Energy Stealing - An Exploration into Unperceived Activities on Mobile Systems

Chi-Hsuan Lin[1], Yu-Ming Chang[1,2], Pi-Cheng Hsiu[3,4], Yuan-Hao Chang[4]

[1] Department of Computer Science and Information Engineering, National Taiwan University, Taiwan

[2] Macronix International Co., Ltd., Emerging System Lab., Taiwan

[3] Research Center for Information Technology Innovation, Academia Sinica, Taiwan

[4] Institute of Information Science, Academia Sinica, Taiwan

r01922015@csie.ntu.edu.tw, sanchang@mxic.com.tw, pchsiu@citi.sinica.edu.tw, johnson@iis.sinica.edu.tw

***Abstract*—Understanding the implications in smartphone usage and the power breakdown among hardware components has led to various energy-efficient designs for mobile systems. While energy consumption has been extensively explored, one critical dimension is often overlooked - unperceived activities that could steal a significant amount of energy behind users' back potentially. In this paper, we conduct the first exploration of unperceived activities in mobile systems. Specifically, we design a series of experiments to reveal, characterize, and analyze unperceived activities invoked by popular resident applications when an Android smartphone is left unused. We draw possible solutions inspired by the exploration and demonstrate that even an immediate remedy can mitigate energy dissipation to some extent.**

***Index Terms*—Energy dissipation, unperceived activities, mobile systems**

I. Introduction

The personal computing landscape is undergoing a massive transition from stationary desktops to mobile devices. Naturally, users would expect those *resident* applications, such as instant messaging (e.g., Line) and social networking (e.g., Facebook), to behave consistently as on desktops. To preserve the applications' semantics, mobile devices, while staying in the standby mode, remain connected to the Internet through their Wi-Fi or 3G network interfaces. This connected standby state, where the screen is off while the network remains on, can deplete a smartphone's battery quickly even if it is left unused [14]. Intuitively, there must be some *unperceived activities* that steal energy bit by bit behind users' back silently, without improving their user experience. Understanding such unperceived activities and their impact on energy dissipation will lead to better power management and application designs for mobile systems.

Energy consumption has been a popular area of research in embedded and mobile systems for many years. Most of prior studies tried to understand *how much energy is consumed by which components* from the perspectives of user behavior [3, 10, 11], application functionalities [7, 12, 15], and/or hardware resources [1, 2, 16]. In particular, many studies aimed to characterize the power consumption of different hardware components through direct measurements to build their power models [2, 10, 16]. As different applications may require diverse combinations of hardware resources, some studies developed profiling tools to analyze the resource usage of one single application [12] or some individual functionalities [7, 15]. Because user behavior drives application execution, which determines the energy consumption, some studies were conducted based on mobile users' daily activity. The analytic results indicated that the screen and the CPU consume most of the energy when users are interacting with their devices and, therefore, are good targets for power optimization [10, 11]. Understanding which components kill the battery has facilitated the empirical design of various energy-efficient schemes, such as those for CPU scheduling [13] and OLED display [6].

Existing energy-efficient schemes, e.g., [6, 13], were intended mainly for mobile systems in the *interactive* state. On average, however, a smartphone is in the standby mode for 89% of the time and accounts for 46.3% of the total energy consumption for a typical daily usage [11]. Recently, *how energy is stolen behind users' back* has attracted research attention. To prolong the standby time of smartphones, a mobile system normally turns off hardware components aggressively unless the component is explicitly requested by some app. This new programming paradigm gives rise to *no-sleep bugs* [8], which will keep smartphones awake. Motivated by the observation, diagnostic tools were developed to automatically detect no-sleep bugs in immature apps at compile time [8] and runtime [4], respectively. Apart from no-sleep bugs, email sync, which will cause normal background activities, can also shorten the standby time of a smartphone by 40% due to the long 3G *tail time* [14]. To reduce the energy spent on the 3G tail, a fast dormancy scheme was proposed to force the 3G interface to sleep adaptively according to email arrival patterns. Moreover, based on large packet traces collected from a commercial cellular carrier, it has been observed that *periodic transfers*, where a mobile device periodically exchanges some data with a remote server, account for only a small portion of the overall network traffic but contribute significantly to the total radio energy consumption [9]. While some attention has focused on finding solutions, e.g., [4, 8, 14], the major sources of unperceived activities remain to be identified and their individual impact on energy dissipation has yet to be studied.

Much of previous research focused on how energy is *used* with respect to user behavior, application functionalities, and/or hardware resources. However, there has been comparatively little research on how energy is *wasted*. In contrast to individual studies on a specific cause, such as no-sleep bugs or periodic transfers, this paper takes the first comprehensive steps towards understanding unperceived activities that drain the energy of smartphones in connected standby. To this end, we design and conduct a series of experiments based on a popular Android smartphone with three most frequently-used mobile apps under two different networks, i.e., Wi-Fi and 3G. First, to unveil unperceived activities' patterns, we employ multiple profiling tools to collect real traces in Android and use a power monitor to measure the smartphone's transient power. Then, we dig into the collected activity and power traces to characterize unperceived activities and identify their major sources. Next, we analyze the activities to obtain their occurrence breakdown and energy breakdown in respect of our classification, as well as draw some implications. Finally, based on the implications, we discuss potential solutions to this energy-stealing problem. We have also implemented an immediate remedy that can reduce the stolen energy by up to 35% and 13% under Wi-Fi and 3G networks, respectively. The results in this exploration could inspire system developers to elaborate sophisticated designs in power management on mobile systems.

The remainder of this paper is organized as follows. Section II provides some background information and a motivational study. In Section III, we present a detailed exploration of unperceived activities. We discuss potential solutions and an exemplary implementation in Section IV. Section V contains our concluding remarks.

II. Background and Motivation

A. Power Management on Mobile Systems

Energy is certainly the most critical resource of mobile devices. To preserve this limited resource, mobile systems have resorted to a paradigm shift in power management. The philosophy is to have every hardware component, such as the CPU and screen, turned off or kept in a low-power dormant mode by default, unless some app explicitly requests the mobile system to keep the component on. To realize this *aggressive sleeping philosophy*, mobile systems normally provide specific mechanisms for apps to express their intention to continue using a certain component. In the Android system, which we target in this exploration, a mechanism called *wakelock* is provided for power management [8]. The mechanism enables the android system to transit between three states, namely, *awake*, *notification*, and *sleep*. For example, if a *full* wake lock is acquired by any apps, the system will remain awake and keep a set of components, including the CPU, on. After the system has been idle for a period of time, it will transit to the notification state and turn off some components (e.g., backlight) registered as *early-suspend* components. Finally, the system will turn off the CPU and go into sleep immediately once all (full or *partial*) wake locks are released. The power required to keep the system in the awake or notification state is much larger than in the sleep state. Based on our measurements, the power required by a Samsung Galaxy S3 to stay awake for idling is about 800–1000 mW, whereas it consumes only 17–30 mW during sleeping.

Most mobile apps rely on wireless connectivity, often via a Wi-Fi or 3G network adapter. To save power by turning off the Wi-Fi interface whenever possible, nearly all commercial smartphones implement the *power save mode* (PSM) as the default management scheme. Under PSM, a smartphone will turn off its Wi-Fi interface after a short idle period on completion of the transmission, and turn on the interface, which will remain in the high power state, again when it has packets to send or receive. However, the Wi-Fi interface must still be activated frequently to perform *idle listening* [17] for packet detection or carrier sensing. Similarly, to achieve power efficiency, a 3G interface transits between four states with different power consumption by adhering to the *radio resource control* (RRC) protocol. However, a 3G interface does not switch from the high to the low power state immediately after each transmission; instead, it waits for a long period to alleviate the switching overhead at a cost of extra *tail energy* [1], in case new transmission will occur before long. Typically, compared to a Wi-Fi interface that consumes more energy for idle listening, a 3G interface consumes more energy per bit for data transmission and requires an extra amount of tail energy after each transmission.

B. A Motivational Study

Even though mobile systems have employed various power management schemes in accordance with the aggressive sleeping philosophy. Users may all have similar experience that the energy of batteries drain quickly even if smartphones are laid aside. To have some sense of how serious the energy-stealing problem is, we conducted a preliminary experiment under two scenarios. The first scenario was conducted with a Samsung Galaxy S3 recovered to its factory settings. In the second scenario, three popular apps, namely LINE (instant messaging), Facebook (social networking), and Chrome (web browsing), were installed on the same smartphone. For both scenarios, we left the smartphone unused for 30 minutes with the screen turned off and the Wi-Fi adapter turned on. During the 30 minutes, no perceivable activities occurred to attract our attention. Note that Wi-Fi does not dissipate a non-negligible amount of tail energy after each transmission; moreover, the apps were developed by professional programmers and thus not supposed to contain no-sleep bugs due to the misuse of wakelock. The energy consumption under each scenario was measured by a power monitor of Monsoon Solutions. As shown in Figure 1, the energy consumption under the two scenarios is 70 joules and 138 joules, respectively. In other words, with these three common apps installed, the standby time of the Samsung Galaxy S3 was shortened by one half. The more apps installed, the more energy that could be stolen behind users' back potentially.

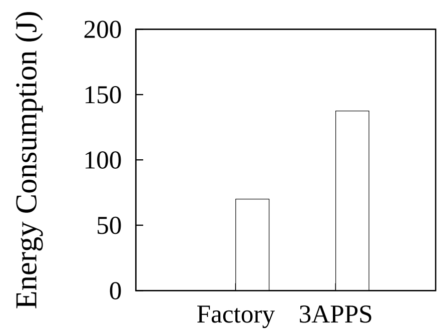


Fig. 1. Energy consumption of a smartphone under two idle scenarios

Intuitively, there must be some *unperceived activities* that steal energy in a silent manner. However, there has been comparatively little research aiming at such activities directly. This observation motivates us to conduct a detailed exploration into unperceived activities on mobile systems. In particular, we conducted a series of experiments to 1) unveil how they deplete a smartphone battery silently, 2) characterize them in terms of their functionalities, as well as 3) analyze their distribution and energy breakdown. This exploration allows us to identify the activities that steal the most energy in different networks and draw other insights, which would be useful for system developers and application programmers to reduce those unperceived activities that consume energy without improving user experience.

III. An Exploration into Unperceived Activities

A. Unveiling Unperceived Activities

1) Definition: In this work, an unperceived activity is defined as a process invoked to run momentarily, without using any user-perceivable hardware components (including the screen, speaker, and vibrator) to attract the user's attention, when a smartphone is left unused. In other words, users are not aware that unperceived activities are taking place just in front of their eyes.

2) Observation Methodology: We explored unperceived activities systematically on the Samsung Galaxy S3, one of the most popular Android smartphones when this study was conducted. The specifications of the related hardware and software are detailed in Table I. As indicted in recent survey reports, instant messaging, social networking, and web browsing are three most frequently-used mobile

TABLE I
SPECIFICATIONS OF SAMSUNG GALAXY S3

Hardware	
Processor	Samsung Exynos (Cortex-A9) 1.4GHz
Memory	RAM: 1 GB
Screen	AMOLED 720 × 1280 pixels 4.8 inches
3G Network	HSDPA 850/900/1900/2100
Data Network	Wi-Fi IEEE 802.11 a/b/g/n
Storage	Flash: 32GB
Battery	Li-Ion 2100 mAh
Software	
OS	Android 4.1.2
	Linux Kernel 3.0.31

applications. Accordingly, we chose three representative mobile apps, namely LINE, Facebook, and Chrome, for investigation. To reduce the potential influence of other apps, the smartphone was recovered to its factory settings before each experiment. We installed and launched an investigated app at a time. Then, we put it into the background and turned off the screen. After the smartphone entered the sleep mode, we started to measure the smartphone's transient power and energy consumption via a power monitor produced by Monsoon Solutions for 30 minutes, because a period of 30 minutes was sufficient to observe the patterns of unperceived activities. An experiment was deemed invalid and restarted if any hardware that can attract user attention was activated during the idle session. The experiments were conducted respectively over Wi-Fi and 3G networks to investigate the influence of different types of networks on unperceived activities. Moreover, the energy consumption of the smartphone without the investigated apps installed was also measured for comparison. The results reported are the average values of 5 independent experiment measurements.

To gain further insights into unperceived activities, we employed a set of tools to collect real traces from the smartphone so that more decisive factors that waste energy could be explored. In particular, two native loggers, named *kmsg* and *syslog*, which reside respectively in the Linux kernel and the Android framework were used. The former can capture the activities triggered by the firmware, e.g., device drivers, while the latter was responsible to record the activities caused by software, e.g., system programs and mobile apps. For network activities, we used *tcpdump*, a common packet analyzer, to monitor the data traffic transmitted or received over Wi-Fi and 3G networks. Moreover, a diagnostic tool called *traceroute* was used to identify whether a packet was intended for an associated sever of the investigated app, and the packets were visualized with *WireShark* to facilitate observation. With regard to storage activities, *inotify*, an application programming interface, was utilized to intercept the read/write requests of each investigated app.

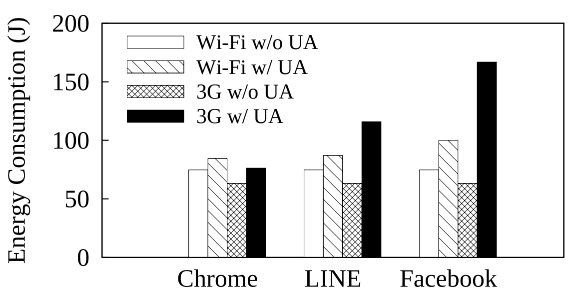


Fig. 2. Energy consumption in various idle scenarios

3) Impact on Energy Consumption: Figure 2 shows the energy consumption of the smartphone during a 30-minute idle session in various scenarios. The label "Wi-Fi w/ UA" denotes that the corresponding app was executed under the Wi-Fi network, while "Wi-Fi w/o UA" represents that the app was not launched. The same labeling rule is applied to the 3G scenarios. Clearly, the energy consumed by unperceived activities in most scenarios accounts for a significant amount of the idle energy consumption. Moreover, the energy wasted in all scenarios are much more under the 3G network than under the Wi-Fi network. The result shows that, unperceived activities can lead to a dramatic increase of 164% of the idle energy when Facebook was launched under the 3G network, compared with the same setting where Facebook was not launched. There is at least a 13% increase when Chrome was launched under the Wi-Fi network. Based on the numerical results, we draw a reasonable inference that a large number of unperceived activities might exist and they could lead to a more significant waste of energy than expected.

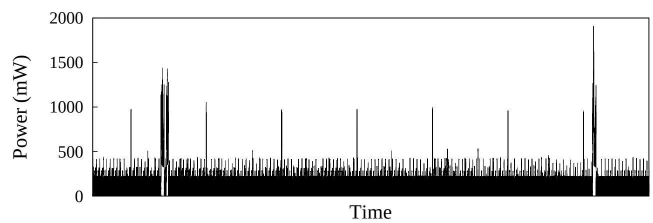


Fig. 3. A six-minute fragment of a power trace

To better understand how the energy is stolen, we picked a 6-minute fragment of a power trace, as shown in Figure 3, to present the smartphone's transient power during an idle session. Obviously, the increased energy consumption in Figure 2 primarily results from those *power pulses*. Even if the power pulses only remain for a short time and look sparse, they contribute significantly to the idle energy consumption, because the transient power incurred by unperceived activities is much higher than the idle power. The idle power is approximately 40 mW; however, a power pulse is often higher than 1000 mW and sometimes up to 2000 mW. This experiment result evidences the existence of unperceived activities, and they could drain the energy quickly in a silent manner.

B. Characterizing Unperceived Activities

1) Classification: Next, we dig into the collected traces to characterize unperceived activities and identify the major sources. Note that, to find out common activities among the investigated apps so that the findings could be applicable to other resident apps as well, the adopted classification should be application-independent. We observed that the power pulses, such as those in Figure 3, have a strong relationship with *system interrupts*, which can be roughly classified into two categories. The first category, called *active activities*, serves the requests registered by system libraries so as to wake up the smartphone to perform specific tasks. Such interrupts originate from system libraries and are triggered by the applications running on the smartphone actively. In contrast, the other category, called *passive activities*, is initiated by external facilities e.g., the access point or base station, rather than the smartphone itself. Passive activities are mainly generated by network interfaces so as to facilitate wireless communications. In the following, we classify the active and passive activities observed in the traces according to their functionalities.

2) Active Activities: When a smartphone stays in connected standby, the mobile system is frequently (and often regularly) awakened to maintain network connections. In particular, if the awakening involves polling data from a remote server or pushing some collected information back, network probing is a must before data synchronization. Apart from those network-related activities, the mobile system is also periodically or occasionally awakened to handle various tasks and routines that apps register via AlarmManager in the Android framework.

- **Network Probing (A1):** This type of activity occurs because mobile apps which connects to the Internet rely on a suite of network protocols (especially ICMP, ARP, and DNS) to probe and acquire some information of the current network environment (e.g., resolving network addresses).
- **Connection Maintenance (A2):** Such activities take place when acknowledgment messages are exchanged between a mobile device and its access point or base station in order to maintain network connections. Typical examples are the TCP and SSL messages used to maintain a stateful connection whose states are affected by the previous connection.
- **Data Synchronization (A3):** Unlike the previous two types, this activity type happens due to data synchronization, which allows a mobile device to establish data consistency with a remote storage system. Such activities are often presented with a burst of data transmissions (usually more than 10 packets).
- **Local Tasks and Routines (A4):** The remaining activities, which do not cause data communications via network interfaces, result mainly from various application requests for specific local tasks or routines, such as system information logging, battery checking, and location positioning.

3) Passive Activities: As a smartphone stays in a network, it will be awakened passively to interpret the messages and packets coming from the external environment. For example, the smartphone has to reply to a packet issued by another device for network probing. Apart from that, if there exists a stateful connection between the smartphone and a remote server, the server may take the initiative in sending data intended for the smartphone via some notification mechanism. Moreover, if the smartphone sends a message, it may expect to receive a response from the server or another device. There are also a variety of other sporadic activities that wake up the smartphone via network interfaces.

- **Network Probing (P1):** Similar to the first type of active activity, this type of passive activity occurs for the purpose of network probing. The difference is that the passive ones are initiated from the external environment; in other words, the smartphone is forcibly awakened to handle or interpret the packets generated by other devices.
- **Notification (P2):** Such activities may happen when a remote server actively pushes data to the smartphone to realize some notification mechanisms like GCM, which is a push notification service provided by Google for Android devices.
- **Response (P3):** Such an activity occurs when a response (e.g., an acknowledge message) under some network handshaking mechanism (e.g., TCP) returns later than expected; as a result, the smartphone has entered the sleep mode. Such activities often requires larger power consumption and extra switching energy than other activities.
- **Others (P4):** The remaining activities, which awaken the smartphone unexpectedly via network interfaces but not system libraries, are categorized into this type.

C. Analyzing Unperceived Activities

1) Metrics: After classifying the major sources of unperceived activities, we conduct statistical analysis from different perspectives to gain further insights. We analyze the average numbers, as well as the detailed *occurrence breakdown*, of active and passive activities per 30 minutes under different networks with various apps. This experiment helps us understand the distribution of different types of activities and how they behave in different networks. Similarly, we investigate the average energy consumption, as well as the detailed *energy breakdown*. The experiment allows us to find out how much energy unperceived activities steal and which types of activities are "principal culprits".

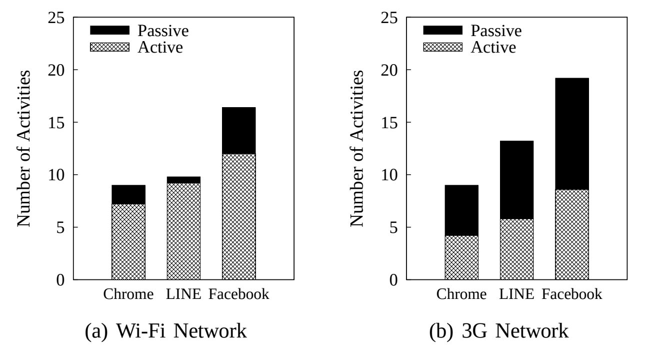


Fig. 4. Average numbers of activities per 30 minutes under different networks with various apps

2) Occurrence Breakdown: Figure 4 shows the total number of unperceived activities originated with each app under a Wi-Fi or 3G network. At a first glance, the network type does not have a significant impact on the total number of activities. However, the number of active activities is much larger under the Wi-Fi network than under the 3G network. Active activities account for 70% to 95% of the total activities, depending on the investigated app, under the Wi-Fi network. Just the opposite is true under the 3G network; passive activities accounts for 50% to 60% depending on the investigated app.

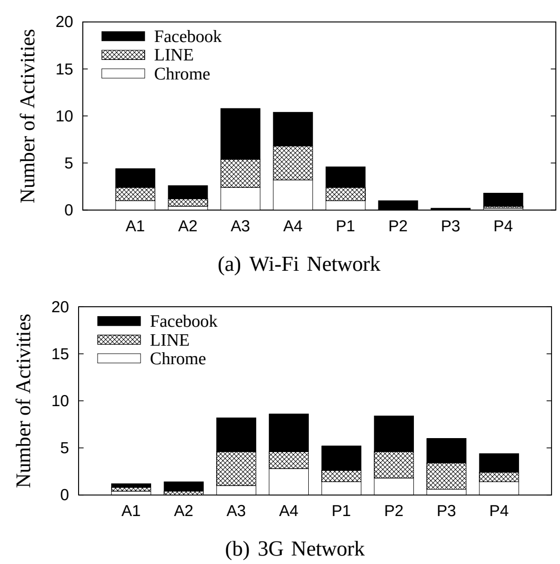


Fig. 5. Occurrence breakdown of unperceived activities under different networks with various apps

To figure out the cause, we further examine the occurrence breakdown of different types of activities, as shown in Figure 5, where the labels Ai and Pi (i = 1, 2, 3, 4) denote the eight activity types listed in Section III-B. We observe a few differences between the two networks. As far as passive activities are considered, the numbers of notification (P2) and response (P3) activities increase substantially under the 3G network. The reason for this interesting phenomenon is that 3G networks are often more unstable than Wi-Fi networks and network packets are more likely to be retransmitted or delayed. This, in turn, leads to more passive activities to awaken the smartphone. Moreover, with regard to active activities, the numbers of activities for network probing (A1), connection maintenance (A2), and data synchronization (A3) decrease greatly under the 3G network. Note that they are all network-related activities. We draw some reasonable inferences from the experiment results: 1) some resident apps may autonomously adjust the trigger frequency of network-related activities according to network status; and 2) the number of passive activities can dramatically increase under a 3G network due to network instability.

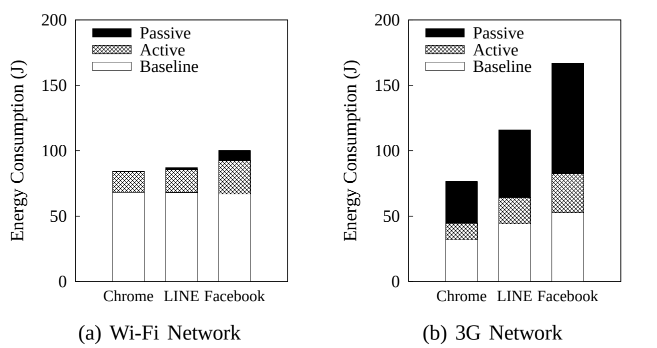


Fig. 6. Average energy consumption per 30 minutes under different networks with various apps

3) Energy Breakdown: Figure 6 shows the energy consumption incurred by unperceived activities under a Wi-Fi or 3G network with various apps, where the label “Baseline” denotes the energy not consumed for unperceived activities. As shown in the figures, the energy consumed by unperceived activities accounts for 20% to 33% of the total energy consumption under the Wi-Fi network, and the ratios increase to between 58% and 68% under the 3G network. The result indicates that unperceived activities can consume a significant amount of energy, especially under a 3G network.

For more in-depth analysis, we further examine the energy breakdown of different types of activities, as shown in Figure 7. Under the Wi-Fi network, the energy is consumed mainly by active activities, especially data synchronization (A3) and local tasks and routines (A4). In contrast, under the 3G network, the energy consumption is dominated by passive activities, like notification (P2) and response (P3), due to network instability. In addition, passive activities for network probing also consume a significant amount of energy under the 3G network. The reason behind the phenomenon is that the smartphone is awakened frequently by network probing from external devices in the same 3G network via a broadcast mechanism. In general, the energy breakdown of unperceived activities has a distribution similar to that of the occurrence breakdown shown in Figure 5. We draw some useful observations from the experiment results: 1) the energy consumption is proportional to the number of activities and network-related activities dissipate most of the energy, 2) active activities usually consume most (approx. 78%-87%) of

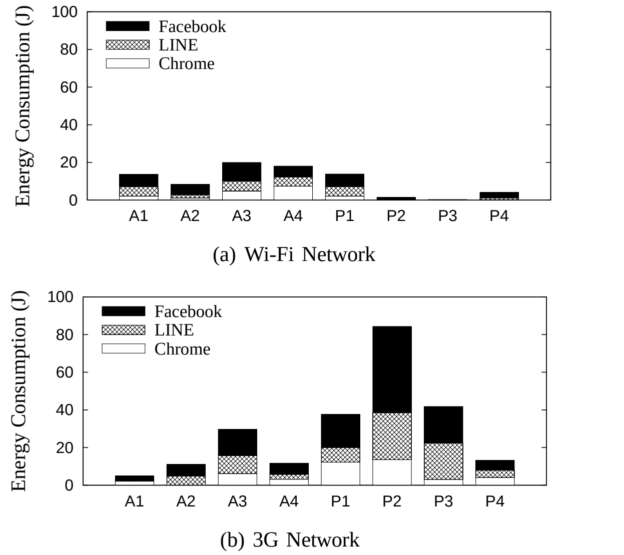


Fig. 7. Energy breakdown of unperceived activities under different networks with various apps

the activity-induced energy under a Wi-Fi network, while passive activities contribute most (approx. 74%-76%) of the energy under a 3G network.

IV. Solutions and Implementations

A. Potential Solutions

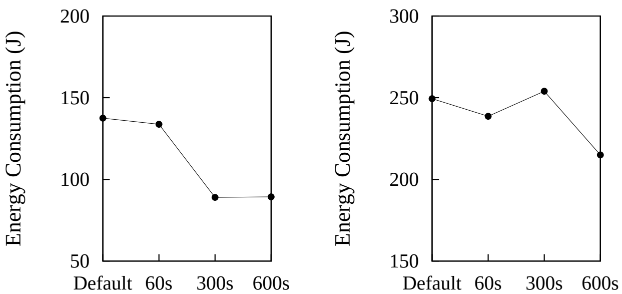
Unperceived activities consume a significant amount of energy. To alleviate unnecessary energy dissipation, we discuss two potential solutions inspired by the observations and implications we draw in the previous section for active and passive activities respectively.

Wake-up Alignment: Active activities may awaken a smartphone regularly and/or frequently, thus resulting in a considerable number of power pulses. With more apps installed, the smartphone would be waked up more frequently by unperceived activities originating from various apps. Thus, a straightforward way to reduce energy dissipation is to prevent the smartphone from being waked up frequently by not granting wake locks all the time. Instead, some *active* activities within a small time interval are aligned into the same group, so called *wake-up alignment* or *timer alignment* [5], when the smartphone is left unused. As a consequence, some unperceived activities that originally awaken the smartphone individually will be handled by the mobile system together. This solution is expected to be more effective under Wi-Fi networks, where active activities usually dominate unperceived activities.

Server-side Throttling: Passive activities may grow dramatically under a 3G network due to network instability. Most of the activities, such as those for notification or response, are initiated by external devices or servers, instead of the smartphone itself. Thus, one sensible solution, called *server-side throttling*, is to enable the server to adjust the frequencies of network probing or pushing data to smartphones adaptively according to their network conditions. Normally, the number of passive activities will decrease as the number of interrupts coming from the outside environment is reduced. Furthermore, applying the solution to smartphones might also be helpful to themselves and the entire network.

B. An Exemplary Implementation

Since the server-side throttling is dependent on how the server interacts with mobile devices and can not be realized in mobile systems, we implemented one solution, namely wake-up alignment, discussed above as an exemplary remedy to reduce the energy stolen by active activities. To this end, we modified a kernel module that takes charge of the real-time clock (RTC) so that the activation timing of unperceived activities can be rearranged. In addition, we exploited a library, called AlarmManager, which is responsible for collecting and managing activity-related information in the Android framework. With the support of the modified RTC module and the AlarmManager library, we aligned the active activities in each fixed time interval strategically and allowed the smartphone to be woken up only at specific time instants.



(a) Wi-Fi Network (b) 3G Network

Fig. 8. Energy consumption under various wake-up alignment settings

To evaluate the efficacy of the simple remedy, we conducted a set of experiments based on the Samsung Galaxy S3 smartphone with the three investigated apps, i.e., Chrome, LINE, and Facebook, installed simultaneously. In the experiments, the awaking period of the smartphone was configured via the modified RTC module and set at 60, 300, or 600 seconds. Note that, with fixed self-awaking periods, the smartphone will not lose any data/messages initiated from the external for the apps. The experiments were conducted under Wi-Fi and 3G networks respectively. Figure 8 shows the energy consumption of the smartphone when it was configured with a specific setting and left unused for a 30-minute idle session, where the energy consumed by the smartphone with its default setting was measured for comparison. As shown in Figure 8(a), the energy consumption decreases significantly (up to 35%) and then becomes saturated when the awaking period increases under the Wi-Fi network. The decrease is as expected because multiple active activities are aligned to be handled by the smartphone at the same awaking time. However, the energy reduction becomes saturated when the awaking period is configured as 300 seconds, and the reduction is negligible when the period is between 300 and 600 seconds. The reason for this interesting phenomenon is that wake-up alignment is only applicable to active activities; thus, the energy reduction will reach a limitation when the energy consumed by passive activities eventually dominates the total energy consumption. This also explains why the reduction is relatively small (up to 13%) under the 3G network, as shown in Figure 8(b). Note that the results under the 3G network are relatively instable because passive activities are generated irregularly from the external environment and involve many mobile devices in the network. In general, the experiment results are consistent with our previous observations that active and passive activities dominate the unperceived activities under Wi-Fi and 3G networks, respectively.

V. Concluding Remarks

We have presented the first detailed exploration of unperceived activities in mobile systems. This exploration unveil how unperceived activities drain energy bit by bit in a silent manner. We classify such activities roughly into two categories, active and passive, each of which is further classified into four types according to their functionalities. Moreover, we analyze their behavior in different networks, and find that active activities usually dominate under a Wi-Fi network while passive activities could increase dramatically under a 3G network. Based on the findings, we implement an exemplary remedy to reduce the energy stolen by active activities in Android. The results show that even a simple remedy can reduce energy dissipation by up to 35% and 13% under the Wi-Fi and 3G networks, respectively. We hope our exploration brings a new angle to this extensively-studied topic and provides useful insights that will inspire more elegant and energy-efficient designs for mobile systems.

Acknowledgment

This work was partially supported by the Ministry of Science and Technology under Grant 102-2221-E-001-007-MY2.

References

[1] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In *Proc. of ACM IMC*, pages 280–293, 2009.

[2] A. Carroll and G. Heiser. An Analysis of Power Consumption in a Smartphone. In *Proc. of USENIX ATC*, pages 21:1–21:14, 2010.

[3] H. Falaki, R. Mahajan, and S. Kandula. Diversity in Smartphone Usage. In *Proc. of ACM MobiSys*, pages 179–193, 2010.

[4] K. Kim and H. Cha. WakeScope: Runtime WakeLock Anomaly Management Scheme for Android Platform. In *Proc. of ACM EMSOFT*, pages 1–10, 2013.

[5] V. Könönen and P. Pääkkönen. Optimizing Power Consumption of Always-On Applications Based on Timer Alignment. In *Proc. of IEEE COMSNETS*, pages 1–8, 2011.

[6] C.-H. Lin, C.-K. Kang, and P.-C. Hsiu. Catch Your Attention: Quality-retaining Power Saving on Mobile OLED Displays. In *Proc. of IEEE/ACM DAC*, pages 1–6, 2014.

[7] 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. ACM EuroSys*, pages 29–42, 2012.

[8] A. Pathak, A. Jindal, Y. C. Hu, and S. P. Midkiff. What is keeping my phone awake? Characterizing and Detecting No-Sleep Energy Bugs in Smartphone Apps. In *Proc. of ACM MobiSys*, pages 267–280, 2012.

[9] F. Qian, Z. Wang, Y. Gao, J. Huang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Periodic Transfers in Mobile Applications: Network-wide Origin, Impact, and Optimization. In *Proc. of ACM WWW*, pages 51–60, 2012.

[10] A. Shye, B. Scholbrock, and G. Memik. Into the Wild: Studying Real User Activity Patterns to Guide Power Optimizations for Mobile Architectures. In *Proc. of IEEE/ACM MICRO*, pages 168–178, 2009.

[11] A. Shye, B. Scholbrock, G. Memik, and P. A. Dinda. Characterizing and Modeling User Activity on Smartphones: Summary. In *Proc. of ACM SIGMETRICS*, pages 375–376, 2010.

[12] N. Thiagarajan, G. Aggarwal, A. Nicoara, D. Boneh, and J. P. Singh. Who Killed My Battery: Analyzing Mobile Browser Energy Consumption. In *Proc. of ACM WWW*, pages 41–50, 2012.

[13] P.-H. Tseng, P.-C. Hsiu, C.-C. Pan, and T.-W. Kuo. User-Centric Energy-Efficient Scheduling on Multi-Core Mobile Devices. In *Proc. of IEEE/ACM DAC*, pages 1–6, 2014.

[14] F. Xu, Y. Liu, T. Moscibroda, R. Chandra, L. Jin, Y. Zhang, and Q. Li. Optimizing Background Email Sync on Smartphones. In *Proc. of ACM MobiSys*, pages 55–68, 2013.

[15] C. Yoon, D. Kim, W. Jung, C. Kang, and H. Cha. AppScope: Application Energy Metering Framework for Android Smartphones Using Kernel Activity Monitoring. In *Proc. of USENIX ATC*, pages 36:1–36:14, 2012.

[16] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones. In *Proc. of IEEE/ACM CODES+ISSS*, pages 105–114, 2010.

[17] X. Zhang and K. G. Shin. E-MiLi: Energy-minimizing Idle Listening in Wireless Networks. In *Proc. of ACM MobiCom*, pages 205–216, 2011.