

Energy-Efficient Rate-Adaptive GPS-based Positioning for Smartphones *

Jeongyeup Paek Joongheon Kim Ramesh Govindan

Embedded Networks Laboratory
Computer Science Department
University of Southern California
{ jpaek, joonghek, ramesh }@usc.edu

ABSTRACT

Many emerging smartphone applications require position information to provide location-based or context-aware services. In these applications, GPS is often preferred over its alternatives such as GSM/WiFi based positioning systems because it is known to be more accurate. However, GPS is extremely power hungry. Hence a common approach is to periodically duty-cycle GPS. However, GPS duty-cycling trades-off positioning accuracy for lower energy. A key requirement for such applications, then, is a positioning system that provides accurate position information while spending minimal energy.

In this paper, we present RAPS, rate-adaptive positioning system for smartphone applications. It is based on the observation that GPS is generally less accurate in urban areas, so it suffices to turn on GPS only as often as necessary to achieve this accuracy. RAPS uses a collection of techniques to cleverly determine when to turn on GPS. It uses the location-time history of the user to estimate user velocity and adaptively turn on GPS only if the estimated uncertainty in position exceeds the accuracy threshold. It also efficiently estimates user movement using a duty-cycled accelerometer, and utilizes Bluetooth communication to reduce position uncertainty among neighboring devices. Finally, it employs celltower-RSS blacklisting to detect GPS unavailability (e.g., indoors) and avoid turning on GPS in these cases. We evaluate RAPS through real-world experiments using a prototype implementation on a modern smartphone and show that it can increase phone lifetimes by more than a factor of 3.8 over an approach where GPS is always on.

Categories and Subject Descriptors

C.3 [Special-purpose and Application-based Systems]: Real-time and embedded systems

*This material is based upon work supported by the National Science Foundation under Grant Nos. 0121778 and 0905596. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiSys'10, June 15–18, 2010, San Francisco, California, USA.
Copyright 2010 ACM 978-1-60558-985-5/10/06 ...\$10.00.



Figure 1: Accuracy of GPS, WPS, and GSM-based Positioning

General Terms

Algorithms, Design, Experimentation, Performance, Measurement

Keywords

Energy-Efficient, Adaptive Positioning, Smartphone, GPS, Sensor

1. INTRODUCTION

Many emerging smartphone applications require position information to provide location-based or context-aware services. Our Urban Tomography [13] system is a good example. It allows a user to capture video clips, tags each with the most recent position information, and then automatically uploads each video to a server. Many participatory sensing applications, where the phone is autonomously recording ambient conditions or user activity, also continuously record position information. Other applications that make continuous use of location or context information are MicroBlog [11], TrafficSense [16], Pothole Patrol [8], MetroSense [7], PlaceIts [23], PeopleNet [17], MyExperience [10].

In these applications, GPS is often preferred over its alternatives such as GSM/WiFi based positioning systems because it is known to be more accurate. Figure 1 plots two example locations from which we have collected positioning data using all three positioning systems: GPS, WPS (WiFi-based positioning system from SkyHookWireless [22]), and GSM-based positioning on a N95 smartphone over the AT&T cellular network. Both locations had a clear view of the sky and usable WiFi access points so that GPS and WPS could work. The figure clearly shows that WPS is less accurate than GPS, and GSM-based positioning has an error as high as 300 meters. For these reasons, the use of GPS for location-based

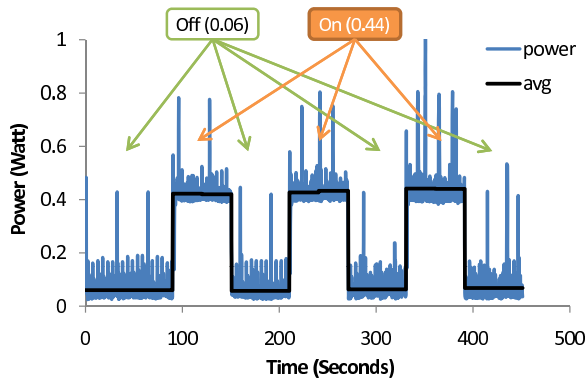


Figure 2: Power consumption of GPS on N95 smartphone (Repeated 30 seconds on and 90 seconds off)

applications is unlikely to diminish; it may, however, be augmented with these other positioning methods.

However, it is also well-known that GPS is extremely power-hungry. Our measurements of the power consumption¹ (shown in Figure 2) agrees with the results published by others [26, 3] and confirms that the internal GPS on Nokia N95 smartphones uses around 0.37 Watt of power on top of ~ 0.06 W idle power. Keeping GPS activated continuously would drain the 1200 mAh battery on an N95 smartphone in less than 11 hours, even in the absence of any other activity. This is clearly a significant roadblock on the way to all-day smartphone usage, and a more intelligent and energy-efficient activation of GPS is the subject of our paper.

The key insight that motivates our work is the observation that, when used by pedestrians in urban areas, GPS can exhibit errors in the range of 100m. GPS inaccuracy in urban “canyons” is well-known, but we have found that, even in relatively benign environments such as college campuses or residential neighborhoods, GPS can exhibit this kind of inaccuracy, especially for pedestrian smartphone usage. Location-based applications will have to deal with this level of error using application-specific methods, such as map-matching or map-snapping. So we ask: if applications can tolerate this position error, why not trade off some position accuracy for reduced GPS energy usage? A simple way to do this is to periodically duty-cycle GPS. This trades-off positioning accuracy for lower energy. However, the key challenge in this periodic GPS duty-cycling is to decide on a suitable time period; for almost any choice, there exists a user mobility pattern that will result in unbounded position error (Section 2).

To avoid this, a thread of research has examined different heuristics to cheaply determine a change of position so that GPS can be selectively activated [12, 5, 27, 9, 6, 3]. Our work follows in this thread, but makes two contributions. First, it introduces novel techniques for cheaply inferring whether and when GPS activations are necessary. More important, rather than merely considering each technique in isolation, we design a complete system that uses a collection of techniques in concert to reduce energy usages.

At the core of our approach is a method to estimate user velocity from a history of previously measured velocities at the same location and the same time-of-day; intuitively, this velocity estimation leverages consistency in user behavior. When the estimated distance traveled approaches a user-specified accuracy bound, our sys-

¹ We have used the Power Monitor device from Monsoon Solutions Inc. for all of our power measurements and cross-verified it with the Nokia Energy Profiler v1.2 software tool.

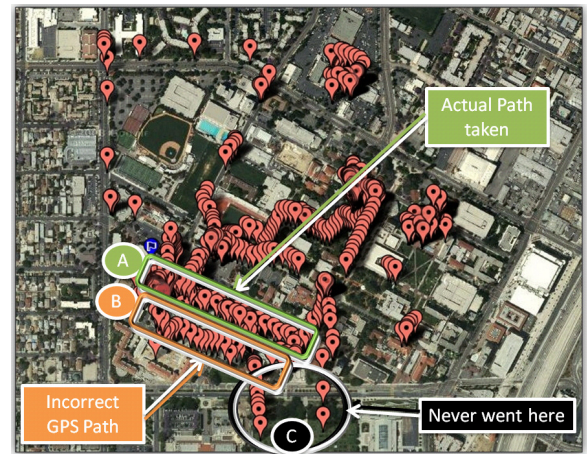


Figure 3: GPS trace plot showing the inaccuracy of GPS in urban areas.

tem (called RAPS, the rate-adaptive positioning system) activates GPS. This decision to activate GPS is delayed if the user’s current average activity level, as measured by a duty-cycled accelerometer, is inconsistent with historical activity at that position and time-of-day. Similarly, RAPS delays activation if the identifier and the signal strength from the currently active cell-tower indicates that previous activation attempts at locations with comparable identifier and signal strength information failed frequently. Finally, RAPS delays GPS activation if it learns, over Bluetooth, of a more recent position fix completed by an opportunistic contact.

We have implemented these techniques in a complete RAPS prototype on the Nokia N95 smartphones and have experimented with it on our university campus. Our evaluation reveals that RAPS has over 3.8x longer lifetime than a scheme in which GPS is always on, and about 1.9x longer lifetime than a periodic GPS scheme with comparable error rate. We also break down the contributions of each technique in RAPS towards these performance gains. Finally, we demonstrate that RAPS can be easily adapted to work atop a WiFi-based positioning system, WPS [22]. In general, our evaluation is encouraging and suggests that RAPS can obtain substantial energy savings without sacrificing the accuracy of periodic GPS.

2. PROBLEM

In this section, we first present our observations on the inaccuracy of GPS readings taken from a smartphone. We demonstrate that, in urban environments and especially for pedestrian use, there is significant uncertainty in GPS-reported positions, and location-aware applications must adapt to this using application-specific methods. We then explore whether, by periodically duty-cycling GPS readings, it is possible to achieve significant energy savings without sacrificing accuracy. This motivates the problem we tackle in this paper, that of designing an energy-efficient rate-adaptive duty-cycling for GPS based positioning.

Figure 3 plots a part of a GPS trace collected for a period of 1 week using a smartphone that continuously logged GPS positions every 1 second. To obtain this figure, we wrote a small program that records raw GPS readings using the Location API provided by the Android OS. Later, we used the Google Maps API v3 to plot the points in the figure. An analysis of this trace reveals an interesting result. The “ground truth” path is labeled ‘A’ in Figure 3, but the trace contains a phantom path, labeled ‘B’ in the figure, parallel to the ground truth path, which was never taken during trace collec-

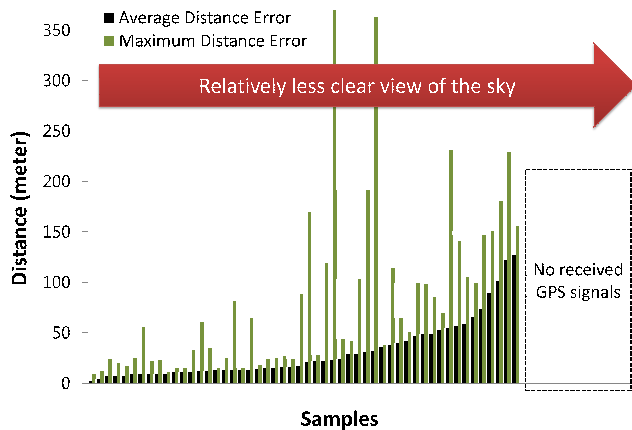


Figure 4: GPS ground truth verification: Distance error of GPS reading vs. GPS ground truth

tion. (That the phantom path aligns with a walkway on our campus is, we believe, accidental: to the best of our knowledge, the Google Maps API does not do any map-matching when provided with the GPS coordinates.) Moreover, the trace had position logs that were much further south on campus, labeled as ‘C’, which were never visited during trace collection.

While GPS inaccuracy for automotive navigation applications in urban canyons is well known, our finding suggests that such inaccuracies might be much more prevalent, and may affect pedestrian or outdoor use of GPS even on campuses or moderately populated neighborhoods with modest-height multi-story structures. To understand the prevalence of this problem we decided to further investigate the accuracy of GPS on smartphones.

We collected GPS position readings from 70 different locations in the greater Los Angeles area including the USC campus, West Los Angeles, Glendale, Korea Town, and other areas. These locations were selected to cover a variety of operating conditions such as wide-open spaces, areas with trees, streets between buildings, indoors near windows, inside vehicles, and so on. We then compared the error in collected positions relative to “ground truth” positions obtained by manually marking each location on Google Maps and extracting the corresponding coordinates. While Google Maps itself might be inaccurate, many location-aware applications do map matching against Google Maps (or other online map services), and our errors are representative of the errors those applications would see.

Figure 4 presents the result of this experiment. Black and gray bars represent the average and maximum error, respectively, for each tested location, and the figure is sorted by increasing average error. Generally speaking, we have found that this ordering corresponds to decreasing visibility of the sky, from left to right. (Our observation is clearly qualitative: without access to the satellites visible to the GPS receiver at each location, clearly we cannot quantify visibility.) With good visibility, GPS errors are relatively low, but still range from 5 to 35 meters and occasionally go up to as high as 120 meters. As we move to the right, the average error increases up to around hundred meters, with maximum error at times of over 300 meters. Finally, there were several places where GPS was completely unavailable. In fact, in the one-week GPS trace from which Figure 3 was drawn, GPS was available for only 11.2% of the time. This result is consistent with other reports [1] of low GPS availability on human-carried devices. This discussion confirms our observation in Figure 3: GPS may provide inaccurate

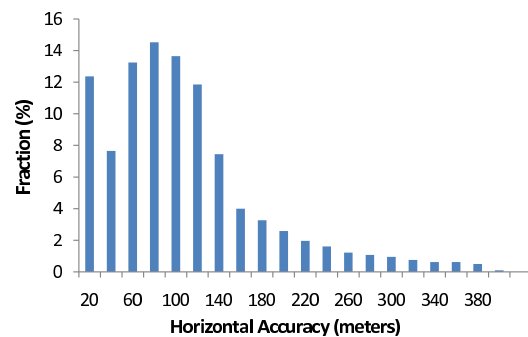


Figure 5: Distribution of horizontal accuracy reported by GPS on N95 smartphone

rate positioning with average errors as high as 100 meters or more across a wide range of urban locations.

Where do these errors come from? A GPS receiver requires signals from 4 satellites, along with their ephemeris data, to calculate the time and position of the receiver in latitude, longitude, and altitude. As long as the receiver has a position fix using complete signals from at least 4 satellites, these errors are known to be within 15 meters. The sources of error in this case are ionospheric effects (± 5 meters), shifts in the satellite orbits (± 2.5 meters), satellite clock errors (± 2 meters), multipath effects (± 1 meters), tropospheric effects (± 0.5 meters), and calculation and rounding errors (± 1 meters) [21]. However, when GPS receivers do not have complete signals or complete ephemeris data from all 4 satellites, then the errors can increase significantly. For example, if signals from only 3 satellites are available, a GPS receiver must guess its location by assuming either perfect time sync or mean sea-level altitude.

When it guesses a position, a GPS receiver provides accuracy estimates. Figure 5, which plots the distribution of the horizontal accuracy estimates obtained from the GPS on the Nokia N95 smartphone, shows that the GPS itself often reports low confidence in its position results. Receivers use proprietary methods for computing accuracy estimates and for correcting errors when less than 4 satellites are visible, so it is difficult to infer the exact cause for the inaccuracy. However, what we do know is that there may exist significant inaccuracy in the position reported by these devices. (So far, we have reported GPS accuracy only on a single platform. In Section 4.4, we demonstrate that this problem may be pervasive: similar inaccuracies exist in at least three other platforms, running different phone operating systems. Moreover, we show that the use of Assisted-GPS does not improve accuracy, but can reduce the time to first-fix.)

Moreover, smartphone GPS receivers are more likely to be inaccurate than, say, automotive GPS systems. Smartphones have smaller antennae, are often carried in clothing or bags, are often indoors, and frequently powered off. As such, location-aware smartphone applications will have to have application-specific ways to deal with these inaccuracies: techniques such as map-matching [4] and map-snapping are, by now, well-established methods for addressing this problem. On the other hand, smartphone GPS use is known to be an energy drain, and many power users likely manually activate and de-activate GPS to conserve battery. In this paper, we ask the question: can we cleverly activate GPS only when necessary, and sacrifice a little accuracy (since location-aware applications will have to deal with this loss of accuracy anyway), in exchange for significant reductions in energy usage by GPS?

The simplest approach to trade-off accuracy for energy in GPS-

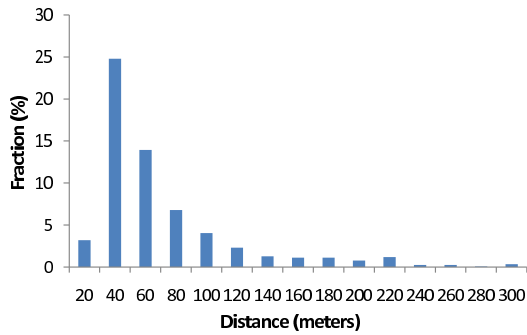


Figure 6: Distribution of distances between GPS updates for periodic GPS with 180 sec update interval.

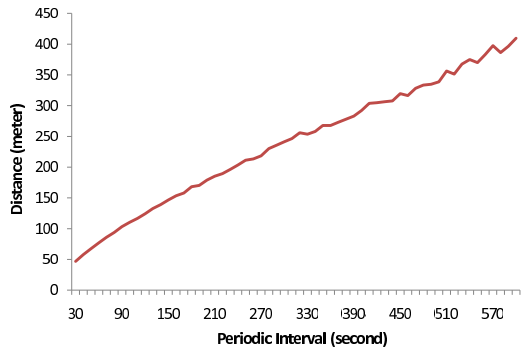


Figure 7: Update interval vs. average distance between GPS updates for periodic GPS.

based locations services is to periodically duty-cycle GPS. This approach is commonly used in many real-world applications. For example, the Urban Tomography application [13] activates GPS every 7 minutes and tags the videos taken by the user with the position information closest in time. Similarly, PEIR [18] collects GPS readings approximately every 30 seconds. However, the key challenge in this periodic GPS duty-cycling is to decide the time interval at which to turn on and off GPS. If the GPS is activated too often, it will waste a lot of energy when the phone is mostly stationary. If the GPS is activated too infrequently, accuracy will suffer.

To illustrate these tradeoffs with periodically duty-cycled GPS, consider Figure 6. To obtain this figure, we conducted an experiment in which, for a specific choice of period (3 mins), we measured the position uncertainty (the reported distance between two GPS readings in our trace taken at 3 minute intervals). The uncertainty is less than 40 meters for 28% of time, but may often exceed 100 meters and go as high as 300 meters. Clearly, this uncertainty depends upon the movement pattern of the user during the experiment. However, because periodic duty-cycles are oblivious to the actual mobility, this approach can incur high error.

Moreover, there exists no single satisfactory value for the duty-cycle period, as shown in Figure 7. Figure 7 plots the average distance between two consecutive position updates as a function of the periodic duty-cycling interval. This figure was generated from an experiment in which we collected GPS reading every 5 seconds for 48 hours. We simulated the different intervals by appropriately sub-sampling the experimental trace. The figure shows that as the interval increases enabling lower energy usage, the uncertainty increases linearly, with no obvious sweet spot.

Another approach is to tune the GPS duty-cycling period to an

application-specified average error bound. In Figure 7 if that an application could tolerate a 100m uncertainty, its GPS could be activated every 80 seconds. However, this may not help reduce energy significantly. GPS can take up to several tens of seconds after activation to get a position fix, and many GPS receivers have a power-down delay of around 30 seconds ([12]). So, an 80 seconds duty-cycling interval will likely result in minimal savings. To be more precise, if we assume an ideal case where first-time-to-fix is always 6 seconds, then the GPS would be on for 36 sec (6 + 30sec power off delay) every 80 seconds, which translates into spending 600 Joule per hour. This means that on N95 smartphones with 1200mAh batteries, the battery will last for less than 27 hours even when GPS duty-cycling is the only task on the phone. In practice, however, GPS times out when satellite views are unavailable. If this happens as frequently as it did in our traces (88.8% of the GPS attempts), then the battery will last less than 13 hours. Even at this cost, the errors are non-negligible.

In summary, we have made two important observations in this section: 1) GPS is generally less accurate in urban areas, and 2) periodic duty-cycling with a fixed interval can introduce significant, potentially unbounded, error without necessarily providing significant energy benefits. This motivates the need for a positioning system that provides position information with reasonable error while expending minimal energy. In the rest of the paper, we present RAPS, a rate-adaptive positioning system for smartphone applications.

3. RATE-ADAPTIVE POSITIONING SYSTEM

In this section, we start by describing the goal of RAPS, a rate-adaptive positioning system for smartphone applications, and discuss how it is designed to meet that goal. We then delve into the details of the individual components of RAPS: 1) user movement detection using a duty-cycled accelerometer, 2) velocity and uncertainty estimation using space-time history, 3) GPS unavailability detection using celltower-RSS blacklisting, and 4) position uncertainty reduction using Bluetooth. We conclude with a discussion of RAPS’s limitations.

3.1 Overview

RAPS is designed for smartphone applications that require position information for location-based context-aware services. It is based on the observation that GPS is generally less accurate in urban areas, so it suffices to turn on GPS only as often as necessary to achieve this accuracy. Thus the goal of RAPS is to reduce the amount of energy spent by the positioning system while still providing sufficiently accurate position information.

To achieve this goal, RAPS uses a collection of techniques to cleverly determine when to turn on GPS, and when not to. First, it uses a duty-cycled accelerometer to efficiently estimate user movement. RAPS detects whether the user is moving or not, and also measures the *activity ratio*, the fraction of time that the user is in motion between two position updates. The movement detection is used to prevent RAPS from activating GPS when the user has been stationary. The activity ratio is used to estimate the current velocity based on historical correlations between velocity and activity (see below); this lets RAPS activate GPS only if the estimated uncertainty in position exceeds the accuracy threshold.

Second, RAPS stores the space-time history of user movements (where the user has been and at what time) to estimate when to activate GPS. Whenever RAPS gets a new position update, it calculates the average velocity relative to the previous position, and associates this velocity and the recent activity ratio with the previous space-time coordinate. It uses averages of the velocity and

activity ratio to estimate, during a subsequent visit to the same location at the same time-of-day, the likely user velocity. This in turn enables RAPS to estimate positioning uncertainty, allowing it to selectively activate GPS.

Third, RAPS employs *celltower-RSS blacklisting* to detect GPS unavailability (e.g., indoors) and avoids turning on GPS in these places. Whenever it succeeds or fails in obtaining new position update, it records the current celltower ID and the received signal strength (RSS) information and associates with that success or failure. Then, when it determines that it is time to activate GPS, it checks the celltower-RSS table for the historical probability of GPS availability and defers GPS activation if it believes that GPS is not likely to be available, thus avoiding unnecessary energy usage.

Finally, RAPS utilizes Bluetooth communication to reduce position uncertainty among neighboring devices. Whenever it receives a new position update, it broadcasts this information to Bluetooth peers so that they can update their position without activating GPS themselves. If a device receives a position update from a peer that has greater uncertainty than its own estimate, it replies with its more accurate information. Eventually, all devices in the neighborhood synchronize to the position information with least uncertainty.

All the techniques that RAPS uses can be realized on the current generation of smartphones. As we discuss later, we have implemented a complete RAPS prototype on a smartphone, the Nokia N95. In the following subsections, we discuss each of these techniques in more detail.

3.2 Movement

Many, if not all, modern smartphones are equipped with an accelerometer. There is a significant literature on the effectiveness of using an accelerometer to detect whether the user is moving or not [12, 27, 3]. Others have explored techniques to estimate velocity [15] using the accelerometer. Given that we are interested in estimating user speed to minimize GPS activation, we too use the accelerometer to detect motion. However, our use of the accelerometer differs from the prior literature in two important ways. First, unlike most prior work that either uses the accelerometer as a binary sensor to detect movement or non-movement, or restricts it to detect pedestrians, we use the accelerometer to measure the *activity ratio*, defined as the fraction of a given time window during which the user is in motion. We then use this activity ratio along with the history of velocity information (Section 3.3) to estimate the current velocity of the user. Second, given the high power consumption of the accelerometer, we carefully duty-cycle it to save energy without significantly sacrificing accuracy. We describe each of these two aspects below in more detail.

From a continuous sequence of accelerometer samples, it is possible to detect whether a user is stationary or not using an onset detection technique [20]. This technique can reliably detect the start and duration of significant user activity. It does this by maintaining running estimates of the signal envelope (dynamic upper and lower bounds of the signal) and compares these against the noise mean (which is 1g, or gravity, in our case) and the standard deviation. It is designed such that the onset is detected as soon as possible after the beginning of a period of significant activity. Moreover, the end of the period is declared conservatively, i.e., only after the envelope of the acceleration magnitude has decayed below and has then not exceeded one standard deviation from the noise level, indicating quiescence. Figure 8 illustrates the behavior of the onset detector where the acceleration signal and its envelope of user movement are plotted in solid gray lines along with the onset detection result in dotted black line. RAPS uses this onset detection method and calculates the *activity ratio* as the fraction of the time during a mea-

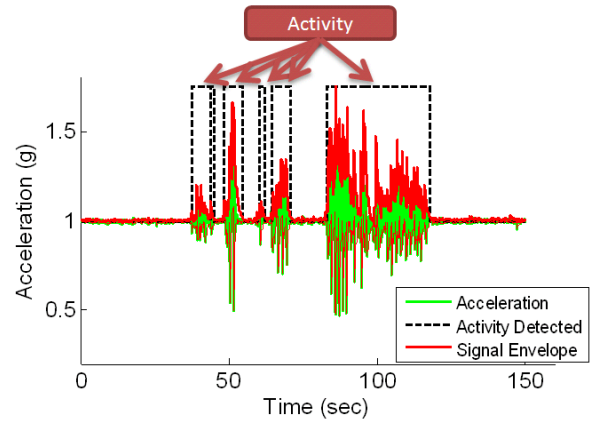


Figure 8: Example behavior of the onset detector where the acceleration signal and its envelope of user movement

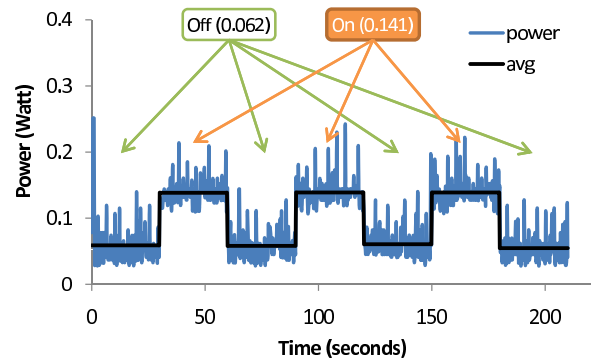


Figure 9: Power consumption of accelerometer sensor on N95 phone

surement period that the user was active. This value is later used to estimate the user velocity, which we describe in Section 3.3.

However, RAPS cannot use the accelerometer continuously because measuring acceleration can incur significant energy usage. Figure 9 plots the power usage from our measurement on an N95 smartphone when the accelerometer was turned on and off at 30 second intervals. This measurement shows that the accelerometer consumes around 0.08 Watt, which means that turning on the accelerometer for 5 minutes consumes more energy than activating GPS for 1 minute. In other words, if the accelerometer is always on and GPS was turned on every 5 minutes on average, then over 50% of the energy would be used by the accelerometer, and we might as well turn on GPS every 2.5 minutes instead.

Given that energy-efficiency is a major goal of our design, we argue that operating the accelerometer continuously is not a viable option. RAPS duty-cycles the accelerometer carefully, using a duty-cycling parameter derived empirically, as described below.

Our method for deriving a good duty-cycling parameter involved first collecting continuous acceleration measurements for five different kinds of common human activities: being stationary, frequent walking and stopping, fast walking, driving in a car, and milling about in a coffee shop. Although these five activities do not cover every possible human movement, our intention is to illustrate a plausible methodology for selecting a duty-cycling parameter. It is possible to optimize the parameter selection method, and we have left that to future work. For each activity, we obtained

a 5-minute long acceleration trace. For each trace, we calculated the reference activity ratio for the always-on accelerometer case using the onset detector. Then, we performed offline analysis that emulated duty-cycling on these logs for various ON and OFF periods, and calculated the error in the activity ratio with respect to the reference always-on case. The parameters we emulated range from 5% ~ 50% for the duty-cycle, and 2 ~ 15 seconds for the ON period. Our goal was to select the lowest duty-cycle parameter which resulted in less than 10% average error across all five activities.

Figure 10 plots the average error (averaged over five activity logs) as a function of the duty-cycle with various ON periods. From this figure, the approximate knee of the bounding curves corresponds to a duty-cycle parameter of 12.5% with 2 and 14 seconds ON/OFF periods; this incurs 3% error on average and less than 10% error across all the activities. Intuitively, RAPS is able to detect human activity which has a timescale larger than multiples of 16 seconds, by simply turning on the accelerometer for a short fraction of that time window. In doing so, the accelerometer consumes 1/8 of its original power, which is 0.01 Watt, which significantly alters the balance of power usage between GPS and the accelerometer.

Could we have used a similar accelerometer duty cycling technique to estimate user speed (and therefore distance), rather than just activity? In theory, distance is a simple integral of velocity, which in turn is another integral of acceleration once we know the exact orientation of the phone.

To determine this, we used the continuous accelerometer traces obtained above, and computed the horizontal displacement of the user using a technique proposed in [15]. To do this, we needed to estimate the gravity vector, which can be obtained by performing a running average of accelerometer samples (when the magnitude of the acceleration vector is close to 1g, the gravity). This gravity vector estimate, in turn, enables estimation of the vertical component and the magnitude of the horizontal component of the user's motion regardless of the orientation of the three-axis accelerometer. Using this idea, we calculate the distance that the user has moved as follows:

$$\begin{aligned} \mathbf{d} &= \mathbf{a} - \mathbf{g} && \text{: user acceleration minus gravity} \\ \mathbf{p} &= \left(\frac{\mathbf{d} \cdot \mathbf{g}}{\mathbf{g} \cdot \mathbf{g}} \right) \mathbf{g} && \text{: vertical component of } \mathbf{d} \\ \mathbf{h} &= \mathbf{d} - \mathbf{p} && \text{: horizontal component of } \mathbf{d} \\ \mathbf{v} &= \int \mathbf{h} \cdot dt && \text{: velocity vector} \\ \text{distance} &= \left\| \int \mathbf{v} \cdot dt \right\| \end{aligned}$$

We conducted extensive experiments that revealed that this calculation is roughly correct when the orientation of the phone does not change too frequently and the acceleration is greater than the noise level during the period of movement. However, it often overestimates distance when the user is handling the phone in her hands, and underestimates distance when sitting quietly on a cushioned car seat.

Furthermore, using a method similar to that described above, we found that a duty cycle of 50% or more is required to estimate the distance moved by the user within 10% error of the always-on case. This makes it a less attractive option for us, so RAPS uses only the *activity ratio* whenever we have sufficient history to perform velocity estimation, and falls back to using both the *distance estimation* and the *activity ratio* only if there is insufficient history. We describe this in more detail later when we explain our RAPS algorithm in full.

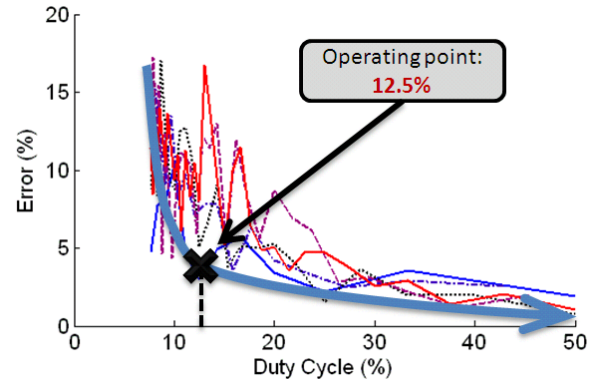


Figure 10: Accelerometer duty-cycle parameter selection: we select 12.5% as our accelerometer duty-cycle which incurs 3% error on average, and less than 10% error for five different activity modes (stationary, walks & stops, fast walking, driving in a car, slow walking with conversation)

3.3 Space-Time History

A key component of RAPS is the space-time table that records the past history of user movements. Specifically, RAPS maintains the average user velocity and the average estimated activity ratio (Section 3.2) associated with each space-time unit in a 3-dimensional (latitude, longitude, time-of-day) coordinate system. Whenever RAPS needs to decide when to activate GPS, it looks up the history of average user velocity and activity ratio associated with the current position and time, and then calculates the current position uncertainty based on the estimated current velocity and the activity ratio.

The intuition behind this idea is that there is, often, consistency in user behavior at a given point in the space/time coordinate system. For example, a person is, in general, more or less stationary for extended durations when at the office during the work hours or at home overnight. A person is likely to move at walking speed at places that she usually walks, and also likely to move fast on roads that she usually drives in a vehicle. Conversely, a person is not likely to be stationary in the middle of a street, and also not likely to walk on a freeway. This is the intuition that lies behind our use of space-time coordinates to associate user movement history, and to estimate speed. (One might obtain more precise estimates by maintaining a week-long history, and using day-and-time-of-week to index into this history: we have left that to future work).

RAPS updates and uses this space-time history as follows. First, for scaling reasons, it quantizes both the space and time dimensions. The space coordinates are quantized into a 2-D grid by rounding latitude and longitude values to 3 decimal places; 0.001' x 0.001' grid box. In the Southern California area, this approximately corresponds to a 92.2m x 111.2m square area. In other locations, RAPS quantizes the history so that the size corresponds to about 100m, the lower bound on the position accuracy we target. The time-of-day dimension is quantized into bins of size 30 mins each. The quantized space-time coordinate system is illustrated in Figure 11.

Next, RAPS associates with each grid box in this quantized coordinate system two quantities: a history of average velocity and the average activity ratio seen in the box. More precisely, whenever it receives a new position P_B at time T_B , it calculates a velocity V_A as $\frac{\text{distance}_{A \rightarrow B}}{\text{time interval}_{A \rightarrow B}}$ relative to its previous known position P_A

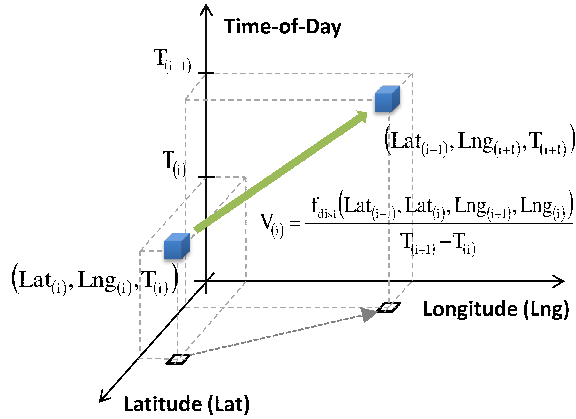


Figure 11: Conceptual view of the 3-D space-time coordinate system used for maintaining history of user mobility.

at time T_A , and uses V_A to update the average velocity \bar{V}_A associated with the space-time grid box $S_{(P_A, T_A)}$. In a similar manner, it also reads the activity ratio during this time interval updates the average activity ratio \bar{R}_A associated with that space-time grid box.

This space-time history is used to estimate uncertainty as follows. Suppose that the user was last known to be at (P_A, T_A) (P_A is the last recorded position fix). To calculate the uncertainty $U(t)$ at some time $t > T_A$, RAPS uses the following equations:

$$V(t) = \bar{V}_A * \frac{R(t)}{\bar{R}_A} \quad (1)$$

$$U(t) = V(t) * (t - T_A) + U_A \quad (2)$$

where \bar{V}_A and \bar{R}_A are the average velocity and activity ratio associated with (P_A, T_A) , and U_A is the last uncertainty of position P_A at time T_A , $R(t)$ is the current activity ratio, and $V(t)$ is the estimate of the current velocity. If RAPS does not find information associated with (P_A, T_A) , it falls back to using only the position information and finds an entry that matches P_A only. This uncertainty calculation is based on the assumption that the average velocity in a space-time coordinate location is proportional to the average activity ratio in that location; i.e. $V_A \propto R_A$.

Intuitively, we use the activity ratio as a surrogate for velocity. This can lead to better estimates of GPS activation times than simply using history or detecting movement, and can have lower energy (as we have discussed earlier) than estimating velocity by keeping the accelerometer always on.

To understand how RAPS works in practice, consider the following example. For illustration, assume that the decisions are made after every unit of time. If a user moves into a position A, is stationary for 4 units of time and moves out of position A in the next unit of time at a velocity of 10m/s and activity ratio of 0.5. Then the velocity and activity ratio associated with position A during this period is $\{0, 0, 0, 0, 10\}$ and $\{0, 0, 0, 0, 0.5\}$ respectively, resulting in average values of 2m/s and 0.1 stored in the history. Later on, when the user revisits position A, there are two possible cases; if the user is stationary, the current activity ratio $R(t)$ will be zero, and thus the estimated velocity $V(t)$ is zero by Eq. 1. When the user is moving out of position A with $R(t)$ close to 0.5, $V(t)$ becomes $2 * 0.5 / 0.1 = 10\text{m/s}$ thereby correctly estimating the user velocity and thus the uncertainty. This approach allows RAPS to cheaply estimate user movement, activating GPS only when user movement may have exceeded the accuracy bound.

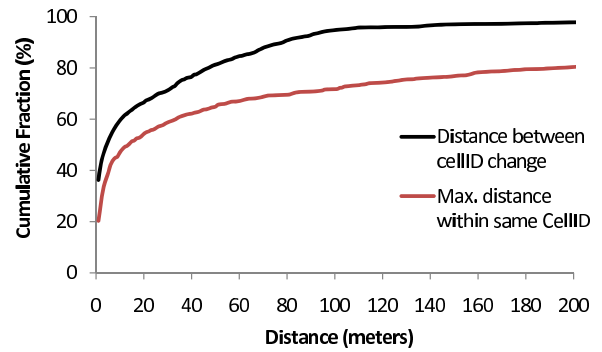


Figure 12: CDF of the distance between two consecutive GPS locations 5 seconds apart; with and without a GSM *cellid* change.

3.4 Celltower-RSS Blacklisting

Prior work has proposed the use of GSM signatures, which include information about visible cell-towers and their RSS (Received Signal Strength), to detect large scale movement [6] or recognize mobility modes [24]. In this section, we investigate whether or not such signatures can be used to detect motion, so we can regulate how often GPS needs to be activated. GSM data comes for free – as long as the phone is on and has an active service subscription, GSM data can be retrieved without incurring additional energy cost. However, most prior work on GSM-based localization has assumed sufficient knowledge of the location and RSS information from all visible cell towers. In practice, this information is often not available to third party application developers on many platforms. At least for both Symbian-OS on Nokia-N95 phone and Android on G1 phone, only one cell tower information is visible at a time, and there is no information about the location of the tower. This is one reason why much of the prior work on GSM localization has been evaluated only in simulation.

To determine the feasibility of using cell-tower information to adaptively activate GPS, we first empirically examined whether information from a single cell-tower can reliably detect user movement. Figure 12 plots the cumulative distribution of the distance between two consecutive GPS locations when there was a change in *cellid*. It also plots the CDF of the maximum distance between two positions within the same *cellid*. Each GPS update was obtained every 5 seconds.

One observation from the figure is that, 58.3% of the time, a less than 10m difference in GPS positions can result in a change in *cellid*. The rest of the time, when a *cellid* changes, the difference in GPS positions can be uniformly anywhere between 10m and 150m. Conversely, the maximum distance between two positions within same *cellid* is greater than 100m for 28.2% of the time (CDF of 71.8% within 100m). Thus, these results imply that simply using the *cellid* itself provides insufficient information about whether the user has moved a significant distance or not.

Our next step was to see whether the signal strength difference within a *cellid* can be an indicator of change in position. Figure 13 plots the maximum, average, and minimum distance between two GPS updates with different RSS values within same *cellid*. Although the average distance plot shows a rough trend in which an increasing RSS difference correlates with increasing distance, the variance of the distance is too high to be used as a measure of distance. This variability is not too surprising given the complexity of the urban environment we live in. These results together show that using the identifier of, and the signal strength from, a single

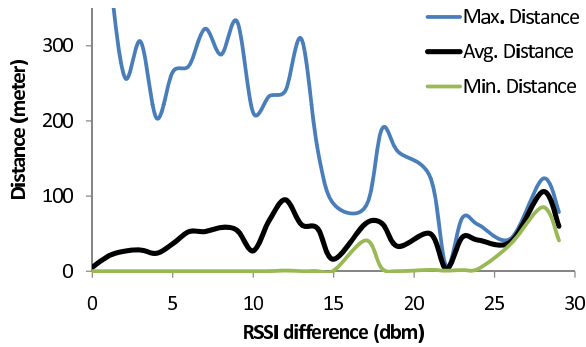


Figure 13: RSS difference vs. distance, between two positions withing same *cellId*

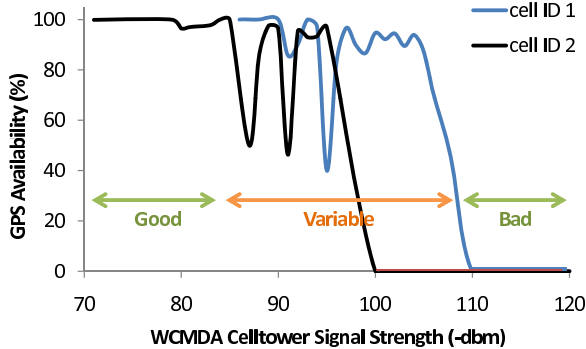


Figure 14: GPS availability probability as a function of signal strength for two different *cellId*'s

cell-tower cannot reliably identify user movement, at least in urban areas with irregular cell deployment.

Since we are merely interested in determining whether and when GPS should be activated, we considered the following question: instead of using cell towers to detect motion, can we directly detect whether users are in an environment (e.g., indoors) where it would be futile to turn on GPS? To understand this question, we first studied the relationship between GSM *cellId* and their RSS values, and GPS availability. We define GPS availability as the fraction of attempts for which it is possible, at a given location, to get a position fix within 2 minutes starting from power-off state. This 2-minute time-to-first fix timeout follows from GPS receiver properties: a GPS receiver requires information from 4 satellites to get a position fix and receiving full ephemeris data for a satellite takes 30 seconds each.

For a given *cellId*, we expected to see some correlation between RSS and GPS availability, since GSM signals indoors are heavily attenuated. Figure 14 plots the GPS availability probability as a function of signal strength for two different *cellId*'s. We note several features. First, there is a *good* RSS range at which GPS is available most of the time, and also a *bad* RSS range at which GPS is mostly unavailable. Examples of extremely bad locations include elevators and underground parking lots. Second, there is a *variable* RSS region where there exists a high variability in the GPS availability. In this region, it is difficult to derive a strong relationship between RSS value and availability, suggesting that GPS availability must be estimated probabilistically. Finally, note that these region boundaries differ for different *cellId*s.

From these results, it is clear that there may not exist a universal way to predict GPS availability from cell-tower information.

However, we believe that maintaining a history of GPS availability per-*cellId* can help accurately predict whether a GPS fix is likely to be successful, given a specific *cellId* and the current RSS reading. Indeed, RAPS uses exactly this idea. Whenever a GPS reading is successfully obtained or the request has timed-out, it stores this information in a celltower-RSS blacklist. To do this, it reads the current *cellId* and RSS values and increments the success or fail counter for the corresponding *cellId* in the list. Then the GPS availability for that particular *cellId* is simply the fraction of successful attempts. It also updates one of two RSS threshold values associated to each *cellId*: RSS_{Good_Thresh} is the best RSS value that resulted in GPS update failure, and RSS_{Bad_Thresh} is the worst RSS value that resulted in successful GPS update. The list is called blacklist because RAPS uses it to maintain locations at which GPS activation is more likely to fail, so it can delay activation and save energy. To do this, RAPS uses an eviction policy that discards the *cellId* entry with the best GPS availability when the list is full. RAPS conservatively assumes GPS availability if no prior history exists in the blacklist.

Then, when the position uncertainty indicates that it is time to get a GPS update, RAPS checks the blacklist for the GPS availability predicted for the current *cellId*-RSS reading. If the list indicates that the user is in the *good* or *variable* region, RAPS turns on GPS with a probability equal to the historically-observed availability. If it finds itself in the *bad* region, it waits until either there is a change in GSM data (either *cellId* or RSS) or until a maximum timeout time has passed.

In summary, the goal of *celltower-RSS blacklisting* is to estimate, based on a history of GPS attempts at locations with specific *cellId* and RSS readings, locations where turning on GPS is unlikely to provide a position fix. Our method can be robust to cell-tower density, unlike schemes that attempt to detect mobility using *cellId* information. However, its performance is, of course, heavily dependent on user motion, on the surrounding environment, and on cell-tower placement. We quantify the performance benefits of this approach in Section 4.

3.5 Bluetooth-based Position Synchronization

In this section, we discuss the use of Bluetooth communication to synchronize position information between neighboring devices. This enables phones to save energy by reducing the number of GPS activations. Consider a scenario where there are two smartphones, *A* and *B*, in direct communication range of each other. If *A* has recently activated GPS and received a position fix, then it is possible for *B* to get the position information from its neighboring node using peer-to-peer communication without the need to activate GPS itself. If *B* happens to have better position (lower uncertainty) than *A*, then it can immediately notify *A* of that fact and *A* can use *B*'s position to update itself. The goal is to opportunistically synchronize the position information with Bluetooth contacts to lower overall uncertainty and reduce energy usage.

Bluetooth is a good candidate for such a position synchronization for several reasons. First, the communication range of Bluetooth in smartphones is in general less than 10 meters. Thus, the added uncertainty in position at the receiving node is less than 10 meters relative to the transmitting node. If the transmitting node transmits its uncertainty along with its last position, then the uncertainty at the receiving node can conservatively be set as the received uncertainty plus the max communication range, 10 meters.

Second, the energy cost of Bluetooth is considerably less than that of GPS. Figure 15 plots the actual power consumption on smartphones for a master-slave pair while running our implementation of the position synchronization protocol (which we describe be-

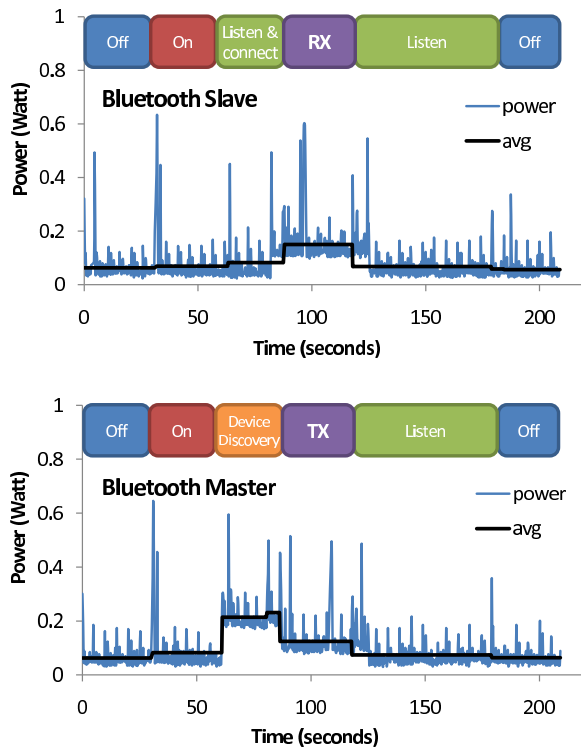


Figure 15: Power consumption of Bluetooth on N95 phone

low). Using Bluetooth, most of the power is consumed on the master node while discovering the nearby devices and their services. This contributes around 0.15 Watt of power relative to idle state for around 15 ~ 20 seconds. The actual transmissions ($\sim 0.07W$) and receptions ($\sim 0.09W$) take less than one second, and listening for and detecting incoming discovery requests consumes almost negligible power, less than 0.01W relative to idle state. During one synchronization cycle depicted in the figure, the slave node used 0.09J, and the master node used 3.07J, averaging 1.58J per node. For comparison, if a GPS receiver was turned on and stayed on for 60 seconds to get a position fix, it would have spent around $0.37W * 60\text{sec} = 22.2J$. Thus, if a single Bluetooth exchange could avoid GPS activation at one of the two nodes, about 43% reduction in energy usage (overall, across both nodes) would be achieved.

Furthermore, due to the master-slave communication architecture of Bluetooth technology, the energy cost of device discovery is amortized over the number of nodes in the neighborhood. For example, if there are 5 nodes in the area and one node acts as a master while others as slaves, then, if a GPS activation at one node, followed by Bluetooth communication can avoid GPS activations at the other 4 nodes, the energy cost for the two cases is $5 * 22.2 = 111J$ versus $22.2 + 3.07 + 4 * 0.9 = 28.87J$ resulting in a 74% reduction in energy.

Finally, Bluetooth is available on almost all mobile phones, not just smartphones, and many users enable Bluetooth for frequent everyday use. We see this likely to persist, with the passage of legislation requiring drivers to use hands-free headsets, many of which use Bluetooth. Thus, we envision a future where many smartphones run our RAPS software to energy-efficiently get their position from neighboring devices. Furthermore, it is also possible to imagine future Bluetooth access points that provide accurate location beacons for fine-grained infrastructure-based location services at pub-

lic places and places where GPS is unavailable (i.e. positioning in tunnels, subways, or for advertisements in shopping malls [2]).

Our Bluetooth-based Position Synchronization protocol (BPS), which we have prototyped on a Nokia N95, works as follows. By default, every node becomes a Bluetooth slave node and stays in idle listening state awaiting incoming device discovery requests. Once a node decides to transmit its position and uncertainty information, it assumes the role of a Bluetooth master and scans its neighborhood for device and service discovery. If any neighboring slave nodes exist, the master connects to all of them, and broadcasts its position information. Each slave, upon reception of the position information from the master, compares its uncertainty and updates its own position if the received position has lower uncertainty. If the received uncertainty is higher than its own, the slave replies to the master with better position and uncertainty values. When the master receives a reply from the slave, it also compares the uncertainty, updates its own position if the received position has lower uncertainty, and rebroadcasts that information. At this point, the uncertainty value of all connected devices are synchronized.

Finally, recall that when a node receives a better position estimate from another node, it adds a 10m uncertainty (the nominal Bluetooth range). If it re-propagates this estimate to another neighbor, that neighbor will conservatively add 10m again to the estimate. However, this uncertainty propagation is not serious: a node accepts a neighbor's estimate only if that estimate, plus the 10m uncertainty, is better than its own estimate. Thus, at every step, Bluetooth synchronization results in improved accuracy estimates.

To determine when to transmit, BPS employs two simple rules. A node decides to become a master and initiate synchronization when either 1) it has received a fresh GPS position update, or 2) when a GPS update was requested but was unavailable for a specified interval. In Section 4.1, we present a result showing that BPS can be effective in reducing the number of required GPS activations.

3.6 Discussion

Three details of RAPS are worth mentioning briefly. First, the user space-time history and the celltower-RSS blacklist must be populated for RAPS to work efficiently. While this data is being populated, RAPS aggressively activates GPS, and frequently measures celltower-RSS information. Second, our velocity estimation based on activity ratio can be misled by handset activity not related to human motion. For users who continuously fiddle with their mobile phone while sitting in one location, for example, RAPS will record an inflated activity ratio with respect to velocity. Subsequent velocity estimates at that location can underestimate the true velocity. Finally, accelerometers on smartphones may need a one-time per-device calibration of the offset and scaling before running RAPS. Before our experiments, we have manually calibrated the accelerometer offline.

Our Bluetooth-based position synchronization requires user cooperation. This protocol raises privacy and security concerns, which we have not considered in this paper: our objective was merely to explore the energy-reduction potential of Bluetooth synchronization. It would also require incentivizing users appropriately to share their position with Bluetooth-neighbors, a topic beyond the scope of this paper. We observe, however that, given the way our protocol works, each node is roughly equally likely to avoid GPS activations as a result of Bluetooth-based synchronization, so there may exist a natural incentive to participate.

	<i>History</i>	<i>Accel.</i>	<i>C-R Blacklist</i>	<i>BPS</i>
<i>RAPS</i>	○	○	○	○
<i>RAPS-B</i>	○	○	○	×
<i>RAPS-BC</i>	○	○	×	×
<i>RAPS-BCA</i>	○	×	×	×
<i>Always-On</i>	Periodic GPS with 20 seconds interval			
<i>Periodic</i>	Periodic GPS with 180 seconds interval			

Table 1: Six different schemes used for evaluation. ○-mark and ×-mark represent enabled and disabled, respectively.

4. EVALUATION

In this section, we present results from real world experiments using our prototype implementation of RAPS on a modern smartphone. We have implemented RAPS in Symbian C++ for the Symbian S60 3rd FP1 devices. In all our experiments, we have used the Nokia N95-3 smartphone, which has GPS, a built-in accelerometer, Bluetooth, WiFi and 3G/EDGE interfaces, and a 2GB micro-SD card. We have experimented with our RAPS implementation in and around the USC campus.

Our evaluation is designed to answer four questions:

- How much, in absolute terms, does RAPS increase lifetime, and how much do each of its components contribute to this improvement?
- Could a periodic GPS activation strategy have performed just as well?
- Is RAPS flexible enough to be incorporated with a WiFi-based positioning system?
- Are the GPS errors that motivated the design of RAPS pervasive?

We discuss these questions in the following sub-sections.

4.1 Quantifying the Benefits of RAPS and its Components

In this sub-section, we present two kinds of evaluation. First, we demonstrate the energy savings achieved by RAPS, relative to an always-on GPS schemes. Second, we quantify the benefits of each of our techniques: mobility detection using a duty-cycled accelerometer, celltower-RSS blacklisting, and Bluetooth-based position synchronization.

Methodology. To conduct this evaluation, we programmed six smartphones to use different GPS activation strategies, as discussed below. All of these six phones were placed in a single bag, and carried by one of the authors for almost two entire days. During a single run of the experiment, the experimenter went about his normal daily activities, mostly inside and around the USC campus. These activities involved a variety of mobility modes, including occasional trips in a car. There was no intentional repetition nor any artificial movement. Moreover, the phones were *not* used for any other activities, such as browsing, or calling. Figure 16 shows the locations visited by the experimenter during a run of the experiment; the geographic spread of the locations is over 3 miles. We conducted several runs, and report the result of one of these; other runs show qualitatively similar results.

Of the six phones, two ran RAPS with all of its components enabled, while three phones ran a variant of RAPS each with a different combination of the heuristics enabled; mobility detection using a duty-cycled accelerometer, celltower-RSS blacklisting, and Bluetooth-based position synchronization. (The space-time history based velocity estimation is a fundamental component of RAPS

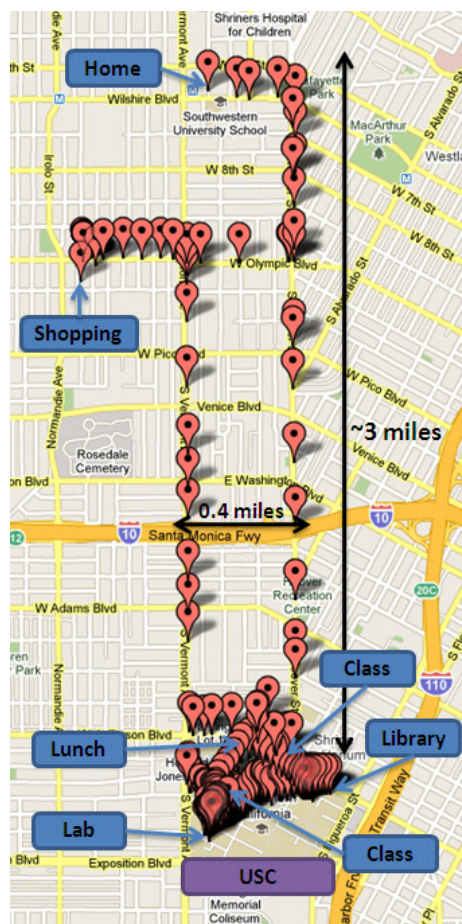


Figure 16: GPS trace plot of our experiment

that cannot be disabled.) In describing our results, we use the following notation (Table 1): *RAPS* denotes results from the phone running RAPS with all of its components enabled; *RAPS-B* denotes RAPS without BPS (Bluetooth-based position synchronization); *RAPS-BC* denotes RAPS without BPS and without celltower-RSS blacklisting; and *RAPS-BCA* is RAPS without BPS, celltower-RSS blacklisting, and accelerometer-based velocity estimation. Finally, another phone was programmed with alternative GPS activation schemes: *Always-On* denotes a scheme in which GPS is never turned off. Our experimental methodology, of carrying six phones in one bag, ensures that each phone sees the same GPS and celltower availability, and user activity. Moreover, all phones are within Bluetooth communication range of each other, and those phones that have BPS installed can benefit from synchronization.

We seeded each phone with two days worth of prior space-time history and blacklist information. The battery was fully charged before starting the experiment, and the experiment ran until the battery capacity dipped below 14% of full charge (level 1, out of 7 levels) at which point we intentionally terminated the application on that phone. This ensured graceful termination of the application, and safe retrieval of experimental logs. Our actual experiment ran for approximately 34 hours for the longest lasting phone while the *Always-On* phone terminated in approximately 9 hours.

Lifetime. Our first performance metric is lifetime: the time from when the experiment was started until the battery indicator for the phone reached 14% of full charge. Figure 17 shows the lifetime

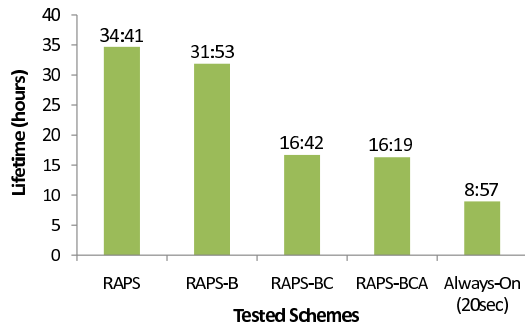


Figure 17: Lifetime of the phones (in hours:min) for each of the tested schemes

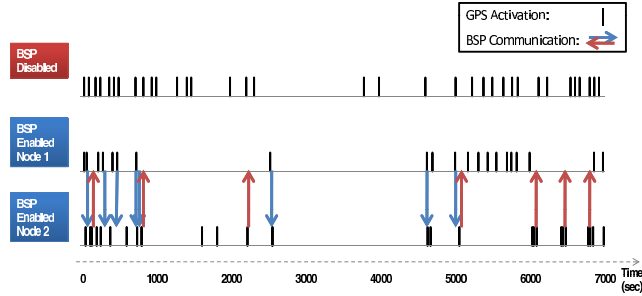


Figure 18: Event timeline of the three phones; two with BPS enabled, and the other with BPS disabled

of each phone from our experimental run. There are several observations to be made in this figure. First, RAPS’s lifetime is 3.87 times longer than that of *Always-On*, extending life of the smartphone on a single charge by more than 25 hours. By comparison, the state-of-the-art scheme [12] shows a 50% energy improvement over *Always-On* in an emulated scenario.

Figure 17 also shows the lifetime of RAPS variants, from which we can infer each RAPS component’s contribution to lifetime savings.

By comparing the lifetime of RAPS with that of RAPS-B, we see that BPS increased lifetime by 2 hours and 48 minutes. This corresponds to about 10.8% of the total RAPS lifetime increase over the *Always-On* case. Although it might seem that our evaluation methodology might overestimate the performance benefits from BPS, since phones were carried in close proximity, we believe this is not always true. If BPS penetration was high, then it is quite conceivable that the average number of Bluetooth contacts seen in an urban or campus setting could be considerably higher than the one that we have in our experiment, and our evaluation could underestimate energy savings.

To understand how BPS enables energy savings, it is instructive to look at a timeline of Bluetooth messaging and GPS activation. Figure 18 shows an excerpt of a timeline containing two phones running full RAPS, and one running RAPS-B. Three kinds of events are shown in the figure: GPS activation, Bluetooth transmission attempts, and Bluetooth reception. Notice how Bluetooth communication clearly defers GPS activation in the two RAPS phones, relative to the RAPS-B phones. Also, notice that BPS benefits are not unidirectional: each RAPS phone benefits from the other. This provides a natural incentive for BPS adoption; by adopting BPS, users will not only benefit others, but will also improve their own phone lifetimes. Of course, in our timeline, a GPS activation event may or may not result in a GPS position update

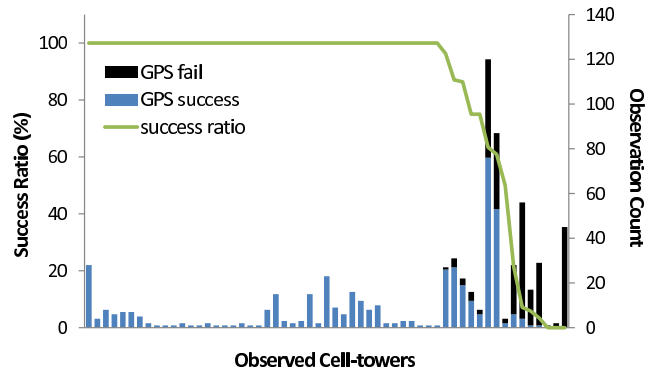


Figure 19: Content of the celltower-RSS blacklist

since GPS may timeout due to unavailability. Nevertheless, activation consumes energy regardless of whether it results in a position fix or not. Moreover, Bluetooth transmission attempts also include both successful and failed transmissions. Failures happen when both nodes attempt to become a master at the same time and neither was listening to become a slave device. This happened several times in our experiment because all phones were in the same bag and their activities and positions were closely synchronized: in this case, our evaluation methodology is somewhat adversarial with respect to BPS. Because of these failure modes, two phones running BPS may activate GPS at different times because they can have different uncertainty estimates.

By comparing RAPS-B and RAPS-BC, we see that celltower-RSS blacklisting contributed a significant increase to lifetime; it extended the lifetime by 15 hours and 11 minutes, or 59.0% of the total lifetime increase. To understand why this RAPS component is effective in increasing lifetime, consider Figure 19. This figure depicts the content of the celltower-RSS blacklist at the end of our experiment. Bar graphs represent the failure/success counts of GPS activations for each celltower that was observed when GPS was activated. The continuous line shows the corresponding success ratio, and celltowers on the x-axis are ordered by decreasing success ratio. Two features are evident from this graph. First, for a majority of cell-towers (those on the middle and left of the plot), GPS position fixing never fails. Second, for a smaller number of cell-towers, which also happen to be frequently visited, GPS failures do occur. We conjecture that this latter set includes cell-towers observed when a user is indoors; in this case, the GPS failures represent the cost of learning the different RSS values to blacklist. As we shall see below, celltower-RSS blacklisting can significantly increase the average interval between GPS activations (or, equivalently, can reduce the number of GPS activations).

By comparing RAPS-BCA with the *Always-On* case, we see that our space-time history-based velocity estimation, together with the idea of trading-off accuracy for energy and allowing uncertainty to be as high as 100 meters, extends the lifetime by 7 hours and 22 minutes over the *Always-On* case, contributing to 28.5% of the energy savings.

On the contrary, the use of accelerometer for velocity estimation actually resulted in only about 1.5% lifetime savings (compare RAPS-BC and RAPS-BCA). As we show later, RAPS-BC does have notably fewer GPS activations than RAPS-BCA. However, for the user in our experiments, clearly the energy cost of using the accelerometer (even a duty-cycled one) almost cancels the benefits of fewer GPS activations. This brings out an important aspect of RAPS; the performance benefits depend strongly on many factors

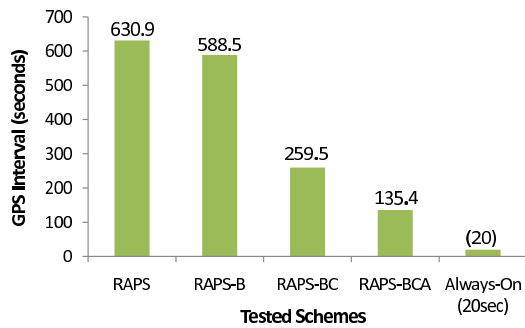


Figure 20: Average GPS activation interval (in seconds) for each of the tested schemes

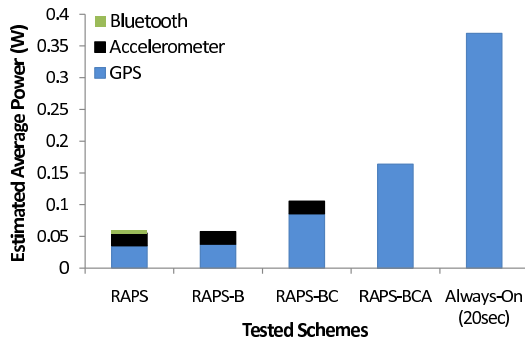


Figure 21: Estimated average power consumption for each of the tested schemes

(human behavior, environmental conditions), and not all components will necessarily provide significant energy savings under all conditions. We have left an exploration of this to future work.

Average GPS Activation Interval. Figure 20 shows how often GPS was activated (in the *Always-On* case, GPS was activated always and a position update was logged every 20 seconds). The figure shows that RAPS activated GPS every 630.9 seconds on average. Moreover, as expected, we also see progressively smaller intervals as we disable components of RAPS one-by-one: RAPS-B activates GPS every 588.5 seconds, RAPS-BC every 259.5 seconds, and RAPS-BCA every 135.4 seconds. Note that the longer experiments lasted for over 24 hours and include the time when the user was sleeping.

Expected Average Power Consumption. The average GPS activation interval is an important indication of energy consumption, but a more complete picture emerges when we consider the breakdown of power consumption across other hardware components (Bluetooth and accelerometer). Figure 21 shows the estimated average power consumption for each of the tested schemes. This figure was generated by averaging out the power usage of all components over the average GPS activation interval in Figure 20. For *Always-On* case, it is the power consumption of the GPS itself. From the figure we see that, as expected, GPS consumes the most power. By contrast, Bluetooth and the accelerometer contribute relatively small portions to the overall power consumption. This validates our strategy of adapting GPS activation intervals, at the cost of a little additional energy expenditure by using other sensors.

Distance Between GPS Position Updates. The goal of RAPS is to trade-off position uncertainty for reduced energy use. Thus, RAPS activates GPS when it estimates the position uncertainty to

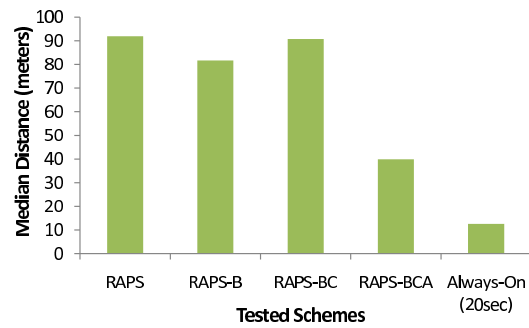


Figure 22: Median distance between two consecutive GPS position updates for each of the tested schemes

be near the 100m limit. Hence, ideally, the distance between two consecutive GPS readings should be exactly 100 meters. Of course, this is not a hard limit and can be exceeded due to various reasons: RAPS uncertainty estimates may not be precise; after GPS is activated, the user might have moved some distance before a fix is obtained; a GPS activation may fail, because GPS is unavailable at the user’s new location. Thus, in practice, we expect that 100m bound to hold only in a statistical sense. Figure 22 shows the median distance between two consecutive GPS position readings for each of the tested schemes. In general, RAPS, RAPS-B and RAPS-BC all have median distances of around 80 to 110 meters, which is an encouraging result. However, RAPS-BCA has a lower median distance than other RAPS variants because it frequently over-estimated the uncertainty, without being able to detect non-movement of the stationary user, and turned on the GPS. Without an accelerometer, RAPS-BCA uses the historical average of the velocity even when a user has been stationary, and this accounts for the overestimate.

4.2 Comparing RAPS to Periodic GPS Activation

In this subsection, we investigate how RAPS compares with periodic GPS activations. To conduct this experiment, we used two phones: one phone ran RAPS with all of its components enabled, and another phone was programmed to collect a GPS reading every 5 seconds. The experiment began with no prior space-time history or blacklist information (an adversarial setting for RAPS), and ran for approximately 8 hours and 2 minutes. The methodology for this experiment was otherwise identical to that discussed in Section 4.1.

In this experiment, RAPS achieved an average GPS activation interval of 465.1 seconds with average position uncertainty of 85.5 meters and a success ratio of 72.2%. *Position uncertainty* is defined as the distance between the two GPS positions, and *success ratio* is the fraction of times that the distance between two GPS readings was within 100 meters (our target uncertainty bound). To compare this to periodic GPS with various intervals, we used the log of GPS readings collected every 5 seconds and simulated the different intervals by sub-sampling at the appropriate frequency.

Figures 23 and 24 depict the average position uncertainty and the success ratio of periodic GPS, respectively, as a function of the periodicity. Using these two metrics, we find the periodicity that matches the performance achieved by RAPS. To have comparable average uncertainty as RAPS, periodic GPS duty-cycling requires its period to be around 270 seconds as indicated by arrows in Figure 23. For comparable success ratio, periodic GPS requires a period of 180 seconds, as shown in Figure 24. We compare RAPS against these two benchmarks.

	Avg. GPS Interval	Avg. Uncertainty (Dev.)	Success Ratio	Avg. Power
RAPS	465.1 sec	85.8 m (69.2 m)	72.2%	0.064 W
Periodic ($T=180$)	180.0 sec	61.9 m (84.6 m)	72.3%	0.123 W
Periodic ($T=270$)	270.0 sec	84.1 m (88.6 m)	69.4%	0.082 W

Table 2: Comparison of the RAPS against periodic GPS with fixed duty-cycle

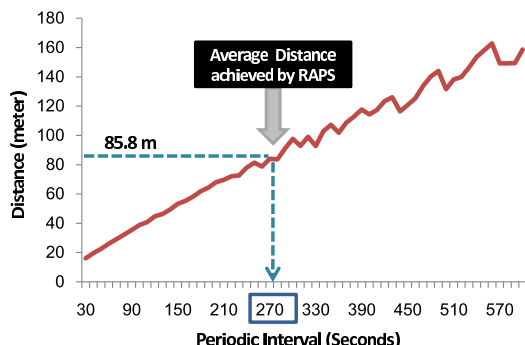


Figure 23: Average position uncertainty of periodic GPS from experiment as a function of duty-cycling interval

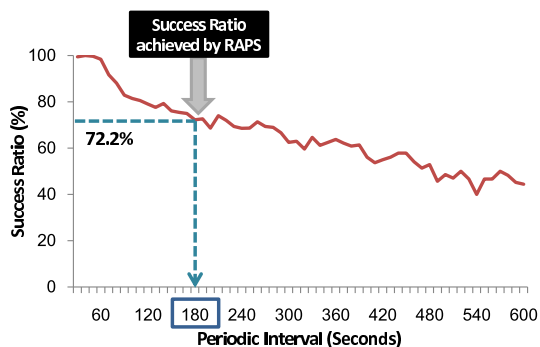


Figure 24: Success ratio of periodic GPS from experiment as a function of duty-cycling interval

Table 2 summarizes the result. In this table, the average GPS interval is the time between GPS activations, regardless of whether they resulted in a successful position fix or not. The average power is the estimated power usage for each case, which includes the power consumed by GPS and the accelerometer (when enabled), averaged over the average GPS time interval. The average uncertainty is the average distance between the two consecutive position updates. The table also shows the deviation of this distance from the 100m uncertainty tolerance target. Although the average location uncertainty and its deviation seem large, the large deviations can be attributed to GPS unavailability, and the results show that RAPS's deviation is comparable to that of periodic GPS.

To achieve a comparable success ratio, periodic GPS consumes 1.92 times the power used by RAPS, while to achieve comparable average uncertainty, it uses 1.28 times the energy used by RAPS. To get some insight into these numbers, let us consider a scenario where these positioning systems are used on an N95 smartphone with 1200mAh batteries without any other service running. Then, the battery will last for 35 hours with RAPS whereas it will last only for 23.8 and 30.8 hours for periodic GPS with intervals 180 and 270 seconds respectively. This result clearly shows the energy saving benefits of RAPS over periodic GPS schemes with fixed period.

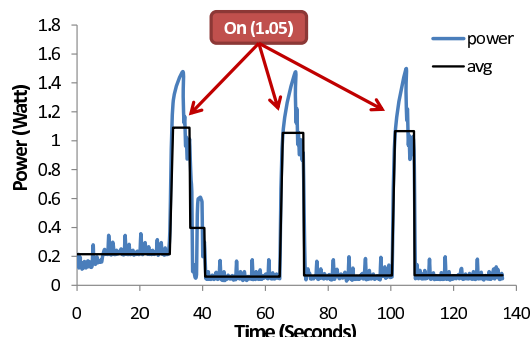


Figure 25: Power consumption of WPS on N95 phone

4.3 Integration with a WiFi Positioning System

RAPS has been designed to work with GPS in mind. However, all the techniques used in RAPS are oblivious to GPS, so RAPS can be used atop another underlying positioning system as long as it provides reasonably accurate positions. To validate this, we have implemented and tested a WPS [22] version of RAPS, namely RAPS-WPS.

WPS is a WiFi-based positioning system from Skyhook Wireless [22]. It determines location based on Skyhook's database of known Wi-Fi access points. The key advantage of WPS is that it can provide position information in urban areas and indoors given that a database exists for those areas. However, since it requires detection of beacons from 3 or more known WiFi APs, it does not work well when moving fast (i.e. driving), nor in places where WiFi APs are sparse. Also, the power consumption of WPS is comparable to GPS because of the cost of WLAN scanning and of communicating with the Skyhook database server, as we show below. Finally, it is generally known that WPS provides less accurate position than GPS at outdoor locations, which agrees with our experiments (Figure 1).

Figure 25 plots the power usage on the phone when WPS positioning was requested three times with 30 seconds interval. It shows that a single WPS positioning request, which involves scanning for visible WiFi APs and communicating with the database server, consumes around 1.05 Watt on average for a duration of approximately 6 seconds per request. Using this number, we can estimate the average power used by RAPS when WPS is used as the underlying positioning system. Table 3 summarizes the result along with the original RAPS for comparison. It shows that RAPS-WPS uses lower power, but has higher average uncertainty. However, higher average uncertainty is not due to the fact that RAPS-WPS activated WPS on fewer occasions, but because WPS is inherently less accurate; in fact, the average positioning interval was smaller for RAPS-WPS, which means that it was activated more frequently. The estimated average power usage of RAPS-WPS is lower than that of GPS because WiFi scanning is faster, so consumes less energy.

To summarize, RAPS can be used with positioning systems other than GPS, such as WPS. It consumes less energy when used with

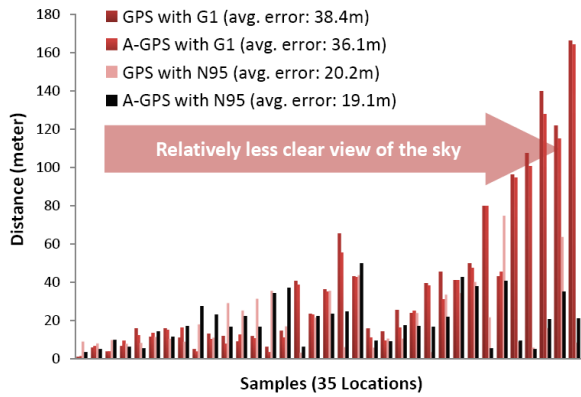


Figure 26: Assisted-GPS vs. GPS : Position error comparison

WPS because it can obtain a position fix faster, but its accuracy is lower than when using GPS because WPS is inherently more inaccurate.

	Avg.Positioning Interval	Avg.Power	Avg.Uncertainty (Dev.)
RAPS <i>GPS</i>	465.1 sec	0.064 W	85.8m (69.2 m)
RAPS <i>WPS</i>	387.3 sec	0.035 W	122.9 m (108.1 m)

Table 3: Comparison of the RAPS with GPS and WPS.

4.4 Are GPS errors Pervasive?

In Figure 4 of Section 2, we showed the inaccuracy of GPS by collecting position readings from several known locations. In this section, we validate this observation by answering two questions:

- Does Assisted-GPS result in lower error?
- Are there significant differences in the error characteristics across different handset platforms?

At least from the experiments we have conducted, the answer to both these questions is negative, suggesting that the instance of relatively high GPS error is a function of the environment, not of the platform.

Assisted GPS. Figure 26 compares the position inaccuracy results for GPS and Assisted-GPS, on both N95 and G1 phones, at 35 different known locations. As the figure shows, we have found negligible difference in position accuracy between A-GPS and GPS for either platform. This result is consistent with our personal experience using N95 phones where Assisted-GPS is superior to GPS only in the Time-To-First-Fix (TTFF) and not in the positioning accuracy. Although the term *A-GPS* is broadly used to include various forms of assistance (which could possibly include sophisticated position estimation techniques), it usually refers to an enhanced version of GPS that retrieves almanac and ephemeris data over a data connection (e.g. GPRS or 3G) for a faster fix. At least for the phones that we have used in our measurements, this does not improve positioning accuracy.

Different Platforms. Figure 27 compares the position (in)accuracy results for four different handset platforms: Android G1, Nokia N95, MotoDroid, and WinMobile HTC. As the figure shows, there isn't a qualitative difference in position accuracy between different types of mobile phone except for the G1 phone; other three phones had similar performance. The Android G1 phone did often report slightly higher distance errors than other phones especially when

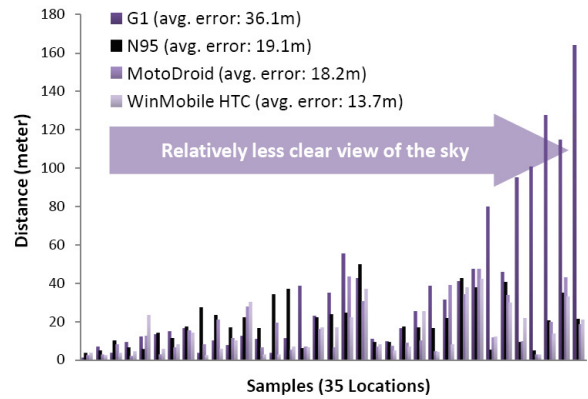


Figure 27: Comparison of position error for four different types of mobile smartphones

there was relatively less clear view to the sky. However, we believe that this fact does not significantly affect our design nor the result of RAPS.

4.5 Summary and Future Work

The results we have presented so far suggest that significant energy-saving gains can be obtained with a collection of techniques that permit delayed GPS activation. Of course, much work remains to be done before RAPS can become truly practical. Our parameter settings (for example, the accelerometer duty-cycling) can be optimized. Although we believe the system has the right incentives for the adoption of Bluetooth synchronization, privacy and security considerations need to be considered. Finally, we intend to perform a more careful evaluation of the parameter space, which is exceedingly large: performance critically depends on user consistency and phone handling behavior, on environmental conditions, GPS visibility, celltower placement density, and so forth.

5. RELATED WORK

A few prior pieces of work have attempted to duty-cycle location determination. The piece of work most closely related to ours is EnTracked [12]². It uses the accelerometer as a binary sensor to distinguish between stationary and in-motion users, and determines user velocity from the estimate reported by GPS. Based on this estimation of mobility, it schedules the next position update to save energy. However, it does not leverage user history nor does it employ the concept of activity ratio as a surrogate for velocity, as we do. It also does not consider the high power consumption of the accelerometer, nor infer GPS availability using celltower information, and it does not adapt to movement changes until the next decision point. Moreover, the system has an inherent circularity which can potentially degrade performance: it uses a velocity estimate to decide when to turn on GPS, but also uses GPS to get the velocity estimate. Most importantly, it does not consider the case where GPS is not available (e.g., indoors), and the system has been evaluated only in scenarios where GPS is always available.

Some work has explored the energy-accuracy tradeoff in ways different from RAPS. EnLoc [5] uses dynamic programming to

²As of this writing, the code for EnTracked is not publicly available. So, we present only a qualitative comparison with this scheme. However, both protocols have compared their performance relative to the case when GPS is always on, so the reader can assess the relative performance difference between RAPS and EnTracked.

find the optimal localization accuracy for a given energy budget and decides which one of GPS/WiFi/GSM localization methods to use. It uses the space-time history in its human mobility profile, but does not associate this with the accelerometer-based user activity. Micro-Blog [11] exploits the accuracy-energy tradeoff of GPS, WiFi, and GSM based localization for energy-aware localization. Specifically, depending on the accuracy requirement of the application, it uses a lower energy method over a higher method when possible. Contemporaneously, a-Loc [14] proposes to dynamically trade-off location accuracy and energy use, using probabilistic models of user location and sensor errors. It uses these models to choose among different localization methods and tune the energy expenditure to meet the dynamic location accuracy requirements specified by applications. It also discusses a method to automatically determine the accuracy requirements for certain applications such as mobile search and social networking. In contrast to all these pieces of work, RAPS leverages the observation that applications will need to tolerate intrinsic GPS errors in urban obstructed environments in order to trade-off accuracy for lower energy.

We have also built upon other duty-cycle determination methods. You et al. [27] propose a signal-strength based indoor localization scheme that adapts the sampling rate to the target's estimated mobility level for energy-efficient operation. This work uses a positioning error model similar to ours and also employs the accelerometer to detect whether the target is stationary or not. However, unlike RAPS, it is tailored for RF-based indoor localization and assumes walking as the only mobility mode. Farrell et al. [9] also use a similar positioning uncertainty model and propose an algorithm that controls when to perform GPS updates for efficient positioning. However, they do not estimate velocity but simply assume a maximum walking velocity for uncertainty calculation. Moreover, this work has been evaluated only in simulations.

There are several pieces of work that are related to individual components of RAPS. Deblauwe and Treu [6] propose GSM signature-based triggering to avoid activating the GPS receiver for as long as possible in order to save energy while still being able to detect entering and leaving a zone. Their basic idea is to compare the device's current GSM measurements with the ones taken the last time the GPS was switched on. This work is based on a similar intuition as ours, but assumes that *cellId*-RSS information from several celltowers are available at the smartphone, which, as we have discussed, contradicts current practice. SenseLess [3] uses accelerometer triggering to detect the existence of user movement and turns off GPS when not moving, and reports that this can be effective in reducing the number of GPS activations. EEMSS [26] employs the idea of using low power sensors (i.e. the accelerometer) to detect user state and context, and trigger activation of high power sensors (i.e. GPS) only if necessary. While doing this, they duty cycle each sensor to further save energy. Concurrent with our own work, Zhuang et al. [29] have proposed a location-sensing framework that includes four design principles – accelerometer-based suppression, location-sensing piggybacking, substitution of location-sensing mechanisms, and adaptation of sensing parameters when battery is low – to improve the energy efficiency of localization on smartphones that run multiple location-based applications.

Several pieces of work have attempted to learn mobility patterns. BreadCrumbs [19] uses the space history of a user to train a mobility model for each specific user and use it to schedule network usage using the connectivity forecasts. Zheng et al. [28] use supervised learning to infer motion modes (e.g., walking, bus, driving) from their GPS logs. Sohn et al. [24] also recognizes mobility

modes, but using coarse-grained GSM data instead of GPS data. Although RAPS does not explicitly infer users' motion modes using GPS or GSM logs, it estimates user velocity (which is a function of the mobility mode).

Finally, several other attempts have been made to address shortcomings with GPS. Pure WiFi-based [1, 22] and GSM-based [25] systems attempt to save energy by inferring location using other means. In contrast, RAPS focuses only on adapting the rate of GPS activations and can, in theory, be incorporated into these systems.

6. CONCLUSIONS

In this paper, we presented RAPS, rate-adaptive positioning system for smartphone applications. It is based on the observation that GPS is generally less accurate in urban areas, so it suffices to turn on GPS only as often as necessary to achieve this accuracy. RAPS uses a collection of techniques that could be implemented on current generation of smartphones to cleverly determine when to turn on GPS. We have evaluated RAPS through real-world experiments using a prototype implementation on a modern smartphone and show that it increases lifetime by more than a factor of 3.8 relative to the case when GPS is always on. Several interesting issues remain for future work including: examining optimal duty-cycling parameters for the accelerometer, better methods for distinguishing human motion from other activities that cause phone motion, and larger-scale evaluations.

Acknowledgments.

We would like to thank our shepherd, Romit Roy Choudhury, and the anonymous referees, for their insightful suggestions for improving the technical content and presentation of the paper.

7. REFERENCES

- [1] Place lab: A privacy-observant location system. <http://www.placelab.org/>.
- [2] L. Aalto, N. Göthlin, J. Korhonen, and T. Ojala. Bluetooth and wap push based location-aware mobile advertising system. In *Proceedings of 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys'04)*, 2004.
- [3] F. B. Abdesslem, A. Philips, and T. Henderson. Less is more: energy-efficient mobile sensing with senseless. In *MobiHeld'09: Proceedings of the 1st ACM SIGCOMM Workshop on Networking, Systems, and Applications for Mobile Handhelds*. ACM, 2009.
- [4] I. Constandache, R. R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *Proceedings of IEEE Infocom*, 2010.
- [5] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. Cox. Enloc: energy-efficient localization for mobile phones. In *INFOCOM Miniconference'09*, 2009.
- [6] N. Deblauwe and G. Treu. Hybrid gps and gsm localization - energy-efficient detection of spatial triggers. In *WPNC'08: Proceedings of the 5th Workshop on Positioning, Navigation and Communication*, 2008.
- [7] S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. Metrosense project: people-centric sensing at scale. In *First Workshop on World-Sensor-Web: Mobile Device Centric Sensory Networks and Applications WSW06, SenSys 06*, 2006.
- [8] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: Using a mobile sensor network for road surface monitoring. In *Proceedings of 6th International Conference on Mobile Systems, Applications, and Services (MobiSys'08)*, 2008.

- [9] T. Farrell, R. Cheng, and K. Rothermel. Energy-efficient monitoring of mobile objects with uncertainty-aware tolerances. In *Proceedings of 11th International Database Engineering & Applications Symposium (IDEAS'07)*, 2007.
- [10] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay. Myexperience: A system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of 5th International Conference on Mobile Systems, Applications, and Services (MobiSys'07)*, 2007.
- [11] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In *Proceedings of 6th International Conference on Mobile Systems, Applications, and Services (MobiSys'08)*, 2008.
- [12] M. B. Kjaergaard, J. Langdal, T. Godsk, and T. Toftkjaer. Entracked: energy-efficient robust position tracking for mobile devices. In *Proceedings of 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, 2009.
- [13] M. H. Krieger, M.-R. Ra, J. Paek, R. Govindan, and J. Evans-Cowley. Urban tomography. *Journal of Urban Technology*, 2010. <http://tomography.usc.edu/>.
- [14] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-accuracy aware localization for mobile devices. In *Proceedings of 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, June 2010.
- [15] D. Mizell. Using gravity to estimate accelerometer orientation. In *ISWC'03: Proceedings of the 7th International Symposium on Wearable Computing*, 2003.
- [16] P. Mohan, V. N. Padmanabhan, and R. Ramjee. "nericell: rich monitoring of road and traffic conditions using mobile smartphones". In *Proceedings of 6th ACM International Conference on Embedded Networked Sensor Systems (SenSys'08)*, Nov. 2008.
- [17] M. Motani, V. Srinivasan, and P. S. Nuggehalli. Peoplenet: Engineering a wireless virtual social network. In *Proceedings of 11th ACM International Conference on Mobile Computing and Networking (MobiCom'05)*, 2005.
- [18] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, pages 55–68, 2009.
- [19] A. J. Nicholson and B. D. Noble. Breadcrumbs: Forecasting mobile connectivity. In *Proceedings of 14th ACM International Conference on Mobile Computing and Networking (MobiCom'08)*, 2008.
- [20] J. Paek, K. Chintalapudi, J. Cafferey, R. Govindan, and S. Masri. A wireless sensor network for structural health monitoring: Performance and experience. In *Proceedings of The Second IEEE Workshop on Embedded Networked Sensors, (EmNetS-II)*, Sydney, Australia, May 2005.
- [21] B. W. Parkinson and J. J. Spilker. *Global Positioning System: Theory and Practice*, volume I. American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1996.
- [22] Skyhook Wireless. <http://www.skyhookwireless.com/>.
- [23] T. Sohn, K. A. Li, G. Lee, I. Smith, J. Scott, and W. G. Griswold. Place-its: A study of location-based reminders on mobile phones. In *UbiComp'05: Proceedings of the 7th International Conference on Ubiquitous Computing*, 2005.
- [24] T. Sohn, A. Varshavsky, A. LaMarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, and E. de Lara. Mobility detection using everyday gsm traces. In *UbiComp'06: Proceedings of the 8th International Conference on Ubiquitous Computing*, 2006.
- [25] A. Varshavsky, D. Pankratov, J. Krumm, and E. de Lara. Calibree: Calibration-free localization using relative distance estimations. In *Proceedings of 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, 2009.
- [26] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, 2009.
- [27] C.-W. You, P. Huang, H. hua Chu, Y.-C. Chen, J.-R. Chiang, and S.-Y. Lau. Impact of sensor-enhanced mobility prediction on the design of energy-efficient localization. *Elsevier Ad Hoc Networks*, 6(8):1221–1237, 2008.
- [28] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on gps data. In *UbiComp'08: Proceedings of the 10th International Conference on Ubiquitous Computing*, 2008.
- [29] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving energy efficiency of location sensing on smartphones. In *Proceedings of 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, June 2010.