Bartendr: A Practical Approach to Energy-aware Cellular Data Scheduling

Aaron Schulman⋆1, Vishnu Navda†, Ramachandran Ramjee†, Neil Spring⋆,
Pralhad Deshpande‡1, Calvin Grunewald⋆, Kamal Jain⋄, Venkata N. Padmanabhan†
⋆University of Maryland †Microsoft Research India ‡Stony Brook University ⋄Microsoft Research

ABSTRACT

Cellular radios consume more power and suffer reduced data rate when the signal is weak. According to our measurements, the communication energy per bit can be as much as 6x higher when the signal is weak than when it is strong. To realize energy savings, applications must preferentially communicate when the signal is strong, either by deferring non-urgent communication or by advancing anticipated communication to coincide with periods of strong signal. Allowing applications to perform such scheduling requires predicting signal strength, so that opportunities for energy-efficient communication can be anticipated. Furthermore, such prediction must be performed at little energy cost.

In this paper, we make several contributions towards a practical system for energy-aware cellular data scheduling called Bartendr. First, we establish, via measurements, the relationship between signal strength and power consumption. Second, we show that location alone is not sufficient to predict signal strength and motivate the use of tracks to enable effective prediction. Finally, we develop energy-aware scheduling algorithms for different workloads—syncing and streaming—and evaluate these via simulation driven by traces obtained during actual drives, demonstrating energy savings of up to 60%. Our experiments have been performed on four cellular networks across two large metropolitan areas, one in India and the other in the U.S.

Categories and Subject Descriptors

C.2.1 [**Computer Communications Networks**]: Network Architecture and Design—*Wireless communication*; B.8.0 [**Hardware**]: Performance and Reliability—*General*

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Bartendr, Cellular, Mobile, Energy, EVDO, HSDPA

MobiCom'10, September 20–24, 2010, Chicago, Illinois, USA.

1. INTRODUCTION

While cellular data networks provide near-ubiquitous coverage across many urban areas, the quality of coverage is far from being uniform. There are areas where the signal is strong and others where it is weak, and the whole spectrum in between [18]. The signal strength has a direct impact on radio energy consumption, which is a significant component of overall energy consumption on mobile devices such as smartphones. For instance, our measurements on a Samsung SGH-i780 smartphone show that the radio, when active, consumes up to *five times* the power of the base device.

The impact of signal strength on cellular radio energy consumption arises from two factors. First, the *power*, i.e., energy per unit time, drawn by the radio increases when the signal is weak. This is because a weak signal often means that the mobile device is near the edge of coverage and hence it must amplify the received signal. Also, it must transmit, both upstream data and acks for downstream data, at a higher power for power control reasons [4] or simply to have its feedback be heard by the tower. Second, the *data rate*, i.e., bits per unit time, decreases when the signal is weak. The reason is that the radio typically switches to a lower rate of modulation to keep the bit error rate low. The net result is that, according to our measurements, the communication *energy per bit* when the signal is weak can be as much as *six times* the energy per bit when the signal is strong.

While WiFi radios also exhibit significant variation in energy cost per bit at different locations, this is mainly due to variations in data rate rather than radio power [2, 8]. Moreover, there is little opportunity to exploit these variations for saving energy since WiFi users are typically stationary. In contrast, we show that a cellular radio experiences significant signal variations, say as the user goes about their daily commute, and these signal variations can be exploited by preferentially communicating when the signal is strong, thereby saving energy.

Scheduling communication when the signal is strong may not always be feasible (e.g., in the case of real-time communication such as VoIP). However, there are significant classes of applications where it is. Certain applications such as email syncing or photo uploads can defer communication, up to a point, without sacrificing service. Other applications such as on-demand streaming can prefetch content in anticipation of future need.

In this paper we present Bartendr, a practical framework for reducing radio energy cost in the above class of applica-

[1]Aaron and Pralhad completed part of this work while interning at Microsoft Research India.

tions by being cognizant of the signal strength "bars". Bartendr saves energy for these applications by addressing the following two key challenges.

First, signal strength must be *predicted* so that opportunities for energy-efficient communication can be anticipated, while taking application deadlines into account. Such opportunities may arise as the user, who may be driving, moves in and out of areas of good coverage. Moreover, such prediction must be energy-efficient, i.e., we cannot rely on energy-expensive sensors such as GPS.

Second, communication scheduling must take into account the energy consumption characteristics of both the CPU and the cellular radio. For example, waking up the phone to schedule can be expensive (e.g., 1 J to awaken and 0.5 J to suspend on a Samsung Omnia). Further, the cellular radio incurs a "tail energy" cost [3, 19], since the radio lingers in a high-energy state even after the end of transmission or reception. These factors make the problem of performing energy-aware data scheduling, driven by the mobile phone, challenging and novel compared to channel-state based scheduling algorithms such as Proportional Fair used in 3G base stations [11].

Our work addresses the above challenges and makes the following contributions:

First, we establish the relationship between signal strength and power consumption, on the one hand, and between signal strength and data rate, on the other. Together these yield the impact of cellular signal strength on energy per bit, which to our knowledge has not been characterized empirically in the literature.

Second, we show that location alone is not sufficient to predict signal strength because of the hysteresis built into cellular handoff decisions, which results in stickiness to the current point of attachment [17]. On the other hand, we show that the pattern of variation in signal strength is quite stable when *location is coupled with direction of travel*. Bartendr, thus, leverages the notion of a track [1], e.g., the route from a user's home to workplace, for its signal prediction. However, using GPS on the phone for identifying its position on a track can be prohibitive in terms of energy cost. Bartendr sidesteps this difficulty by representing a track solely in terms of the identifies of the base stations encountered, together with the corresponding cellular signal strength values. The device's current position and future signal values are identified by *matching against the recorded tracks in the signal dimension*, thereby completely avoiding the need, and hence the energy cost, of determining the device's physical location.

Third, we develop energy-aware scheduling algorithms for different application workloads. For a syncing workload, where little data is transmitted or received, scheduling is based on finding intervals in which the signal strength exceeds a threshold. For a streaming workload, on the other hand, we develop a dynamic-programming-based algorithm to obtain the optimal data download schedule. This algorithm incorporates the predictions of signal quality, and thereby energy per bit, and also the tail energy cost. Note that in this paper we focus on the common case of data download, although we believe energy-aware scheduling is equally applicable to the case of data upload.

We evaluate Bartendr using extensive simulations based on signal and throughput measurements obtained during actual drives. Our experiments have been performed on four cellular networks across two large metropolitan areas, Bangalore in India and Washington DC in the U.S, and spans 3G networks based on both EVDO and HSDPA. Our evaluation demonstrates energy savings of up to 10% for the email sync application, even when the sync operation results in no email being downloaded, implying that the energy savings in this case results only from the lowering of the radio power when the signal is strong. In contrast, our evaluation shows energy savings of up to 60% for the streaming application, where energy-aware scheduling helps save energy by both lowering the radio power used and by cutting the duration of radio activity owing to the higher throughput enabled by a strong signal.

provider	country	type	device	precision
Sprint	USA	EVDO	Pre	~80
Reliance	India	EVDO	EC1260 USB	6
Verizon	USA	EVDO	Omnia	~80
AT&T	USA	HSDPA	SGH-i907	~80

Table 1: Sources of measurement data, including different technologies in different countries. "Precision" represents the number of unique signal strength values the device reports.

2. MOTIVATION

In this section, we argue, based on measurements, that exploiting variation in signal strength can yield energy savings. Table 1 lists the mobile devices and networks that we measured. These devices expose signal strength in one of two ways: some provide fine-grained, raw received signal strength indication (RSSI), others provide only six coarse signal levels, corresponding to the (0–5) "bars" displayed on phones. These reported values lack meaningful units.

Our first step is to show that energy consumed by communication varies with reported signal strength. Next we show that signal strength varies in practice, and that this variation is consistent. Finally, we describe how a few typical applications may be sufficiently flexible in scheduling their communication to match the periods of good signal strength.

We focus on saving energy for applications that mostly download because these are most prevalent on mobile devices (e.g., email, news, streaming audio and video). While there are applications that primarily upload (e.g., photo sharing), we do not discuss these here or present measurements of the upload power consumption on mobile phones.

Measuring the energy consumption of mobile phones in motion requires a power measurement apparatus that is both portable and can be connected between a mobile phone's battery and the device. Our setup consists of a USB oscilloscope that measures current by observing the voltage drop over a .01 Ω precision shunt resistor. The resistor connects the phone's power lead to its battery.

2.1 Strong signal reduces energy cost

Communicating when the signal is strong reduces the energy cost by cutting both the power drawn by the radio and the communication time. Communication in strong signal takes less *power*, for both transmission *and* reception, although our focus here is primarily on the reception power. A strong signal also makes feasible advanced modulation schemes that yield higher throughput. Thereby cutting the

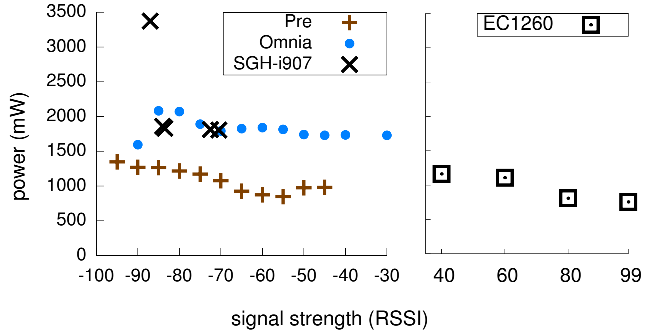

Figure 1: Power consumed by mobile devices report fine- (left) or coarse-grained RSSI (right). The EVDO devices are measured while driving; the i907 was measured statically at several locations. The very high i907 power was observed while communicating indoors.

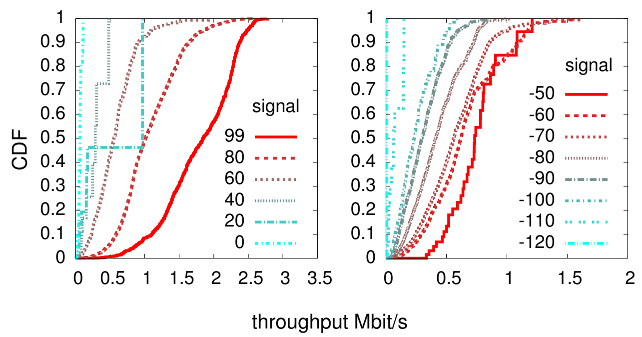

Figure 2: Signal affects throughput. Throughput over 9 drives on Reliance Telecom's EVDO network in Bangalore, India (left) and a 4 hour drive using Sprint's EVDO network on US interstate 95 (right).

time needed to complete the communication and potentially allowing the device to sleep eagerly. In this section, we address power and time, in turn.

The radio draws more current to operate in low signal locations. One reason is that the power amplifier switches to a high power mode to counter the drop in signal strength [7]. This applies not only for transmission, but also for reception since the mobile client continuously reports the received signal strength to the base station, 800 to 1600 times per second (the base station uses this feedback to choose an appropriate modulation and data rate). Figure 1 depicts the power consumption of different devices while receiving a packet flood, when operating at different signal strengths. Communication at a poor signal location can result in a device power draw that is 50% higher than at good signal locations. Moderate signal strength values (-90 to -70 RSSI) are more common than extremes.

Communication can take less time by exploiting high throughput available in locations of good signal. Strong signal allows for high rates, and thus short data transfer times. For example, EVDO Rev A uses one of 14 rates ranging from 38 Kbps to 3.1 Mbps depending on signal strength. Figure 2 depicts cumulative distributions of receive throughput for various

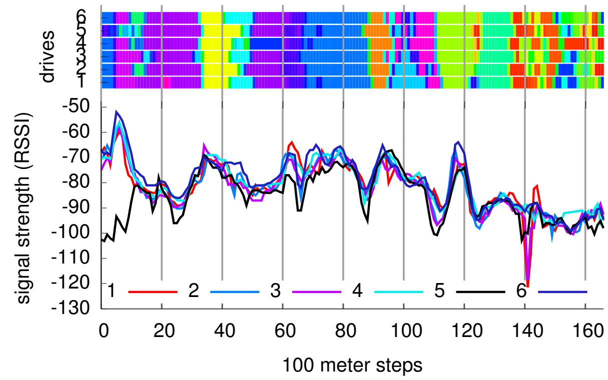

Figure 3: Signal varies by location for 6 drives over 17 km. Colors on top indicate base station id.

signal strengths. The measurements are 2-second (Reliance) and 3-second (Sprint) UDP flood throughput samples.

For the device on Sprint's network, RSSI values are rounded "up," i.e. values between -59 and -50 appear as -50 while the device on Reliance's network reported signal values already in one of six bins. The median throughput increases dramatically with signal strength; there is a factor of four difference in the median throughput between 60 and 99 RSSI for the Reliance network in India and a similar factor appears between -50 and -110 RSSI for the Sprint network in the U.S. However, the CDF also shows a wide range of throughputs for each signal quality bin, likely due to variations in sharing of aggregate cell capacity with other users.

In summary, when the signal is weak, not only does data transfer take longer to complete, but the radio is also simultaneously drawing higher power. These two factors are cumulative, so the overall energy required to transfer a fixed chunk of data, i.e. energy per bit, can be as much as *six times higher* (25% throughput and 50% more power) while communicating from poor signal locations compared to good signal locations.

2.2 Signal varies by location

Cellular signal strength varies depending on location because of the physics of wireless signal propagation. Signal strength degrades sharply with distance from the base station and is impeded by obstructions such as trees and buildings. Although the "bars" displayed on phones have made everyone aware that cellular signal varies across locations, this variation needs to be both significant and consistent for signal strength based scheduling to be effective.

Figure 3 plots the signal strength reported by the Palm Pre on each of five drives along a 17 km highway path from Washington, DC to College Park, MD collected on different days. The colored bars above the graph represent which base station the device is associated with during each track. In presenting these repeated drives of approximately the same path, we take for granted that humans are creatures of habit and that their paths are predictable [10]. Figure 3 shows graphically that, despite potential changes in handoff behavior or environmental effects, recorded traces of signal variation may be able to predict future signal variation along the same path. A majority of the signal variation across drives are small (<5 RSSI) while a few variations are significant, for example, at the start of drive 5. For this particular drive,

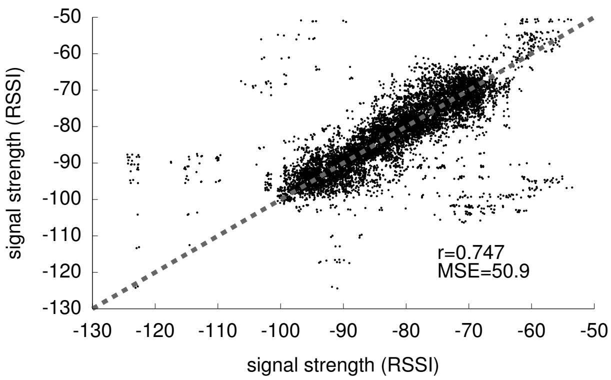

Figure 4: Signal variations are consistent for 6 drives over 17 km. Signal correlation for 25 m steps in all drive-pairs. To better illustrate the density of integral signal strength points, we add random noise of -0.5 to 0.5 RSSI to each pair.

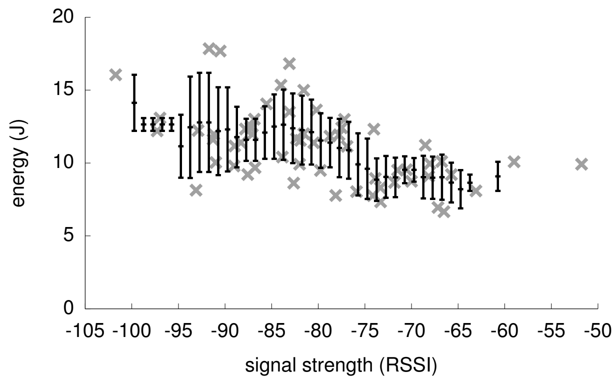

Figure 5: Average signal and energy for 62 SSL email syncs on the Palm Pre. Error bars represent μ and σ for windows of $\pm$2.5 RSSI. Above -75 RSSI, the sync energy is consistently low. Below this threshold, the sync energy is higher and variable.

unlike the others, the Pre does not keep a steady association to the base station represented by blue. Instead it switches between base stations until close to step 15 when it steadily associates with the violet (dark gray) base station.

For each 25 m step, we present a scatter plot of signal strength values across all pairs of drives in Figure 4. Perfect linear correlation is represented by the 45-degree line; one can see that most of the points in the figure are clustered around this line. The overall correlation coefficient is 0.75, which indicates that there is significant linear correlation of signal strength values across drives. This validates our hypothesis that signal variation along a path is consistent between drives.

Figure 3 shows that the variation of the signal strength along the drive is also significant. The highest and lowest strength values are -50 and -120 RSSI and there are frequent signal strength variations between -90 and -70 RSSI. The cost of communicating at -90 instead of at -70 RSSI entails the use of about 20% additional power (Figure 1) and a median throughput that is 50% lower (Figure 2). This results in an energy per bit on the Pre that is 2.4 times higher at -90 RSSI compared to at -70 RSSI. Thus, if applications were to preferentially communicate at -70 RSSI instead of at -90 RSSI, the potential communication energy savings are 60% ($1 - \frac{1}{2.4}$).

3. SUITABLE APPLICATIONS

The approach to saving energy in Bartendr is to defer communication, where possible, until the device moves into a location with better signal strength, or conversely, to prefetch information before the signal degrades. However, not all communication is amenable to adjustments in timing. We now look at two types of common applications that can make use of this approach.

3.1 Sync

The first application class that we consider is background synchronization, where the device probes a server periodically to check for new mail, updated news, or similar pending messages. Within each periodic interval, the syncing operation can be scheduled whenever the signal is strong. Many devices perform this background synchronization whenever powered on, meaning that even small savings in energy consumption for each background synchronization operation has the potential to yield large cumulative energy savings.

In order to schedule individual synchronization attempts, we assume that the synchronization interval is flexible. For example, while the application may be configured to check for new mail every five minutes, we allow each individual sync to be spaced, say, four or six minutes apart, while ensuring that the average sync interval remains five minutes. In our evaluation, we focus on the common case of a sync operation that does not result in any new data (e.g., email) being fetched. This represents a lower bound on the potential energy savings. We expect that, by choosing to perform sync more often when the signal is good, actual data communication (e.g., downloading of emails and attachments), if any, would also likely happen during periods of good signal, resulting in additional energy savings.

Figure 5 shows the energy consumed for 60 syncs while driving on main roads in two states in the U.S. (20 in Michigan, 40 in Maryland). The signal shown is the average signal over the sync. The median sync time is 10 seconds, the maximum is 13.2 s and the minimum is 7.4 s. Although the signal strength during a given sync operation is typically stable (median variation is 3 RSSI), it can also be highly variable at times (maximum variation is 22 RSSI). The error bars represent average and standard deviation for $\pm$2.5 RSSI windows. Nevertheless, in Figure 5, we can set a threshold of approximately -75 RSSI, such that syncs performed when the signal is above the threshold consistently consume less energy than when below the threshold. The average sync at a location with signal above -75 requires 75.3% of the energy when the signal is below -75. The decrease in average sync energy as signal increases matches the power profile of the Pre in Figure 1.

Although the differences in energy cost are significant, designing a system to realize these savings is challenging. In the extreme case, when there are no other applications executing on the device, the entire device could be asleep in between each sync operation, so the device must predict when to sync before it goes to sleep, and while it sleeps the prediction can not be updated. A simple syncing schedule that checks precisely every five minutes might miss opportunities to sync in

run	power (mW)	time (s)	energy (J)
	-93 RSSI		
1	1969	85	167
2	1983	83	164
3	1904	82	156
	-73 RSSI		
4	1655	86	142
5	1539	68	104
6	1532	187	286
7	1309	85	111
8	1400	76	106
9	1403	71	99

Table 2: Even while playing a YouTube video on the Palm Pre, efficient communication saves significant energy. Energy consumed while playing a one minute YouTube video in low (-93 RSSI) and high (-73 RSSI) signal.

strong signal locations, but for those five minutes, the device uses very little power. Further, any energy expended as part of the computation to predict where the good signal strength location five minutes in the future detracts from the potential savings, if any.

The alternative of push-based notification is not necessarily more energy efficient than a pull-based approach. Push notification requires the device to be in a (higher-energy) state where it can receive such notifications and also expend energy to maintain a connection to the server as the device traverses many cells. In contrast, a pull-based approach can keep the device in a very low-power suspended state between each sync. Finally, even in the case of a device that uses push notifications, Bartendr could be used to decide when to schedule the downloading of large messages, the availability of which is learned through the push notification mechanism.

3.2 Streaming

Streaming applications such as Internet radio and YouTube are another class of applications that permit flexible communication scheduling, so long as application playout deadlines are met. Streaming sites typically transmit pre-generated content over HTTP. In addition, some of these sites throttle the rate at which the data is streamed to the client, while keeping the client-side buffer non-empty to avoid playout disruptions. We can save energy for these applications by modulating the traffic stream to match the radio energy characteristics: downloading more data in good signal conditions and avoiding communication at poor signal locations. The challenge, however, is to ensure that every packet is delivered to the client before its playout deadline, to avoid any disruption being experienced by the user.

One might question whether the variation in power due to signal strength is significant relative to the baseline power consumed by display and processor during video playback. In fact, video playing is an important worst-case example because it exercises the most energy consuming hardware on the device. To address this question, Table 2 presents the total energy cost of downloading and playing a one minute YouTube clip on the Pre at two locations with different signal strengths. In general, energy consumed at -93 RSSI is about 50% higher than the energy consumed at -73 RSSI. In other words, communication energy savings are significant, even while most of the phone's hardware is active. However, in run 6, energy consumed at -73 RSSI is higher than energy consumed at -93 RSSI, because of lower bandwidth at -73 RSSI for this experiment (perhaps due to competing users). This bandwidth variation can pose a significant challenge in delivering the full energy savings of scheduling at good signal locations.

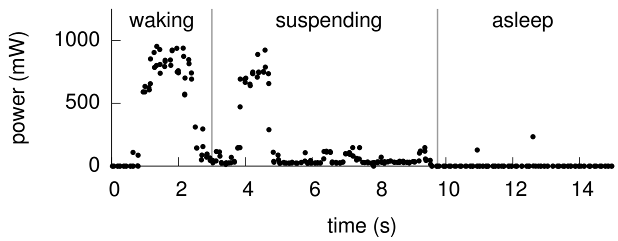

Figure 6: Omnia's wakeup and suspend power.

Given the above findings, in later experiments where we evaluate Bartendr, we will ignore the processor and display power consumption, and focus our measurement solely on the communication energy. Doing so also helps us avoid noise due to fluctuations in the energy consumed by the other components.

Although our focus is on streaming, with its intermediate deadlines for maintaining uninterrupted playback, bulk transfers could be equally amenable to scheduling. Consider the tasks of uploading photographs and downloading podcasts: although these tasks should be completed within a reasonable time, a user might appreciate if the transfers also placed a minimal cost on battery life.

4. PHONE ENERGY MODEL

Bartendr reduces the energy consumed to communicate by taking advantage of variations in power indicated by signal. However, to save radio energy, it will need to spend energy to predict signal strength and schedule communication. One consequence of this tradeoff is that we must forgo GPS as a means of determining location.

The goal of this section is to identify the power costs of various features of the phone that we use to inform the design of Bartendr. We expect these power features are representative of the current and future state of phone technology. Where possible, we provide power costs measured on the Palm Pre and Samsung Omnia exemplars, and expect their relative values on other phones will be be qualitatively similar. This section comprises two parts: the power of devices on the phone, and a power model for the radio.

4.1 Processor and device power

Processors can sleep in a very low power state. The Omnia uses power unmeasurably small by our equipment when sleeping, while when awake it requires approximately 144 mW. Sleep precludes any activity until the processor is awakened by an interrupt. The focus of aggressive power saving is thus to keep the processor in suspended state for as long as possible.

Unfortunately, the transition between sleep and active states requires more energy than simply powering the processor. We show an example transition on the Omnia in Figure 6. The peaks before and after the awake state represent the power cost of restoring state and saving state for devices. The energy cost of these transitions is approximately 1 joule

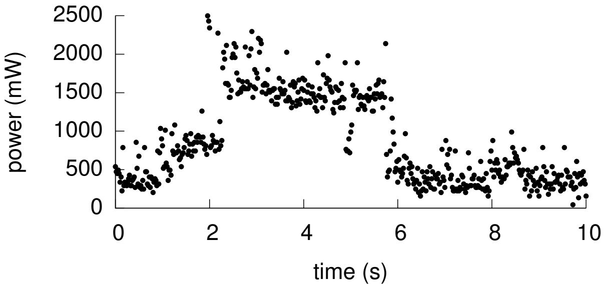

Figure 7: The Palm Pre's tail energy. The Pre initiates an ICMP Ping at 2 s and it completes at 3 s; the tail continues until 6 s. Following the tail, the baseline power of the phone is 400 mW with the display dimmed.

to restore from sleep and 0.5 joules to return to sleep. In short, the processor cannot simply be put to sleep for short intervals to save power.

GPS devices can provide precise position. However, GPS measurement suffers from high latency to obtain a fix from the satellites, and high energy to interpret their weak signals. Constandache et al. report 400 mW baseline power for GPS on a Nokia N95 [6].

Cellular signal strength provides an alternative method to position the device [12] with much lower power requirements. The radio measures signal strength as part of the normal operation of the phone, for example, to receive incoming phone calls and perform registrations. It performs this measurement even while the processor is suspended.

Accelerometer measurements are nearly free, but can only be used when the processor is powered on. Their measurements could improve mobility prediction. In Section 8.2. we discuss incorporating accelerometer measurements into the signal prediction. Accelerometer measurements have been previously used to position devices [5].

4.2 Radio power states

In this subsection, we describe the cellular radio characteristics that influence power consumption but are independent of signal variation.

The cellular radio mostly remains in a low power state, ready to receive incoming phone calls. At an intermediate power state, the radio is ready to transmit and receive data packets. Finally, in the highest power state, the radio is actively transmitting or receiving data. Apart from these states, when the received signal is very poor, the phone may expend energy continuously searching for a tower with good signal.

Figure 7 depicts the power drawn by a Palm Pre device over time when a single ICMP Ping message is initiated from the device around time 2 s. Prior to 2 s, the radio remains in a low power state (the phone, dimmed display and radio consume less than 400 mW). When the ping is initiated, signaling messages are exchanged for resource allocation and the radio transitions to the high power state (power drawn goes up to 2000 mW). The ping is sent and a ping response is received between 2 and 3 s. The radio then remains in an intermediate power state, ready to transmit and receive further packets, until about 6 s (power drawn around 1500 mW). Finally, the radio transitions back to the low power state.

The radio remains active in the intermediate power state for a preconfigured timeout duration after each communication episode, consuming what is known as Tail Energy [3, 19] (3 s to 6 s in Figure 7). Cellular network providers typically control this timeout value, though some mobile devices use a technique called fast dormancy to reduce the duration. The duration of this timeout, which ranges from a few seconds to ten seconds or more, is chosen to balance the cost of signaling for resource allocation to move a radio into active state (and the resulting latency and energy costs on the device) and the wasted resources due to maintaining a radio unnecessarily in active state. Since the tail energy cost is incurred after every communication episode, sporadic communication can be a significant energy drain on mobile devices.

5. BARTENDR ARCHITECTURE

In this section, we introduce Bartendr. Bartendr strives for energy efficiency by scheduling communication during periods of strong signal. To accomplish this, it predicts signal strength minutes into the future. For example, Bartendr can predict efficient times to wake up and sync email, and intervals when data should be downloaded. First, we describe how Bartendr uses prior tracks of signal strength to predict future signal strength. Then, we compare tracks to alternate methods of signal prediction based on location and history. Finally, we present algorithms that use the predicted signal strength to efficiently schedule syncs and streaming media.

5.1 Predicting signal with signal tracks

Bartendr predicts signal strength for a phone moving along a path. As we saw in Figures 3 and 4, signal strength is consistent at the granularity of 25 and 100 m steps along a path. This consistency means that Bartendr could, in principle, predict signal strength along a path using previous signal measurements, captured while traveling along the same path. Before discussing the challenges involved in accomplishing this, we lay out our assumptions. We assume that users will, in general, store several of these *signal tracks* on their phone, corresponding to the paths that they frequently travel on. We further assume that Bartendr can infer the current track of the mobile phone. This can be identified with high probability using mobility prediction techniques [10].

Predicting signal strength on a signal track requires two steps: finding the current position of the phone on the track, and predicting the signal in the future starting from that position. GPS could be used to locate a phone on a track, but doing so would drain considerable energy and would detract from the energy savings sought by Bartendr. Instead, Bartendr locates itself on a signal track by finding the measurement in the track that is *closest* to its current signal measurement. Signal measurements come at no extra energy cost because the phone's cellular protocol needs them for handoff [21]. Of course, there may be several points on a signal track with the same signal strength, so signal measurements also include a neighbor list: a list of all the phone's neighboring base stations sorted by signal strength. Bartendr's current position in the track is the one that has the most matching neighbors (in order) and the closest signal strength. While this approach of signal tuple-based matching in Bartendr is similar to that used for localization in prior work [12], all of the computation in Bartendr is confined to signal space, without any reference to physical location.

We find that the closest match heuristic works well for Bar-

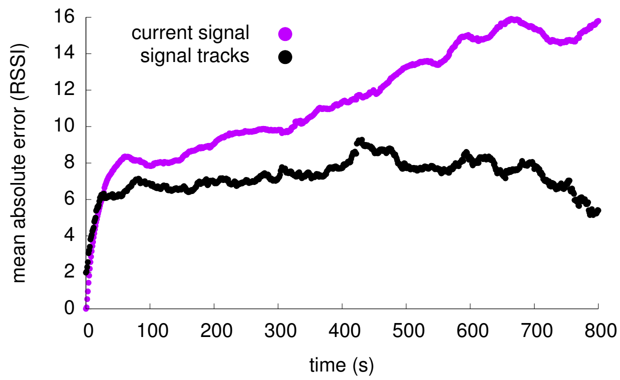

Figure 8: Average prediction error from all 25 m steps in six 17 km tracks. Signal tracks predict signal strength with lower error than predicting future signal to be the same as the current signal. Signal tracks can also be used for long term predictions.

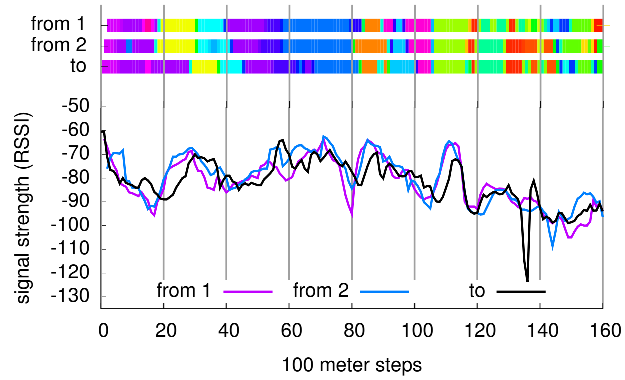

Figure 9: Tracks from opposite directions may not align in signal strength. Two "from" tracks compared to a representative "to" track from Figure 3.

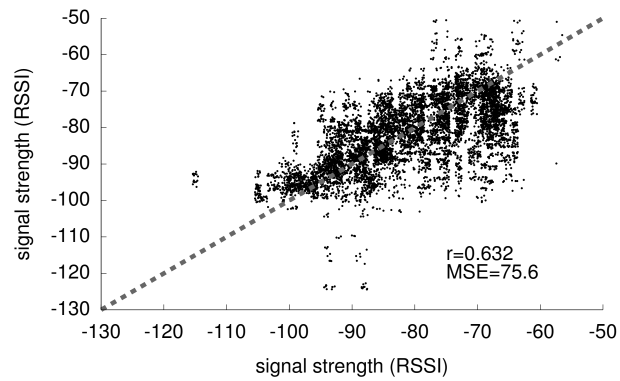

Figure 10: Signal correlation of 25 m steps in all pairs of the two "from" tracks with the six "to" tracks. The dashed line represents the ideal correlation.

tendr's needs, but errors are possible. The signal strength approaching and leaving a cell may be similar, and the closest neighbor list might not disambiguate these two positions. Further, if the signal is not changing often, perhaps in an environment of sparsely populated terrain, the closest match may be ambiguous. We observed very few occurrences of such errors in our testing. We discuss other approaches for phone positioning in Section 8.2.

Once Bartendr determines the phone's location on the track, it predicts signal strength minutes into the future. With signal tracks this means simply looking ahead in time from the current location. Although signal is consistent for locations along a path, the time to travel to a location along the path is not. For example, if the phone is traveling along a road it may travel at different speeds or stop at different times. This problem can be mitigated by constantly updating the phone's location on the track (discussed further in Section 8.1). For the evaluation of Bartendr in Section 6, Bartendr skips over stops in the signal tracks, although this provides minimal benefit.

We now compare Bartendr's signal track predictor with a simple predictor that only uses the current observed signal strength, which is averaged over few seconds, to predict future signal strength. We choose a position every 25 m on each of the six 17 km tracks as the starting position for the prediction. Using the two methods, we computed the error of signal strength predictions up to 800 s in the future. We ran both predictors on all of the starting positions in all six tracks. We also used all six tracks as previous tracks for the signal track predictor.

Figure 8 shows the average absolute error (y-axis) of all signal predictions 0 s to 800 s in the future (x-axis). Signal appears to vary about 6 RSSI over short intervals (< 20 s). We find that predicting past 20 s in the future, the signal track based predictor has lower average error than the current signal based predictor. Signal tracks can also predict signal strength 800 s in the future without a significant increase in error.

5.1.1 *Location alone is not sufficient*

One might expect that the precise location of GPS, if made for "free" in terms of energy cost, would yield better estimates of signal than would be possible with Bartendr's signal track predictor. However, our measurements show that signal strength can be significantly different at a location based on how the device arrived there, for example, the direction of arrival. Figure 9 depicts the average signal strength for each 100 m step of tracks collected while traveling in *opposite* directions ("from" and "to"). Compared to signal strength values over a track when traveling in the same direction (Figure 3), it is clear that there is less correlation when traveling in opposite directions. The lower correlation values of a signal at a location when traveling in opposite directions (Figure 10) compared to traveling in the same direction (Figure 4) provides further evidence that location alone is not sufficient for signal strength prediction.

The identity of the cellular base station that the phone is attached to at a given location (color-coded at the top of the graph) does not match at many steps across tracks in opposite directions. We believe that the hysteresis in the cellular handoff process explains this effect. A phone switches from its current attached base station only when its received signal strength dips below the signal strength from the next base station by more than a threshold [17]. Thus, phones traveling in different directions may be attached to different base stations at a given location. This observation implies that incorporating direction of travel with location, i.e., a track [1], is necessary for accurately predicting signal strength on a path.

5.2 Scheduling sync

After locating the device on a stored signal track, we must determine when to schedule the next sync. Recall that for sync, our goal is to put the processor to sleep for a calculated interval, so that the phone wakes up when it is at a strong signal location. If the prediction is incorrect, perhaps because the device traversed a track more quickly or more slowly than expected, the device may sync in a location of poor signal or attempt to defer for a few seconds to to catch the good signal if it is soon to come.

We propose two threshold-based techniques to find the location to sync given our current location in the track, *first above threshold* and *widest above threshold*. The threshold used in both techniques represents significant power savings: in Figure 5, reasonable threshold values span -76 to -74 RSSI, from which we picked -75 RSSI. The *first* approach waits to sync until the first time -75 RSSI is crossed in the stored track; the *widest* approach waits (potentially longer) for the time at which the stored track exceeds -75 RSSI for the widest interval. *First* could limit prediction mistakes if it is easier to predict near-term events; *widest* could limit prediction mistakes if the wide intervals represent easy targets.

Once the future time for sync is predicted by one of these techniques, the device is suspended to a very low power state until the predicted time. If the user on the track travels at a speed significantly different from that of the historical tracks, the wakeup could occur at a different signal strength value compared to prediction, possibly resulting in diminished energy savings.

5.3 Scheduling streaming

We next look at scheduling communication for the streaming application. When streaming, the device remains powered on and continuously determines position, so that unlike syncs, errors due to speed variations can be compensated for dynamically.

We now look at how to efficiently schedule a data stream of size S over certain duration of time T with minimal energy. To make the problem tractable, we divide the input stream into fixed size chunks of N frames, and time T is divided into slots. A slot is defined as the period of time where a single frame can be transmitted. Since data rates are not fixed, each slot can be of variable width depending on the expected datarate at that time. The power consumed to transmit a frame in a slot is also variable. We use the predicted signal strengths and median observed throughput values for the scheduling interval T to estimate the slot widths and average power consumption for each slot. This is illustrated in the Figure 11, which depicts signal strength variation over time and power consumed during transmission of frames A, B, C, and D at times t_1,t_2, t_3, and t_4. Frame A is scheduled when the signal is low, and hence it incurs higher power and longer time to complete compared to all the other frames. Given a predicted $signal_\ell$ in slot ℓ, we calculate the communication energy as follows:

$$\frac{Signal\_to\_Power(signal_l) * \frac{S}{N}}{Signal\_to\_Throughput(signal_l)}$$

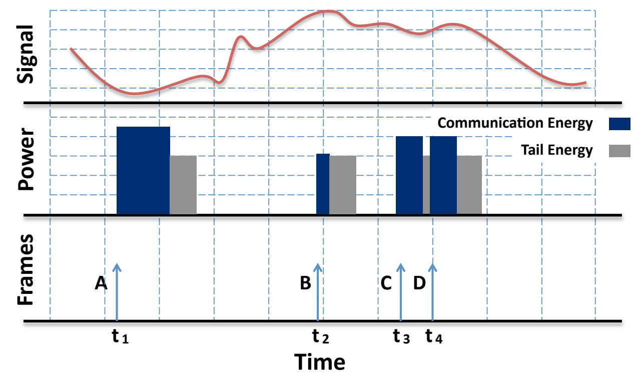

Figure 11: A sketch of the signal-based stream scheduling algorithm.

The two functions in this expression map a signal value to the corresponding power value and median throughput value. The mapping is done based on empirical measurements as described in Section 2.

Given N frames and M slots, where $N \leq M$, the optimal scheduling problem seeks to find an assignment for each frame to one of the slots, such that the total energy required to transmit N frames is the minimum of all possible assignments. One approach is to greedily schedule the frames in the best N slots which incur the least energy for communication. However, this approach ignores the cost of tail energy incurred every time there is a communication. When multiple frames are scheduled in consecutive slots, the entire batch of frames incur only one tail, as opposed to a tail for each frame if they are spaced out in time. A greedy approach that ignores the radio tail behavior can be very inefficient.

Thus, the scheduling algorithm should take into account both the energy required for communication and the tail energy incurred for the schedule. Let us look at how to compute the tail energy overhead for a schedule. When there are no frames scheduled for a certain period of time prior to the current slot, the radio is in an idle state at the beginning of the slot. If a frame is sent during this slot, the radio switches to its active state, and remains in the active state for the duration of at least one tail period (e.g., Frame A in Figure 11). However, if a frame is scheduled in a slot when the radio is already in active state due to some transmission in the prior slots, none or only a fraction of the tail energy needs to be accounted (Frame D). We now describe a dynamic programming formulation that computes the minimum energy schedule given the energy cost of transmission in each slot, accounting for these various tail overheads.

Let $E_{k,t}$ be the minimum energy required to transmit k frames in t timeslots. Corresponding to this minimum energy schedule, the variable $Last_{k,t}$ stores the slot number where the kth frame is scheduled. Let $ESlot_\ell$ be the sum of the communication energy required to transmit a frame in slot ℓ and the incremental tail energy cost, given the transmissions that occurred in the previous slots. For example, in Figure 11, frames A, B, and C are all scheduled at times when the radio is idle. As a result, each of them incur the cost of a full tail. However, frame D is scheduled immediately after frame C, and hence incurs no additional tail energy.

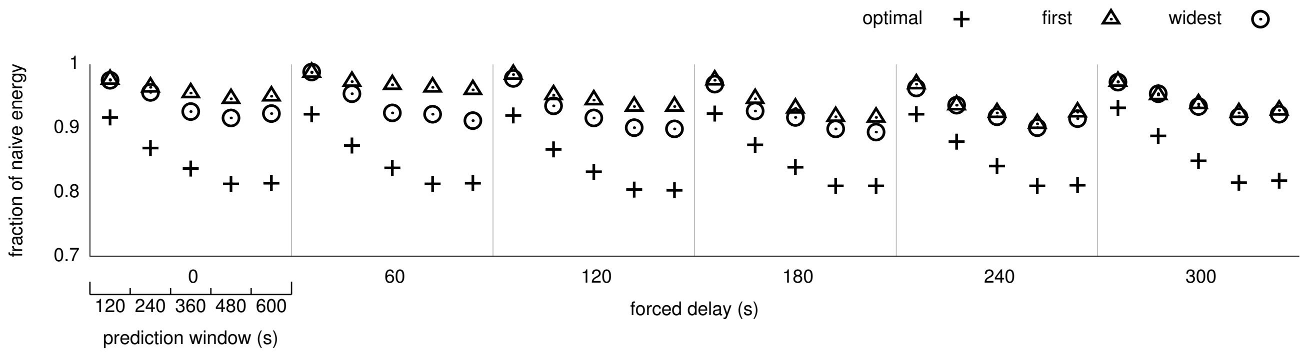

Figure 12: Based on the Pre's radio power profile, predicting signal with previous traces can save energy for email syncs. Median energy for 42 pairs of experiment and training traces with a maximum ten minute scheduling window. Y-axis starts at our expected best savings.

The dynamic programming algorithm that computes the minimum energy schedule is as follows:

Initialization
for $t = 1$ to M **do**
  $E_{0,t} = 0$
end for

Computing optimal schedules
for $k = 1$ to N **do**
  for $t = k$ to M **do**
    $E_{k,t} = \min_{\ell=k-1}^{t-1}(\ E_{k-1,\ell} \ + \ ESlot_{\ell+1})$
    $Last_{k,t} = \ell$ value for which the previous quantity was minimized
  end for
end for

The intuition behind the dynamic programming algorithm is that the minimum energy to transfer k frames in time t, $E_{k,t}$, is simply the minimum of sum of transferring $k-1$ frames in time ($k-1$ to $t-1$) and the cost of transferring the $k'th$ frame in the time remaining, including incurred tail costs, if any. Thus, the optimal substructure property holds and the solution to the dynamic program is the same as the optimal solution. Additional timing constraints for a frame can be easily incorporated in this algorithm by restricting the search for minimum value within the arrival and deadline slots for the frame as follows:

$E_{k,t} = \min_{\ell=Arrival(k)-1}^{Deadline(k)-1}(\ E_{k-1,\ell} \ + \ ESlot_{\ell+1})$

The value $E_{N,M}$ is the minimum energy for the predicted schedule, which can be computed by tracing backwards from $Last_{N,M}$. The order of the above algorithm is $O(M^2 \times N)$.

Finally, the algorithm could suffer from two kinds of errors: 1) the speed of the current drive being different from speed of the previous drive, and 2) the expected throughput at a slot being different from the median throughput in the track. Fortunately, since the device continues to remain powered on for running this application, we can simply re-run the dynamic programming algorithm from the point of discrepancy and recompute the optimal schedule. In our evaluations, this recomputation helped avoid significant deterioration in energy savings on account of the above errors.

6. SIMULATION-BASED EVALUATION

In this section, we simulate Bartendr with the 17 km tracks shown in Figure 3. While driving these tracks we collected the Palm Pre's signal strength and throughput. We model the phone's power with the measurements shown in Figure 1. The advantage of using a simulator is that we can compare the performance of different approaches against each other as well as compare their performance against an optimal algorithm that has full knowledge of future signal.

6.1 Syncing

In this section, we show that the variation of signal is amenable to energy savings by an optimal algorithm and that the prediction algorithms (first and widest, Section 5.2) are able to approach that optimal reduction. We run the simulation on all pairs of seven 17 km tracks (training and experiment). Although likely valuable, we do not yet have a scheme for combining different tracks to form a refined track model.

At every ten second interval of the experiment track, we execute the prediction given two constraints: the *forced delay* represents the minimum time that must be slept before performing a sync, while the *prediction window* represents the interval of time after the forced delay where a sync may occur. The decomposition into these two parameters allows us to model various application-level choices about how to constrain the schedule, and provides information about how much latitude is required and whether prediction accuracy degrades with time.

We assume all syncs take a fixed time of 10 seconds (the median observed in Section 3.1). It is possible that syncs can take more or less time because of latency variations.

Figure 12 presents the total sync energy for *widest*, *first*, and optimal. It shows the sync energy for these techniques relative the naive approach (always sync immediately after the end of the forced delay period, equivalent to a prediction window of zero). Optimal scheduling can provide up to 20% sync energy savings, when it is possible to choose over prediction windows longer than six minutes. Optimal uses future knowledge of the current track to make decisions, and will always choose a low-energy period if one is available. We note from this graph the potential energy savings resulting from increased flexibility in scheduling communication events.

The *widest* scheduling approach generally outperforms *first*, offering up to 10% reduction in sync energy. We suppose that the advantage of widest is due to short-term variations in mobility that do not accumulate over long enough intervals. That is, the variations may cancel each other out.

While these savings are modest, they represent savings for the case where the sync operation does not result in downloading of any content. Thus, the energy savings are only due to reduced transmission power from good signal locations and does not benefit from better data rates available at these locations. When the sync operations result in downloading of updated content, the higher data rates available at good signal locations should substantially improve energy savings (see next section on scheduling streaming).

Finally, we expect that the effectiveness of these scheduling approaches could be improved by better aggregation of training data (omitting track 5 seen in Figure 3 in particular alters the results substantially). Further, by explicitly considering variations in mobility we would have further potential to increase scheduling accuracy. We discuss a particle-filter-based approach for doing so in Section 8.2.

6.2 Streaming

We compare the performance of a naive approach to Bartendr's signal-based scheduling algorithm for the Streaming application through data driven simulations. Again, we use the data collected from real driving experiments, which consist of signal strength, instantaneous throughput and power consumption measurements while receiving TCP streams over several drives, to drive the custom simulator. We analyze the energy consumed to download streams of varying bitrates (corresponding to popular audio/video encoding rates) and varying stream lengths (120 s to 600 s of play length). The bitrates play a role in how fast the application consumes the downloaded data, and thus impacts the how long a data frame in the stream can be delayed. The stream length determines the total number of data frames in the stream that need to be downloaded.

In the naive case, all data frames in the stream are downloaded in one shot from the beginning of the stream until completion. In the signal-based scheduling algorithm, we plugin the real signal values from the track. We then map these signal values to median throughput and power consumption numbers from prior tracks corresponding to the same drive, which are used in the computation of $ESlot_\ell$ values. For each frame in the input stream, we also compute the deadline before which it needs to be scheduled for transmission based on the stream bitrate. We illustrate how the deadlines are calculated through an example. Consider a 5 min audio stream encoded at 128 Kbps (total size $\approx$ 5 MB), and assume the number of data frames N to be 25 (frame size $\approx$ 200 KB). Since the application consumes data at an average rate of 128 Kbps, at most one data frame is needed every 12 seconds to ensure that the playout buffer doesn't run out of data. Thus, in this example the deadlines for each consecutive frame occurs every 12 s within the 5 min window. After computing the deadlines, we then run the dynamic programming algorithm to compute the schedule for downloading each frame. During the execution of the schedule, if we find that there is a deviation in the actual and expected throughput, we rerun the dynamic programming algorithm with an updated number of frames to be scheduled in the reminder of the interval along with their corresponding deadlines.

To implement the signal-based approach, we need to be able to start and stop the stream download based on the

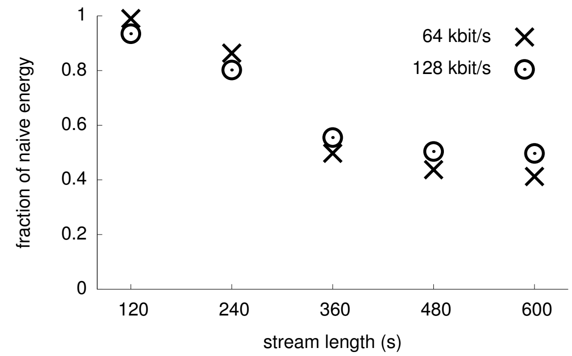

Figure 13: Energy savings with signal-based scheduling for 64 and 128Kbps data streams.

schedule computed by the dynamic programming solution. We achieved this using a network proxy that starts to download the data from the streaming server as soon as the client initiates a request. The schedule basically consists of start and stop times for downloading each frame from the proxy.

We stagger the start to arbitrary times in the tracks and present the average results for over hundred runs. Figure 13 plots the energy savings of the signal-based scheduling schemes compared to the naive case. As the stream length increases from 120 s to 600 s, there are more number of data frames that are farther away from the start of the stream. These frames have longer deadlines and provide more opportunities to search for energy efficient time slots to download them within their specified deadlines. We see that energy savings of up to 60% are achievable by scheduling using the Bartendr framework.

7. RELATED WORK

Several studies have results that are relevant to the Bartendr theme, including: mobile prediction of wireless network quality, stability of ubiquitous wireless network quality measurements, scheduling and other approaches for mobile energy savings.

7.1 Predicting wireless network quality

The goals of Breadcrumbs [16] closely resemble those of Bartendr: predict network quality at a physical location, and use this knowledge to make applications use the network more efficiently. However, the two systems differ in many ways.

Bartendr seeks to provide energy savings on ubiquitous cellular networks, while Breadcrumbs is tailored to wireless LANs. While Breadcrumbs indexes WiFi bandwidth availability by GPS location (similar to the work of [18] where location is a key context used in determining whether to scan for WiFi availability), we find that for cellular networks, *location coupled with direction of arrival* is necessary for predicting signal strength, and thus available bandwidth.

Furthermore, because of the dynamic nature of wireless LANs, Breadcrumbs requires an energy consuming measurement framework that periodically scans for WiFi bandwidth availability to predict network quality at a location. Bartendr leverages the fact that cellular signal strength information can be obtained very inexpensively on the mobile (since the cellular radio must remain on to receive phone calls) to schedule communication at appropriate locations.

7.2 Stability of cellular signals

Recent studies have quantified the service-level stability of ubiquitous wireless networks [14, 20]. They found that bandwidth changes over time, even though signal strength measurements remain stable. In the earliest study, Tan et al. discovered that over three weeks, even though signal strength measurements were stable at a location, network bandwidth measurements vary by 54% [20]. Later, Liu et al. [14], using more detailed measurements of an EVDO network, found that signal strength and the downlink data rate used by the provider were highly correlated over long timescales. However, they also noticed that over about a months time, the rates fluctuate between 500Kbps and 3Mbps at a given location.

Given this information, can an approach like Bartendr still be effective? First, since we find that energy efficiency is dependent on signal strength, relative signal strength stability and high long term correlation with data rates is certainly helpful. Second, while some bandwidth availability variation is to be expected on a commercial network with competing users, applications like email or RSS feeds that we expect to benefit from Bartendr do not require the maximum throughput of the link.

7.3 Energy-efficient cellular data scheduling

The proportional fair scheduler [11] used in 3G networks today already uses signal conditions at the mobile nodes to preferentially schedule nodes with higher signal strength. Bartendr uses the same principle of channel state-based scheduling but differs from the proportional fair scheduler in two ways. First, while proportional fair schedules traffic at a fine granularity of milliseconds, Bartendr schedules traffic over time intervals of tens of seconds to several minutes or more. Thus, while proportional fair is *reactive*, relying on continuous feedback of channel state from the mobile nodes, Bartendr must *predict* future channel state in order to schedule effectively. Second, proportional fair is used today only to schedule traffic on the downlink while Bartendr schedules both uplink and downlink traffic, leveraging a network-based proxy for buffering downlink traffic.

Another aspect unique to cellular radios is the tail energy overhead, i.e., the lingering of cellular radio in a high power state for a few seconds after each communication episode. Approaches like TailEnder [3] and Cool-Tether [19] create energy savings by reducing the cellular *tail* overhead; TailEnder performs batching and prefetching for email and web search applications, respectively, while Cool-Tether performs aggregation for web browsing. In contrast to these approaches, Bartendr takes into account both the tail overhead and signal strength information for saving energy.

8. DISCUSSION

We now discuss a few potential enhancements to Bartendr.

8.1 Alternate models of energy conservation

Based on the energy conservation potential presented in this paper, it is worth considering whether the phone-based logic we use to demonstrate feasibility is in fact the correct architecture for taking advantage of signal variation. Here, we discuss two alternate models: first, what could be possible if simple programmability were exposed by the (always-active) radio firmware; second, what designs might be reasonable with provider cooperation.

Bartendr, for synchronization, attempts to carefully choose when to awaken the processor and radio to perform communication when signal is strong. The radio logic on a mobile phone, however, is constantly active and always able to determine signal quality, even when the processor is inactive. A straightforward approach, then, would be to use this information to awaken the processor, not just after a sleep interval has elapsed or on receiving a phone call, but also whenever signal quality exceeded some threshold. With some potentially simple rules, one might be able to perform communication as soon as signal is good, without a need to predict variation. Such a threshold-based approach does little for planning in the context of streaming applications, but could augment the gain in very low power operation. One might imagine that an accelerometer could provide additional information about mobility, or that the history of signal measurements could be recorded without processor involvement to generate training data. These operations, of course, would require slightly more work, but would be valuable in improving predictions made on the device.

A second approach to positioning and prediction may simply be to place the prediction logic at the provider. Providers are reasonably aware of the paths taken by clients and can observe per-client signal strength variation. Such observations have led to research in better selection of handoff locations [13], and other provisioning scenarios. It could be reasonable to migrate the logic of signal strength variation prediction to the well-provisioned side of the wireless link. To place this logic at the provider might also enable exploitation of information from diverse users who follow similar paths. Cellular data providers have a direct interest in providing such a service since Bartendr helps conserve spectrum: using faster communication with hosts that are near a base station avoids using slower modulations and data rates for the same effective service, permitting service to more subscribers. At the same time, extending battery life for subscribers may aid to the service they provide.

8.2 Particle filters

The challenge in predicting signal variation comprises two interrelated problems: determining current location along a trace and predicting future locations. Inspired by research literature [9, 15], we applied a particle-filter-based approach to both problems, intending to explicitly capture the uncertainty in measurement and prediction. Inferring position based on cell signal information alone, while avoiding power-hungry GPS, had the potential to yield many candidate locations of comparable likelihood. If so, particle filters would help to predict signal based on all possible current locations, which would yield a better estimate of sleep time. Similarly, predicting movement along a track despite uncertain speed and stop durations would be facilitated by particle filters, since different probabilities can be assigned for faster and slower movements than the baseline training data.

In our experience, somewhat surprisingly, particle filter approaches did not yield better outcomes. For the problem of determining current location, the closest match by signal already provided us a reasonably accurate estimate of location; the uncertainty captured by particle filters did not turn out to be useful beyond this. For the problem of predicting future locations, it turned out that mobility along our automobile-based commute paths was not deterministic enough—speeds and stop durations varied—to make any algorithm significantly better at prediction than any other. Completely deterministic progress would mean that a simple model of matching progress through the training data will work well. Somewhat deterministic progress would enable particle filters to outperform. The wide variation in progress through the training data that we observed did not permit particle filters to yield significantly better selection of good signal periods. Thus, the added computation cost was also not compensated. Despite this negative result, we believe that particle filters can be useful in other scenarios. For example, in order to resolve conflicting position information or to help incorporate low energy accelerometer data into the scheduling algorithms.

9. CONCLUSION

Signal strength has a direct impact on cellular radio energy consumption, which by far dominates the base energy consumption of mobile devices such as smartphones. The variations in cellular signal strength as a user drives around coupled with the presence of flexible applications, such as email syncing, photo sharing, and on-demand streaming, presents a significant opportunity to save energy. We have presented Bartendr, a practical framework for scheduling application communication to be aligned with periods of good signal. Bartendr addresses a number of challenges and makes novel contributions, including track-based, energy-efficient prediction of signal strength and a dynamic programming-based procedure for computing the optimal communication schedule. Our simulations demonstrate significant energy savings of up to 10% for email sync and up to 60% for on-demand streaming.

In future work, we plan to investigate more sophisticated approaches to signal strength prediction and, separately, the design of appropriate APIs for exposing radio energy cost to applications that wish to make an intelligent choice.

10. ACKNOWLEDGMENTS

We thank our shepherd Prasun Sinha and the anonymous reviewers for their insightful comments and discussion. Additionally, we thank the summer 2009 interns of Microsoft Research India for many beneficial discussions. We extend our gratitude to the cab drivers at Microsoft Research India for their assistance with the data collection. This work was supported in part by NSF Awards CNS-0643443 and CNS-0626629.

11. REFERENCES

[1] G. Ananthanarayanan, M. Haridasan, I. Mohomed, D. Terry, and C. A. Thekkath. StarTrack: A framework for enabling track-based applications. In *ACM MobiSys*, 2009.

[2] Power consumption and energy efficiency comparisons of WLAN products. White paper, Atheros Communications, Inc., 2003.

[3] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile smartphones: A measurement study and implications for network applications. In *ACM IMC*, 2009.

[4] M. Chiang, P. Hande, T. Lan, and C. W. Tan. Power control in wireless cellular networks. *Foundations and Trends in Networking*, 2008.

[5] I. Constandache, R. R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *IEEE INFOCOM*, 2010.

[6] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. C. (Duke). EnLoc: Energy-efficient localization for mobile phones. In *IEEE INFOCOM Mini-conference*, 2009.

[7] S. C. Cripps. *RF Power Amplifiers for Wireless Communications*. Artech House, 2006.

[8] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *IEEE INFOCOM*, 2001.

[9] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation and tracking. *T-SP*, Feb 2002.

[10] J. Krumm and E. Horvitz. Predestination: Where do you want to go today? *IEEE Computer Magazine*, Apr 2007.

[11] H. Kushner and P. Whiting. Convergence of proportional-fair sharing algorithms under general conditions. *IEEE Transactions on Wireless Communications*, July 2004.

[12] A. LaMarca et al. Place Lab: Device positioning using radio beacons in the wild. In *IEEE Pervasive Computing*, May 2005.

[13] T. Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless atm networks. *IEEE JSAC*, Aug 1998.

[14] X. Liu, A. Sridharan, S. Machiraju, M. Seshadri, and H. Zang. Experiences in a 3G network: interplay between the wireless channel and applications. In *ACM MobiCom*, 2008.

[15] L. Mihaylova, D. Angelova, S. Honary, D. R. Bull, C. Canagarajah, and B. Ristic. Mobility tracking in cellular networks using particle filtering. *IEEE Transactions on Mobile Computing*, Oct 2007.

[16] A. J. Nicholson and B. D. Noble. Breadcrumbs: Forecasting mobile connectivity. In *ACM MobiCom*, 2008.

[17] G. P. Pollini. Trends in handover design. *IEEE Communications Magazine*, Mar 1996.

[18] A. Rahmati and L. Zhong. Context-for-wireless: Context-sensitive energy-efficient wireless data transfer. In *ACM MobiSys*, 2007.

[19] A. Sharma, V. Navda, R. Ramjee, V. Padmanabhan, and E. Belding. Cool-Tether: Energy efficient on-the-fly WiFi hot-spots using mobile phones. In *ACM CoNext*, 2009.

[20] W. L. Tan, F. Lam, and W. C. Lau. An empirical study on 3G network capacity and performance. In *IEEE INFOCOM*, May 2007.

[21] N. D. Tripathi, J. H. Reed, and H. F. VanLandingham. Handoff in cellular systems. *IEEE Personal Communications*, Dec 1998.