Wireless Pers Commun (2014) 78:333–346
DOI 10.1007/s11277-014-1756-5

A Measurement-Based Prioritization Scheme for Smartphone Applications

Yantao Li · Gang Zhou · George Ruddy · Bruce Cutler

Published online: 12 April 2014

Abstract Applications labeled with priorities can reduce energy consumption on smartphones for network traffic, by conserving up to 56 % energy under typical usage patterns. In this paper, we present a measurement-based prioritization scheme for smartphone applications, which labels each application with a priority. More specifically, we first conduct a measurement of application usage on Android smartphones based on the implementation of *SystemSens*. Based on the measurement results, we observe that two key features of receiving rate (*RX rate*) and screen touch rate (*ST rate*) extracted from *netlog* and *screen* data can reflect the network interactivity and improve the accuracy of the prioritization scheme as well. Then, with the two selected features, we propose an online solution that prioritizes applications on smartphones to conserve energy consumption. Finally, we evaluate the proposed prioritization scheme with data from a user study, and the results demonstrate that our scheme can accurately prioritize applications on smartphones most of the time.

Keywords Measurements · Smartphone applications · RX rate · ST rate · Prioritization

1 Introduction

The preferred platforms for a user's daily computing needs are shifting from traditional desktops or laptops towards mobile smartphones and tablet devices based on the statistics of shipments by CNN in 2011 [1]. With the widespread development, smartphones are able to provide a wide range of applications for common people and professionals, such as emails, web browsing, calendar management, navigation, and location-specific information, as well as traditional phone calls and text messages. As a result of various applications, smartphone usage is becoming more diverse and the amount of time spent on interacting with these devices

Y. Li (✉)
College of Computer and Information Sciences, Southwest University, Chongqing 400715, China
e-mail: yantaoli@swu.edu.cn;yantaoli@foxmail.com;liyantao@live.com

G. Zhou · G. Ruddy · B. Cutler
Department of Computer Science, College of William and Mary, Williamsburg, VA 23187, USA

is increasing, which bring on a growing need to prioritize network traffic based on application priority, thereby conserving limited energy. Moreover, since smartphone applications span a broad and diverse collection of functionality, many of them rely heavily on the ability to transmit data from one smartphone to another, or from a smartphone to a web server. Therefore, smartphone application prioritization and energy efficiency are crucially important fields of study.

While there are significant existing works on smartphones, most of approaches do not consider the application prioritization. Some approaches focus on smartphone-based applications [2–9], such as *LifeMap* [2], which is able to identify points of interest by using smartphones equipped with sensors, and application preference [8], and which examines the popularity of smartphone applications. Other works dedicate to energy efficiency on smartphones [10–18]. For instance, *NAPman* [16] provides an energy-aware fair scheduling algorithm to minimize WiFi energy consumption for mobile devices, even in the presence of competing traffic, and *SAPSM* [17] introduces a smart adaptive PSM solution that prioritizes network traffic based on application priority. Motivated by *NAPman* and *SAPSM*, we provide a novel solution for smartphone application prioritization. Several works devote to smartphone usage [19–23]. For example, *MyExperience* [19] is the first to measure device usage and context information *in situ*, and *ProfileDroid* [22] is used to monitor smartphone applications in user level. Compared to such approaches, we put heavy emphasis on how to extract features that capture network interactivity and design the real-time prioritization scheme.

We are initially motivated by *SAPSM* [17], a Smart Adaptive PSM that prioritizes network traffic based on application priority. The application priority is defined as: if an application is set to high priority, the network traffic of the application is permitted to adaptively switch to Constantly Awake Mode (CAM), while if a low priority, the network traffic of the application is not permitted to switch to CAM, but will instead remain in Power Save Mode (PSM). The reason is that CAM consumes 20 times power than PSM when idle [24]. Accordingly, key challenge of *SAPSM* is how to determine which applications are high priority and which are low priority. To address this challenge, *SAPSM* labels each application with a priority with the assistance of a machine learning technique Support Vector Machine (SVM), which is able to differentiate applications based on the collected information. However, one serious drawback that exists in *SAPSM* is that it uses an offline-trained classifier with the collected information of application usage from a specific user study.

To achieve online application prioritization, we present a measurement-based prioritization scheme for smartphone applications. More specifically, we first conduct a measurement of the application usage over six weeks on Android smartphones based on the implementation of *SystemSens* [23], which is designed by researchers in CENS at UCLA. Through the measurement, we obtain two key features: receiving rate (*RX rate*) and screen touch rate (*ST rate*) extracted from *netlog* and *screen* data, which reflect the network interactivity and improve the accuracy of the prioritization scheme as well. Then, with the two features, we propose an online solution that prioritizes applications on smartphones, thereby conserving energy consumption. Finally, we evaluate the proposed prioritization scheme by comparing with a user study, and the results demonstrate that our scheme can accurately prioritize applications on smartphones most of the time.

To the best of our knowledge, we are the first to investigate the online prioritization scheme for smartphones. The main contributions of this work can be summarized as follows:

- We conduct measurements of the application usage over six weeks on smartphones based on the implementation of *SystemSens*. Based on the analysis of measurement results, we obtain that *RX rate* and *ST rate* can reflect the network interactivity and improve the accuracy of the prioritization scheme.

- We present an online prioritization scheme for smartphone applications to conserve energy consumption.
- We evaluate the proposed scheme, and the results show that it can accurately prioritize applications on smartphones most of the time.

The remainder of this paper is organized as follows: Sect. 2 covers related works, and Sect. 3 details the measurements of data collection and data analysis on smartphone applications. Then, we present a lightweight and online prioritization scheme for smartphone applications in Sect. 4, and evaluate the proposed prioritization scheme in Sect. 5. Finally, Sect. 6 presents conclusions.

2 Related Work

There are significant existing works focusing on smartphones, which fall under three broad categories: smartphone-based applications, energy efficiency on smartphones, and smartphone usage.

Some approaches focus on smartphone-based applications. In [2], the authors develop *LifeMap*, a smartphone-based context provider, which fuses accelerometer, digital compass, WiFi, GPS to track and automatically identify points of interest with room-level accuracy. The authors in [3] present *Cells*, a virtualization architecture for enabling multiple virtual smartphones to run simultaneously on the same physical smartphone in an isolated and secure manner, which introduces a usage model of having one foreground virtual phone and multiple background virtual phones. In [4], the authors propose a novel system that uses dynamic time warping and smartphone based sensor-fusion (accelerometer, gyroscope, magnetometer, GPS, video) to detect, recognize and record these actions without external processing. The authors in [5] address the limitations by collecting anonymized IP-level networking traces in a large tier-1 cellular network in US for one week in August 2010. In contrast to previous work, they use signatures based on HTTP headers (included in the IP-level trace) to distinguish the traffic from different applications. Due to the format of User-Agent in HTTP headers when mobile applications use standard platform APIs, this technique gives them the ability to gather statistics about each individual applications in a marketplace, not just categories of network traffic characterized by port number. Moreover, they examine the spatial and temporal prevalence, locality, and correlation of applications at a national scale, not just in one area or over a small population of users. In [6], the authors present a novel architecture geared for privacy-sensitive applications, where personal information is shared among users and decisions are made based on given optimization criteria. The authors in [7] present the scenario related to a context-aware restaurant recommender for Android smartphones. In [8], the authors study fourteen users of a particular demographic to examine which applications are popular in that demographic, where the phone is used, and how it is shared among users. They find that recreational usage (games, media player, camera) are the most popular type of applications within their study, demonstrating the importance of examining the prioritization of applications that exhibit minimal network usage. The authors in [9] characterize intentional user activities interactions with the device and the applications used and the impact of those activities on network and energy usage by using detailed traces from 255 users. Nevertheless, we are different in that we attempt to examine whether we can utilize application usage information, specifically touchscreen interactions, to help in the process of application prioritization.

Some works devote to energy efficiency on smartphones. The authors in [10] present an efficient power saving scheme, which can minimize power consumption while guaranteeing the delay constraint during call signaling and talk time. In [11], the authors propose an architecture called *Cool-Tether* that harnesses the cellular radio links of one or more mobile smartphones in the vicinity, builds a WiFi hotspot on-the-fly, and provides energy-efficient, affordable connectivity. The authors in [12] present an adaptive location-sensing framework that significantly improves the energy efficiency of smartphones running location-based applications. The underlying design principles of the proposed framework involve substitution, suppression, piggybacking, and adaptation of applications location-sensing requests to conserve energy. In [13], the authors present *RAPS*, a rate-adaptive positioning system for smartphone applications. The authors in [14] build *EET*, the energy emulation toolkit that allows developers to evaluate the energy consumption requirements of their applications against real users energy traces based on a large-scale user study on BlackBerry smartphone users. In [15], the authors introduce *SmartDC*, a mobility prediction-based adaptive duty cycling scheme to provide contextual information about a users mobility: time-resolved places and paths. The authors in [17] propose *SAPSM*, a smart adaptive PSM solution that prioritizes network traffic based on application priority, which labels each application with a priority (high/low) using the assistance of a machine learning classifier. Only high priority applications affect the client's behavior to switch to CAM or Active mode, while low priority traffic is optimized for energy efficiency. Results within this work demonstrate that there are substantial energy savings to be had when applying this technique. However, although this work has been proposed for the prioritization of applications with moderate to high network usage levels, no work has been proposed which examines the possibility of prioritizing applications which are used frequently yet exhibit minimal network usage. Given this, we propose a solution which prioritizes such applications in a lightweight and online manner. In [18], the authors study the power consumption characteristics of 20 users. They examine how battery drain is connected to the activity and behavior of the end-user. In addition, they infer properties such as which components in the phone consume most power and also explore optimizations based on these characteristics. In spite of that, we attempt to examine how applications that are used frequently but exhibit minimal network usage can be prioritized to reduce the overall energy footprint. We gain our understanding of how such applications should be prioritized by examining recorded user touchscreen interactions within specific applications.

Several works propose measurement tools for monitoring smartphone usage. The authors in [19] present *MyExperience*, which is one of the earliest tools built to measure device usage and context information *in situ*, and which runs on Windows Mobile smartphones and supports active context triggered experience sampling. Similarly, *LiveLab* [20] is a research tool implemented for the iPhone platform, which measures usage and different aspects of wireless network performance. A key feature of *LiveLab* is "in-field programmability", which provides the convenience of being able to update the logging tool while it is deployed out in the field. In [21], the authors present a comprehensive analysis of real smartphone usage during a six-month study of real user activity on the Android G1 smartphone. In [22], the authors propose *ProfileDroid*, a comprehensive, multi-layer system for monitoring and profiling applications, which examines application usage at the user-level, focusing on touches that result from interaction between the user and the device. The authors in [23] develop *SystemSens*, a measurement tool designed for Android devices, which collects and logs smartphone usage parameters in the wild in an unobtrusive and expandable way. *SystemSens* allowed us to collect an extensive amount of valuable data, including detailed information about application network traffic and user touch interactions with the smartphone. It is an end-to-end system that includes a web-based visualization and authentication service to provide feedback to users while preserving their privacy. Unlike the *LiveLab* logger which requires a “jailbroken” iPhone device, *SystemSens* is particularly attractive as it can run on stock Android devices. However, our focus in this paper is not on the development of logging tools, but on analyzing their data to aid in the process of effectively prioritizing applications.

3 Measurement

In order to properly identify and analyze the usage of smartphone applications with minimal network traffic, we attempt to gather a sufficient amount of information on smartphone usage with a measurement. In this section, we first focus on data collection by implementation of a smartphone usage monitoring tool *SystemSens* [23], which will help to accentuate the use of applications with conservative network usage on the smartphone, and then analyze the collected data to extract two key features to improve the accuracy of a prioritization scheme.

3.1 Data Collection

We start with a brief introduction of a client implemented with *SystemSens*, and then describe the setup of a server for collecting data.

Client This part is largely predicated on the implementation of a smartphone usage monitoring tool *SystemSens* designed by Falaki H. et al. in CENS at UCLA, which consists of an Android logging client and a visualization web service. It is a lightweight and dynamic background tool for the Android platform designed to collect and log smartphone usage parameters in the wild in an unobtrusive and expandable way. The tool is designed to be unobtrusive: it has no user interface to minimize impact on usage; and it has a small footprint in terms of memory, CPU, and energy consumption. As a result, we implement the *SystemSens* on a client, which is a brand-new Motorola Droid X2 smartphone running Android operating system and serviced by Verizon. Also, note that on occasions a period of data will be flushed from the records on account of resource limitations on the smartphone. It is likely that the design of *SystemSens* causes the overwriting of data after a certain log limit is reached, which may occur over a three-day period data generation without plugging in the smartphone. Through the remainder of testing the subject consistently charges the smartphone at night in order to ensure that no more data are lost. In the measurement, we choose only one single user and a single privately provided client device, due to the limited time and resources, and choose Verizon service that limits the interactivity in order to avoid compromising the users’ smartphone usage. Though this measurement only consists of a single subject, it would be trivial to increase the magnitude as discussed in Sect. 6. The intent behind gathering these resources is diverse, though in this environment we are most concerned with measuring touch frequency and the correlation of touch screen usage and applications with conservative network usage. This will prove to be useful in helping us identify alternative prioritization opportunities to better optimize energy consumption on the smartphone when heavy use of low-network applications exists.

Server Given that Android is based on the Linux operating system, we are able to collect lots of useful information from Linux logs, such as /proc and /sys with the implementation of *SystemSens* on the client through a collection of classes known as sensors. In addition, *SystemSens* is able to capture smartphone usage context includes human interactivity (touches), application usage, battery consumption, battery voltage, battery current, battery temperature, all transmissions made to and from the client including disconnect records, resource usage, memory usage, and GPS usage. Through these sensors and polls, there will be an abundant amount of context data generated even for a single subject. The generated data cannot be stored on the client for a long time, because of the limited resources on the smartphone. Due to the importance of the large amount of data, it must be uploaded to a server for permanent storage, which is only carried out during smartphone charges. The use of *SystemSens* requires a storage location for maintaining data outside the phone client. This is necessary for two reasons: 1) data tend to build up quickly on the phone if an upload does not occur for multiple days; and 2) the server side *SystemSens* application has a convenient graphing tool which retrieves and plots client data by user. Note that there is a large enough gap between uploads and it is possible that resource limitations would cause *SystemSens* to purge old data. In order to backup all the data collected by a client, we set up a server, which can be accessed through a HTTP connection. The server is located in Room 223 of Millington Hall at the College of William and Mary, and is running Ubuntu with an Apache/MySQL setup and the maintenance typically is occurred through an ssh connection. It should be noted for those familiar with the design of *SystemSens* that the server is set up to accept POST data on port 80 as opposed to the suggested HTTPS port 443. Logging data to the server typically took place from 9 PM to midnight daily between October 20th, 2011 and November 16th, 2011.

3.2 Data Analysis

With the measurement of the server logged records from the user majoring in Computer Science over a period of October 21st through November 15th, 2011, we obtain over 200,000 data records that can be classified into 24 types of data. We list the measurement records in Table 1 with 3 columns and 24 rows. As illustrated in Table 1, each row is composed of the data type, the number of corresponding records, and the percentage of records for each type, which is arranged in a descending order of the number of corresponding records. As demonstrated, within all the data types, *netlog* and *screen* have 6.93 and 1.18 % of the total records, respectively, which are of greatest interest in this paper, because these are the most effective features to capture user's usage context. The *netlog* data type collects all instances of network activity including transmission and reception of data for specific processes and applications running on the client device, and the *screen* data type records instances of touch screen usage and associates each interaction with the active application.

In order to conduct a concrete analysis, we specify the dates between the 26th and 28th of October 2011 for touch screen usage information in a high-level perspective. The reason why we choose this specific time period is that it contains a large number of screen touch interactions, which provide the best opportunity to examine the relationship between application usage and network activity data captured by the *netlog* data type associated with the same time period.

With the data records from the *netlog* data type over the specified two-day period, we plot the receiving rate (*RX rate*) and transmitting rate (*TX rate*) of eleven applications in Fig. 1. As illustrated in Fig. 1, the first major point of note is that *RX rates* are considerably higher than *TX rates* for all the applications. This can be explained by the fact that smartphones are more receivers rather than producers of information, and the time spent on receiving is usually much longer than that of transmitting, as noted by the authors in [17]. A further observation of the applications in the figure is that Twitter and Firefox are network intensive, which have the highest *RX rates* with 4,930 bytes/s and 5,353 bytes/s, respectively, while Settings, Contacts, and Words with Friends are with lower network interactivity, which have

Table 1 Measurement records

Data type	Number of corresponding records	Percentage of records (%)
cpu	37,527	14.07
memory	18,536	6.95
meminfo	18,532	6.95
netdev	18,527	6.95
appresource	18,522	6.95
netiflog	18,520	6.95
recentapps	18,512	6.94
servicelog	18,506	6.94
activitylog	18,491	6.93
netlog	18,490	6.93
celllocation	9,724	3.65
signal	6,832	2.56
wifiscan	3,810	1.43
screen	3,154	1.18
battery	2,343	0.88
network	1,805	0.68
dataconnection	1,789	0.67
callstate	438	0.16
sms	357	0.13
servicestate	356	0.13
call	346	0.13
message	102	0.04
systemsens	97	0.04
callforwarding	94	0.04

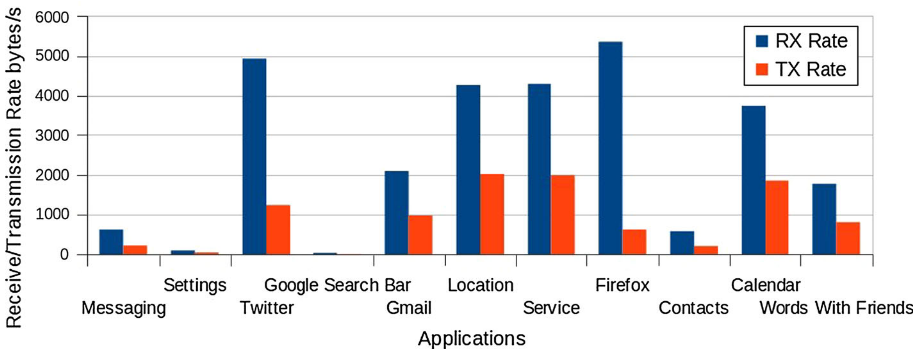


Fig. 1 Application usage

more conservative *RX* or *TX rates*. Therefore, *RX* or *TX rates* from *netlog* data can capture the network intensive applications. On the other hand, with the data records from the *screen* data type, we plot the percentage of screen touch interactions per hour for the same eleven applications in Fig. 2, where the x-axis indicates the percentage of interactions per hour and the y-axis represents the corresponding eleven applications. As shown in Fig. 2, Calendar

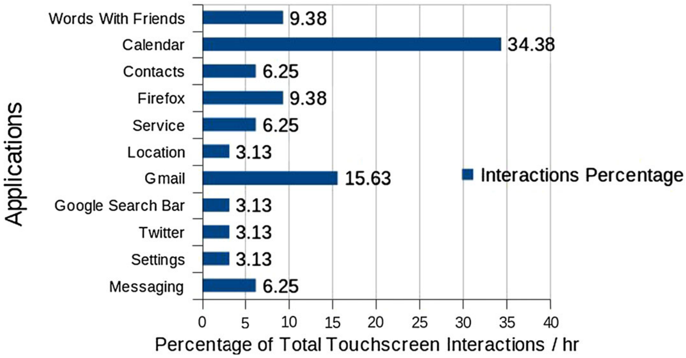


Fig. 2 The percentage of touchscreen interactions per hour

and Gmail have the first two highest interaction percentages per hour with 34.38 and 15.6 %, respectively. Comparing with Twitter in Fig. 1, where it shows a high *RX rate*, it shows a lowest interaction percentage within the same test period in Fig. 2. Such a fact demonstrates that there are substantial amounts of background traffic associated with the Twitter application. Therefore, screen touch rate (*ST rate*) from *screen* data can capture users' preference on touch screen.

In addition to the statistics of the two application traffic features (*RX rate* and *TX rate*), we conduct a small user survey to gauge user opinion on whether applications are thought of as delay tolerant or delay sensitive. The purpose of the survey is to determine whether user opinion on applications matches up with the application traffic features based on prioritization scheme presented later. We randomly interview five application users on the Richmond road in the survey. Note that if we choose users from different demographic population, we believe our following findings still hold, because we select a new feature compared with existing works, which is able to capture the users' usage patterns. Based on the results of this survey, the interactive game Words with Friends produces some discrepancies with the current simplistic prioritization scheme. Though its inbound and outbound traffic would classify Words with Friends to a low priority application, the application touch screen usage information combined with survey results indicate that this may be a misclassification. At around 10 %, this specific application demonstrates a relatively high percentage of total screen touch interactions relative to other applications. Considering the results from Figs. 1 and 2, Words with Friends is an example of an application which we believe would be mislabeled as a low priority application under current circumstances. We attribute these a few mislabeled cases to users' preferences, because different users have different usage patterns and preferred applications. The analysis of measured *netlog* and *screen* data shows that the application traffic features (*RX* and *TX rates*) associated with *ST rate* feature can improve the accuracy of prioritization schemes.

Based on the analysis of the measurement and the results of user survey, we find that: 1) a number of applications the users deemed delay sensitive would be misclassified as low priority if only consider application traffic (*RX or TX rates*); and 2) further analysis shows that this could be resolved by examining the regularity of application usage through the consideration of *ST rates*. The authors believe that this information, considered in tandem with *RX* and *ST rates*, could provide a lightweight and online prioritization scheme which can be reapplied intermittently as application measurements change.

4 The Prioritization Scheme

In this section, we first define the priority of applications used in this paper, and then present a lightweight and online prioritization scheme for smartphone applications.

A prioritization scheme for smartphone applications is able to prioritize the smartphone applications according to application usage frequency, data traffic, delay sensitivity and so on, thereby improving user's experience and reducing energy consumption. As defined in *SAPSM* [17], when an application is set to high priority, the application's network traffic is permitted to adaptively switch to CAM; a low priority setting for an application means that application's network traffic is not permitted to switch to CAM, but will instead remain in PSM. Based on the measurement in Sect. 3.2, we select two features *RX rates* and *ST rates* to capture the network traffic of smartphone applications.

Algorithm 1: The Prioritization Scheme

Input: A set of applications on smartphone
Output: Prioritized applications in two sets: H and L

$H = \emptyset$ *// set of applications with high priority*
$L = \emptyset$ *// set of applications with low priority*
Monitor the RX rates RX_i for all the applications over a specified time period
for $i = 1$ ***to*** $ApplicationNumber$ **do**
    if $RX_i \geq$ *mean(RX rates)* **then**
        $H = H \cup \{Application_i\}$
    end
    else if $RX_i <$ *mean(RX rates)* **then**
        $L = L \cup \{Application_i\}$
    end
end
Examine the ST rates ST_j for all the applications classified in set L
for $j = 1$ ***to*** $|L|$ **do**
    if $ST_j \geq$ *mean(ST rates)* **then**
        $H = H \cup \{Application_j\}$
    end
end
return H *and* L

After detailing the definitions of high-priority and low-priority applications, we present a lightweight and online prioritization scheme as discussed in Algorithm 1, which uses *RX rates* and *ST rates* to prioritize applications on smartphones. As demonstrated in [17] that smartphones are more receivers rather than producers of information and the time spent on receiving is usually much longer than that of transmitting, we select *RX rate* feature to capture the usage of applications in the proposed schemes. The prioritization scheme for smartphones begins with monitoring *RX rates* of all the applications over a specified time period by *SystemSens*, where we use RX_i to denote the *RX rate* of application i and $ApplicationNumber$ to indicate the number of applications on the testing smartphone. Then, we compute the mean value of *RX rates* of all the applications denoted as $mean$(*RX rates*). Based on $mean$(*RX rates*), we classify all the applications into two sets: a high-priority set denoted as H and a low-priority set indicated as L. More specifically, we classify application i into set H if $RX_i \geq mean$(*RX rates*), and classify application i into set L if $RX_i < mean$(*RX rates*). As we discussed in Sect. 3.2, only *RX rate* feature is not sufficient for the classification. We next examine *ST rates* of the applications classified in set L, where we use ST_j to indicate *ST rate* of application j and $|L|$ to denote the number of elements in set L. Based on the calculation of the mean value of iteration times denoted as $mean(ST\, rates)$, we reclassify application j from set L into set H if $ST_j \geq mean(ST\, rates)$. Finally, we obtain the prioritized application in sets of H and L.

Compared with the SVM solution in [17], the proposed scheme is: 1) a lightweight solution, which can be implemented on a smartphone with minimal effort; and 2) an online scheme, where the applications are prioritized by *RX rates* and *ST rates* that can be calculated in real time.

5 Evaluation

In this section, we first present a basic evaluation of the proposed prioritization scheme by a random user study, and then compare the results of the prioritized application with that of the user study.

We implement the proposed prioritization scheme on a testing smartphone, which is a brand-new Motorola Droid X2 smartphone running Android operating system and serviced by Verizon, and then conduct a user study by using this smartphone. A random mixture of ten technical and non-technical users participate in this user study, and the major distribution of all the participants is summarized in Table 2. Note that the user attending the data collection did not participate in this user study. In the study, each application is set to low priority as the default configuration. During the user study, each participant is required to use each of the eleven applications for ten minutes and we assign each participant a set of instructions that they should follow as well. We vary the degrees of interactivity among the instructions of all participants and ask them to determine whether they feel the observed latency is acceptable. The answers from participants are used as labels for the applications. If a participant feels the observed latency is unacceptable, this application is labeled as high priority. Otherwise, if the perceived latency does not impact the users' experience, the application will be labeled as low priority. At the end of the user study, we do a brief survey to collect participants' basic information such as their majors and their experience about smartphones. At the same time, the proposed prioritization scheme installed on the same smartphone outputs the prioritization of all the applications. To evaluate the accuracy of the proposed scheme, we compare the user study results with the outputs of the scheme, and the corresponding comparison results are listed in Table 3. As we can see from Table 3, for each testing application in every row, we demonstrate user study results and results of the proposed prioritization scheme associated with each application. Note that H denotes high priority and L represents low priority. Especially, in the column of user study results, $H(i)$; $L(j)$ denotes the number of participants labeling high priority or low priority to the testing applications based on their experience in the user study. For instance, $H(7)$; $L(3)$ associated with the application Gmail

Table 2 The major distribution of participants

Majors	Number
Computer Science	3
Mathematics	2
Physics	2
Finance	2
Psychology	1

Table 3 The comparison of results of user study and the proposed scheme

Applications	User study results	The scheme results
Messaging	$H(4)$; $L(6)$	L
Settings	$H(2)$; $L(8)$	L
Twitter	$H(10)$; $L(0)$	H
Google search bar	$H(2)$; $L(8)$	L
Gmail	$H(7)$; $L(3)$	H
Location	$H(4)$; $L(6)$	H
Service	$H(5)$; $L(5)$	H
Firefox	$H(10)$; $L(0)$	H
Contacts	$H(2)$; $L(8)$	L
Calendar	$H(8)$; $L(2)$	H
Words with friends	$H(5)$; $L(5)$	H

represents that seven participants label it with high priority and three with low priority. There is a discrepancy in the results of user study and the scheme for the application Location, which indicates that the proposed scheme may have further improvement.

Therefore, based on the comparison, the proposed scheme can accurately prioritize applications on smartphones most of the time.

6 Conclusions and Future Work

In this paper, we explore alternative smartphone traits in an attempt to create a lightweight and online scheme for the effective prioritization of applications. More specifically, we first conduct a measurement of the application usage on smartphones based on the implementation of *SystemSens*. We observe that two kinds of features *RX* rate and *ST* rate extracted from *netlog* and *screen* data can improve the accuracy of the prioritization scheme. Then, based on the measurements, we propose a lightweight and online prioritization scheme for smartphone applications. Finally, we evaluate the proposed prioritization scheme by comparing with a user study, and the results demonstrate that our scheme can accurately prioritize applications on smartphones most of the time. For some inaccurate cases, we believe these are caused by users' preferences, because there are a few occasions that users disable using the proposed features.

In future work, we plan to broaden the participants encompassing users from different demographic populations for the user study. For example, we can choose five participants from North America, five from Asia, and four from Europe, which can mitigate the culture and regional differences, thereby improving the accuracy of the prioritization scheme.

Acknowledgments The work was supported in part by the Fundamental Research Funds for the Central Universities (Grant Nos. XDJK2013C116, XDJK2013B030, SWU113019, SWU111047), the National Natural Science Foundation of China (Grant No. 61173178).

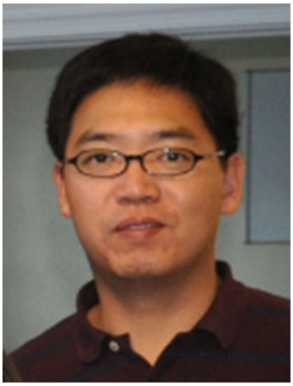
References

1. CNN. (2010). Industry First: Smartphones Pass PCs in Sales. http://tech.fortune.cnn.com/2011/02/07/idc-smartphone-shipment-numbers-passed-pc-in-q4-2010.

2. Chon, Y., & Cha, H. (2011). LifeMap: A smartphone-based context provider for location-based services. *IEEE Pervasive Computing*, *10*(2), 58–67.
3. Andrus, J., Dall, C., Hof, A. V., Laadan, O., & Nieh, J. (2011). Cells: A virtual mobile smartphone architecture. In *Proceedings of the 23rd ACM symposium on operating systems principles (SOSP'11)*. pp. 173–187, October 2011.
4. Johnson, D. A., & Trivedi, M. M. (2011). Driving style recognition using a smartphone as a sensor platform. In *Proceedings of the 14th international IEEE conference on intelligent transportion systems (ITSC'11)*. pp. 1609–1615, October 2011.
5. Xu, Q., Erman, J., Gerber, A., Mao, Z. M., Pang, J., & Venkataraman, S. (2011). Identifying diverse usage behaviors of smartphone apps. In *Proceedings of the 2011 internet measurement conference (IMC'11)*. pp. 329–344, November 2011.
6. Cristofaro, E. D., Durussel, A., & Aad, I. (2011). Reclaiming privacy for smartphone applications. In *Proceedings of the IEEE international conference on pervasive computing and communication (PerCom'11)*. pp. 84–92, March 2011.
7. Gallego, D., Woerndl, W., & Huecas, G. (2013). Evaluating the impact of proactivity in the user experience of a context-aware restaurant recommender for Android smartphones. *Journal of Systems Architecture*. doi:10.1016/j.sysarc.2013.02.004.
8. Rahmati, A., & Zhong, L. (2009). A longitudinal study of non-voice mobile phone usage by teens from an underserved urban community. Tech. Rep. 0515–09, Rice University, 2009.
9. Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R., & Estrin, D. (2010). Diversity in smartphone usage. In *Proceedings of the 8th international conference on mobile systems, applications, and services (MobiSys'10)*. pp. 179–194, June 2010.
10. Jang, S. B., & Kim, Y. G. (2011). An energy-efficient delay reduction technique for supporting WLAN-based VoIP in SmartPhone. *Journal of Systems Architecture*, *57*(10), 934–944.
11. Sharma, A., Navda, V., Ramjee, R., Padmanabhan, V. N., & Belding, E. M. (2009). Cool-tether: Energy efficient on-the-fly WiFi hot-spots using mobile phones. In *Proceedings of the 5th ACM international conference on emerging networking eXperiments and technologies (CoNEXT09)*. pp. 109–120, December 2009.
12. Zhuang, Z., Kim, K. H., & Singh, J. P. (2010). Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th annual international conference on mobile systems, applications and services (MobiSys10)*. pp. 315–330, June 2010.
13. Paek, J., Kim, J., & Govindan, R. (2010). Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of the 8th annual international conference on mobile systems, applications and services (MobiSys10)*. pp. 299–314, June 2010.
14. Oliver, E., & Keshav, S. (2011). An empirical approach to smartphone energy level prediction. In *Proceedings of the 13th international conference on ubiquitous, computing (UbiComp11)*. pp. 345–354, September 2011.
15. Chon, Y., Talipov, E., Shin, H., & Cha, H. (2011). Mobility prediction-based smartphone energy optimization for everyday location monitoring. In *Proceedings of the 9th ACM conference on embedded networked sensor systems (SenSys11)*. pp. 82–95, April 2011.
16. Rozner, E., Navda, V., Ramjee, R., & Rayanchu, S. (2010). NAPman: network-assisted power management for wifi devices. In *Proceedings of the 8th international conference on mobile systems, applications, and services (MobiSys'10)*. pp. 91–106, June 2010.
17. Pyles, A. J., Qi, X., Zhou, G., Keally, M., & Liu, X. (2012). SAPSM: Smart adaptive 802.11 PSM for smartphones. In *Proceedings of the 12th international conference on ubiquitous computing (UbiComp'12)*. pp. 11–20, September 2012.
18. Shye, A., Sholbrock, B., & Memik, G. (2009). Into the wild: Studying real user activity patterns to guide power optimization for mobile architectures. In *Proceedings of the 42nd annual IEEE/ACM international symposium on microarchitecture (MICRO'09)*. pp. 168–178, December 2009.
19. Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., & Landay, J. A. (2007). MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the 5th international conference on mobile systems, applications, and services (MobiSys'07)*. pp. 57–70, June 2007.
20. Shepard, C., Tossell, C., Zhong, L., & Kortum, P. L. (2010). LiveLab: Measuring wireless networks and smartphone users in the field. *ACM SIGMETRICS Performance Evaluation Review*, *38*(3), 15–20.
21. Shye, A., Scholbrock, B., Memik, G., & Dinda, P. A. (2010). Characterizing and modeling user activity on smartphones: Summary. In *Proceedings of the ACM SIGMETRICS international conference on measurement and modeling of computer systems (SIGMETRICS10)*. pp. 375–376, June 2010.
22. Wei, X., Gomez, L., Neamtiu, I., & Faloutsos, M. (2012). ProfileDroid: multi-layer profiling of android applications. In *Proceedings of the 18th annual international conference on mobile computing and networking (Mobicom'12)*. pp. 137–148, August 2012.

23. Falaki, H., Mahajan, R., & Estrin, D. (2011). SystemSens: A tool for monitoring usage in smartphone research deployments. In *Proceedings of the 6th ACM international workshop on mobility in the evolving internet architecture (MobiArch'11)*. pp. 25–30, August 2011.
24. Pyles, A., Ren, Z., Zhou, G., & Liu, X. (2011). SiFi: exploiting VoIP silence for WiFi energy savings in smart phones. In *Proceedings of the 13th international conference on ubiquitous, computing (UbiComp'11)*. pp. 325–334, September 2011.

Yantao Li is an Associate Professor in the College of Computer and Information Sciences at Southwest University, China. He received the Ph.D. degree from the College of Computer Science at Chongqing University, in December 2012. From September 2010 to September 2012, he was a visiting scholar supported by China Scholarship Council working with professor Gang Zhou at the Department of Computer Science in the College of William and Mary, USA. His research area includes wireless communication and networking, sensor networks and ubiquitous computing, and information security.



Gang Zhou is an Associate Professor in the Department of Computer Science at the College of William and Mary, USA. He received his Ph.D. degree from the University of Virginia in 2007 under Professor John A. Stankovic. He has published more than 40 wireless communication and sensor networking papers in prestigious conferences and journals. The total citations of his papers are more than 2400 according to Google Scholar, among which the MobiSys'04 paper has been cited about 600 times. Also, he has 8 papers each of which attracts more than 100 citations since 2004. Dr. Zhou served program vice chair, doctor colloquium panelist, or technical program committee positions for 45 academic conferences, including ACM SenSys, IEEE INFOCOM, IEEE RTSS, etc. He is a reviewer for 30 journals and 50 conferences. Dr. Zhou served as NSF, NIH, and GENI proposal review panelists multiple times during 2008 2012. He received an award for his outstanding service to the IEEE Instrumentation and Measurement Society in 2008. He is also a recipient of the Best Paper Award of IEEE ICNP 2010.

George Ruddy is working toward the Ph.D. degree in the Department of Computer Science at the College of William and Mary. He received the master's degree in the Department of Computer Science at the College of William and Mary, in May 2013. His research is with wireless sensor networks and smartphone applications.

Bruce Cutler is working toward the Ph.D. degree in the Department of Computer Science at the College of William and Mary. He received the master's degree in the Department of Computer Science at the College of William and Mary, in May 2013. His research interests include wireless sensor networking and smartphone energy consumption.