption of TRANSMIT and RECEIVE modes with adaptive ones.

Second, the power analysis for hardware at design stage has been widely studied, but not for software. We believe it is worth investigating the tools that can offer suggestions to application developers for improving the energy efficiency of mobile applications at design stage, especially with respect to the energy efficiency of wireless network transmission. Mobile device emulators are widely used for testing mobile applications. The power models presented in Publication I and II can be integrated into the emulators in order to compare the energy-efficiency of different design choices, if the model variables can be obtained.

Third, most of the existing strategies have been designed and evaluated for a specific application in mind. In real life, different kinds of applications might access networks from a mobile device concurrently, which means the wireless network interface might be serving more than one application at the same time. In those scenarios, optimizing part of the data transmission does not necessarily lead to the minimization of the overall transmission cost. Therefore, the application-specific strategies and the other power management software on the mobile device need to cooperate to optimize the holistic energy efficiency of the mobile device. In practice, the complex event processing presented in Publication V can be used for analyzing the aggregated traffic on the basis of the traffic information of each single application.

Fourth, contextual information collected from mobile devices can be shared among themselves through central web services or peer-to-peer information sharing. The sharing of context information, such as the SNR traces and location, can help reduce the energy consumption spent in context monitoring and processing. Crowd-sourcing techniques can be utilized for implementing such collaborative power management solutions [139, 133]. On the mobile side, our event-driven framework presented in Publication V can be extended to support the collection and sharing of different kinds of contexts.

4. Conclusions

Energy consumption is an important issue for mobile devices, applications and networks. It has attracted much attention from the industry and academia. Considerable efforts have been put into developing low-power hardware, energy-efficient software and wireless networks, but despite these efforts, the research achievements could still not address all the challenges posed by the emerging mobile devices, applications and networks.

In this thesis, we have focused on understanding and describing the energy consumption of mobile devices, with special focus on the energy consumption caused by wireless data transmission. We have shown that transmission costs are dependent on various contexts, including Internet traffic characteristics, power-saving mechanisms, and network environment. We have leveraged the power models by developing concrete strategies based on reducing or scheduling the workload of wireless data transmission.

We have also investigated the management of energy-efficient strategies on mobile devices and have proposed an event-driven architecture for implementing context-aware and policy-based power management. We introduced complex event processing into power management and have shown its advantage through a proof-of-concept power management platform.

The evaluation of energy-efficient techniques consists of the analysis of the effects the techniques have on the energy consumption and the application performance. The contributions of this thesis not only focus on the potential energy savings and the performance improvement, but also on how to assist the development of energy-efficient techniques. We believe that the solutions presented in this thesis could provide great insight into the development of energy-efficient mobile systems in the future.

Bibliography

[1] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages C1 –1184, June 2007.

[2] 3rd Generation Partnership Project (3GPP). Radio resource control (rrc) protocol specification. *3GPP TS 25.331*, 2006.

[3] Nevine AbouGhazaleh, Daniel Mossé, Bruce R. Childers, and Rami Melhem. Collaborative operating system and compiler power management for real-time applications. *ACM Trans. Embed. Comput. Syst.*, 5:82–115, February 2006.

[4] Yuvraj Agarwal, Ranveer Chandra, Alec Wolman, Paramvir Bahl, Kevin Chin, and Rajesh Gupta. Wireless wakeups revisited: energy management for voip over wi-fi smartphones. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, pages 179–191, New York, NY, USA, 2007. ACM.

[5] Susanne Albers. Energy-efficient algorithms. *Commun. ACM*, 53:86–96, May 2010.

[6] Dheeraj Kumar Anand and Shiva Prakash. A short survey of energy-efficient routing protocols for mobile ad-hoc networks. In *Proceeedings of the 2010 International Conference on Advances in Recent Technologies in Communication and Computing*, ARTCom'10, pages 327 –329, October 2010.

[7] Manish Anand, Edmund B. Nightingale, and Jason Flinn. Ghosts in the machine: interfaces for better power management. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, MobiSys '04, pages 23–35, New York, NY, USA, 2004. ACM.

[8] Manish Anand, Edmund B. Nightingale, and Jason Flinn. Self-tuning wireless network power management. *Wirel. Netw.*, 11:451–469, July 2005.

[9] Ganesh Ananthanarayanan and Ion Stoica. Blue-fi: enhancing wi-fi performance using bluetooth signals. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pages 249–262, New York, NY, USA, 2009. ACM.

[10] H. S. Ashwini, Amit Thawani, and Y. N. Srikant. Middleware for efficient power management in mobile devices. In *Proceedings of the 3rd International Conference on Mobile technology, Applications, and Systems*, Mobility '06, New York, NY, USA, 2006. ACM.

[11] Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani. 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*, IMC'09, pages 280–293, New York, NY, USA, 2009. ACM.

[12] Kenneth Barr and Krste Asanović. Energy aware lossless data compression. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, MobiSys '03, pages 231–244, New York, NY, USA, 2003. ACM.

[13] Luca Benini, Alessandro Bogliolo, and Giovanni De Micheli. A survey of design techniques for system-level dynamic power management. readings in hardware/software co-design. pages 231–248. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[14] Luca Benini, Robin Hodgson, and Polly Siegel. System-level power estimation and optimization. In *Proceedings of the 1998 International Symposium on Low Power Electronics and Design*, ISLPED '98, pages 173–178, New York, NY, USA, 1998. ACM.

[15] Davide Bertozzi, Luca Benini, and Bruno Ricco. Power aware network interface management for streaming multimedia. In *Proceedings of the 2002 IEEE Wireless Communications and Networking Conference*, volume 2 of *WCNC'02*, pages 926–930, March 2002.

[16] Enrico Bini, Giorgio Buttazzo, and Giuseppe Lipari. Minimizing cpu energy in real-time systems with discrete speed management. *ACM Trans. Embed. Comput. Syst.*, 8:31:1–31:23, July 2009.

[17] Paleologo Benini Bogliolo, G. A. Paleologo, L. Benini, A. Bogliolo, and G. De Micheli. Policy optimization for dynamic power management. In *Proceedings of the 35th Conference on Design Automation*, pages 182–187. ACM Press, 1998.

[18] David Brooks, Vivek Tiwari, and Margaret Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th Annual International Symposium on Computer architecture*, ISCA '00, pages 83–94, New York, NY, USA, 2000. ACM.

[19] Aaron Carroll and Gernot Heiser. An analysis of power consumption in a smartphone. In *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, USENIX ATC'10, pages 21–21, Berkeley, CA, USA, 2010. USENIX Association.

[20] Surendar Chandra. Wireless network interface energy consumption: implications for popular streaming formats. *Multimedia Syst.*, 9(2):185–201, 2003.

[21] Surendar Chandra and Amin Vahdat. Application-specific network management for energy-aware streaming of popular multimedia formats. In

Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference, USENIX ATC'02, pages 329–342, Berkeley, CA, USA, 2002. USENIX Association.

[22] R. Chandramouli, S. Bapatla, K. P. Subbalakshmi, and R. N. Uma. Battery power-aware encryption. *ACM Trans. Inf. Syst. Secur.*, 9:162–180, May 2006.

[23] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. Clonecloud: elastic execution between mobile device and cloud. In *Proceedings of the 6th Conference on Computer Systems*, EuroSys '11, pages 301–314, New York, NY, USA, 2011. ACM.

[24] Cisco. The future of hotspots: Making wi-fi as secure and easy to use as cellular, 2011.

[25] Gilberto Contreras and Margaret Martonosi. Power prediction for intel xscale processors using performance monitoring unit events. In *Proceedings of the 2005 International Symposium on Low Power Electronics and Design*, ISLPED '05, pages 221–226, New York, NY, USA, 2005. ACM.

[26] Igor Crk, Fahd Albinali, Chris Gniady, and John Hartman. Understanding energy consumption of sensor enabled applications on mobile phones. In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, EMBC'09, pages 6885–6888, September 2009.

[27] Igor Crk, Mingsong Bi, and Chris Gniady. Interaction-aware energy management for wireless network cards. *SIGMETRICS Perform. Eval. Rev.*, 36:371–382, June 2008.

[28] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. Maui: making smartphones last longer with code offload. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 49–62, New York, NY, USA, 2010. ACM.

[29] Dinesh Datla, Xuetao Chen, Timothy R. Newman, Jeffrey H. Reed, and Tamal Bose. Power efficiency in wireless network distributed computing. In *Proceedings of the IEEE 70th Vehicular Technology Conference*, VTC 2009-Fall, pages 1 –5, September 2009.

[30] Vinay Devadas and Hakan Aydin. On the interplay of dynamic voltage scaling and dynamic power management in real-time embedded applications. In *Proceedings of the 8th ACM International Conference on Embedded Software*, EMSOFT '08, pages 99–108, New York, NY, USA, 2008. ACM.

[31] Thanh Do, Suhib Rawshdeh, and Weisong Shi. pTop: A Process-level Power Profiling Tool. In *Proceedings of the Workshop on Power Aware Computing and Systems*, HotPower'09, New York, NY, USA, October 2009. ACM.

[32] Fahad R. Dogar, Peter Steenkiste, and Konstantina Papagiannaki. Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices. In *Proceedings of the 8th International Conference on Mo-*

bile Systems, Applications, and Services, MobiSys '10, pages 107–122, New York, NY, USA, 2010. ACM.

[33] Mian Dong and Lin Zhong. 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*, MobiSys '11, pages 335–348, New York, NY, USA, 2011. ACM.

[34] J. Ebergen, J. Gainsley, and P. Cunningham. Transistor sizing: how to control the speed and energy consumption of a circuit. In *Proceedings of the 10th International Symposium on Asynchronous Circuits and Systems*, ASYNC'04, pages 51 – 61, April 2004.

[35] S. Ehsan and B. Hamdaoui. A survey on energy-efficient routing techniques with qos assurances for wireless multimedia sensor networks. *IEEE Communications Surveys and Tutorials*, PP(99):1 –14, 2011.

[36] Hossein Falaki, Dimitrios Lymberopoulos, Ratul Mahajan, Srikanth Kandula, and Deborah Estrin. A first look at traffic on smartphones. In *Proceedings of the 10th Annual Conference on Internet Measurement*, IMC '10, pages 281–287, New York, NY, USA, 2010. ACM.

[37] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings of the 12th Conference on Computer Communications*, volume 3 of *INFOCOM'01*, pages 1548 –1557, 2001.

[38] F.I.Kushnirskii and M.E.Primak. Regression with nonnegative coefficients. *Cybernetics and Systems Analysis*, 9, January 1973.

[39] Frank H.P. Fitzek and Frank Reichert. *Mobile Phone Programming and its application to wireless networking*. Springer, 2007.

[40] Jason Flinn and M. Satyanarayanan. Powerscope: A tool for profiling the energy usage of mobile applications. In *Proceedings of the 2nd IEEE Workshop on Mobile Computer Systems and Applications*, WMCSA '99, pages 2–, Washington, DC, USA, 1999. IEEE Computer Society.

[41] Jason Flinn and M. Satyanarayanan. Managing battery lifetime with energy-aware adaptation. *ACM Trans. Comput. Syst.*, 22:137–179, May 2004.

[42] Rodrigo Fonseca, Prabal Dutta, Philip Levis, and Ion Stoica. Quanto: tracking energy in networked embedded systems. In *Proceedings of the 8th USENIX Conference on Operating Systems Design and Implementation*, OSDI'08, pages 323–338, Berkeley, CA, USA, 2008. USENIX Association.

[43] Roy Friedman, Alex Kogan, and Krivolapov Yevgeny. On power and throughput tradeoffs of wifi and bluetooth in smartphones. In *Proceedings of the 30th Conference on Computer Communications*, INFOCOM '11, Shanghai, China, 2011.

[44] G Shanker Rao. *Numerical Analysis*. New Age International Publishers, 3rd edition, 2006.

[45] Yan Gao, Zheng Zeng, and P.R. Kumar. Joint random access and power selection for maximal throughput in wireless networks. In *Proceedings of the 29th Conference on Computer Communications*, INFOCOM'10, pages 1 –5, March 2010.

[46] M. Gundlach, S. Doster, H. Yan, D.K. Lowenthal, S.A. Watterson, and Surendar Chandra. Dynamic, power-aware scheduling for mobile clients using a transparent proxy. In *Proceedings of the 2004 International Conference on Parallel Processing*, ICPP'04, pages 557 – 565, August 2004.

[47] Selim Gurun, Chandra Krintz, and Rich Wolski. Nwslite: a light-weight prediction utility for mobile devices. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, MobiSys '04, pages 2–11, New York, NY, USA, 2004. ACM.

[48] Henry Haverinen, Jonne Siren, and Pasi Eronen. Energy consumption of always-on applications in wcdma networks. pages 964–968, April 2007.

[49] Paul J.M. Havinga and Gerard J.M. Smit. Energy-efficient wireless networking for multimedia applications. *Wireless Communications and Mobile Computing*, 1(2):165–184, 2001.

[50] Yong He, Ruixi Yuan, Xiaojun Ma, Jun Li, and C. Wang. Scheduled psm for minimizing energy in wireless lans. In *Proceedings of the 2007 IEEE International Conference on Network Protocols*, ICNP'07, pages 154–163, 2007.

[51] Inki Hong, Gang Qu, M. Potkonjak, and M.B. Srivastavas. Synthesis techniques for low-power hard real-time systems on variable voltage processors. In *Proceedings of the 19th IEEE Real-Time Systems Symposium*, pages 178–187, December 1998.

[52] Mohammad Hoque, Matti Siekkinen, and Jukka K. Nurminen. On the energy efficiency of proxy-based traffic shaping for mobile audio streaming. In *Proceedings of the 2011 IEEE Consumer Communications and Networking Conference*, CCNC 2011. IEEE, 2011.

[53] Razak Hossain, Menghui Zheng, and Alexander Albicki. Reducing power dissipation in cmos circuits by signal probability based transistor reordering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(3):361 –368, March 1996.

[54] Young-Si Hwang, Sung-Kwan Ku, Chan-Min Jung, and Ki-Seok Chung. Predictive power aware management for embedded mobile devices. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, ASP-DAC '08, pages 36–41, Los Alamitos, CA, USA, 2008. IEEE Computer Society Press.

[55] Sandy Irani, Sandeep Shukla, and Rajesh Gupta. Online strategies for dynamic power management in systems with multiple power-saving states. *ACM Trans. Embed. Comput. Syst.*, 2:325–346, August 2003.

[56] C. Isci and M. Martonosi. Runtime power monitoring in high-end processors: methodology and empirical data. In *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-36, pages 93 – 104, December 2003.

[57] Inwhee JOE, Won-Tae KIM, and Seokjoon HONG. A network selection algorithm considering power consumption in hybrid wireless networks. *IEICE Transactions on Communications*, E91.B(1):314–317, 2008.

[58] Christine E. Jones, Krishna M. Sivalingam, Prathima Agrawal, and Jyh Cheng Chen. A survey of energy efficient network protocols for wireless networks. *Wirel. Netw.*, 7:343–358, September 2001.

[59] J.R.Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, October 1998.

[60] Seungwoo Kang, Jinwon Lee, Hyukjae Jang, Hyonik Lee, Youngki Lee, Souneil Park, Taiwoo Park, and Junehwa Song. Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In *Proceeding of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 267–280, New York, NY, USA, 2008. ACM.

[61] Aman Kansal and Feng Zhao. Fine-grained energy profiling for power-aware application design. *SIGMETRICS Perform. Eval. Rev.*, 36:26–31, August 2008.

[62] Imre Kelényi and Jukka K. Nurminen. Bursty content sharing mechanism for energy-limited mobile devices. In *Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, PM2HW2N '09, pages 216–223, New York, NY, USA, 2009. ACM.

[63] Imre Kelényi and Jukka K. Nurminen. Cloudtorrent: Energy-efficient bittorrent content sharing for mobile devices via cloud services. In *Proceedings of the 7th IEEE Consumer Communications and Networking Conference*, CCNC '10, pages 1 –2, January 2010.

[64] Donnie H. Kim, Younghun Kim, Deborah Estrin, and Mani B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 43–56, New York, NY, USA, 2010. ACM.

[65] Mikkel Baun Kjaergaard, Sourav Bhattacharya, Henrik Blunck, and Petteri Nurmi. Energy-efficient trajectory tracking for mobile devices. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 307–320, New York, NY, USA, 2011. ACM.

[66] Ronny Krashinsky and Hari Balakrishnan. Minimizing energy for wireless web access with bounded slowdown. *Wirel. Netw.*, 11:135–148, January 2005.

[67] Branislav Kveton, Prashant Gandhi, Georgios Theocharous, Shie Mannor, Barbara Rosario, and Nilesh Shah. Adaptive timeout policies for fast fine-grained power management. In *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence*, pages 1795–1800. AAAI Press, 2007.

[68] Ana Belen Lago and Iker Larizgoitia. An application-aware approach to efficient power management in mobile devices. In *Proceedings of the 4th*

International ICST Conference on Communication System Software and Middleware, COMSWARE '09, pages 11:1–11:10, New York, NY, USA, 2009. ACM.

[69] Johann Laurent, Nathalie Julien, Eric Senn, and Eric Martin. Functional level power analysis: An efficient approach for modeling the power consumption of complex processors. In *Proceedings of the Conference on Design, automation and test in Europe - Volume 1*, DATE '04, pages 10666–, Washington, DC, USA, 2004. IEEE Computer Society.

[70] Benjamin C. Lee and David M. Brooks. Accurate and efficient regression modeling for microarchitectural performance and power prediction. *SIGOPS Oper. Syst. Rev.*, 40:185–194, October 2006.

[71] Tao Li and Lizy Kurian John. Run-time modeling and estimation of operating system power consumption. In *Proceedings of the 2003 ACM SIGMETRICS International conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '03, pages 160–171, New York, NY, USA, 2003. ACM.

[72] Xiaodong Li, Ritu Gupta, Sarita V. Adve, and Yuanyuan Zhou. Cross-component energy management: Joint adaptation of processor and memory. *ACM Trans. Archit. Code Optim.*, 4, September 2007.

[73] Jiayang Liu and Lin Zhong. Micro power management of active 802.11 interfaces. In *Proceeding of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 146–159, New York, NY, USA, 2008. ACM.

[74] Xiaotao Liu, Prashant Shenoy, and Mark Corner. Chameleon: application level power management with performance isolation. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 839–848, New York, NY, USA, 2005. ACM.

[75] Xiaoan Lu, T. Fernaine, and Yao Wang. Modelling power consumption of a h.263 video encoder. In *Proceedings of the 2004 International Symposium on Circuits and Systems*, volume 2 of *ISCAS'04*, pages 77–80, May 2004.

[76] Gregor Maier, Fabian Schneider, and Anja Feldmann. A first look at mobile hand-held device traffic. In *Proceedings of the 11th International Conference on Passive and Active Measurement*, PAM'10, pages 161–170, Berlin, Heidelberg, 2010. Springer-Verlag.

[77] Justin Manweiler and Romit Roy Choudhury. Avoiding the rush hours: Wifi energy management via traffic isolation. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 253–266, New York, NY, USA, 2011. ACM.

[78] Robert N. Mayo and Parthasarathy Ranganathan. Energy consumption in mobile devices: Why future systems need requirements-aware energy scale-down. In *Lecture Notes in Computer Science*, 2003.

[79] Jung-hi Min, Hojung Cha, and Jongho Nang. Energy management for interactive applications in mobile handheld systems. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 795–799, New York, NY, USA, 2007. ACM.

[80] S. Mohapatra, N. Dutt, A. Nicolau, and N. Venkatasubramanian. Dynamo: A cross-layer framework for end-to-end qos and energy optimization in mobile handheld devices. *IEEE Journal on Selected Areas in Communications*, 25(4):722 –737, May 2007.

[81] Shivajit Mohapatra, Radu Cornea, Nikil Dutt, Alex Nicolau, and Nalini Venkatasubramanian. Integrated power management for video streaming to mobile handheld devices. In *Proceedings of the 11th ACM International Conference on Multimedia*, MULTIMEDIA '03, pages 582–591, New York, NY, USA, 2003. ACM.

[82] Shivajit Mohapatra and Nalini Venkatasubramanian. Parm: Power aware reconfigurable middleware. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, ICDCS '03, pages 312–, Washington, DC, USA, 2003. IEEE Computer Society.

[83] J.P. Monks, V. Bharghavan, and W.-M.W. Hwu. Transmission power control for multiple access wireless packet networks. In *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks*, LCN'00, pages 12 –21, 2000.

[84] Sergiu Nedevschi, Lucian Popa, Gianluca Iannaccone, Sylvia Ratnasamy, and David Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pages 323–336, Berkeley, CA, USA, 2008. USENIX Association.

[85] L. Negri, M. Sami, Que Dung Tran, and D. Zanetti. Flexible power modeling for wireless systems: power modeling and optimization of two bluetooth implementations. In *Proceedings of the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks,*, WoWMoM'05, pages 408 – 416, June 2005.

[86] Jeongyeup Paek, Kyu-Han Kim, Jatinder P. Singh, and Ramesh Govindan. Energy-efficient positioning for smartphones using cell-id sequence matching. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 293–306, New York, NY, USA, 2011. ACM.

[87] Abhinav Pathak, Y. Charlie Hu, Ming Zhang, Paramvir Bahl, and Yi-Min Wang. Fine-grained power modeling for smartphones using system call tracing. In *Proceedings of the 6th Conference on Computer systems*, EuroSys '11, pages 153–168, New York, NY, USA, 2011. ACM.

[88] Trevor Pering, Yuvraj Agarwal, Rajesh Gupta, and Roy Want. Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, MobiSys '06, pages 220–232, New York, NY, USA, 2006. ACM.

[89] Gian Paolo Perrucci, Frank H.P. Fitzek, Giovanni Sasso, Wolfgang Kellerer, and Jorg Widmer. On the impact of 2g and 3g network usage for mobile phones' battery life. In *Proceedings of the 15th European Wireless Conference*, EW'09, pages 255 –259, May 2009.

[90] Christopher Pluntke, Lars Eggert, and Niko Kiukkonen. Saving mobile device energy with multipath tcp. In *Proceedings of the 6th International workshop on Mobility in the Evolving Internet Architecture*, MobiArch '11, pages 1–6, New York, NY, USA, June 2011. ACM.

[91] Christian Poellabauer and Karsten Schwan. Energy-aware media transcoding in wireless systems. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications*, PerCom'04, pages 135 – 144, March 2004.

[92] Christian Poellabauer and Karsten Schwan. Energy-aware traffic shaping for wireless real-time applications. In *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS'04, pages 48 – 55, may 2004.

[93] N.R. Potlapally, S. Ravi, A. Raghunathan, and N.K. Jha. A study of the energy consumption characteristics of cryptographic algorithms and security protocols. *IEEE Transactions on Mobile Computing*, 5(2):128 – 143, feb. 2006.

[94] Bodhi Priyantha, Dimitrios Lymberopoulos, and Jie Liu. Eers: Energy efficient responsive sleeping on mobile phones. In *Proceedings of the International Workshop on Sensing for App Phone*, PhoneSense '10, 2010.

[95] Feng Qian, Z. Morley Mao, Zhaoguang Wang, Subhabrata Sen, Alexandre Gerber, and Oliver Spatscheck. Characterizing radio resource allocation for 3g networks. In *Proceedings of the 10th Annual Conference on Internet Measurement*, IMC '10, pages 137–150. ACM, 2010.

[96] Feng Qian, Zhaoguang Wang, Alexandre Gerber, Zhuoqing Mao, Subhabrata Sen, and Oliver Spatscheck. Profiling resource usage for mobile applications: A cross-layer approach. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 321–334, New York, NY, USA, June 2011. ACM.

[97] Daji Qiao, Sunghyun Choi, and Kang G. Shin. Interference analysis and transmit power control in ieee 802.11a/h wireless lans. *IEEE/ACM Trans. Netw.*, 15:1007–1020, October 2007.

[98] Gang Qu, Naoyuki Kawabe, Kimiyoshi Usami, and Miodrag Potkonjak. Function-level power estimation methodology for microprocessors. In *Proceedings of the 37th Annual Design Automation Conference*, DAC '00, pages 810–813, New York, NY, USA, 2000. ACM.

[99] R. H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics, 2nd edition, 2006.

[100] Moo-Ryong Ra, Jeongyeup Paek, Abhishek B. Sharma, Ramesh Govindan, Martin H. Krieger, and Michael J. Neely. Energy-delay tradeoffs in smartphone applications. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 255–270, New York, NY, USA, 2010. ACM.

[101] Moo-Ryong Ra, Anmol Sheth, Lily Mummert, Padmanabhan Pillai, David Wetherall, and Ramesh Govindan. Odessa: enabling interactive perception applications on mobile devices. In *Proceedings of the 9th Interna-*

tional Conference on Mobile Systems, Applications, and Services, MobiSys '11, pages 43–56, New York, NY, USA, 2011. ACM.

[102] Vijay Raghunathan, Trevor Pering, Roy Want, Alex Nguyen, and Peter Jensen. Experience with a low power wireless mobile computing platform. In *Proceedings of the 2004 International Symposium on Low Power Electronics and Design*, ISLPED'04, pages 363–368, New York, NY, USA, 2004. ACM.

[103] Vijay Raghunathan, Mani B. Srivastava, and Rajesh K. Gupta. A survey of techniques for energy efficient on-chip communication. In *Proceedings of the 40th Annual Design Automation Conference*, DAC '03, pages 900–905, New York, NY, USA, 2003. ACM.

[104] Ahmad Rahmati, Angela Qian, and Lin Zhong. Understanding human-battery interaction on mobile phones. In *Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '07, pages 265–272, New York, NY, USA, 2007. ACM.

[105] Ahmad Rahmati and Lin Zhong. Context-for-wireless: context-sensitive energy-efficient wireless data transfer. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, pages 165–178, New York, NY, USA, 2007. ACM.

[106] Ahmad Nazir Raja, Zhihua Jin, and Matti Siekkinen. Energy efficient client-centric shaping of multi-flow tcp traffic. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, GREENCOM-CPSCOM '10, pages 260–267, Washington, DC, USA, 2010. IEEE Computer Society.

[107] Daler Rakhmatov and Sarma Vrudhula. Energy management for battery-powered embedded systems. *ACM Trans. Embed. Comput. Syst.*, 2:277–324, August 2003.

[108] Kishore Ramachandran, Ravi Kokku, Honghai Zhang, and Marco Gruteser. Symphony: synchronous two-phase rate and power control in 802.11 wlans. In *Proceeding of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 132–145, New York, NY, USA, 2008. ACM.

[109] Balaji Raman and Samarjit Chakraborty. Application-specific workload shaping in multimedia-enabled personal mobile devices. In *Proceedings of the 4th International Conference on Hardware Software Codesign*, CODES'06, pages 4–9. ACM Press, 2006.

[110] Enrico Rantala, Arto Karppanen, Seppo Granlund, and Pasi Sarolahti. Modeling energy efficiency in wireless internet communication. In *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds*, MobiHeld '09, pages 67–68, New York, NY, USA, 2009. ACM.

[111] Nishkam Ravi, James Scott, Lu Han, and Liviu Iftode. Context-aware battery management for mobile phones. In *Proceedings of the 6th Annual*

IEEE International Conference on Pervasive Computing and Communications, PerCom'08, pages 224–233, Washington, DC, USA, 2008. IEEE Computer Society.

[112] M.U. Rehman, X. Chen, C.G. Parini, and Z. Ying. Effects of on-pcb location of radiating element on the performance of mobile terminal gps antennas in multipath environment. In *Proceedings of the 5th European Conference on Antennas and Propagation*, EUCAP'11, pages 410 –414, April 2011.

[113] A. Rice and S. Hay. Decomposing power measurements for mobile devices. In *Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications*, PerCom'10, pages 70 –78, April 2010.

[114] Oriana Riva. Contory: A Middleware for the Provisioning of Context Information on Smart Phones. In *Proceedings of the 7th ACM International Middleware Conference*, volume 4290 of *Middleware'06*, pages 219–239. Springer, 2006.

[115] Peng Rong and Massoud Pedram. Hierarchical power management with application to scheduling. In *Proceedings of the 2005 International Symposium on Low Power Electronics and Design*, ISLPED '05, pages 269–274, New York, NY, USA, 2005. ACM.

[116] Arjun Roy, Stephen M. Rumble, Ryan Stutsman, Philip Levis, David Mazières, and Nickolai Zeldovich. Energy management in mobile devices with the cinder operating system. In *Proceedings of the 6th Conference on Computer Systems*, EuroSys '11, pages 139–152, New York, NY, USA, 2011. ACM.

[117] Eric Rozner, Vishnu Navda, Ramachandran Ramjee, and Shravan Rayanchu. Napman: network-assisted power management for wifi devices. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 91–106, New York, NY, USA, 2010. ACM.

[118] Aaron Schulman, Vishnu Navda, Ramachandran Ramjee, Neil Spring, Pralhad Deshpande, Calvin Grunewald, Kamal Jain, and Venkata N. Padmanabhan. Bartendr: a practical approach to energy-aware cellular data scheduling. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 85–96, New York, NY, USA, 2010. ACM.

[119] Curt Schurgers, Vijay Raghunathan, and Mani B. Srivastava. Power management for energy-aware communication systems. *ACM Trans. Embed. Comput. Syst.*, 2:431–447, August 2003.

[120] Sanaa Sharafeddine and Rakan Maddah. A lightweight adaptive compression scheme for energy-efficient mobile-to-mobile file sharing applications. *J. Netw. Comput. Appl.*, 34:52–61, January 2011.

[121] Eugene Shih, Paramvir Bahl, and Michael J. Sinclair. Wake on wireless: an event driven energy saving strategy for battery operated devices. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, MobiCom '02, pages 160–171, New York, NY, USA, 2002. ACM.

[122] Alex Shye, Benjamin Scholbrock, and Gokhan Memik. 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*, MICRO 42, pages 168–178, New York, NY, USA, 2009. ACM.

[123] Eero Sillasto, Le Wang, and Jukka Manner. Using compression energy efficiently in mobile environment. In *Proceedings of the 2010 IEEE/ACM Internationall Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing*, GREENCOM-CPSCOM '10, pages 9–16, Washington, DC, USA, 2010. IEEE Computer Society.

[124] Harkirat Singh and Suresh Singh. Energy consumption of tcp reno, newreno, and sack in multi-hop wireless networks. *SIGMETRICS Perform. Eval. Rev.*, 30:206–216, June 2002.

[125] Dr.ir. G.J.M. Smit. Qos scheduling for energy-efficient wireless communication. In *Proceedings of the International Conference on Information Technology: Coding and Computing*, ITCC'01, page 167, Washington, DC, USA, 2001. IEEE Computer Society.

[126] David C. Snowdon, Etienne Le Sueur, Stefan M. Petters, and Gernot Heiser. Koala: a platform for os-level power management. In *Proceedings of the 4th ACM European Conference on Computer Systems*, EuroSys '09, pages 289–302, New York, NY, USA, 2009. ACM.

[127] David C. Snowdon, Stefan M. Petters, and Gernot Heiser. Accurate on-line prediction of processor and memory energy usage under voltage scaling. In *Proceedings of the 7th ACM & IEEE International Conference on Embedded Software*, EMSOFT '07, pages 84–93, New York, NY, USA, 2007. ACM.

[128] Jacob Sorber, Nilanjan Banerjee, Mark D. Corner, and Sami Rollins. Turducken: hierarchical power management for mobile devices. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, pages 261–274, New York, NY, USA, 2005. ACM.

[129] Enhua Tan, Lei Guo, Songqing Chen, and Xiaodong Zhang. Psm-throttling: Minimizing energy consumption for bulk data communications in wlans. In *Proceedings of IEEE International Conference on Network Protocols*, ICNP'07, pages 123 –132, October 2007.

[130] T. K. Tan, A. Raghunathan, and N. K. Jha. Energy macromodeling of embedded operating systems. *ACM Trans. Embed. Comput. Syst.*, 4:231–254, February 2005.

[131] Ying Tan and Qinru Qiu. A framework of stochastic power management using hidden markov model. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '08, pages 92–97, New York, NY, USA, 2008. ACM.

[132] Arvind Thiagarajan, James Biagioni, Tomas Gerlich, and Jakob Eriksson. Cooperative transit tracking using smart-phones. In *Proceedings of the*

8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10, pages 85–98, New York, NY, USA, 2010. ACM.

[133] Arvind Thiagarajan, Lenin Ravindranath, Hari Balakrishnan, Samuel Madden, and Lewis Girod. Accurate, low-energy trajectory mapping for mobile devices. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 20–20, Berkeley, CA, USA, 2011. USENIX Association.

[134] Cheng-Lin Tsao and Raghupathy Sivakumar. On effectively exploiting multiple wireless interfaces in mobile hosts. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pages 337–348, New York, NY, USA, 2009. ACM.

[135] Narseo Vallina-Rodriguez, Pan Hui, Jon Crowcroft, and Andrew Rice. Exhausting battery statistics: understanding the energy demands on mobile handsets. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds*, MobiHeld '10, pages 9–14, New York, NY, USA, 2010. ACM.

[136] Vasanth Venkatachalam and Michael Franz. Power reduction techniques for microprocessor systems. *ACM Comput. Surv.*, 37:195–237, September 2005.

[137] Bokyung Wang and Suresh Singh. Analysis of tcp's computational energy cost for mobile computing. *SIGMETRICS Perform. Eval. Rev.*, 31:296–297, June 2003.

[138] Andreas Weissel, Matthias Faerber, and Frank Bellosa. Application characterization for wireless network power management. In *Proceedings of the International Conference on Architecture of Computing Systems*, ARCS'04, January 2004.

[139] Yu Xiao, Pan Hui, Petri Savolainen, and Antti Ylä-Jääski. Cascap: cloud-assisted context-aware power management for mobile devices. In *Proceedings of the 2nd International Workshop on Mobile Cloud Computing and Services*, MCS '11, pages 13–18, New York, NY, USA, 2011. ACM.

[140] Yu Xiao, R.S. Kalyanaraman, and A. Ylä-Jääski. Energy consumption of mobile youtube: Quantitative measurement and analysis. In *Proceedings of the 2nd International Conference on Next Generation Mobile Applications, Services and Technologies*, NGMAST'08, pages 61 –69, sept. 2008.

[141] Wen-Hsin Yang, You-Chiun Wang, Yu-Chee Tseng, and Bao-Shuh P. Lin. Energy-efficient network selection with mobility pattern awareness in an integrated wimax and wifi network. *Int. J. Commun. Syst.*, 23:213–230, February 2010.

[142] Chenjie Yu and Peter Petrov. Energy- and performance-efficient communication framework for embedded mpsocs through application-driven release consistency. *ACM Trans. Des. Autom. Electron. Syst.*, 16:8:1–8:39, November 2010.

[143] Heng Zeng, Carla S. Ellis, Alvin R. Lebeck, and Amin Vahdat. Ecosystem: managing energy as a first class operating system resource. *SIGOPS Oper. Syst. Rev.*, 36:123–132, October 2002.

[144] Zheng Zeng, Yan Gao, and P.R. Kumar. Sofa: A sleep-optimal fair-attention scheduler for the power-saving mode of wlans. In *Proceedings of the 31st International Conference on Distributed Computing Systems*, ICDCS'11, pages 87 –98, June 2011.

[145] Jiansong Zhang, K. Tan, Jun Zhao, Haitao Wu, and Yongguang Zhang. A practical snr-guided rate adaptation. In *Proceedings of the 27th Conference on Computer Communications*, INFOCOM'08, pages 2083 –2091, 2008.

[146] Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P. Dick, Zhuoqing Morley Mao, and Lei Yang. 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*, CODES/ISSS '10, pages 105–114, New York, NY, USA, 2010. ACM.

[147] Sushu Zhang, Karam S. Chatha, and Goran Konjevod. Near optimal battery-aware energy management. In *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design*, ISLPED '09, pages 249–254, New York, NY, USA, 2009. ACM.

[148] Jianli Zhuo and Chaitali Chakrabarti. Energy-efficient dynamic task scheduling algorithms for dvs systems. *ACM Trans. Embed. Comput. Syst.*, 7:17:1–17:25, January 2008.

Aalto-DD 139/2011

ISBN 978-952-60-4429-3
ISBN 978-952-60-4430-9 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science and Engineering
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS