Power Monitor v2: Novel Power Saving Android Application

Soumya Kanti Datta, Christian Bonnet, Navid Nikaein, *Member, IEEE*
Mobile Communication Department, EURECOM, France
Emails: {dattas, bonnet, nikaeinn}@eurecom.fr

***Abstract*-- This paper presents a novel scheme to derive power saving profiles based on the usage patterns of the Android devices. The entire architecture is developed as an Android app "Power Monitor v2" and is deployed to the smart devices. A monitoring module of the app periodically collects several data from the devices and stores them locally. A learning engine then operates on the raw data to generate multiple usage patterns over time and space, which characterizes the user contexts. The engine then processes the patterns to generate power saving profiles dynamically within the devices. The profiles contain several system settings of the smart devices and intelligently optimize power consumption. We also present a real life usage pattern and the power saving profile. The overall battery life for the device estimated to increase by 82%.**

I. INTRODUCTION

The Android smartphones and tablets are being adopted at a phenomenal pace. Due to the availability of desktop-like features, the activities of the users are becoming more and more smart device centric. Prolong use of those features limit the battery life to a few hours. Thus it is necessary to study the usage pattern to propose better power saving solutions.

Current literature provides several avenues to collect user specific data, transmit them to a remote server which generates the usage patterns [4], [6], [7]. Such patterns reveal several useful information including the frequently used applications, total network usage, battery charging and discharging behavior and device interaction time. But several factors contributing to usage pattern remains un-addressed. These include correlation of user behavior with location and CPU characteristics. Also, uploading personal usage data to a remote server raises concern on privacy.

Power Monitor v2 employs a monitoring module to collect data covering the entire smart device features. The data are collected periodically and are portrayed in Table I.

TABLE I. LOST OF MONITORING FEATURES

Name of module	Collected data
Application monitor	The applications running and their CPU load
Battery monitor	Battery level and status (discharging / AC charging / USB charging
Context monitor	System date, time and coarse location
CPU monitor	CPU load and operating frequency
Display monitor	Brightness level, screen timeout and device interaction time
Network monitor	Status of Wi-Fi, mobile data, GPS and amount of network traffic used by the apps

The app monitors the devices continuously for seven days, stores the data locally and employs a learning engine which generates multiple usage patterns within the device. Each usage pattern is associated with a day of the week (d), time interval of a day (t) and location (s). Then power saving profiles are generated for each pattern dynamically. This method preserves data privacy as all the computations are done locally. This work is a continuation of [5].

The research work is sponsored by French project Smart 4G Tablet Pole SCS.

II. MULTIPLE USAGE PATTERNS GENERATION

Most of the users remain subscribed to a small set of locations in their daily lives. Thus we have characterized the usage patterns based on (d, t, s). For each (d, t, s), a pattern is generated and the algorithm is described below.

For each of the seven days of monitoring, repeat the following.

I. Count the number of distinct locations from the data.
II. Read and store the time interval(s) associated with each distinct location. It is possible to have multiple intervals corresponding to a location (e.g. home).
III. For each pair of time interval and location, repeat the steps below which produces the respective usage pattern.
IV. Determine the running applications and their CPU load.
V. Determine the change in battery level and status.
VI. Calculate the amount of network usage, status of network (i.e. on/off) and the duration of GPS usage.
VII. Read the CPU loads and corresponding operating frequencies.
VIII. Compute the average brightness level 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 as the learning engine of Power Monitor v2. According to the power consumption values in [1], energy 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.

III. POWER SAVING PROFILES GENERATION

The power saving profiles contain several settings of the smart devices and are activated intelligently to optimize power consumption. One profile is generated for each pattern. The settings recommended by the profiles can be activated either by the users or the app itself. Given a profile, following algorithm creates the corresponding power saving profile.

I. If computationally intensive apps (characterized by high CPU and memory) are running, then the users are presented with choices to kill those applications.
II. If the pattern has high device interaction time, the brightness and screen time out values are reduced.

III. Set a limit for daily network usage which reduces power consumption at network interfaces.

IV. If the battery level is below 50% and is discharging, the bulk data transfer is done over high speed network to reduce power consumption.

V. If battery is critically low (i.e. below 15%)
   a. Tone down brightness and timeout to min.
   b. Turn off network usage.
   c. In jail broken devices, scale down freq to min and kill CPU intensive apps.

VI. Auto-sync is periodically switched on during the day. The time period can be configured by user or the app.

VII. If interaction time is really less, turn off network when screen is off and for jail broken devices scale down CPU frequency.

VIII. If GPS is on for long, ask the user to switch it 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.

IV. EVALUATION

In order to evaluate the battery gain, Power Monitor v2 is deployed to several Android smartphones and tablets. The app calculates the initial battery life during the monitoring phase and after the power saving profiles are activated. The battery life gain is the difference between the obtained values. We have compared the gain of Power Monitor v2 with that of two popular apps JuiceDefender and Easy Battery Saver [2], [3]. The results are presented in Table II.

TABLE II. COMPARISON OF AMOUNT OF BATTERY LIFE GAIN

Device name and Android version	Initial battery life (hours)	Increase in battery life (hours)		
		Power Monitor v2	Juice Defender	Easy Battery Saver
Samsung GT-I9100 with Android 2.3.4	10	8.2	6	5.6
Nexus 7 with Android 4.2.1	48	10	8	7.8
Archos 101 G9 with Android 4.0.4	14	2.5	2	2.1

The above results show that our approach is feasible and improves the battery life of the devices better than the other apps. Power Monitor v2 offers several other advantages compared to other power saving apps. Our app generates personalized power saving solutions tailored to fit the demand of each individual user while other apps have same static profiles for all users. Any change in usage patterns modifies the power saving profiles automatically since they are produced analyzing the patterns.

V. TEST CASE

A real life usage pattern for Samsung GT-I9100 running Android 2.3.4 is provided. The number of distinct locations on the weekdays is four. Out of these locations, we present the following pattern corresponding to the interval 6:15PM - 9:00PM at location home. The device is not jail broken.

I. Facebook, Gtalk, SmartVoip and Bowling 3D are the applications running.

II. Battery level reduces from 75 to 49.

III. Brightness level is 65, screen timeout 60 seconds and interaction time is 87 - 110min.

IV. Average CPU load and operating frequency are 54 and 800.

V. Total network usage is about 20 to 22MB and the device is connected using mobile data network. GPS is off.

The profile has registered maximum network usage and interaction time. The power saving profile has the following settings.

I. Reduce brightness to 35 and screen time out to 30sec.

II. Provide options to kill background apps.

III. Turn off mobile data if the screen goes off.

IV. Limit the network usage to 20MB for the duration.

V. Auto-sync is switched on every 3 hours.

There are five more patterns and profiles generated for the device. Overall our app is able to increase the battery life by 82%.

VI. CONCLUSION

The paper introduces a novel approach for dynamic generation of usage patterns and power saving profiles. The algorithm of the learning engine described. It is reported that the battery life gain in case of Power Monitor v2 is better than that of Juice Defender and Easy Battery Saver. Some other key advantages of our app are identified. A real life scenario is also presented. As future research, we are extending the work to the users to travel a lot and the number of distinct locations is higher. This poses some unique challenges for the learning engine. Also we are deploying the app to several other devices to evaluate the result with respect to other popular power saving apps.

REFERENCES

[1] http://www.google.com/events/io/2009/sessions/CodingLifeBatteryLife.html

[2] https://play.google.com/store/apps/details?id=com.latedroid.juicedefender

[3] https://play.google.com/store/apps/details?id=com.easy.battery.saver

[4] 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.

[5] Datta, S.K.; Bonnet, C.; Nikaein, N.; , "Android power management: Current and future trends," Enabling Technologies for Smartphone and Internet of Things (ETSIoT), 2012 First IEEE Workshop on , vol., no., pp.48-53, 18-18 June 2012.

[6] H. Falaki, D. Lymberopoulos, R. Mahajan, R. Govindan, S. Kandula, and D. Estrin. Diversity in Smartphone Usage.In *Proc. ACM MOBISYS*, 2010.

[7] J. Lee, H. Joe, H. Kim. “Smart phone power model generation using use pattern analysis.” In IEEE Intenational Conference on Consumer Electronics, 2012, pp. 412 – 413.