

Personalized Power Saving Profiles Generation Analyzing Smart Device Usage Patterns

Soumya Kanti Datta, Christian Bonnet, Navid Nikaein

EURECOM

Biot, France

Emails: {dattas, bonnet, nikaiein}@eurecom.fr

Abstract—Despite much advancement in mobile computing technologies, the smart devices still suffer from battery limitations. The power consumption largely depends on how the end-users interact with their smart devices. Continuous usage of exotic hardware, inbuilt sensors, network and bright colorful display shorten battery life significantly. Thus the key to develop power saving solutions is to understand the user interactions of smart devices. This paper describes a client-server architecture that proposes personalized power saving profiles by analyzing individual usage patterns. The client is an Android app named “Power Monitor” running on the Android devices. It periodically records usage information from the devices and communicates them to a remote server after a week of monitoring. The server applies an algorithm to generate the usage pattern for each user. Further processing of the patterns provides to power saving profiles and they are sent back to the smart devices. These profiles are highly personalized since they are developed by analyzing individual usage patterns. The profiles will automatically evolve if the usage patterns change over time. We also present two real-life usage patterns and the respective power saving profiles to demonstrate the efficiency of the architecture. The battery level gain using Power Monitor is compared with two popular power saving Android applications. It is shown that Power Monitor can increase the battery life by almost 90 percent. Some related privacy issues are also addressed and privacy preserving usage pattern generation is also discussed.

Keywords—Android; Client-server architecture; Power saving profiles; Privacy; Usage pattern.

I. INTRODUCTION

The Android smartphones and tablets are being adopted at a phenomenal pace. These smart devices empower users with continuous network access, audio-video recording and playback, navigation, sharing of digital content through social media apps and more. These devices also feature powerful CPU, HD camera, various sensors and GPS. The application developers exploit the features to provide state of the art user experience and services through mobile applications. Thus smart devices are becoming increasingly popular among users throughout the world and user practice is undergoing significant change. For instance, users socialize using mobile applications of Google+, Facebook & Twitter, use smartphone camera to take pictures and share them from the devices, connect among peers and navigate maps using GPS. Thus, the activities of users are becoming more and more smart-device centric. Simultaneous use of several such features for long hours consumes much battery. It has been researched that the

in-app advertisements consume up to 30% of the total power consumed by the application [1]. The Android operating system (OS) itself attempts to optimize the battery life by making selective use of the resources. This is mostly implemented through the use of a rigorous power saving architecture, automatic control of display backlight, dynamic scaling of CPU frequency and dynamically controlling the networking technologies. Still energy is a bottleneck for continuous usage of smart devices. The key to power saving is therefore to understand the individual usage patterns and then propose solutions to reduce power consumption.

Current literature reports a study on user charging behavior [2], several avenues to interpret usage patterns [3] and relative power consumption of smartphone hardware. But several other features that contribute to significant energy consumption remain un-addressed: (i) frequency of user interaction with smart device, (ii) amount of network traffic is generated by the apps, (iii) the duration for which wireless networks are being used, (iv) underlying pattern of using network and location, (v) at what battery level does the user choose to recharge and (vi) the duration which corresponds to maximum energy drain. In order to investigate, a major importance is to be given to monitor several aspects of mobile devices.

The goal of the paper is to provide a methodology to derive personalized power saving profiles by analyzing smart device usage patterns. We present the methodology using client-server architecture in Figure 1. The architecture employs an Android application named “Power Monitor” (acting as client) which gathers usage information from smart devices, communicates them to a server which processes them to generate personalized power saving profiles. These profiles contain system settings which on being activated reduce power consumption of the devices and thereby extend battery life. Main innovative aspects of the architecture are twofold: (i) if the usage pattern changes over time, the power saving profiles will also evolve automatically and (ii) the profiles are personalized and tailored to suit the need of individual users.

The rest of the paper is organized as follows. Section II describes the Android application and the information collected from the smart devices. The algorithms to develop the usage patterns and power saving profiles are explained in Section III. Section IV provides real life usage patterns from two different

Android devices and their corresponding power saving profiles. The overall gain in battery lives of Android devices are reported in Section V while Section VI describes related privacy issues. Section VII compares our approach with state-of-the-art to point out the innovative aspects of the proposed architecture. Finally Section VII concludes the paper.

II. USAGE INFORMATION COLLECTION

It is well established that power consumption in smart devices is dependent on how the resources of the devices are used by user as well as mobile applications. To extract the usage information, we have developed Power Monitor. It comprises of four modules (as seen from Figure 1) which are explained below.

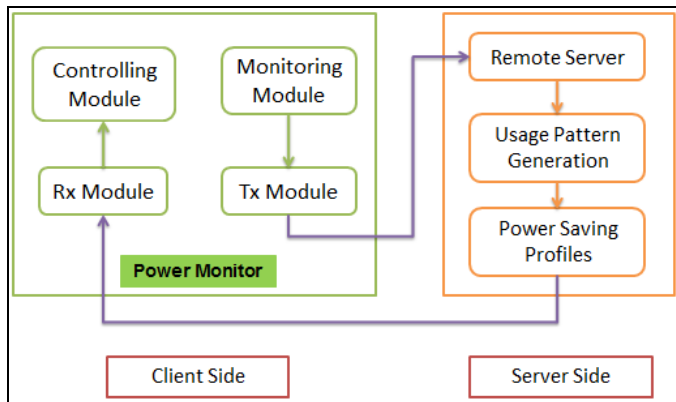


Fig. 1. Client-server architecture of the proposed system.

- Monitoring Module:** It records how the device is being used on daily basis and stores usage information locally. It is further elaborated in subsequent section. It also assigns every device a unique ID (UID). It is generated by hashing several parameters from the smart devices. It can also be obtained from Google Cloud Messaging. Initially the module collects data for a week and they are sent to the server for processing. Then this module is activated in every fortnight to collect user data to detect any change in usage pattern. If a change is detected, the usage logs are communicated again to the server for generating updated power saving profiles.
- Tx Module:** It is responsible for establishing a secure connection to a remote server. Then the collected data i.e. the usage log and the unique ID are communicated to the remote server.
- Rx Module:** The remote server process the usage logs and generates personalized power saving profiles. Receiving (Rx) module connects to the server periodically to download the profiles corresponding to the unique ID.
- Controlling Module:** The downloaded profiles are intelligently applied to the Android devices to increase

the battery life. This module initially parses the profiles which contain activation time, termination time and a list of settings to be applied. The module activates the settings of a profile based on the activation time. The system is restored to default settings at termination time of a profile.

A. Monitoring Module of Power Monitor

A complete monitoring of smart devices is essentially the first step towards the usage pattern derivation. Thus the monitoring module extensively records data related to several features of the devices. Figure 2 depicts the architecture of the module which is implemented using a background service collecting data in every five minutes for seven continuous days. The data consist of time series values of several features of the devices and are stored locally in a secure storage to protect the privacy of sensitive user information. We have used an alarm callback to suspend the service between two callbacks. This further minimizes the power consumption and resource utilization by Power Monitor. Most of the data are obtained using the APIs provided by Android and reading several system files in “/proc”. The module collects coarse location information available from network connections (Wi-Fi, Mobile data) and avoid the use of power hungry GPS. The locations are used to define the profiles along with timestamps.

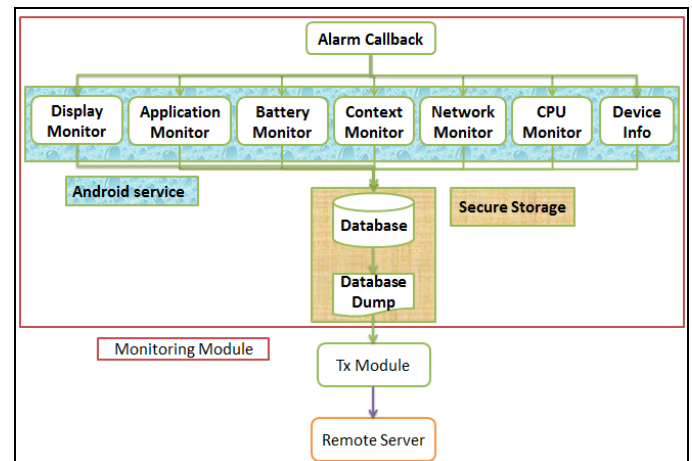


Fig. 2. Monitoring module of Power Monitor collecting information from smart devices.

Table 1 lists the monitored features in smart devices. The following subsections throw light on the reasons for such extensive monitoring of smartphones and tablets along with the information revealed by the above modules.

B. Application monitor

This identifies the running applications and engagement of users with different applications over time. The drain of battery depends a lot on the type of application. For example, games typically have higher interaction time with users and therefore have higher CPU and display usage. These contribute of much higher power consumption. Again, there are applications which heavily depend on network usage and have high demand of power.

TABLE I. LIST OF MONITORING FEATURES

Name of Module	Monitoring Features
Application monitor	<ul style="list-style-type: none"> The running applications (e.g. Gmail)
Battery monitor	<ul style="list-style-type: none"> Battery level Battery status (AC/USB charging, discharging)
Context monitor	<ul style="list-style-type: none"> System date and time Coarse location
CPU monitor	<ul style="list-style-type: none"> CPU load Operating frequency
Display and audio monitor	<ul style="list-style-type: none"> Brightness level Screen timeout value Total interaction time with the device Audio volume level
Network monitor	<ul style="list-style-type: none"> Status of Wi-Fi, mobile data network, Bluetooth and GPS Amount of network traffic generated by apps.
Device information	<ul style="list-style-type: none"> Android OS version Device manufacturer Model number Maximum & minimum CPU frequencies

C. Battery charging and discharging behavior

The data collected by "battery monitor" enables us to understand the battery charging and discharging pattern:

- The time duration during which battery is discharging, is noted along with corresponding locations.
- The remaining battery level at which the user starts to recharge is noted along with the location and method of charging (AC or USB).

The above are two key points to calculate the average battery life and calculate the time during which the remaining battery can support the device. Also the next recharging opportunity can be predicted using the location information.

D. CPU characteristics

Higher the operating frequency of CPU, higher will be the power consumption. Android operating system dynamically scales the CPU frequency based on the load of CPU. The CPU monitor module stores the CPU load and operating frequency. Our interest is to see if the operating frequency changes with CPU load or the former is always at a specific interval irrespective of CPU load. It is found that for a particular device and Android version, the operating frequency is always the highest irrespective of the CPU load and accounts for short battery life of the device.

E. Context monitor

Context monitor extracts the system date, time and the coarse location using cellular network and/or Wi-Fi. The information has been used to classify the patterns as described in Section III.

F. Display and audio module

It is common practice to set the brightness value of the display at a higher value and the colorful bright display consumes significant power. Thus a device with higher amount of interaction time (as calculated by display monitor) and high

brightness will have shorter battery life. The volume level of audio also affects the overall power consumption.

G. Network usage

Continuous network usage limits the battery life. Majority of the applications require network connection for their operation. Thus using the network monitor module, we extract the following data:

- The duration through which Wi-Fi and mobile data are used and the corresponding location information are noted.
- Usage of GPS is recorded.
- Amount of traffic generated per day is also recorded.

Together the information point to the network usage pattern of users. The durations when devices are connected and disconnected to network are useful for power saving profiles. Use of GPS drains much battery and the location in which it is used, is also important information.

III. USAGE PATTERNS AND POWER SAVING PROFILES

Once the monitoring for a week is over, the collected usage logs are uploaded to a remote server using HTTPS connection opened by the Tx Module. The unique identification corresponding to each device is also uploaded at the same time. The server initially stores the received logs in appropriate database and then following processing takes places to generate the usage pattern.

Usage patterns are characterized based on (i) a day of the week (d), (ii) time interval of a day (t) and (iii) location (s). They are retrieved using the context monitor module. For each (d, t, s), a pattern is generated and the algorithm is described below.

A. Algorithm to generate usage patterns

For each day of the week of individual users, repeat the following.

- Count the number of distinct locations that the user is subscribed to from the collected data.
- Determine the time interval(s) associated with each distinct location. It is possible for a location (e.g. home) to be associated with multiple time intervals at different parts of the day.
- For each pair of time interval and location, repeat the steps 4-9 which produces the respective usage pattern.
- Determine the running applications and their CPU load.
- Determine the change in battery level and status.
- Determine the volume level for audio functions.
- Calculate the amount of network usage, status of Bluetooth, mobile data, Wi-Fi and GPS (i.e. on/off) and the duration of GPS usage.
- Read the CPU loads and corresponding operating frequencies.
- Record the brightness level, screen timeout value and the interaction time with the device.

The patterns having similar time duration and location over several days are clubbed together. The above algorithm is implemented in the remote server. Power consumption estimation is done for each pattern to inform the user. Further information can be obtained by comparing the profiles such as - the profile(s) corresponding to (i) very high device interaction time, (ii) higher network usage, (iii) battery charging and (iv) intense CPU operations. The knowledge thus procured is intelligently converted into power saving profiles.

B. Personalised Power Saving Profile Generation

The power saving profiles are essentially a set of settings that regulate the network technologies and amount of network data, brightness, limit the network traffic per app and scale CPU frequency dynamically. The profiles are activated intelligently to reduce power consumption in the smart devices. One profile is generated for each pattern of the devices. The settings recommended by the profiles can be activated either by the users or by Power Monitor itself. Given a profile, following algorithm creates the corresponding power saving profile. The algorithm is also running in the remote server. A lightweight version of the engine can be ported to the smart devices which can generate the usage patterns and such profiles within the devices [17].

1. For computationally intensive apps (characterized by high CPU load and memory consumption), provide the users with choices to kill the unnecessary applications.
2. If the pattern registers high device interaction time (more than one hour), the brightness and screen time out values are reduced to save power at display hardware.
3. Set a limit for the amount of daily network usage. This can also be extended limit the background data consumption for network intense applications. It reduces power consumption significantly at the network interfaces.
4. If the pattern registers very less interaction with the device, the network can be switched off when the screen is off and can be switched on when the screen is on.
5. During the night when there is no interaction with the devices, turn off the wireless network which save considerable amount of battery.
6. If the Bluetooth is not being used, turn it off.
7. Provide options to turn off some of the feedbacks – audible touch tones and haptic feedback.
8. Reduce the volume level for incoming call, media play (e.g. sound level when YouTube is played), system and notifications.
9. When the battery level is below a certain threshold (e.g. 50%) and is discharging, the bulk data transfer should be done over high speed network to reduce power consumption.
10. If battery is critically low (i.e. below 10%)

- a. Tone down brightness and timeout to min.
- b. Turn off network usage.
- c. In jail broken devices, scale down frequency to minimum and kill CPU intensive apps.
- d. Turn off any vibration feedback for notification, haptic feedback and audible touch tones and selections.

11. Auto-sync is periodically switched on during the day. The time period can be configured by user or the app.
12. If GPS is on for long, ask the user to switch it off because GPS cannot be automatically turned off.

Each profile is activated at the starting time of a pattern and remains active throughout the entire duration of the pattern. Then another appropriate profile is activated. This cycle continues for every smart device.

C. Novel Aspects

Such computation of usage pattern and power saving profile has several novel aspects as mentioned below.

- **Dynamic computation:** Each user has different power expenditure pattern and studying usage behavior reveals that pattern. The power saving profiles are dynamically computed based on these individual patterns. This is an important innovation provided by the application which differentiates itself from other power saving applications that offer static power saving settings [4].
- **Detection of change in user behavior:** Power Monitor initially monitors every device for seven days and the usage logs are uploaded to the server. The server pushes back the usage patterns and power saving profiles. The monitoring phase continues after that and after every fortnight the monitoring is activated in the background. For each pattern, the monitoring module analyses new usage logs to determine if there is a change in behavior. If such changes are detected, the new behavior is collected and reported to the server along with the UID of the particular device.
- **Evolution of profiles:** On receiving new behavior from any smartphone or tablet, the server processes them using the same algorithm to generate new profiles for the pattern. Thus the power saving profiles automatically evolve and are communicated back to the device corresponding to the UID. This is another novel aspect proposed through the application.
- **Usage pattern based feedback:** An intelligent feedback on how battery lifetime is being utilized by the mobile applications is given to the users. The feedback is also personalized as it is derived from the individual usage patterns. This enables the users to understand the power expenditure in their smart devices.

IV. REAL LIFE USAGE PATTERNS AND POWER SAVING PROFILES

The prototype mobile application is deployed to thirty Android devices. In this section, we present two real life usage patterns from two different Android devices and describe in detail the corresponding power saving profiles. Also another test case is explained where a change in usage pattern is noticed and it is shown that our algorithm generated the evolved power saving profile.

A. Usage Pattern and Power Saving Profile – 1

The device under test is Samsung Galaxy S2 running Android version 2.3.4. The maximum and minimum CPU frequencies of the device are 1200 MHz and 200 MHz respectively. The profile described below corresponds to Monday during 18:04 – 21:09 hours at location home. The usage pattern and power saving profile characteristics are compared in Figure 3.

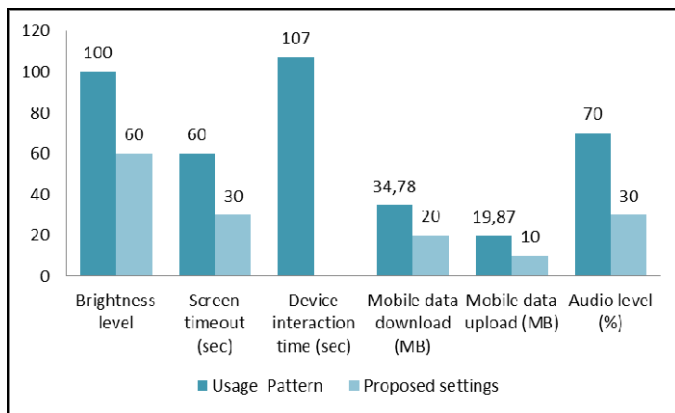


Fig. 3. Characteristics of the usage pattern and proposed profile for Samsung Galaxy S2.

The usage pattern is explained below.

1. The user is interacting with several Android applications: Facebook, Gtalk, SmartVoip, Opera, Whatsapp, Gmail, Temple Run and Techcrunch.
2. The battery is discharging during the time interval and drops to 21 from 48.
3. The device brightness is 100, screen time out is set to 60 seconds and total interaction time with the smartphone is calculated to be 107 minutes. The audio functionalities for media (i.e. playing song/video) are not used.
4. As for wireless network connections, Wi-Fi and Bluetooth are not used. Mobile data connection is used. The amount of traffic downloaded is 34.78 MB and uploaded is 19.87 MB. GPS is off during the course of the pattern.

The underlying pattern as understood from the above is that the user is engaged with the smartphone for really long time and the brightness level is too on the higher side. The total amount of network traffic generated during the time period is 54.65MB. Thus considerable power is spent at the network

interface and display hardware. To limit the power consumption, following settings are proposed as the power saving profile. The settings are activated at 18:00 hours and restored to default settings at 21:15 hours on Monday.

1. Reduce brightness level to 50.
2. Set screen timeout value to 30 seconds.
3. Set limits to network data download (20MB) and upload (10MB). When 80% of the limit is crossed, the user is notified if he/she wants to continue to use mobile data connection.
4. The user is given option to operate on Wi-Fi which provides faster connection than mobile data network.
5. Options to kill mobile applications are given. In this case, the user has to terminate the applications.

As seen from the Figure 3, the profile settings reduces the power dissipation at display hardware and network interfaces as they are the main power consuming features for the time duration.

B. Usage Pattern and Power Saving Profile – 2

This use case is extracted from a Nexus 7 running Android 4.2.1 with maximum and minimum CPU frequencies being 1300 MHz and 51 MHz respectively. Unlike the previous pattern, we witness both discharging and AC charging in this case which makes the power saving profile different. The usage pattern corresponds to Sunday from 14:56 to 18:21 and location is home. Figure 4 provides the comparison of the usage pattern and corresponding profiles characteristics. Following is the usage pattern.

1. The running applications are Facebook, Chrome, Facebook Messenger, Bowling 3D, Scrabble, and YouTube.
2. The battery is discharging from 50 to 17 and then it is charged with AC source to 29.
3. The brightness value is 179, screen timeout is 120 seconds and device interaction time is 126 minutes.
4. The audio for media is used and the volume level is 85%.
5. The Wi-Fi is used throughout the time duration while mobile data connection and Bluetooth remain switched off. The amount of downloaded data is 125MB and uploaded data is 17.7MB. It is observed that YouTube accounted for 87MB of network traffic.
6. GPS is on during the entire period.
7. CPU load is portrayed in Figure 4.

In this case the generated power saving profile is given below. The activation and terminations times are 14:50 and 18:20 hours respectively.

1. Present options to kill applications to the user.
2. Prompt the user to switch GPS off.
3. When the battery is discharging –
 - a. Set brightness to 50.
 - b. Set screen timeout 30 seconds.

- c. Limit the network usage to 50MB. The limit is configurable by the user.
 - d. Set the volume for media to 30%.
4. When the battery is charging –
 - a. Set brightness to 75.
 - b. Set screen timeout 30 seconds.
 - c. Limit the network usage to 75MB. The limit is configurable by the user.
 - d. Set the volume for media to 50%.

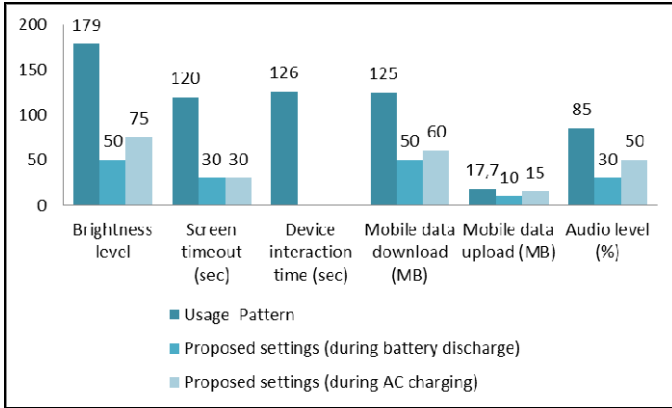


Fig. 4. Characteristics of usage pattern and proposed profiles for Nexus 7.

In this case, the profile offers strict power saving settings when the battery is discharging and less strict settings when the battery is being charged with AC power source. Thus the application provides flexibility in terms of user experience and this is another user innovation in Power Monitor.

C. Change in Usage Pattern

We have detected a change in usage behavior for the same smart device mentioned in Section IV-B. This change is observed one month after installing Power Monitor. The evolved usage pattern for the same day, time duration and location is given below.

1. The applications running are – Facebook, SmartVoip and Gmail.
2. Battery is discharging from 45 to 28. There is no charging happening during the time.
3. Brightness is set to 100 with screen timeout at 30 seconds. The interaction time with the device has reduced to 37 minutes.
4. Wi-Fi is used during the time which accounts for total 18.2 MB network traffic. Bluetooth, mobile data and GPS are not used.
5. The media volume level is 50%.

After Power Monitor detected the evolution in usage behavior, the corresponding log file was communicated to the remote server which generated the profile with modified settings.

1. Set brightness to 50.

2. Switch off Wi-Fi when screen is off and switch it on when screen is on.

Since there is very limited interaction with the device (only 37 minutes during 215 minutes), the Wi-Fi is turned off when the user is not interacting with the device. Also reduced brightness further helps to save power. Figure 5 compares the characteristics for both the patterns and their corresponding profiles.

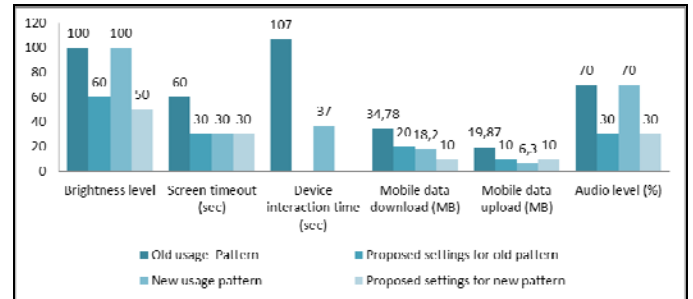


Fig. 5. Characteristics of old and new usage patterns and proposed profiles for Samsung Galaxy S2.

D. Power Consumption Reduction

We computed the amount of power saved by reducing brightness and limiting network traffic and these results are portrayed in following figures.

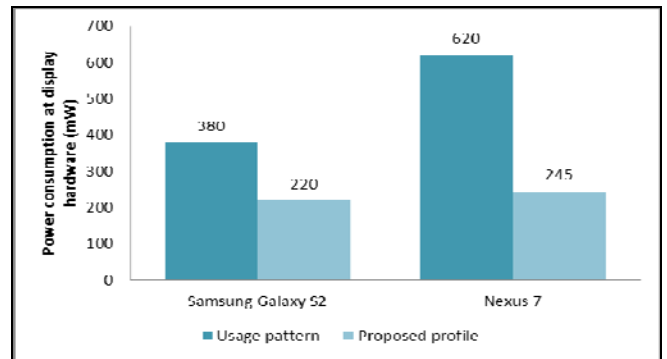


Fig. 6. Reduced power consumption after applying the power saving profiles.

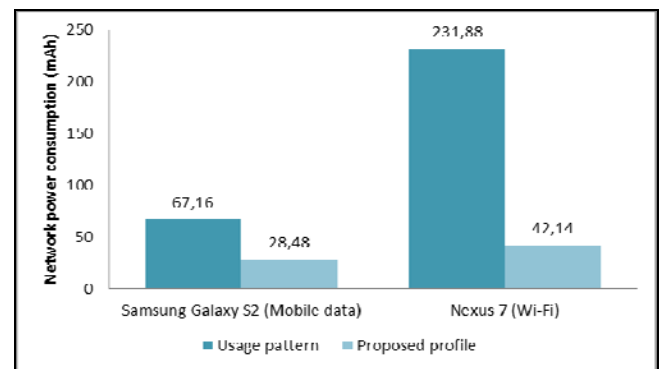


Fig. 7. Reduced power consumption at network interfaces after applying the power saving profiles.

The power consumption data for brightness are retrieved by Power Tutor [11] and that for mobile data is computed from the values presented in [16]. From the figures above, it is clear that the proposed profiles can reduce the overall power consumption in smart devices.

V. EVALUATION

To evaluate the overall battery life gain, Power Monitor is deployed to several Android smartphones and tablets. During the monitoring phase, the application determines the initial battery life. Again battery life is measured after the power saving profiles are applied. The difference between the obtained values gives the gain in battery life. We have also separately measured the gain with two popular power saving applications Juice Defender and Easy Battery Saver that are available in Google play app store [14-15]. The battery gain results are presented in following figure.

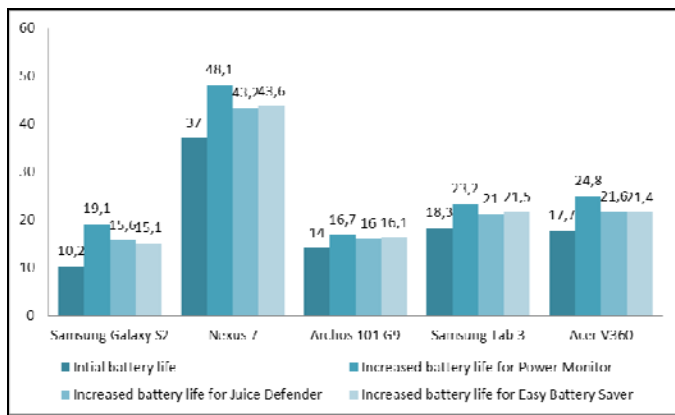


Fig. 8. Comparison of increased battery life for smart devices using Power Monitor, Juice Defender and Easy Battery Saver.

The above comparison establishes that the proposed architecture is feasible and provides better gain in battery life than well-known power saving applications. The main reasons behind such improvement are attributed to personalized profile computation based on user behavior and automatic evolution of the profile with change in behavior.

VI. PRIVACY ISSUES

The process of monitoring Android powered devices and recording usage related data raises privacy issues. The actual behavior of users, applications and coarse location information are stored by the Power Monitor. In a rooted device, the stored information could be accessed by another Android application and send the log to another server. It could easily spam the device with location based advertisements. The location privacy of users is also compromised [4].

To eliminate the privacy concerns, following steps are to be taken.

- The collected data must be encrypted and stored.
- The app randomly decides when to send the usage log to the remote server.
- The user should be given some control over how much usage information is to be recorded and sent to

the remote server. For instance, user may not like to provide the location information.

- Such usage logs should never be revealed to any third party.

VII. STATE-OF-THE-ART

Significant amount of researches have been carried out to understand power consumption of mobile devices and propose methods to reduce it [5]. In this section, we provide three important research directions dealing with the power consumption based on data collected from smart devices.

A. Power model generation using use pattern analysis

Author of [5] have proposed a power model generation technique analyzing the smartphone usage patterns. Such patterns are logged and divided into several “chunks” and are evaluated for power model generation. The authors have compared their model with that of Power Tutor [10-11] and reported better accuracy in estimating power consumption. But the paper reports only modeling of CPU and display hardware. An Android application Power Doctor was developed that collect the following data at every 1% discharge of battery: average power consumption, CPU utilization and frequency, display activity and brightness and average battery voltage.

However the research does not take into account other parameters like network traffic, battery discharging/charging pattern and applications running in the mobile devices. Neither has it considered their effect on power consumption. In our approach, we consider the most important factors having major contribution to power dissipation. The power saving profiles dynamically control these factors to prolong battery life.

B. Usage pattern analysis

The paper [3] analyzes usage pattern based on variety of data: voice call status, screen status, network information, battery level and status. The authors also propose a method to predict the battery lifetime based on usage pattern in [13]. Another plethora of works have concentrated on gathering usage logs and present the diversified usage pattern [6-9].

But none of these methods are studying the complete user behavior in terms of applications used, network status & amount of traffic, display hardware characteristics, battery charging and discharging profile and CPU characteristics. As explained in this research, all these factors contribute to the actual usage patterns and provide us information about the battery lifetime. Another factor is that, we are extending the study of usage pattern to propose personalized battery saving solutions. This is the main innovative aspect of our approach.

C. Battery charging behavior

The research [2] portrays battery charging behavior of 4035 users (over a period of a month) who agreed to share the data. OverCharged [12] – an Android application was developed to record the following data: charging activity, battery level, device type, temperature & voltage of battery and the uptime. Based on the information, the paper presents several statistical

results like charging duration, charging schedule, how charging happens (USB or AC) and more.

We have taken a similar approach and generating statistics of battery level and status form a part of our usage pattern derivation. We have also extended the study on usage patterns to collect more usage information. Also, main innovation lies in proposing solutions automatically that reduces power consumption in mobile devices. The change in usage pattern is also detected by Power Monitor and the server can modify the existing power saving profiles based on the new usage behavior.

VIII. CONCLUSION

In a nutshell, the paper presents client-server architecture to collect usage logs from Android powered devices. The usage log contains data related to user behavior with the smart devices. All these data are further processed to generate usage pattern of users at a remote server. Based on the patterns, power saving profiles are generated which are personalized to suit the need of individual devices. The profiles contain different settings of the devices and intelligently control the resources of the device to minimize power dissipation. Two real life usage patterns are then presented along with corresponding profiles. Another example is given to show that Power Monitor is capable of detecting any change in user behavior automatically and the profile corresponding to that pattern is automatically modified and applied to the device. The gain in battery life is computed for several devices and compared to that with popular power saving applications. It is seen that use of Power Monitor is beneficial to users as it is able to prolong the battery life more than other applications.

ACKNOWLEDGMENT

The authors would like to thank the persons who participated in the project and shared their smart device usage information. The work is sponsored by French research project "Smart 4G Tablet" Pole SCS.

REFERENCES

- [1] A. Pathak, Y. C. Hu and M. Zhang. "Where is the energy spent inside my app? Fine grained energy accounting on smartphones with eprof." In Proc. of ACM EruoSys'12, Bern, Switzerland, 2012.
- [2] K. Lyons, J. Hightower, and E.M. Huang (Eds.). "Understanding human-smartphone concerns: a study of battery life." Pervasive 2011, LNCS 6696, pp. 19–33, Springer-Verlag Berlin Heidelberg 2011.
- [3] J.M. Kang, S. Seo and J. Hong. "Usage pattern analysis of smartphones." In 13th Asia-Pacific Network Operations and Management Symposium, 2011, pp. 1-8.
- [4] Datta, S.K.; Bonnet, C.; Nikaiein, N., "Android power management: Current and future trends," First IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT), 2012, pp.48-53.
- [5] J. Lee, H. Joe, H. Kim. "Smart phone power model generation using use pattern analysis." In IEEE International Conference on Consumer Electronics, 2012, pp. 412 – 413.
- [6] H. Falaki, D. Lymberopoulos, R. Mahajan, R. Govindan, S. Kandula, and D. Estrin. Diversity in Smartphone Usage. In *Proc. ACM MOBISYS*, 2010.
- [7] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, S. Venkataraman. "Identifying diverse usage behaviors of smartphone apps." In Proc. of ACM SIGCOMM conference on Internet Measurement Conference, 2011, pp. 329 – 344.
- [8] N. Vallina-Rodriguez, P. Hui, J. Crowcroft, A. Rice. "Exhausting battery statistics." In Proc. of ACM MobiHeld, 2010, pp. 9 – 14.
- [9] A. Oulasvirta, T. Rattenbury, L. Ma, E. Raita. "Habits make smartphone use more pervasive." In *Pers Ubiquit Comput* (2012) 16:105–114.
- [10] L. Zhang, et al. "Accurate online power estimation and automatic battery behavior based power model generation for smartphones." In Proc. Of ACM CODES+ISSS'10, Arizona, USA, 2010, pp. 105-114.
- [11] <https://play.google.com/store/apps/details?id=edu.umich.PowerTutor>
- [12] <https://play.google.com/store/apps/details?id=com.dymsoft.com.overcharged>
- [13] J.-M. Kang, C.-K. Park, S.-S. Seo, M.-J. Choi, and J. W.-K. Hong, "User-centric prediction for battery lifetime of mobile devices," Challenges for Next Generation Network Operations and Service Management, pp. 531–534, 2008.
- [14] <https://play.google.com/store/apps/details?id=com.easy.battery.saver>
- [15] <https://play.google.com/store/apps/details?id=com.latedroid.juicedefender>
- [16] Datta, S.K.; Bonnet, C.; Nikaiein, N., "Minimizing energy expenditure in smart devices," IEEE Conference on Information & Communication Technologies (ICT), 2013, pp.712-717.
- [17] Datta, S.K.; Bonnet, C.; Nikaiein, N., "Power monitor v2: Novel power saving Android application," 17th IEEE International Symposium on Consumer Electronics (ISCE), 2013 pp.253-254.