Automatic Battery Interface-based Energy Modeling for Wireless Interface on Smartphones

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Chang Ye

Graduate Program in Department of Electrical and Computer Engineering

The Ohio State University
2015

Master’s Examination Committee:
Prof. Xiaorui Wang, Advisor
Prof. Ümit V. Çatalyürek
Abstract

Energy and power consumption has emerged as a key limitation in smartphone design and usage. At the same time, device functionality is increasing more and more fast. The rich functionality increases the pressure on battery lifetime, and also the demand for effective energy management. Thus, optimizing energy consumption has become a critical design issue in both hardware and software of smartphones.

A major fraction of the energy consumption in smartphones comes from the Wi-Fi and cellular interface, which can account for more than 50% of the total device power budget under typical usage pattern [11] and can quickly drain the phone’s battery when transmitting/receiving data at a high rate. The issue becomes particularly critical today due to a variety of network-intensive applications and the high-speed Internet access. Thus, understanding and optimizing wireless interface energy consumption becomes essential in order to maximize the battery lifetime of smartphones. Precisely estimating the energy consumption of wireless interface can allow better utilization of battery, help identify energy-inefficient software design for developers and assist in avoiding battery-unfriendly applications for end users.

In this thesis, we propose battery interface-based energy model for wireless interface on smartphones. Our energy model can be simply installed as a service in Android system without any change at system/kernel level. The result and evaluation show that our model achieves a high estimation accuracy with a negligible overhead.
Dedication

This document is dedicated to my family.
I would never have been able to finish my thesis without the guidance of my committee members, help from friends and colleagues, and support from my family.

I would like to express my gratitude to my advisor, Dr. Xiaorui Wang, for his guidance and patience for the past year. He provided insight and expertise that greatly assisted my research.

I would like to thank my committee member, Dr. Prof. Ümit V. Çatalyürek, for his support and suggestion to my research.

I would like to thank my partner Zichen Xu who always help in system design and test cases design.

I would express my sincere thanks to Dr. Wladimiro Villarroel, for his help and continuous encouragement.

I am also grateful to my friends and colleagues for assistance of programming and for comments that greatly improved the research.

I would like to thank all the authors of references that are foundation to my work.

Finally, I take this opportunity to express gratitude to my parents for their unceasing support. And also thank my girlfriend, Wenting Cao, I would never complete my research without her patience and understanding for the past year.
Vita

July 2008……………………………………Wenling High School, China

July 2012……………………………………B.S. Communication Engineering, Wuhan University of Technology, China

December 2013……………………………..B.S. Electrical and Computer Engineering, The Ohio State University

June 2014 - present………………………….Graduate Research Assistant, Electrical and Computer Engineering, The Ohio State University

Fields of Study

Major Field: Electrical and Computer Engineering
# Table of Contents

Abstract ........................................................................................................................................ ii

Dedication .................................................................................................................................. iii

Acknowledge ............................................................................................................................... iv

Vita ................................................................................................................................................ v

List of Tables ............................................................................................................................. viii

List of Figures .......................................................................................................................... ix

Chapter 1: Introduction ............................................................................................................ 1

1.1 Background ....................................................................................................................... 1

1.2 Scope and Motivations .................................................................................................... 2

1.3 Contribution and Outline ............................................................................................... 4

Chapter 2: Related Work ......................................................................................................... 7

Chapter 3: Energy Model Construction .................................................................................. 10

3.1 System Design .................................................................................................................. 10

3.2 Energy Modeling .............................................................................................................. 11

3.3 Power States of Wireless Interface ............................................................................. 12

3.4 Battery Interface ............................................................................................................. 14

3.5 Model Construction ....................................................................................................... 16

Chapter 4: Result and Evaluation ......................................................................................... 18
List of Tables

Table 1 Power Model of WiFi and Cellular................................................................. 17
Table 2 LG Nexus 5 Platform...................................................................................... 19
Table 3 Error rate of test cases................................................................................. 26
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hardware architecture of smartphones [10]</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Global smartphones usage trend during 2009-2013 [6]</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Hardware Component Usage Analyzer</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>System Architecture</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Power states transition of Wi-Fi</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Power states transition of Cellular</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>Log file of battery interface readings</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>Process of model construction</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>Monsoon Power Monitor [1]</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>Test environment</td>
<td>21</td>
</tr>
<tr>
<td>11</td>
<td>Measured energy consumption and estimated energy consumption of YouTube video streaming</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>Measured energy consumption and estimated energy consumption of Skype video chat</td>
<td>23</td>
</tr>
<tr>
<td>13</td>
<td>Measured energy consumption and estimated energy consumption of Google Drive downloading</td>
<td>23</td>
</tr>
<tr>
<td>14</td>
<td>Power trace of Wi-Fi in active, PSM and idle states</td>
<td>24</td>
</tr>
<tr>
<td>15</td>
<td>Power trace of Cellular in FACH and DCH states</td>
<td>25</td>
</tr>
<tr>
<td>16</td>
<td>Overhead in power</td>
<td>26</td>
</tr>
</tbody>
</table>
1.1 Background

A smartphone is a mobile phone built on mobile computing platform with more advanced computing ability and connectivity than a feature phone [7]. The initial smartphone were devices which mainly combined the functions a personal digital assistant (PDA) and a mobile phone. Today’s models also serve to combine the functions of media players, cameras, video and global positioning system (GPS). In addition, modern smartphone typically also include high-resolution touch screens, web pages rather than just mobile-optimized web sites, and high-speed network access via Wi-Fi, Bluetooth and cellular interfaces. Fig. 1 [10] is the basic architecture of a smartphone including application processor and its subsystems.

Figure 1 Hardware architecture of smartphones [10]
Smartphone is more and more popular in the world. Based on recent statistics, there are 30% of mobile phone users using smartphone at the end of 2013 which is shown in Fig. 2 [6]. And the number is increasing at a high rate. In this way, a lot of research has been conducted on smartphones including battery life, high resolution screen, high speed internet access, smartphone sensing, CPU/GPU performance, memory and so on.

![Global Smartphone Quarterly Unit Shipments & Smartphone Users as % of Mobile Phone Users, 2009 – 2013](image)

*Figure 2 Global smartphones usage trend during 2009-2013 [6]*

### 1.2 Scope and Motivations

Energy consumption is becoming critical metrics in the design and usage of smartphones. That is because the battery capacity is severely restricted by the size and weight of the device. Also in the modern world, smartphones are part of our daily and frequently used. This implies that energy efficiency of smartphones is very significant to
their usability. Hence, optimal management of power consumption of these devices is critical.

At the same time, device functionality is increasing rapidly. Modern high-end mobile phones combine the functionality of a pocket-sized communication device with PC-like capabilities, resulting in what are generally referred to as smartphones [7]. These integrate such diverse functionality as telecommunication, audio and video streaming, web browsing, short-message and email, media, gaming, location service, sensing and more. The problem becomes particularly critical because some of them are power-hungry. The rich functionality increases the pressure on battery lifetime, and deepens the need for effective energy management.

A major fraction of the energy consumption in smartphones comes from the WiFi and cellular interface, which can account for more than 50% of the total device power budget under typical usage pattern [13] and can quickly drain the phone’s battery when transmitting/receiving at high rates. Thus, understanding and optimizing wireless interface energy consumption becomes essential in order to maximize the battery lifetime of smartphones. The problem becomes particularly critical today due to a combination of reasons. Today’s smartphones run a variety of network intensive application which result in a large growth of network traffic. Besides high rate internet access is another reason to make people use smartphone more frequently instead of PC.

It is important to precisely estimate the energy consumption for several reasons. First, it allows better utilization of battery. For example, some smart scheduling policy of OS can be developed to optimize the performance-power tradeoff. Second, accurate energy model can help identify energy-inefficient phase in software. It would be helpful
for software developer to optimize their software design. Furthermore, for end users, precise energy estimation can provide a handy tool to identify which applications are energy-efficient, and which squander energy.

The most important metrics of a model are accuracy, rate and overhead.

- **Accuracy**: The importance of a high accuracy is obvious. High accuracy would make researchers and developer have a better understanding of where energy go. The more fine-grained power management framework can be developed based on accurate energy model.

- **Rate**: A high rate is also important for many reasons. First, energy models are foundation to energy management and optimization in operating system and software. If the rate is low, it would suffer delay in reading the change of power states. In this way, optimization for both OS and application are not efficient. Second, low-rate model can suffer poor accuracy in fine-grained energy estimation. i.e., for video streaming, the time interval at which the sampling is performed can be too course-grained compared to the duration of power states change of wireless interface.

- **Overhead**: High-overhead model can reduce the battery life and sometimes even larger than the energy saved by power management framework.

### 1.3 Contribution and Outline

Existing energy modeling methodologies are however limited in several important ways. First, a lot of existing methods generate the model in the lab using high quality external power meter such as Monsoon Power Monitor. Such method [18] are not only
labor-intensive but also produce fixed energy models that are just fit for a specific smartphone. Second, wireless interface (Wi-Fi/Cellular) always have tail power states where utilization is zero at this time. In this way, API-based power model [12, 13] would miss the tail power and cause error in these scenarios. Third, energy model based on system call [14] might have delay and high overhead problem as the delay of interface drivers. Finally, AppScope [20] proposed an energy model based on the hardware activity information form kernel level. It is fine-grained, but it is not user-friendly because we have to recompile the kernel of smartphones.

In this thesis, we conduct to attempt at using power states of wireless interface and energy readings of battery interface to estimate energy of wireless interface. The work described in this thesis makes four main research contributions:

- We make a detailed analysis of most current energy modeling techniques which are still widely used in the related field. We make a comprehensive evaluation about their advantages and limitation before our own work.
- Realization of automatic energy modeling for wireless interface based on battery interface without external power meter.
- Our design provide high accuracy (less than 10% error rate) with a negligible overhead both in time (less than 0.5 s) and power (total 67 mW).
- Our energy model can be simply installed as a service in Android system without any change at system/kernel level.

The rest of the thesis is organized as follows. Chapter 2 discusses related work in energy modeling. Chapter 3 presents the system design, detailed energy modeling
methodology and implementation details. Chapter 4 presents the evaluation results and conclude the thesis in Chapter 5.
Current efforts on energy modeling on battery-powered devices have mainly focused on the system architecture [15], battery analysis [16, 17], hardware utilization [18, 19, 20], system call tracing [14] and related API [12, 13].

PowerScope [12], which provided the energy consumption of applications at a fine-grained level, but required post-processing using an external device. The developer also needs to import related APIs for energy metering. PowerProf [13] required information on the API usage for each application in order to estimate energy consumption. Both of the two methods fail to estimate the power tail of wireless interface which would influence the accuracy of model.

PowerTuor [18] proposed an online estimation method for hardware utilization, but still required external power meter to calculate power coefficients. It is labor-intensive and the fixed energy model is not a general model for smartphones. This model would cause problem for other types of smartphones with different hardware components.

AppScope [20] is an Android-based energy metering system. This system monitors application’s hardware usage at the kernel level and accurately estimates energy consumption. AppScope is implemented as a kernel module and uses an event-driven monitoring method that generates low overhead and provides high accuracy. AppScope focus on the accuracy of utilization statistics of each hardware component instead of the power coefficient. In order to estimate the utilization statistics of each
hardware, the system analyzes the traces of a system call (Kprobe [21]), as well as the messages for Android binder inter-process communication (IPC). It collects usage information based on an event-driven approach, hence, the energy consumption of each application at a fine-grained level. Fig. 3 is the hardware component usage analyzer of AppScope. This model is accurate and fine-grained. However, we have to recompile the kernel and develop a kernel module to implement AppScope. Thus it is not appropriate and friendly for end users to install it on their own smartphones.

By addressing those method concerns, our study presents an advanced energy modeling approach. Our model is self-constructed based on battery interface without any
external assistance. Also our model, which is only a service installed in Android system, is user friendly without any change in system/kernel level. In addition, compared with several state-of-art energy model approaches, it achieves high estimation accuracy with a negligible overhead both in time and power.
Chapter 3: Energy Model Construction

3.1 System Design

We developed a self-modeling methodology in which a mobile system automatically generates its energy model for wireless interface without any external power meter. Our model achieves the possibility of self-energy measurement through the battery interface.

It has three major components:

- **Activity data collection**: proc, sys, dev file system in Android can provide information of Wi-Fi and cellular interface such as packet rate, data rate and queue size at a specific sampling rate. These data can be used to determine the power states of wireless interface.

- **Energy readings**: smart battery interface can provide us with the battery voltage, current, remaining capacity and temperature. We utilizes the battery interface to figure out the energy consumption through current and voltage readings of battery I/O.

- **Model construction**: power coefficients are derived through variables and response we collected. In our model, variable is the power state based on the activity data of wireless interface and response is the energy consumption based on the current/voltage readings from battery interface.

Fig. 4 is the architecture of the system design.
3.2 Energy Modeling

An energy model estimates the energy consumption by a mobile system in a given time period. An energy model is actually accumulation of power consumption during this given time. For hardware components, they act differently in different power states which
are defined by the activity data we collected. Thus mathematically for a single component, a power model can be modeled as a linear combination of multiple power states and corresponding power coefficients.

In order to estimate the energy consumption of the wireless interface (Wi-Fi/Cellular), we start by introducing a power model of a linear combination of all power states \( x_i \) and corresponding power coefficient \( \beta_i \). The power model \( P \) can be expressed as:

\[
P = \sum_i (x_i \times \beta_i)
\]

Where \( x_i \) is the variable of \( i \)th power state and \( \beta_i \) is the corresponding coefficient which are actually power consumption in this state. Then the total energy model \( E \) in time interval \( T \) can be expressed as:

\[
E = \sum_{j=1}^{T} \sum_{i=1}^{m} (x_i^j \times \beta_i)
\]

Where \( m \) is the total number of different power states for wireless interface. \( x_i^j \) is variable of \( i \)th power state in \( j \)th second. Corresponding power coefficient \( \beta_i \) keeps the same during the time interval \( T \).

### 3.3 Power States of Wireless Interface

**Wi-Fi:**

There are two important parameters to be considered: packet rate and power tail. Based on our experiment, the power consumption of Wi-Fi is related to packet rate instead of data rate. So the Wi-Fi model is derived by exchanging packets between a smartphone and a PC at the packet rate from 0 to 30 packets/per second and then stop the
program for 30 seconds. Based on our observation, we can find that there are three power states for Wi-Fi interface:

- **Idle State**: There is no transmission in this state.
- **Power Saving Mode (PSM) State**: PSM is defined in 802.11 protocol [22]. It is a power state that has lower power consumption after transmitting/receiving data.
- **Active state**: In this state, Wi-Fi is transmitting/receiving data and has a high power consumption.

As shown in Fig. 5, the transition between different power states is based on power tail and packet rate.

![Figure 5 Power states transition of Wi-Fi](image)

**Cellular:**
The network parameters to be considered in cellular model are queue size and power tail which can be test in the same way as Wi-Fi. But this time, we controlled the byte rate instead of packet rate. Thus, the cellular interface contains three important states for the communication:

- **IDLE**: In this state the cellular interface only receives paging messages and does not transmit/receive data.

- **CELL_FACH**: In this state the cellular interface shares a communication channel to the base station. It can access the random/forward access (CELL_RACH/CELL_FACH) common channels. It can be simply regarded as the low-transmission state.

- **CELL_DCH**: In this state, the cellular interface transmits/receives data at high transmission rate with a dedicated channel [23].

The states transition depends on the tail time and trigger event (queue size), as shown in the Fig. 6 below.

![Figure 6 Power states transition of Cellular](image)

### 3.4 Battery Interface

Lithium-ion batteries have only been in mass production since about 1997, following the resolution of various technical problems during their development. Because
they offer the highest energy density with respect to volume and weight, they are widely used in systems ranging from mobile phones to electric cars [25].

A battery interface includes both hardware and software components. The key hardware is a fuel gauge IC that can measure battery-related information [24]. In software, the fuel gauge IC exposes several registers that store the values of related battery information and updates these at a specific rate.

For our platform LG Nexus 5, we can get the following information as shown in Fig. 7: discharge current, remaining capacity, voltage, temperature, and update rate 0.5 Hz.

![Figure 7 Log file of battery interface readings](image)
3.5 Model Construction

We implemented the energy model as a service on Android 4.4.4 to collect data and construct the model. Packet rate and data rate are updated in 

/sys/class/net/interface/statistics/ file and battery current and voltage are updated in

/sys/class/power_supply/battery/ file. A 10 minutes test program, which is frequently transmitting/receiving data between a PC and smartphone at different packet rate and data rate, is used to calculate the power coefficients at different states as shown in Fig. 8. The test covers all 6 power states of Wi-Fi and cellular interface. And it will repeat for 10 times automatically through shell script to get the average value.

Figure 8 Process of model construction

Thus power model of Wi-Fi and Cellular can be expressed as a linear combination of product of power states and power coefficients (Table 1). Variables are updated every 1
second and 2 second for response. So we regard the response as constant during each 2 second interval. After the determination of power coefficient, the energy/power estimation is updated at 1 Hz.

Table 1 Power Model of WiFi and Cellular

<table>
<thead>
<tr>
<th>Power Model</th>
<th>( P_{\text{wifi}} = \text{WiFi}<em>{\text{idle}} \times \beta</em>{\text{idle}} + \text{WiFi}<em>{\text{PSM}} \times \beta</em>{\text{PSM}} + \text{WiFi}<em>{\text{active}} \times \beta</em>{\text{active}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P_{\text{cellular}} = +\text{Cellular}<em>{\text{IDLE}} \times \beta</em>{\text{IDLE}} + \text{Cellular}<em>{\text{FACH}} \times \beta</em>{\text{FACH}} + \text{Cellular}<em>{\text{DCH}} \times \beta</em>{\text{DCH}} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component</th>
<th>Variable</th>
<th>Range</th>
<th>Power Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>WiFi</td>
<td>R(_{\text{packet}})</td>
<td>0-(\infty)</td>
<td>N/A</td>
</tr>
<tr>
<td>WiFi</td>
<td>WiFi(_{\text{idle}})</td>
<td>0,1</td>
<td>(\beta_{\text{idle}}=8) mW</td>
</tr>
<tr>
<td>WiFi</td>
<td>WiFi(_{\text{PSM}})</td>
<td>0,1</td>
<td>(\beta_{\text{low}}=293) mW</td>
</tr>
<tr>
<td>WiFi</td>
<td>WiFi(_{\text{active}})</td>
<td>0,1</td>
<td>(\beta_{\text{active}}=517) mW</td>
</tr>
<tr>
<td>Cellular</td>
<td>R(_{\text{data}})</td>
<td>0-(\infty)</td>
<td>N/A</td>
</tr>
<tr>
<td>Cellular</td>
<td>Queue(_{\text{uplink}})</td>
<td>0-(\infty)</td>
<td>N/A</td>
</tr>
<tr>
<td>Cellular</td>
<td>Queue(_{\text{downlink}})</td>
<td>0-(\infty)</td>
<td>N/A</td>
</tr>
<tr>
<td>Cellular</td>
<td>Cellular(_{\text{IDLE}})</td>
<td>0,1</td>
<td>(\beta_{\text{IDLE}}=15) mW</td>
</tr>
<tr>
<td>Cellular</td>
<td>Cellular(_{\text{FACH}})</td>
<td>0,1</td>
<td>(\beta_{\text{FACH}}=301) mW</td>
</tr>
<tr>
<td>Cellular</td>
<td>Cellular(_{\text{DCH}})</td>
<td>0,1</td>
<td>(\beta_{\text{DCH}}=750) mW</td>
</tr>
</tbody>
</table>
Chapter 4: Result and Evaluation

4.1 Methodology

As we mentioned before, the most important factors for an energy are accuracy, rate and overhead. The rate has been set to 1 Hz. Thus in this section, we evaluate the performance of our energy model including accuracy and overhead. We make the estimation of some network intensive application through our energy model framework.

- Youtube: A web-based video streaming application that uses WiFi/Cellular.
- Skype: A telecommunications application in providing video chat through WiFi/Cellular.
- Google Drive: An online dropbox that uses WiFi/Cellular for downloading.

To evaluate the accuracy of the model, we use the error metric:

\[ E = \frac{\text{estimation} - \text{measurement}}{\text{measurement}} \]

To evaluate the overhead (power) of the model, we separate the whole energy modeling process into three phases: variable collection, response collection and model construction. So we developed three services and each of them just have one phase. In this way, we implement them in Android system and test them individually. The overhead in each phase can be derived through getting the average value of 30 tests.

4.2 Experiment Setup
We made the measurement on a smartphone (LG Nexus 5) with a quad-core Qualcomm Snapdragon 800 2.26GHz “Krait” processor. It is powered by a Li-Po 2300 mAh battery [5]. Table 2 above shows the technical specifications. And we use Monsoon Power Monitor, as shown in Fig. 9, as the external power meter. The Power Monitor hardware and Power Tool software provide a robust power measurement solution for any
single lithium (Li) powered mobile device rated at 4.5 volts (maximum 3 amps) or lower [1]. In our experiment we set 3.7 volts and 500 Hz sampling rate.

To avoid noise in measurement, we disabled the power management service and location service, set the screen brightness level to 100 and set the audio amplifier to the lowest level. As shown in Fig. 10, it is our test environment with a LG Nexus 5 Android phone, Monsoon Power Monitor and a PC to record and analyze the data. In the tests, we use “osuwireless, 57 Mbps” as the Wi-Fi network and “AT&T, 3G” as the cellular network.
4.3 Result and Evaluation

First we measure the energy consumption of all three test cases as the baseline. Then we conduct the following test cases and estimate the energy consumption by our energy model.

**Test case 1** (Video streaming of YouTube) as shown in Fig. 11:

- Video file: mp4, 19.9 MB and 587 second.
- Play YouTube video under the following conditions with only Wi-Fi on and with Cellular on.
• Each test is repeated 30 times to get the average value.

• Compare the estimated energy consumption with baseline to get the error rate.

**Test case 2** (Skype video chat) as shown in Fig. 12:

• Five minutes video chat.

• Have video chat through Skype with only Wi-Fi on and with Cellular on.

• Each test is repeated 30 times to get the average value.

• Compare the estimated energy consumption with baseline to get the error rate.

**Test case 3** (Google Drive Downloading) as shown in Fig. 13:

• File: .rmvb and 119 MB.

• Download a file from Google Drive with only Wi-Fi on and with Cellular on.

• Each test is repeated 30 times to get the average value.

• Compare the estimated energy consumption with baseline to get the error rate.

---

*Figure 11 Measured energy consumption and estimated energy consumption of YouTube video streaming*
Figure 12 Measured energy consumption and estimated energy consumption of Skype video chat

Figure 13 Measured energy consumption and estimated energy consumption of Google Drive downloading
Power trace of Wi-Fi and Cellular:

As shown in Fig. 14, it is the power consumption of Wi-Fi in active, PSM and idle states. Fig. 15 represents the power consumption of cellular interface in CELL_FACH and CELL_DCH state. For both figures, the black line which is our estimation is very close to measured power consumption and don’t have much delay. Thus our model is not only accurate in long term (energy), but also accurate in small interval (power).

Figure 14 Power trace of Wi-Fi in active, PSM and idle states
4.4 Error Rate and Overhead Analysis

The results in Table 3 indicate that our energy model has a high accuracy with less than 10% error rate. The error distribution is stable within 20% compared with measured baseline.

The overhead comes from three different phases which are response collection, variable collection and model construction. Among the three phases, model construction has the most overhead of 32 mW. Our model have a total overhead in power which is only 67 mW which is much lower than any active power states of wireless interface as shown in Fig. 16. Also after the determination of power coefficient, the overhead is only 52 mW which can be negligible. The overhead in time is less than 0.5 s which is much lower than the update interval 1 s.
Table 3 Error rate of test cases

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Error Rate (avg.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YouTube (Wi-Fi)</td>
<td>8.37%</td>
</tr>
<tr>
<td>YouTube (Cellular)</td>
<td>9.08%</td>
</tr>
<tr>
<td>Skype (WiFi)</td>
<td>7.4%</td>
</tr>
<tr>
<td>Skype (Cellular)</td>
<td>9.17%</td>
</tr>
<tr>
<td>Google Drive (WiFi)</td>
<td>5.05%</td>
</tr>
<tr>
<td>Google Drive (Cellular)</td>
<td>6.02%</td>
</tr>
</tbody>
</table>

Figure 16 Overhead in power
Chapter 5: Conclusion and Discussion

In this thesis, we develop an automatically constructed energy model for wireless interface (Wi-Fi/Cellular) on smartphones. Our model is based on Wi-Fi/Cellular power states and energy readings from battery interface. Our model achieve high accuracy (less 10% error rate) and a negligible overhead (total 65 mW) at 1 Hz.

In addition, there are still some problem related to our model. First, we didn’t consider the influence of channel rate and signal strength (RSSI). We conjecture that such factors can be considered in our modeling to further improve the accuracy. Besides, our model is much more accurate for the application with relative stable power states such as downloading/uploading of Google drive because of the update rate is only 1 Hz. In this way, it would cause problem for some application that the power states change frequently in short interval like video streaming and web browsing. The update rate is restricted by the overhead both in time and power. Thus the update rate can be further improved through reducing the overhead in data collection and model construction phases.
Bibliography


