Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems

Nilanjan Banerjee[1], Ahmad Rahmati[2], Mark D. Corner[1], Sami Rollins[3], and Lin Zhong[2]

[1,3] Dept. of Computer Science, [2]Dept. of Electrical and Computer Engineering
[1]Univ. of Massachusetts, Amherst, [2]Rice University, [3]Univ. of San Francisco

Abstract. Battery lifetime has become one of the top usability concerns of mobile systems. While many endeavors have been devoted to improving battery lifetime, they have fallen short in understanding how users interact with batteries. In response, we have conducted a systematic user study on battery use and recharge behavior, an important aspect of user-battery interaction, on both laptop computers and mobile phones. Based on this study, we present three important findings: 1) most recharges happen when the battery has substantial energy left, 2) a considerable portion of the recharges are driven by context (location and time), and those driven by battery levels usually occur when the battery level is high, and 3) there is great variation among users and systems. These findings indicate that there is substantial opportunity to enhance existing energy management policies, which solely focus on extending battery lifetime and often lead to excess battery energy upon recharge, by adapting the aggressiveness of the policy to match the usage and recharge patterns of the device. We have designed, deployed, and evaluated a user- and statistics-driven energy management system, Llama, to exploit the battery energy in a user-adaptive and user-friendly fashion to better serve the user. We also conducted a user study after the deployment that shows Llama effectively harvests excess battery energy for a better user experience (brighter display) or higher quality of service (more application data) without a noticeable change in battery lifetime.

1 Introduction

It is clear to any mobile user that the reliance on a battery and charging cord has a significant impact on usability, affecting when, where, and how people use mobile systems. Despite its importance, we understand little about how users replenish the energy on their devices. As a result, systems employ ad-hoc solutions for controlling power consumption, regardless of when users charge their devices or the lifetime they hope to achieve. Further, solutions are typically static, ignoring variance in the usage patterns exhibited by different users as well as differences in usage patterns across different devices. We believe that a better understanding of user-battery interaction will help us to ensure that systems are not overly conservative or aggressive, and adequately adapt to changing user behavior and device modalities.

To address this deficiency, we present the results of a systematic study of how users manage batteries. The goal of our efforts is to identify patterns in user behavior that can be leveraged to build systems that adaptively balance the quality of the user experience with longevity. We have collected battery traces using automatic logging tools, conducted a series of user interviews on battery management, and collected results from an *in situ* survey that asks about charging context. We have collected data from users of 56 laptops and 10 mobile phones. To our knowledge, this is the largest public study of battery use and recharge behavior in mobile systems.

This study has yielded three notable results. The first is that the test subjects *frequently* recharged their devices with a large percentage of their battery remaining. The second is that the test subjects' charging behavior was driven by one of two factors: context, such as location and time, or battery levels that are much higher than an empty battery. This can be contrasted with the fact that they are only occasionally driven by a truly low battery level. The third is that there are significant variations in patterns exhibited by users and particular mobile systems. For instance, laptop users typically use either very little of the battery capacity or almost all of it, whereas the mobile phone users generally use a greater portion of their battery, but rarely run completely out. These results highlight the problem of existing energy management policies, which are designed to extend battery lifetime without considering user-battery interaction.

Based on these observations, we have designed, implemented, and deployed an experimental adaptive system, named Llama, to help manage energy consumption in mobile systems. Because users frequently have excess energy remaining in their batteries at recharge, we hypothesize that existing energy management policies are often too conservative as they are designed to simply extend the battery lifetime. For instance, a laptop may reduce the brightness of its screen when unplugged even though the user will charge the device in the near future. In contrast, Llama estimates, for a particular user and device, how much battery is likely to go unused and adaptively adjusts the quality of service to meet the predicted requirement. In a deployment of Llama, we employed display brightness and data synchronization (health monitoring and web browsing data) as the example services for which Llama will use excess battery energy.

Based on a test deployment of Llama using 10 laptop and 10 mobile phone users, we present three results. First, Llama rarely caused the system to run out of energy causing a loss of working time. Second, as intended, users did recharge their laptops at a lower battery level, although it did cause some users to recharge their devices more often. This is because many users charge based on context, and others based on battery levels. Third, Llama provided improved quality of service, i.e., brighter displays and more health and web data. More importantly, the users were qualitatively pleased with the system, with only one user in twenty noting a change in charging behavior.

	Laptop		Phone	
	Users	Data	Users	Data
Trace Collection	56	15–150 days	10	42–77 days
User Interviews	10	N/A	10	N/A
In Situ Survey	10	30 days 415 responses	10	10–45 days 91 responses

Fig. 1. The number of participants and the amount of data for each research method of the user study.

2 User Study of Mobile Battery Use

In this study of user-battery interaction, we primarily wanted to examine where, when, and why people charge mobile systems. To this end, we have employed three complementary methods. The first method is an automatic trace collection tool that samples and records battery related information. This yields quantitative data on recharging behavior, but does not reveal information about why users do what they do. Thus, we also conducted user interviews to collect qualitative experiences with mobile battery use. Finally, as interviews rely on users' imperfect memories, we developed and deployed an *in situ* survey tool that delivers a questionnaire to participants at the moment they plug in their devices.

In this section we describe each of the three data collection methods, then present a summary of our findings with regard to aggregate and individual user behavior. All of our studies were conducted in parallel on laptop computers and mobile phones. The total number of participants and amount of data is shown in Table 1.

2.1 Methodology

Trace Collection: Our first method was a passive logging tool that periodically recorded the battery level and charging status. The laptop implementation is Java-based, runs on both Microsoft Windows and Apple OS X, and is downloadable and installable by the users themselves [1]. It samples the state of the machine every five minutes and the results are reported to our server once per day. Given the latency, high energy cost, and fragility of suspension and hibernation, we have made the tool completely passive: it records measurements only when the system is in an active or idle state and does not wake it. This leaves some gaps in the traces, such as plugging then unplugging the device while suspended, but we believe these cases are uncommon.

The phone logging tool is written in C++ and runs on Microsoft Windows Mobile, recording information every minute. We collected the results manually as not all users had data plans. Due to the well-known difficulties in producing software portable for mobile phones—especially when using low-level APIs—we chose to distribute the logging tool pre-installed on T-Mobile MDA phones [12]. In the case of phones, transition to suspension and other low-power modes is

[1] `http://prisms.cs.umass.edu/llama.html`

much more reliable, so the logging tool is more aggressive; it wakes the phone every 1 minute to record the battery and charging status of the system. The logging tool reduces the phone battery lifetime to approximately two days. Given that participants had little or no prior experience with this particular phone, they had no preconceived expectations of its battery lifetime.

For the laptop study, we have made every effort to gather a large pool of participants. We recruited participants from a large number of academic departments, friends and relatives, as well as community mailing lists and forums. For the mobile phone study, we recruited ten engineering undergraduate and graduate students. All but one were males with ages between 20 and 26. There was no overlap in participants for the laptop and phone studies. Given the method by which participants were recruited for each study, we make no claims about the randomness or represented demographics of the selection process, something we hope to improve in future work.

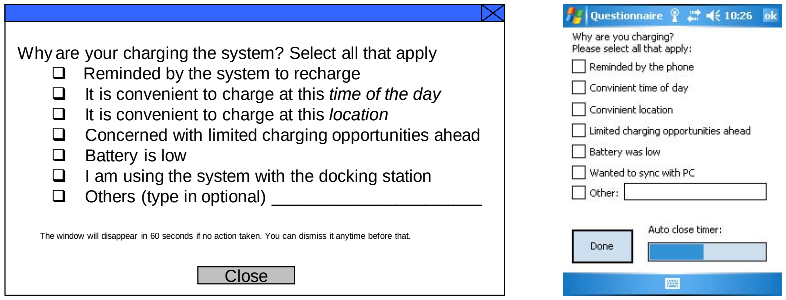
To gather information about the type of users in the laptop study, we asked them to fill out a short survey when downloading the tool. Of the respondents, 75% claimed to use their laptop as their primary machine at home and 52% said it was their primary machine at work. Only 3% of the participants said that they used multiple batteries in their laptops.

The total amount of data and number of participants are shown in Table 1. The laptop users have contributed between 15 and 150 days of data with an average of 68 days. The phone users have contributed 42 to 77 days of data with an average of 59 days. Our analysis and experiments are independent of the amount of data collected per-user, although the results for users with more data can be used with greater confidence.

User Interviews: Our second method was to interview users for qualitative data regarding their battery usage. Our goal in the interviews was to obtain more information about the context of battery usage and the subjective experience. 10 of the 56 laptop users and all 10 phone users participated in the interviews. The 10 laptop users consisted of 3 female and 7 male participants with age ranging between 20 and 30.

In the interviews, we provided some sample scenarios to think about and asked the participants about the last time they were in each scenario, what they were doing with the system, why it happened, how it impacted their future behavior. We encouraged the interviewees to tell their stories and anecdotes.

***In Situ* Survey:** Our third method was to ask users *in situ* why they recharge their system. All 10 phone users and the 10 laptop users that participated in the interviews were asked to install a tool that displays a pop-up survey, as illustrated in Figure 2. The window appears each time the system is plugged in. To minimize the intrusion and encourage users to supply only honest answers, the window can be easily dismissed and will disappear if there is no response in 60 seconds. We filtered out any intervals between charges that were less than 5 minutes to account for times when users accidentally unplugged the system and plugged it in again. We collected 415 responses from the 10 laptop



(a) Laptop (b) Phone

Fig. 2. *In situ* survey design for (a) laptops and (b) mobile phones.

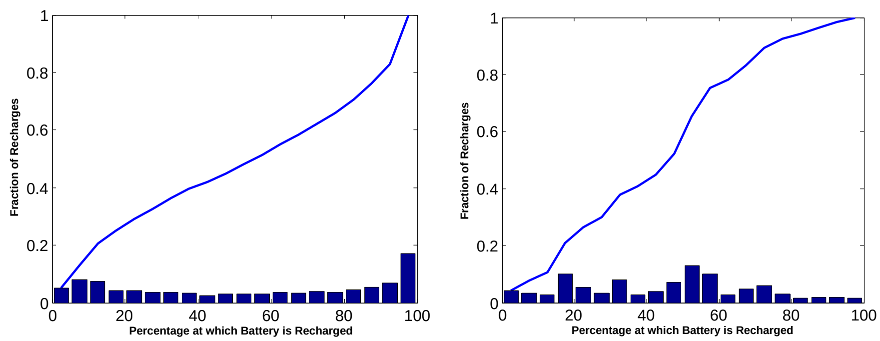
users and 91 responses from the 10 mobile phone users over an average period of 30 and 28 days, respectively.

2.2 Findings Regarding Battery Use and Recharge Behavior

Battery use and recharge behavior is an important aspect of user-battery interaction. Using a combination of data from the trace study, survey, and *in situ* questionnaire, we have reached several conclusions about the recharging behavior of the participants in the study. When attempting to correlate a combination of interviews, trace collection, and *in situ* questionnaires, we were often faced with difficulties in correlating large amounts of imperfect data, resolving discrepancies between collected and quoted information, and a seemingly unlimited number of questions and conclusions. In each instance, we have attempted to distill the highest confidence conclusions and those with the greatest implications for building systems.

The conclusions are as follows: First, when users plug in their devices to charge them, there is typically a significant amount of energy left in the battery. Second, charging of both laptops and phones was mostly and equivalently driven by context and battery levels significantly greater than empty, rather than low battery alarms. Third, there is significant variation among users and between devices. For instance, laptop users typically use very little of the battery capacity or almost all of it, whereas the mobile phone users generally use a greater portion of their battery but rarely run completely out.

The majority of recharges occur with a significant portion of the battery remaining: Figure 3, drawn from the automatic traces, shows the histogram and cumulative distribution of the battery remaining at recharge for both laptops and mobile phones. For each type of device, more than 50% of recharges occur when the battery is more than 50% full. Further, nearly 70% of laptop recharges and nearly 80% of phone recharges occur when the battery is more than 20% full.



(a) Laptops (b) Mobile phones

Fig. 3. Histograms and cumulative distribution of remaining battery upon recharge in the collected traces.

Charging is mostly and equally driven by context and battery levels rather than low battery alarms: The automatic traces cannot explain why users charge their devices, so we must draw results from the *in situ* questionnaire and user interviews. The results from our questionnaires are shown in Figure 4. The results indicate that *around half* of all recharges are triggered by context, including location, time of day, and for the case of phones, to synchronize with a PC. This was corroborated by our interviews; most laptop interviewees stated they usually recharge at the office, at home, and/or at night, driven by context, or based on the battery reaching a certain level. From the phone interviews, four out of ten participants claimed they charge the phone once or twice per day, without even looking at the battery level indicator. This is unsurprising, as many mobile phone users do not typically carry the phone charger thus only charge in a single location. For instance, mobile phone users said:

"I always recharge every night."
"I recharge every night, unless I forget."
"I usually recharge every night, or the other night if I have forgotten."
"I always keep my phone connected [to the USB/charge cable] when I'm working behind my computer."

Conversely, 28% of the laptop and phone responses indicated the reason for charging was a "low battery". At first this seems incongruent with the trace data shown in Figures 3(a) and 3(b). However, a cross examination of the responses and battery traces shows that when users select "low battery" as their reason for recharging, the average remaining battery level was actually 40%. This indicates that although users indicated that they weren't concerned with limited recharge opportunities ahead (7% and 5% of responses for laptops and phones), they were still acting in a very conservative manner. Six of the ten phone users indicated similar behavior, such as:

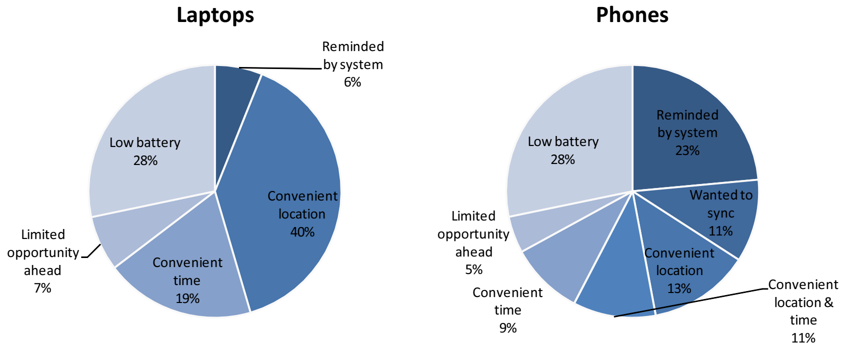
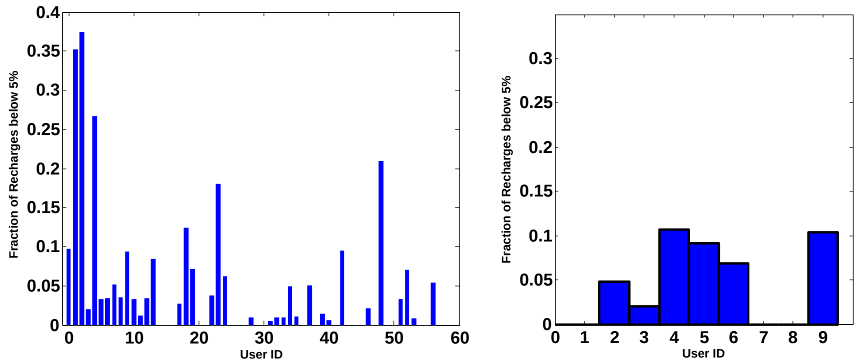


Fig. 4. *In situ* survey results.

> "I usually charge in the office, when the indicator shows 1 [of 2] bars."
>
> "I check the extra battery information screen and recharge around 40%, or when I want to sync"
>
> "I recharge when I get the low battery warning, [since] I still have plenty of time left after that."

In contrast, it is less common for laptop users to charge triggered by low battery alarms. Figure 5 illustrates, for each user, the number of times the device was recharged when the battery was below 5%. These scenarios are likely the result of a low battery warning or automatic system hibernation. Users reported that such scenarios occurred only during elongated trips without recharge opportunities. In these scenarios, the users took all measures to elongate the battery lifetime to focus on accomplishing key tasks, for example minimizing display brightness and turning off the network interface. Most users indicated that they usually mitigated the effects of these situations by fully charging their device beforehand.

Users and devices demonstrate significant variation in battery use and recharge behavior: Figure 6 is a box-and-whiskers plot of the remaining battery upon recharge for each participant. The graph shows the median, 25th & 75th percentiles, max-min values within 1.5x of the interquartile range, and outliers. We observe that not only is there significant variation across users, each user demonstrates variation in her own recharge pattern as well. We also note that there are significant differences between laptop and mobile phone charging patterns, as Figures 6 and 3 clearly show. Laptop users tend to use a larger portion of their energy and, as shown in Figure 5, they encounter low battery scenarios more commonly than mobile phone users. We suspect this is due to the fact that mobile phones often have a longer battery lifetime offering more physical opportunities for charging.



(a) Laptop users (b) Mobile phone users

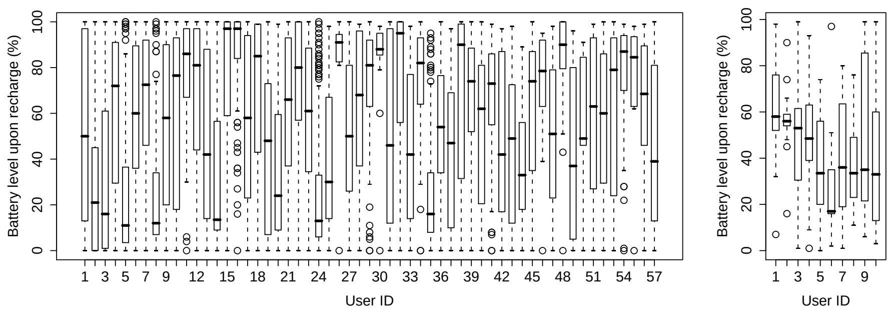
Fig. 5. The fraction of the time the battery falls below 5% for our participants.

2.3 Summary and Motivation for Adaptive Energy Management

The findings from our traces, interviews and questionnaire indicate several opportunities to provide an adaptive energy management system for mobile systems. The most compelling conclusion is the first one: users frequently charge their systems with a significant amount of energy remaining. If the system can perfectly predict how much energy the user will leave in the battery at recharge, it can proactively use the remaining energy to improve the quality of service given to the user. Such a system must be adaptive across users and systems, a necessity dictated by the third set of conclusions that such variations are common. By increasing the quality of service, fueled by so-called "excess energy", systems can deliver better usability such as increased screen brightness or lower latency responses.

However, such a system forms an implicit feedback loop with the user: if it uses extra energy, the user may recharge earlier. Even worse, it may cause the system to run out of energy prematurely, frustrating the user with an unexpectedly short battery lifetime. From the second set of conclusions, we see that there are primarily three kinds of behavior: charging based on context, charging based on conservative battery levels, and charging based on true low-battery conditions. Before designing an adaptive system, one can speculate that for context charging, an adaptive system will have little or no effect on the users—they will charge at the same times regardless of the adaptive system. In the case of charging based on battery levels, the system may have more of an effect, causing the user to charge the device more frequently. However, the adaptive system must carefully avoid the third case, true low-battery conditions, as they will be the most frustrating to the user.

Along the same lines, one should *not* interpret our findings as showing that longer battery lifetime is not desirable. Instead, longer battery lifetime will help



(a) Laptops (b) Phones

Fig. 6. Boxplot of remaining battery upon recharge for each participant. The middle dark line shows the median value, the boxes show the 25 & 75th percentiles, while the whiskers show the max and min values within 1.5x of the interquartile range. The circles show outliers.

users better deal with the current true low-battery conditions. More importantly, users will adapt to it with different battery use and recharge behavior, as our user study showed that they deal with laptops and mobile phones in very different ways. Most recharges happen with substantial battery because mobile users have developed realistic expectations and learn to effectively deal with the limited battery lifetime. The goal of an energy adaptive system is to find how long the user *needs* the device to last, and conform to that expectation. We certainly do not consider this a closed subject, and our work is a step along a new direction of research in *user-centric* power management.

In the next section we describe the design of a novel adaptive energy management system, named Llama, that addresses all of these requirements.

3 Llama: User-Driven Energy Management

Based on the results of our user-study, we have designed a system, named Llama, to manage battery energy on mobile computing systems. Llama tracks how much of the battery is typically used on a system, and uses it to predict the *excess battery energy* of the system, or how much energy will be left in the battery when the system begins its next recharge cycle. One other system has attempted to use this excess energy, SMERT [15], an energy-efficient multimedia messaging system on mobile phones. However, unlike Llama, it is based on pre-set knowledge of expected battery lifetime without automatically tuning to a user's patterns.

Llama, in its current incarnation, assumes that the default system power management decides the minimum acceptable level of power consumption. Llama then devotes the predicted excess battery energy to extra, non-critical services or applications. Such a Llama application can be anything that has a fidelity-power tradeoff. Examples include increased screen brightness, periodic synchronization with a distributed file server, periodic collection and transfer of sensor readings, providing services to peers in a cooperative system, and web prefetching. Llama applications can either be continuous in their use of energy, such as screen brightness, or discrete, such as periodic sensor readings.

3.1 Energy-Adaptive Algorithm

First, we formally define the optimization problem that the Llama algorithm solves. If E_b and E_f are the energy consumed by the Llama application and all other applications between two consecutive recharges, the algorithm tries to maximize E_b—subject to $Pr[E_b + E_f > C] \leq p$—where C is the total battery capacity and p is the confidence of Llama not exceeding the battery capacity. Though we have used the number of times the user runs out of battery as the optimization constraint on Llama, other criteria could also be used to quantify the effect of Llama on user behavior.

There are two key features in the Llama system design. First, Llama uses a *probabilistic* algorithm, ensuring that the Llama application impacts the user-perceived lifetime of the system only a small percentage of the time. Second, Llama is *adaptive*; it responds to short-term and long-term changes in user behavior. The main component of Llama is a *predictor* that estimates the excess battery energy using a histogram of previous battery usage and the current battery capacity. The histogram is measured using the same technique used in the user-study—when the system is awake, Llama periodically records the battery level and tracks when a recharge begins. Using the predicted excess battery energy, the predictor can decide how much energy to devote to the Llama application. The process is adaptive: it recalculates the energy to devote to Llama tasks periodically, as the battery drains.

As an example of how the Llama algorithm operates, suppose a user wants to assure with a confidence of 95% that their battery will not run out before the next expected recharge. Llama determines the current battery capacity to be 30%. It consults the histogram of recharges and determines that 95% of the time that the battery drains below 30% the user recharges at or above 10%. It then allows the Llama application to use up to 10% of the battery.

Algorithm 1 Energy-Adaptive Algorithm

Confidence of not exceeding battery capacity $= p$
Histogram for CDF of recharges given present battery remaining $C_p = H$
Size of Histogram bin $= \Delta H$
Find x such that $H(x) \leq (1 - p) \leq H(x + \Delta H)$
Excess energy for Llama tasks $= x$
Energy for foreground tasks $E_f = 100 - x$

The algorithm, shown as Algorithm 1, tracks the probability distribution of when the user recharges to determine the amount of energy that can be spent on the Llama application. For a continuous application, such as screen brightness, the application spreads this energy usage over the remaining time before the next recharge, and uses energy at the rate $\frac{E_b}{T_d}$, where T_d is the expected time before the next recharge. Llama currently uses the mean time between past recharges as the measure for T_d.

3.2 Supporting Discrete Background Tasks

Llama also enables a new kind of functionality not typically used in laptops, PDAs, and portable music players: *adaptive, self-initiated wake-up.* Generally, once a system places itself in a low-power state, such as suspension, hibernation, or off, it requires user intervention to start again. However, Llama can schedule wake-sleep cycling to proactively run background tasks while the user is not actively using the system. Note that at a hardware level this is already well-supported, but concerns over using excess energy has kept it from being widely adopted. Thus, Llama can execute a background task, such as taking sensor readings, or checking for new email, on a periodic basis by waking up the system and executing the task.

As the transition cost from sleeping to awake can be significant in any platform, Llama must take this cost into account when deciding how often to initiate a wake-up. In such cases, the interval after which a background operation should occur is calculated as $\frac{T_d}{E_b} \cdot (e_b + E_t)$, where e_b is the energy for one background task, and E_t is the cost to transition the system to the active state to complete the task. The system has a predetermined set of time intervals after which a task can be executed. These intervals determine how aggressive the task is. For example, the intervals could determine how often a perfecting occurs. T_d is mapped to the closest time interval. However, E_t will vary depending on the beginning state of the system; it costs more to transition a system from suspended to active than from idle to active. Since we do not know what state the system will be in when we execute a task, we predict it will be in the same state as the last time a task was executed. If the the last time Llama initiated a background operation the system was active, then $E_t = 0$. Otherwise we use the energy needed to resume and then suspend the system, which can be measured online using information provided by the operating system.

3.3 Measuring Energy Usage

Llama's calculation relies upon knowledge of the power consumption of the Llama application and the rest of the system, as well as the measured histogram of recharges collected by the logging tool. Llama measures the power consumption of the system and the Llama application by observing the power or energy of the system as it executes the Llama application at different rates. For example, if the screen brightness level of the system is mapped to an interval $[0, 1]$, where 0 is totally dark and 1 is fully bright, Llama measures the power consumption at 0, 0.25, 0.5, 0.75 and 1.0. It can then set the brightness according to its estimate of excess battery energy and T_d. If the task is periodic it must measure the energy to execute the task once, and use that to estimate the power at different intervals.

Recall that Llama may create a feedback loop with the user, causing them to recharge their device earlier, rather than accepting the excess energy use. However, due to the obvious complexities and ambiguities introduced by trying to factor the user into the algorithm, we have designed the algorithm based on an idealized system that does not contain such feedback. Only through deployment and experimentation, the subject of the next section, can one discover how this relationship bears out.

4 Llama Evaluation

To evaluate the efficacy of the Llama system, we built a working implementation and deployed it on 10 laptops and 10 mobile phones for approximately one month. Before the addition of Llama, mobile systems are already set with a default energy management policy, or one previously tuned by the user. For the purposes of this evaluation we take this policy as the minimum acceptable quality of service and Llama will only use more energy than this policy, not less.

4.1 Llama Deployment

To test the Llama algorithm, we tried a variety of applications, some noticeable to the user and some invisible.
Laptop Screen Brightness application: For Mac laptops we employed a screen brightness adjustment application. We chose screen brightness levels in discrete intervals of $\{0.25, 0.5, 0.75, 1.0\}$ where 0.25 is the least amount of backlight and 1 is full brightness. After installation, the application trains itself to learn the amount of power consumed by the laptop at each level. From then on, the predictor wakes up every 5 minutes to estimate the amount of battery capacity that will remain in the battery when the next recharge takes place. The predictor does so through the history of recharges stored on the machine from the readings taken by our tracing tool. The scheduler then calculates what the screen brightness level should be such that the laptop does not run out of battery with a probability $p = 0.9$. Consequently, the scheduler sets the screen brightness to the level using a script.
Laptop Web Prefetching application : For Windows laptops, we used a web prefetching application that downloads a random webpage from a set of 10 preconfigured choices. The application only runs when the device is active or idle and does not wake it from a suspended state. In this case, the user did not interact with the application and we did not serve prefetched pages to the user. The downloading interval determines the aggressiveness of the application, chosen as once every 30, 60, 120, or 180 seconds. Similar to the screen brightness application the predictor determines the battery capacity at the next recharge. It then

uses the excess energy to determine how often the web prefetching application should run given the excess energy.

Mobile Phone background task: For the mobile phones, we employed a remote health monitoring application used in a previous project—the original application periodically uploaded data from a wireless electrocardiogram (ECG) sensor. We have replaced the sensor with preprogrammed data, and adapted it to report at variable intervals from once every 5 minutes to once every 60 minutes. This leads to an average extra phone power consumption of 3.3 to 40mW. These intervals reduce the two-day idle battery lifetime of the phone by 1.5 to 14 hours respectively.

4.2 User Studies after Llama Deployment

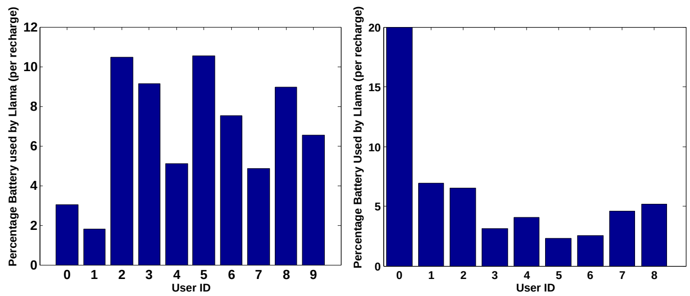
We deployed Llama for both laptop and phone users, and collected battery usage and Llama operation traces for approximately 30 days. We interviewed participants to gather additional subjective experience regarding Llama. The interviews were semi-structured and were conducted in very casual fashion, focusing on eliciting stories from our participants to gauge the effectiveness and user-friendliness of Llama. We first asked the same set of questions regarding different charging scenarios, as described in Section 2.1. For the laptop participants, we then asked whether the interviewee was comfortable with automatic adjustment of brightness and whether they were concerned with its battery impact. At the end, we asked both laptop and phone participants whether Llama had impacted their battery lifetime, charging behavior, and their work.

Effectiveness of Llama in Using Excess Energy: Figure 7 shows the energy use of Llama per recharge for the laptop and mobile phone applications. Figure 8 shows the cumulative distribution of the recharges for approximately 30 days before and 30 days after installation of Llama software.

For laptops, users 2 through 9 correspond to the web prefetching application while users 0 and 1 had the screen brightness application installed on their Mac laptops. Llama used variable amounts of energy, varying from 2% to 11%, depding on the amount of battery remaining at recharge.

From Figure 8(a) we find that the percentage at which users recharge their laptop goes down after installation of Llama. For example, for more than 50% of the recharges, users recharge their laptops at 5% less than they did before installation of Llama. Beneficially, the use of Llama led to an average of 629 webpages of average size 90 KB prefetched each day for the Windows laptop participants and a 3 times brighter display for 16% of the time per day for the two MAC participants. Though we do not directly evaluate improved user experience as a result of Llama, these results strongly indicate that Llama can provide significant benefit to the user.

For mobile phones, Figure 7(b) shows that Llama effectively employed excessive battery energy and enabled the reporting of an average 23 MB of data per recharge for each participant. The average Llama transfer interval for different participants was between 13 and 59 minutes. The average among all phone par-



(a) Laptops (b) Mobile phones

Fig. 7. The figure shows the amount of energy used by Llama for the web-prefetching and screen brightness applications for the laptop, and the health monitoring application for the phones.

ticipants was 26 minutes, corresponding to a power consumption of 7.6mW or about 5% of the battery capacity per day.

User-perceived effect: During the interviews, we found that none of the laptop users had noticed a change in battery life after Llama deployment. After we suggested that Llama may induce extra battery usage, we noted two comments:

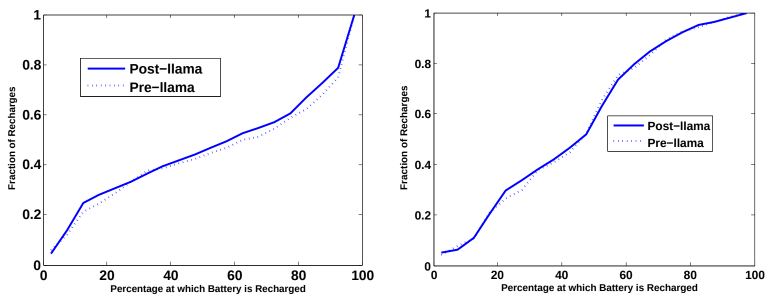
> "It must have been small, since I didn't notice it."
> "Even though I didn't notice it, I would definitely care in situations where I require maximum battery life."

These and other similar comments confirm that Llama has been successful in only employing excessive battery energy, although in the second case the user expressed some concern over using the system.

The mobile phone users also expressed similar satisfaction. Only one mobile phone participant noticed a shorter battery lifetime. He also indicated in the user interview that he always checks the extra battery information screen and charges at around 40%. He offered the only negative comment about the system:

> "The battery lifetime was better last month. I have to recharge it every day now, but it used to be every day and a half.

It is important to note that no participants noticed increase in the time taken to fully recharge the battery, although the average remaining battery level is lower after Llama deployment. One reason is that the difference in remaining battery level often leads to much smaller difference in the time required to fully recharge. For example, 10% lower battery level may only need 3% longer charge time. The second reason is that participants typically keep their devices plugged



(a) Laptops (b) Mobile phones

Fig. 8. The figure shows the cumulative distribution of all recharges before and after installation of Llama

	Before Llama	After Llama
All Laptop Users	6.5	7.8
All phones	10.1	8.9

Fig. 9. Average charge attempts per week on phones and laptops before and after Llama.

in for more than 3-4 hours, enough time to recharge any battery, as we observed from the field-collected battery traces.

Recharge behavior change: Table 9 shows the average number of recharges per week, pre- and post-Llama. As Llama employed considerable battery energy, we did observe an increase in the recharge frequencies in laptops, but not so in mobile phones. Further, the pre-Llama traces show that show 1% of recharges for laptops and 4% for mobile phones occur with the battery below 5%. In contrast, after Llama was installed the battery ran below 5%, in 1% of recharges for laptops and 7% for mobile phones. We have more confidence in the laptop results, but in both cases the number of instances of an empty battery is small

We believe that this increase in charging activity is due to users charging their laptops driven by the battery indicator. This is an example of the feedback loop we speculated at the end of Section 2.2. For users that are primarily driven by context such an effect does not occur, and for those primarily driven by the battery indicator, it will cause an increase in charging activity, but not an increase in dead batteries. It is our view that users have learned over time to be quite conservative with their batteries because the default, static policies of mobile systems have trained them to be so. We feel that users will adapt to Llama to charge their systems at lower levels over a long term. The results also motivate the need for further study to determine if Llama can be more aggressive without the increased recharging rate frustrating the users, or if there are ways to improve the prediction capabilities of Llama to use battery power without increasing recharges. Our interview results also suggested the need of a "maximum battery mode" for a user to override Llama when they anticipate a scenario like air travel.

5 Related Work

Extensive research has been devoted to energy-efficient design of mobile systems; however, little is known about why and when users recharge their batteries. Froehlich and Chen presented a small-scale battery study on four participants for two weeks as a case study of their *in situ* survey tool MyExperience [6]. While limited in scope and scale, they also found there was still significant battery upon recharge and less than 30% of the recharges were driven by "low battery". With a focus on understanding mobile battery use and recharge behavior, we present a systematic study at a much larger scale with multiple research methods, including trace logging, interviews as well as *in situ* surveys for both laptops and mobile phones. We reported a complementary study of user-battery interaction on mobile phones with a focus on revealing mobile phone design problems that lead to ineffective user-battery interaction [10]. In this work, however, our focus is on understanding how mobile users decide to recharge and on applying the new knowledge to user-adaptive system energy management, for both mobile phones and laptops.

Related to our strategy of considering human factors in energy management, HP Labs researchers selectively darkened part of organic light-emitting diode (OLED) displays for energy reduction [9] and evaluated the user acceptance for this technology [7, 1]. Cheng *et al.* [2] exploited limitations in human visual perception to reduce the power consumption of traditional LCDs and Vallerio *et al.* studied the effect of user-interface design on energy efficiency [13]. All of these projects focus on improving the battery lifetime of mobile systems. In contrast, our work focuses on understanding mobile battery use and how to make better use of the available battery energy.

Similar research methodologies have been developed as studies on tools for studying mobile users in their natural settings [3, 4, 8, 6]. In particular, Demumieux and Losquin presented a mobile phone logging tool for studying human-computer interaction on mobile phones [4]. We have employed similar logging to specifically to capture battery use and recharge behavior. MyExperience described an *in situ* survey tool, similar to the one employed in our work. *In situ* survey tools make event-based user experience assessment possible by delivering inquiries at the time of interest [3] . While each of these have contributed to the development of tools, our focus is on the use of these tools for studying mobile battery use. More importantly, we employ a triangulation of research methods, including logging, interviews, and *in situ* surveys, to provide a combined strength in studying how mobile users deal with the limited battery lifetime. Our work presents an example of leveraging the complementary strengths of both quantitative measurements and qualitative inquiries as well as both monitoring and self-reporting.

We note that many projects, including Odyssey [5], ECOSystem [14], and work by Simunic *et al.* [11], attempt to balance performance and system-wide energy usage. The general problem of how to allocate energy, or power, to competing system components and applications, is orthogonal to our work. Ultimately, we envision the integration of Llama with such systems to provide a power management solution that adapts to individual user behavior and can appropriately allocate energy based on competing needs.

6 Future Work

We wish to expand our work in several ways: (i) a larger number of test subjects, particularly mobile phone users, (ii) a less biased subject selection method, or perhaps one that is demographically weighted, (iii) more types of mobile systems including portable music players, (iv) linking user behavior, thus adaptive energy management, with other contextual clues, such as location, mobility, and work patterns, and (v) longer-term studies of Llama to see if users can be retrained to use adaptive systems more effectively. We also hope that researchers will use the results of our study in building new systems and designing new user studies.

7 Conclusions

In this paper we have presented the results of an extensive trace collection and user study that provides a first glimpse into the battery use and recharge behavior of mobile systems, in particular laptops and mobile phones. We have made three key observations in three comprehensive user studies: (i) many users frequently leave excess energy in the battery when recharging devices, (ii) charging behavior is more often than not driven by opportunity, context, and conservative behavior, rather than low battery conditions, and (iii) significant variations occur across mobile users and systems. Based on these three observations, we have created an adaptive energy management system, named Llama that can scale energy usage to user behavior, probabilistically matching energy consumption with the expected recharge time. We have deployed this tool on a number of laptops and mobile phones and received generally positive feedback.

We fully realize that the concept of "excess energy" in a mobile device is not without controversy. After extensive casual conversations with many users on the pros and cons it is clear that, *prima facie*, such a system may work well for some, but perhaps not all users. However, we believe that our research on user behavior and the Llama system is a first step in discovering better adaptive energy policies in mobile systems.

Acknowledgements

We would like to thank our shepherd Adrian Friday and the anonymous reviewers for their valuable comments. We would like to thank all the test subjects who participated in the user study. We would also like to thank Denitsa

Tilkidjieva, Maria Kazandjieva and Nitin Ramamurthy who helped with the development of the llama software and an earlier version of the paper. This research was supported by NSF grants CNS-0519881, CNS-0520729, CNS-0447877, and CNS-0509095. A. Rahmati and L. Zhong were supported in part by Enhancing Rice through Information Technology program, a seed grant funded by Sheafer/Lindsay from the Computer and Information Technology Institute at Rice University and the TI Leadership University Innovation Fund.

References

1. L. Bloom, R. Eardley, E. Geelhoed, M. Manahan, and P. Ranganathan. Investigating the relationship between battery life and user acceptance of dynamic, energy-aware interfaces on handhelds. In *Proc. Int. Conf. Human Computer Interaction with Mobile Devices & Services (MobileHCI)*, pages 13–24, Sept. 2004.
2. W.-C. Cheng and C.-F. Chao. Perception-guided power minimization for color sequential displays. In *Proc. Great Lake Symp. VLSI*, pages 290–295, 2006.
3. S. Consolvo and M. Walker. Using the experience sampling method to evaluate ubicomp applications. *IEEE Pervasive Computing*, 2(2):24–31, 2003.
4. R. Demumieux and P. Losquin. Gather customer's real usage on mobile phones. In *Proc. Int. Conf. Human Computer Interaction with Mobile Devices & Services (MobileHCI)*, pages 267–270, 2005.
5. J. Flinn and M. Satyanarayanan. Energy-aware adaptation for mobile applications. In *SOSP 99*, pages 48–63, Kiawah Island, SC, USA, Dec. 1999.
6. J. Froehlich, M. Chen, S. Consolvo, B. Harrison, and J. Landay. MyExperience: A system for in situ tracing and capturing of user feedback on mobile phones. In *Proc. Int. Conf. Mobile Systems, Applications, and Services (MobiSys)*, June 2007.
7. T. Harter, S. Vroegindeweij, E. Geelhoed, M. Manahan, and P. Ranganathan. Energy-aware user interfaces: An evaluation of user acceptance. In *Proc. Conf. Human Factors in Computing Systems*, pages 199–206, Apr. 2004.
8. S. Intille, E. Tapia, J. Rondoni, J. Beaudin, C. Kukla, S. Agarwal, L. Bao, and K. Larson. Tools for studying behavior and technology in natural settings. In *Proc. Int. Conf. Ubiquitous Computing (UbiComp)*, pages 157–174, 2003.
9. S. Iyer, L. Luo, R. Mayo, and P. Ranganathan. Energy-adaptive display system designs for future mobile environments. In *Proc. Int. Conf. Mobile Systems, Applications, and Services (MobiSys)*, pages 245–258, May 2003.
10. A. Rahmati, A. C. Qian, and L. Zhong. Understanding human-battery interaction on mobile phones. In *Proc. Int. Conf. Human Computer Interaction with Mobile Devices & Services (MobileHCI)*, Sept. 2007.
11. T. Simunic, L. Benini, P. Glynn, and G. DeMicheli. Dynamic power management for portable systems. In *MobiCom*, Boston, MA, USA, Aug. 2000.
12. T-Mobile MDA. *http://www.t-mobile.com/*. 2006.
13. K. S. Vallerio, L. Zhong, and N. K. Jha. Energy-efficient graphical user interface design. *IEEE Trans. Mobile Computing*, 7(5):846–859, July 2006.
14. H. Zeng, C. Ellis, A. Lebeck, and A. Vahdat. ECOSystem: managing energy as a first class operating system resource. In *ASPLOS*, San Jose, CA, USA, Oct. 2002.
15. L. Zhong, B. Wei, and M. J. Sinclair. SMERT: energy-efficient design of a multimedia messaging system for mobile devices. In *Proc. ACM/IEEE Conf. Design Automation (DAC)*, pages 586–591, 2006.