10-2015

Smartphones and BLE Services: Empirical Insights

Meera RADHAKRISHNAN
Singapore Management University, meeralakshm.2014@phdis.smu.edu.sg

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

Rajesh Krishna BALAN
Singapore Management University, rajesh@smu.edu.sg

Youngki LEE
Singapore Management University, YOUNGKILEE@smu.edu.sg

DOI: https://doi.org/10.1109/MASS.2015.92

Citation
RADHAKRISHNAN, Meera; MISRA, Archan; BALAN, Rajesh Krishna; and LEE, Youngki. Smartphones and BLE Services: Empirical Insights. (2015). *IEEE International Conference on Mobile Ad Hoc and Sensor Systems MASS 2015: October 19-22, Dallas, Texas: Proceedings*. 226-234. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/3119

Published in IEEE International Conference on Mobile Ad Hoc and Sensor Systems MASS 2015: October 19-22, Dallas, Texas: Proceedings. pp. 226-234.
https://doi.org/10.1109/MASS.2015.92

Smartphones & BLE Services: Empirical Insights

Meera Radhakrishnan, Archan Misra, Rajesh Krishna Balan, and Youngki Lee
School of Information Systems, Singapore Management University
80 Stamford Rd, Singapore 178902
Email: {meeralakshm.2014, archanm, rajesh, youngkilee}@smu.edu.sg

***Abstract*—Driven by the rapid market growth of sensors and beacons that offer Bluetooth Low Energy (BLE) based connectivity, this paper empirically investigates the performance characteristics of the BLE interface on multiple Android smartphones, and the consequent impact on a proposed BLE-based service: continuous indoor location. We first use extensive measurement studies with multiple Android devices to establish that the BLE interface on current smartphones is not as "low-energy" as nominally expected, and establish that continuous use of such a BLE interface is not feasible unless we choose a moderately large scan interval and a low duty cycle. We then explore the implications of such constraints, on the parameters of a smartphone's BLE stack, on the accuracy of a BLE-based indoor localization techniques. We show that while RF-based indoor location can be highly accurate (80% of estimates have errors less than or equal to 4 meters) for stationary users only if the density of beacons is high, the combination of (large scan interval, low duty cycle) causes the location error to degrade significantly for moving users. These results provide practical insights into the use cases and limitations for future BLE-based mobile services.**

I. Introduction

Bluetooth Low-Power (*BLE*) based devices have gained significant attention and popularity recently, especially after Apple™introduced built-in support in iOS7 for their iBeacon™protocol, which allows BLE based beacons to be used for *proximity-driven* location-based services. BLE is designed specifically to support shorter-range, but much lower energy-overhead and lower latency communication, and is targeted towards sensor & IoT devices.

Many BLE-based applications use the smartphone as a *querying* device, that either *scans* for advertisements (beacons) from nearby BLE devices, or *connects* to such BLE devices to retrieve or exchange information. Motivated by such use cases (e.g., indoor location tracking [14]), we explored the possible issues that may arise from the continuous use of BLE scanning on commercial smartphones, and how additional deployment/environmental artifacts (such as the beacon deployment density or the movement speed of individuals) affect the overall performance metrics. More specifically, our experimental investigations were motivated by our preliminary observations (in [15]) on the **asymmetry of the BLE energy overhead**: devices advertising their IDs or profiles consume much less power, whereas the act of *scanning* (for potential nearby devices), by state-of-the-art mobile devices, is significantly more energy-intensive.

Overall Goals: In this paper, we study two key issues:

- *Energy and Packet Loss Tradeoffs:* We develop a deeper experimental understanding of how parameter settings on a smartphone's BLE stack affect critical performance metrics, such as energy overhead, beacon miss rate, etc., under various environmental conditions. More importantly, we perform fine-grained, accurate power measurements (using a Monsoon power monitor), in contrast to previously reported Web articles [2] that use software-based battery drain estimates (which are known to be inaccurate due to the well-known non-linear drain characteristics of smartphone batteries [13]).
- *Indoor Location:* We focus on relatively long-lived continuous indoor location tracking (especially in places such as shopping malls & retail supermarkets) as a popular use case for BLE beacons, and understand how the beacon deployment density and the smartphone BLE settings collectively affect the resulting *location accuracy*.

To understand these issues, we conducted extensive measurement based studies on our university campus, using the commercially-available Estimote™BLE beacons [1]. We studied variations across 3 different Android devices (Samsung's Galaxy S5, S3 and Note 3 devices), each with a different Android version and BLE chipset, and across different environments/ layouts (e.g., different occupancy levels, locations, etc.). (Additional comparative studies with iOS are deferred for later, as the studies would require jailbroken iPhones because the iOS APIs do not currently permit user-driven setting of various parameters).

Key Findings & Contributions: Our empirical studies provide the following key insights:

- **Higher-energy overhead on commodity smartphones:** We first confirmed that continuous BLE-scanning, as a background service, does indeed incur high energy overhead (power consumption of 200mW or higher) on Android devices; in fact, it is just as expensive as performing active sensing (e.g., collecting gyroscope data). We then investigated the impact of the following two key BLE parameters on the smartphone's energy overhead: (i) the *scan interval (T_s)*, which determines the periodicity with which scanning is performed, and (ii) the *duty cycle* (f_D), which indicates the fraction of time within each T_s period that a phone actively scans. Our investigations show, on Android devices, that the energy overhead is (as expected) proportional to the duty cycle, but increases rapidly for shorter scan intervals (below 1 sec).
- **Marginal benefit for BLE vs. classic-BT (as presently implemented):** We studied the relative energy efficiency of the more recent BLE stacks on Android smartphones, compared to the default implementation of the older

Technical Spec.	Classic	BLE
Frequency	2400 to 2483.5 MHz	2400 to 2483.5 MHz
No. Data Channels	79	37
No. Advertising Channels	32	3
Encryption	64 / 128 bit	AES 128 bit
Range	100 m	$>$ 100 m
App Throughput	0.7–2.1 Mbit/s	$<$ 0.3 Mbit/s
Latency to Connect	≈100 ms	6 ms
Min. Time to Send Data	100 ms	3 ms
Power Consumption	1 W (reference)	0.01 to 0.5 W
Peak Current	22–40 mA	10–30 mA

TABLE I. CLASSIC BLUETOOTH VERSUS BLE [12], [16]

"classic-Bluetooth" (*classic-BT*) protocol on identical devices. Surprisingly, the performance savings turned out to be minimal, with the use of BLE saving less than 10% power, compared to the classic-BT implementation.

- **Moderately-high beacon miss rates:** We also investigated how the combination of (T_s, f_D) values affected the beacon miss rate (the percentage of transmitted beacons that a smartphone fails to receive). Overall, the beacon miss rate is (as expected) inversely proportion to the duty cycle, and insensitive to the scan interval. However, the key finding is that this beacon miss rate increases when the number of in-range beacons increases, and is *significantly higher in the presence of additional BLE devices* (reaching as high as 35% under continuous scanning in our experiments).
- **Establish parameters for sustainable continuous operation:** Given the observed energy vs. miss-rate tradeoffs, we derive a set of "sustainable" choices of (T_s, f_D) for continuous "full-day" BLE operation, such that the resulting energy drain is at most 30% of the typical energy drain observed in daily consumer smartphone usage (usage numbers derived from 132 *LiveLabs* [5] users). Very specifically, if BLE scanning is required to be continuously active, we see that the scan interval for Android devices can be lowered only to around 5secs, as more frequent scanning results in sharp spikes in power consumption.
- **Performance of BLE-based indoor location tracking:** We studied how various choices of the BLE parameter settings affect the accuracy of a fingerprinting-based BLE indoor location tracking service, for both *stationary* and *moving* users. We demonstrate that accuracies of approx. 2 meters are possible for stationary users only under *dense BLE-beacon deployments* (in our case, 9 beacons within a $66m^2$ area). Moreover, due to the higher beacon loss rates associated with sustainable choices of (T_s, f_D), the accuracies degrade, to a median location error of ± 9 meters, for *moving* users.

Our goal is to inform the community about the practical tradeoffs and design choices that arise when using BLE for *continuous* smartphone-based applications and services.

II. QUICK INTRODUCTION TO BLE

BLE, as briefly explained earlier, is a modification to the standard Bluetooth protocol to allow for short range, low bandwidth, low latency, very energy efficient communication. BLE was driven by the realization that a large class of devices are appearing on the market that periodically broadcast small amounts of data to the surrounding region. These devices include proximity beacons (used for indoor location), heart monitors (used for health monitoring), and other IoT type devices that sense and broadcast modest quantities of information periodically (e.g., the level of water in a reservoir). These types of devices have very tight energy budgets as they need to be small (thus only small batteries are possible), and need to last without replacement for a long time (months or even years).

To enable this, BLE makes 3 fundamental changes compared with the standard Bluetooth protocol. First, BLE uses less channels (40) compared to Bluetooth (79). Each channel is also large (2 MHz in width versus 1 MHz) giving BLE a larger range than standard Bluetooth. In addition, BLE uses just 3 channels to scan for other BLE devices (versus regular Bluetooth which uses 32 channels). These 3 channels are also chosen to not interfere with either Bluetooth or Wi-Fi. Using just 3 channels allows BLE to scan for devices extremely quickly (taking just 6 ms to finish a scan versus 100 ms for regular Bluetooth). In addition, BLE uses a different connection protocol that allows it to receive data even without connecting to the sending device. This allows BLE receivers to receive data very quickly (data is received during the 6 ms scanning window effectively) while allowing BLE receivers to save energy by not needing to connect to data receivers.

Second, BLE enforces strict energy controls through heavy duty cycling of the sending device and by optimizing the transmission protocol for small amounts of data (the normal use case for most BLE devices). Third, BLE focuses on specific application use cases that rely on short periodic transmissions of data. This is unlike regular Bluetooth which also supports functionality, such as pairing of a Bluetooth headset, that requires continuous high bandwidth data. Table I shows a more detailed comparison of BLE with Classic Bluetooth; additional details can be obtained from various web presentations [12], [16].

III. EXPERIMENTAL METHODOLOGY

In this paper, our specific focus is on obtaining a deeper understanding of how the key parameter settings in a smartphone's BLE stack affect higher-level performance metrics, and how this effect is moderated by environmental factors, such as the density of BLE beacons, the presence of additional BLE transceivers or the indoor movement speed of the user.

A. Devices Chosen

To conduct a thorough study, we varied the parameters of the BLE stack on 3 different Android-based smartphones. Coincidentally, the phones differed both in the underlying Bluetooth chipset, as well as the deployed OS version. This allowed us to determine, especially for our energy efficiency studies, if the results were artifacts of specific or older hardware, firmware or versions, or are instead broadly representative of currently available commercial smartphones. Table II lists the various devices, summarizing the OS choice, the chipset use and the

Smartphone	OS version	Bluetooth Chipset	Key Features
Samsung S3	Android 4.3	Broadcom BCM4334	Bluetooth 4.0 +HS; the first BLE specification
Samsung S5	Android 4.4.2	Broadcom BCM4354	Bluetooth 4.0+ HS
Samsung Note3	Android 4.4	Broadcom BCM4339	Bluetooth 4.0+ HS
iPhone 5S (illustrative only)	iOS 7.1.1	Broadcom BCM43342	Bluetooth 4.0 +HS

TABLE II. DIFFERENT BLE SMARTPHONES USED

notable differences across the devices. (Even though we do not conduct detailed experiments with iPhones, the iPhone 5S specifications are provided for illustration. Notably, the iPhones seem to be using comparable Broadcom chipsets, and thus are likely to exhibit the same operating characteristics as our representative Android platforms.)

B. Experimental Methodology

For all our studies on Android devices, we implemented our own code, using the Android Bluetooth Low Energy developer specifications, that allowed us to modify the T_s and f_D values of the BLE stack. More specifically, every scan interval, we used the *startLeScan* method on the *BluetoothAdapter* object to initiate a scan, that was then stopped after the active scan duration (i.e., $T_s * f_D$) had elapsed. Note also that the specs indicate that, even within an active scan duration, the Android BLE implementation might perform its own micro-duty cycling (by default operating in LOW_POWER mode, while also allowing the developer (in more recent revisions) to specify alternative LOW_LATENCY or BALANCED modes). The API we used did not allow us to change this default LOW_POWER mode; instead, our f_D parameter refers to the *macro* (or application-level) duty cycle that typically turns the BLE adapter ON/OFF over coarser and longer timescales. The average power consumption values are obtained by removing the battery and connecting the smartphone directly to the Monsoon Power Monitor. Unless specifically mentioned, the readings provided refer to the mean power readings obtained across 3 separate test episodes, with each episode lasting for 10 mins (the average power of each episode is provided directly by the Monsoon monitor).

Screen and foreground/background settings: As part of our experimental studies, we discovered (discussed later in Section V) that the power consumption of the BLE stack itself is affected by two additional choices: (i) whether the phone's screen is ON or OFF, and (ii) whether the BLE scanning App is run in the foreground or background. As the most common usage model for BLE-based services envisions a continuously running background App, the vast majority of our studies are conducted with the screen turned OFF, and the App running as a background service.

IV. BLE PARAMETERS & BEACON MISS RATES

In this subsection (and the next), we report on studies that understand the impact, on the beacon *miss-rate* (i.e., the percentage of beacon readings missed by a smartphone), of the following parameters of the smartphone BLE stack:

- *Scan interval (T_s)* The scan interval is the basic period of BLE scanning: every T_s seconds, the phone will activate its BLE interface and perform scanning for a period of time less than or equal to T_s (as decided by the duty cycle).
- *Duty cycle (f_D):* The duty cycle is the fraction of time within a scan interval that the phone stays active, i.e., it repeatedly cycles through the 3 advertisement channels, looking for advertisements from BLE beacons. In alternative formulations, the duty cycle is implicitly defined via a *scan period*, the contiguous duration (within each T_s) that the BLE interface is active, i.e., $f_D = \frac{\text{scan period}}{T_s}$. In general, a smaller duty cycle indicates a proportionately smaller period of the BLE interface being active, and should result in lower energy overhead at the expense of a higher rate of missed beacons.

All measurements reported here use the Estimote^{TM}beacons, running on 32-bit *ARM*®Cortex M0 CPU with 2.4 GHz Bluetooth 4.0 bidirectional radio, set (unless otherwise specified) to a default transmission power (-12dBm) and a 101 msec advertising period.

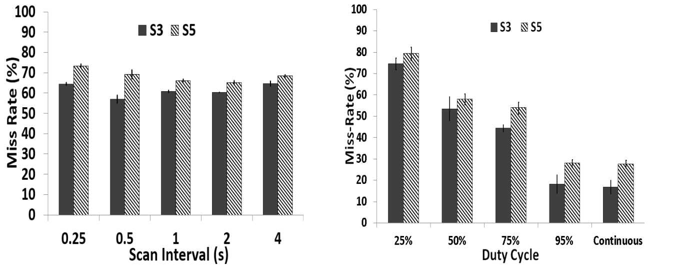
A. RSSI Variation & Miss Rate vs. Distance

We first studied the variation in the beacon miss rate (the fraction of beacon advertisements that were not reported by the phone, as measured over an observation period of 10 minutes), as well as the measured RSSI, as a function of the distance between a beacon and a phone (which was configured to continuously scan for beacons). Table III tabulates both the measured RSSI (the average, as well as std. deviation) of all the beacon readings, as well as the beacon miss rate, as a function of distance. (These studies were conducted in level 4 of our SIS building, where no additional BLE devices (other than our BLE beacons) were found to be active.) It is evident that the RSSI values gradually fall off with distance; more importantly, the packet loss rate is relatively steady at $\approx 15\%$ for a distance of 2 meters or less, but then begins to increase, reaching as high as 27% when the beacon-phone distance is around 5 meters for Galaxy S3. The packet loss rate is $\approx 10\%$ higher with Galaxy S5.

	Device	Distance(m)					
		0.5	1.0	2.0	3.0	4.0	5.0
RSSI (-dBm)	S3	-66.47 (3.73)	-68.15 (2.88)	-71.57 (4.53)	-72.37 (4.34)	-75.18 (4.98)	-78.05 (5.82)
	S5	-77.90 (2.24)	-82.55 (3.33)	-85.05 (4.31)	-85.37 (4.69)	-86.29 (5.27)	-89.73 (4.72)
Miss-rate (%)	S3	13.38 (0.29)	14.74 (1.43)	16.86 (2.67)	21.76 (1.93)	25.67 (3.98)	27.54 (5.02)
	S5	25.26 (1.34)	26.09 (2.53)	27.51 (1.02)	30.64 (2.58)	35.78 (3.72)	39.46 (5.12)

TABLE III. RSSI VARIATION & BEACON MISS RATE VS. DISTANCE

In addition, we also studied this relationship between distance and RSSI/miss-rate as a function of whether the user was stationary or moving (either slowly or fast). The results indicated that this relationship was effectively *insensitive* to the movement speed–as long as the distance of the phone to the beacon did not change, the packet loss rates were very consistent.



(a) T_s: scan interval ($f_D = 50\%$) (b) f_D: Duty Cycle (T_s=1sec)

Fig. 1. Beacon Miss Rate vs. (T_s, f_D) at 2 meter distance

B. Duty Cycle & Scan Interval Effects

We now report on experiments that studied the impact of changing the phone's operating parameters (in particular, its scan interval and its duty cycle) on the beacon miss rate. Figure 1a) plots the variation (for the S3 and S5 devices) in beacon miss rate as a function of the scan interval (for a duty cycle value of 50%), whereas Figure 1b) plots the dependence of the beacon miss rate on the duty cycle (with the scan interval set to 1sec).

In general, it is seen that the beacon miss rate is independent of the scan interval T_s, except for very small values of T_s (which may relate to the transient period involved in turning the phone's BLE interface on). However, as expected, the miss rate is clearly dependent on the duty cycle f_D: in general, if the interface is active only for an f_D fraction of time, it can be expected to miss (1-f_D) of the uniformly spaced beacons. This pair of observations will become relevant when we later consider the *energy implications* of these parameters–we shall see that larger values of f_D (which provide a greater fraction of beacon readings) is possible only at larger values of T_s, which in turn implies a larger worst-case latency ($T_s * (1 - f_D)$) in obtaining measurements from a specific beacon.

C. The Impact of Multiple Beacons/BLE Devices

The studies in the previous two subsections measured the miss rate associated with a single BLE beacon. However, many use cases (including the localization service to be described in Section VII) typically utilize multiple BLE beacons, often deployed in close proximity to one another. Accordingly, we also studied the impact of the number of neighboring BLE beacons on this beacon miss rate.

Figure 2 plots the beacon miss rate (for the Samsung S5) as a function of the number of proximate beacons (all the beacons are within hearing range of the phone, resulting in a *clique*). We can see that the beacon miss rate increases marginally as the number of BLE beacons is varied between $\{1, 5, 9\}$.

In a separate study, we also studied the beacon miss rate of the smartphone with a single BLE beacon, but with *additional BLE-equipped devices* in the neighborhood. More specifically, we repeated the experiments for a single beacon, in our lab, where several other BLE devices (typically, 6 additional smartwatches, used by other researchers) were active and within earshot. Figure 3 plots the resulting loss rate for different values of f_D. Compared to Figure 1(b), we see that the beacon

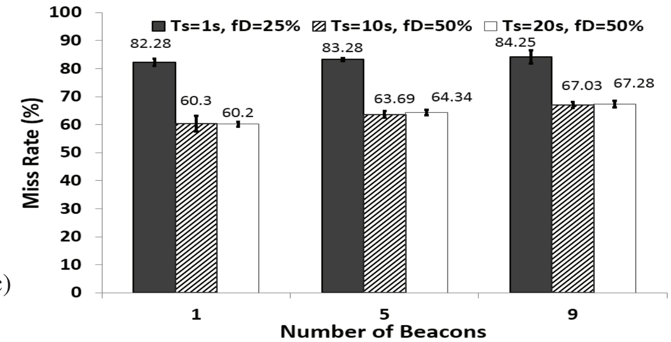


Fig. 2. BLE Miss Rates vs. No. of Beacons (Galaxy S5)

loss rate is significantly higher (almost 10% higher). We thus infer that *the presence of additional interfering BLE devices (which are engaged in active communication) can significantly impair the ability of a scanning smartphone to receive BLE beacons.*

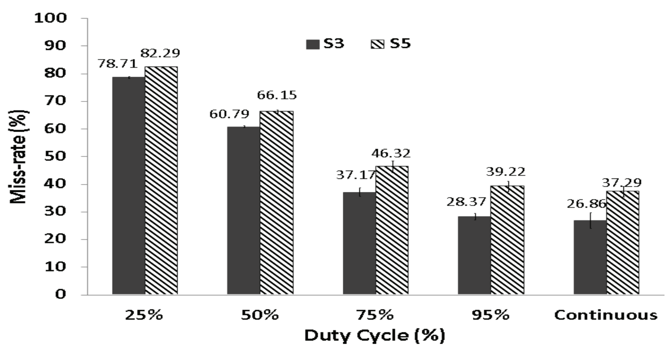


Fig. 3. BLE Miss Rates (BLE-Active Environment), (Ts= 1sec)

V. BLE Parameters & Energy Overheads

We now focus on understanding the impact of the key BLE stack parameters (namely T_s and f_D) on the smartphone's energy overhead. We shall see that, beyond just the values of (T_s, f_D), several considerations affect this power consumption profile, including (a) whether the scanning App runs in the foreground or background (for Android), (b) how many nearby beacons are monitored during a scan, (c) whether the phone performs a *scan* vs. whether it *connects* to the BLE device.

Background Power Consumption: Figure 4 plots the average power consumption of the different Android devices (under continuous scanning). The average power consumption overhead is obtained from the difference between the phone having a background App run in scanning mode vs. not having the App. The figure shows this power overhead for two different configurations: one, where the display is completely turned OFF, and (b) where the display is kept on continuously at maximum intensity. We see that the power consumption with the display being OFF (which would be the typical scenario for both our continuous *Location* tracking scenario) is appreciably higher (by 10-30%). This discrepancy is due to the overheads associated with the *wake lock*–when the display is ON, the phone CPU is already active, and hence the acquisition of the wake lock does not incur any significant overhead. To further scrutinize this discrepancy, Figure 5 plots the power consumption time-series for the display OFF case with T_s=1sec and f_D=50%. We can see that acquiring wakelock consumes ~70 mW, significantly contributing to the power consumption of scanning.

In general, we see that, across all the Android devices, the average power overhead of continuous scanning is around 240mW with screen off. In recent studies, we have shown [4] that the power consumption of individual inertial sensors (e.g., compass, accelerometer, or gyroscope) can consume around 60-140mW on the same commodity smartphones. Hence, our results establish that continuous operation of the BLE stack on the smartphone is *even more expensive than continuous sensing*, and is thus likely to impose a prohibitive energy overhead.

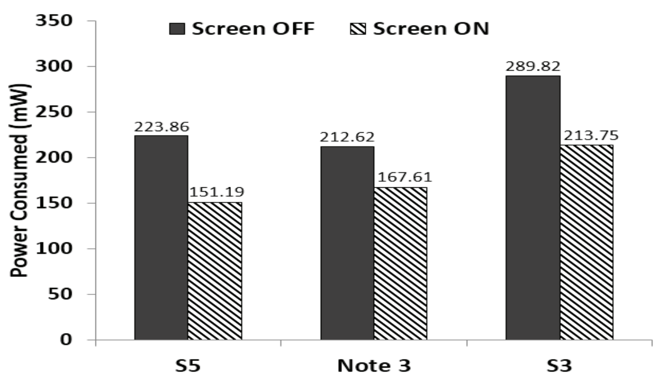


Fig. 4. BLE Scanning Power (Different Devices Continuous Scan)

Note: For the rest of this section, we report the power numbers measured with the screen off, including the baseline power, unless specified otherwise; the baseline power for Galaxy S5, Galaxy Note 3, and Galaxy S3 are 17 mW, 22 mW, and 20 mW, respectively.

A. Energy Overheads vs. (T_s, f_D)

Figure 6 plots the average power consumption of BLE scanning for different values of the scan interval T_s and the duty cycle f_D (for the 3 Android devices). It is clear that the power consumption is dramatically higher for smaller values of T_s; as reported in earlier studies, this is due to the high

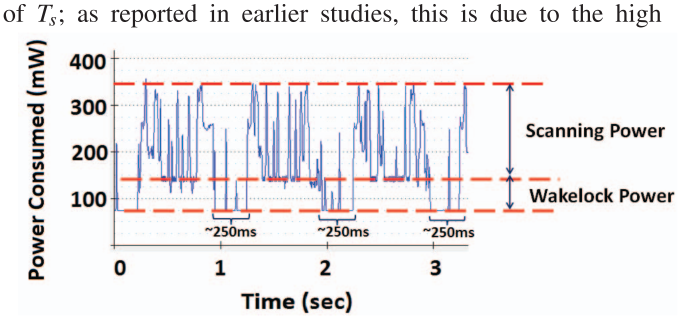


Fig. 5. Power Consumption Plot (Galaxy S5 (Screen OFF), T_s=1s, $f_D = 50\%$)

energy transients associated with turning the BLE interface on and off more frequently. Figure 5 shows the impact of this transient power consumption more vividly. As T_s = 1sec and f_D =50% for the measurement, the non-active cycle (sleep state) is supposed to continue for 500 ms in an ideal case, but the figure shows that the sleep state continues only for around ~250ms. (In fact, the phone never goes back to the sleep state when the scan interval is set to a smaller value such as 250ms.)

More pertinently, the graphs illustrate the tradeoff between T_s and f_D: to maintain a target power consumption rate, one must choose between a smaller (T_s, f_D) (which provides lower latency between successive readings from the same beacon, but misses a greater fraction of beacons) vs. a larger (T_s, f_D) combination (which misses less beacons, but suffers a large worst-case latency).

B. BLE vs. regular BT

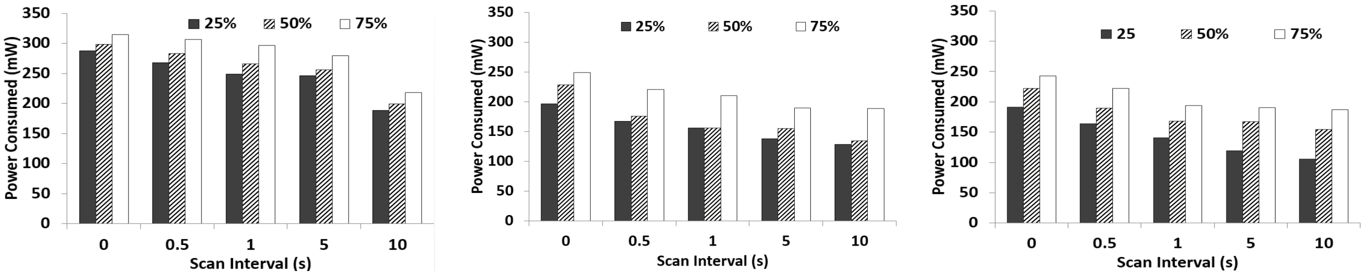
Our measurements also reveal that the power drain associated with the use of the BLE interface on the phone is roughly the same as using the regular (i.e., classic) BT interface! Experimental results suggest that this is because the Android API does not allow an App to turn on the BLE interface exclusively; whenever the BLE interface is activated, the regular BT interface is enabled as well. Figure 7 plots the average scanning power overhead (for T_s = 20secs and $f_D = 0.5$) for the 2 different Android devices (S5 and Note 3), for the use of the BLE vs. BT interface. Compared to classic BT, BLE is seen to save only approx. 20 mW (or $\sim$ 10%) power. Thus, *while BLE saves significant energy on the Estimote beacons, Android-based smartphones currently do not benefit from this "low-energy" protocol stack.*

C. Scan vs. Connect

We also investigated two alternative modes of BLE interaction on the Android devices: (i) *Scan*, where the phone's BLE radio sequentially scans the 3 BLE channels for broadcast Advertisement messages, and (ii) *Connect*, where the phone explicitly makes a connection to a specific BLE device and exchanges communication messages (BLE ostensibly for faster and simpler connections). This study was performed keeping the phone screen on as the current Estimote Android SDK does not provide the support to establish a connection to the beacon in the background. So the BLE connect power numbers are compared vs. the BLE continuous scan power with screen on. The numbers given in Table IV are after subtracting the baseline power for screen on, which is about 550 mW for all the 3 Android devices. The results, however, shows that the power consumption for establishing a connection is 1.5–2 times higher than an equivalent scanning operation for all 3 Android devices. Thus BLE-Connect does *not* provide a more energy-efficient way for the phone to retrieve data from previously known BLE beacons (for example, for indoor location tracking).

D. Impact of Multiple Beacons

We additionally repeated all of the experiments above by varying the number of beacons located within range of the



(a) Galaxy S3 (b) Galaxy S5 (c) Galaxy Note 3

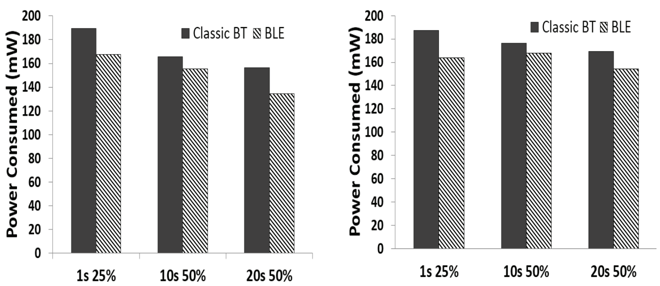
Continuous Scan Power = 311.615 mW (Galaxy S3); 240.583 mW (Galaxy S5); 234.83 mW (Galaxy Note3)

Fig. 6. BLE Power Consumption for Different (T_s, f_D) combinations

Smartphone	BLE Scan (mW)	BLE Connect (mW)
Samsung S5	151.19	297.33
Samsung Note3	167.61	321.72
Samsung S3	213.752	351.167

TABLE IV. SCAN VS. CONNECT ENERGY (SCREEN ON)

phone. Figure 8 plots the power consumed by Samsung Galaxy S5 for scanning at three different settings (T_s=1sec, f_D=0.25, T_s=10sec, f_D=0.50 and T_s=20sec, f_D=0.50) when the number of beacons is varied between 1,5,9. We can see that the power consumed only marginally increases with increase in the number of beacons. Similar behavior was observed for Galaxy Note 3 and Galaxy S3 devices (results are therefore omitted) suggesting that Android devices are robust to the number of beacons in terms of the power consumed.



(a) Galaxy S5 (b) Galaxy Note 3

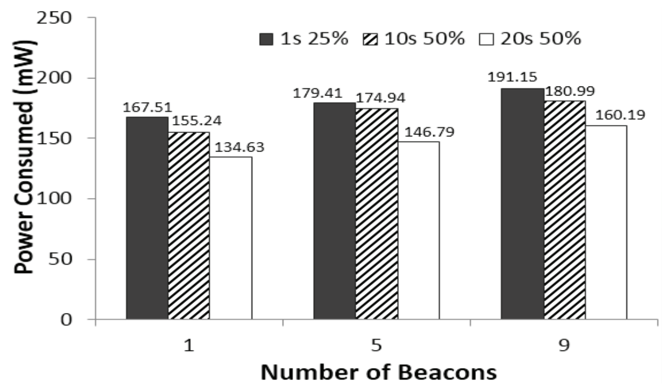
Each bar is a different (T_s, f_D) combination (as stated)

Fig. 7. Scanning Power (BLE vs. BT)

VI. (T_s, f_D) CHOICES FOR CONTINUOUS OPERATION

The experimental studies in the two previous sections quantitatively establish the tradeoff between energy overhead and beacon miss-rate, that results from different choices of (T_s, f_D). Given our intended use of BLE beacons for continuous monitoring, we set ourselves a target of ensuring that our App (running in the background) does not cause an energy drain that is more than 30% of the typical daily battery drain observed by a user.

To obtain the typical battery drain, we analyzed the battery profile of 132 Android users belonging to our LiveLabs testbed over a period of 1 month. Our analysis showed a 95^{th} percentile drain rate of approx. 50% of the battery capacity. Translating



Each bar is a different (T_s, f_D) combination (as shown in legend)

Fig. 8. Power Overhead vs. No. of Beacons (Galaxy S5)

this into a requirement that a continuously running BLE-leveraging App should thus not drain more than 15% (30% of 50) of the battery capacity over a 12 hour typical day, we arrive at a lower lifetime bound of approx. 3 days (72 hours). Figure 9 plots for Samsung Galaxy S5, the battery lifetime values associated with each of the (T_s, f_D) combinations, compared to the 72 hour lower bound. From the figure, we see that the most 'aggressive" parameter values that satisfy this lifetime bound are $T_s = 5secs, f_D = 50\%$ (obviously, larger values of T_s will exceed this bound). Accordingly, for "sustainability", we indicate that the feasible set of (T_s, f_D) values may be one of the representative set: (20secs, 50%), (10secs, 50%) or (5secs, 50%).

The permitted choices can, obviously, be more aggressive if we reduce the overall daily duration for which the BLE interface is active. For example, if the BLE interface can be triggered to be active only 4 hours/day, the overall lifetime bound would effectively reduce to 18 hours and allow the BLE stack to be operated more continuously (e.g., $(T_s, f_D) = (0.5sec, 75\%)$). The key insight thus is: *pervasive applications must have a way of intelligently determining when the BLE interface is to be activated or de-activated, and thus bound the overall daily period of active BLE operation.*

VII. INDOOR LOCALIZATION

We now study the implications that these insights into the operating parameters of the smartphone BLE stack have on a very common BLE-based service: *continuous indoor location*.

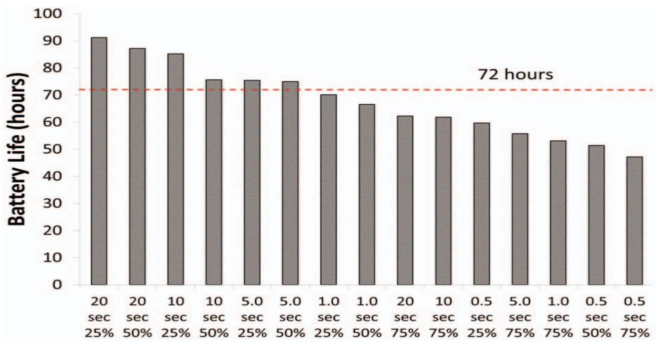


Fig. 9. Sustainable (T_s, f_D) choices (Galaxy S5)

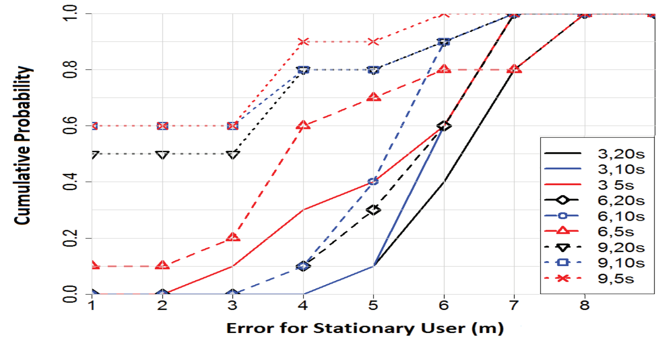
BLE, with its small range and its lower energy overhead, has been widely touted as an enabler of accurate indoor location tracking, in places such as retail stores (e.g., determining the shopper's aisle-level location) and public museums (e.g., tracking a visitor's location at exhibit level location). For us, the key question is: to what extent does the "sustainability constraints" (i.e., assuring that the smartphone battery does not drain significantly) outlined in Section VI affect the accuracy of a BLE beacon-based indoor location system?

In our studies, we use RADAR [3], a widely-studied *fingerprinting* based technique for indoor location. While significantly more sophisticated algorithms have been proposed, we choose RADAR for (i) its simplicity: its performance and limitations are widely benchmarked, and it does not require any additional sensor data (e.g., inertial tracking); (ii) ease of comparison: we have a Wi-Fi based version of RADAR operationally deployed on our campus, which allows us to study its comparative performance. To study these properties, we deploy a varying number of our Estimote BLE beacons on one floor of our school building, and then derive the RSSI fingerprints at a set of landmarks (our landmarks are aligned with ceiling-mounted sprinklers and are 3 meters apart).

A. Spot Location Accuracy

We first study the location accuracy at each of those fingerprinted landmarks, for various devices, as a function of the number of beacons, and for 3 "sustainable" parameter settings: $T_s = \{5$ sec, 10 sec, 20 sec$\}$, $f_D = 0.5$. These studies measure the "spot location" error: the measurements are taken by a stationary user who stands at each of the landmarks multiple times during the test phase.

Figure 10 plots the CDF of the location error for 3 different values of the total number of beacons ($N_B = \{3,6,9\}$), as measured on the Galaxy S5. (Results for other Android devices are comparable and are omitted.) From the figure, we can see that the location accuracy is excellent (80^{th} percentile of error being $\leq$ 4meters) when the number of beacons is high, independent of the parameter settings of the phone. However, the location accuracy degrades appreciably (with 50^{th} percentile of error being $>$ 6meters, i.e., 2 landmarks) when the phone can hear from only 3 beacons. Somewhat surprisingly, the location accuracy is better for smaller scan intervals (T_s), even if the phone is stationary–this is likely due



Scan Interval (s)	No. of Beacons 3	6	9
5	4.31m (4.46)	3m (3.44)	0m (1.62)
10	4.62m (4.75)	4.41m (4.27)	0m (1.02)
20	5.32m (5.18)	4.73m (4.55)	1.5m (1.92)

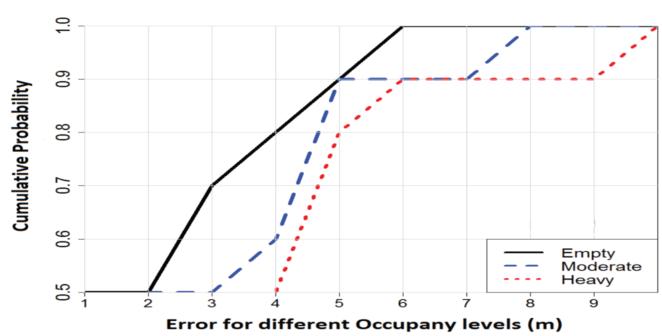
Table shows median value (in meters) and stdev (in brackets).

Fig. 10. Location Error Distribution (Stationary) (Galaxy S5, $f_D = 50\%$)

to the effects of slow-fading, which imply that RSSI readings from the same beacon can be uncorrelated if the measurement interval exceeds a few seconds.

B. Robustness to Occupancy Effects

As RSSI-fingerprint based solutions are well known to be susceptible to environmental changes, especially when the density of human occupants varies significantly, we next tested the accuracy of our BLE-based RADAR implementation under different occupancy scenarios: *empty*, with only 1-2 occupants in the test region (this mimics the conditions under which the fingerprint was created), *moderately crowded*, with 5-6 occupants in the test region and *heavily crowded*, when the test region contained 20+ occupants. Figure 11 plots the error CDFs (again for the S5) for these three occupancy states. We can see that an increase in occupancy impairs both the median location error (which doubles to 4 meters under high occupancy), and especially, the tail of the location error (the 95^{th} percentile error range doubles to almost 10 meters). BLE thus suffers from the same problems as Wi-Fi based fingerprinting systems (where changes in occupancy density are known to cause $\pm 10 - 12$dBm offsets in our campus).

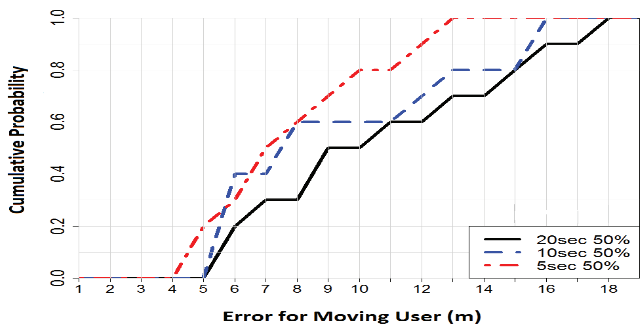


Empty: 0.72m (1.74); Moderate: 1.62m (2.12); Heavy: 3.1m (2.62) (median error (in meters) with stdev (in brackets)).

Fig. 11. Location Error vs. Occupancy (Galaxy S5, N_B =9)

C. Accuracy for Moving Users

We now study how the BLE-based location system when the user continually moves, at regular indoor pedestrian speeds. A macro-duty cycle of 0.5 implies that the phone has a contiguous period of $0.5T_s$, when its BLE interface is off and it is unable to receive beacons. Given the relatively high values of T_s $\{5,10,20\}$ secs configured on the phone, this implies that the freshness of location estimates on the phone might be delayed by as much as 10 secs (0.5*20secs). Given human movement speeds of around 1 meter/sec, this can translate into significant estimation errors.



5s: 6.25m (4.24); 10s: 7.0m (3.97); 20s: 9.0m (2.90) (median error (in meters) with stdev (in brackets)).

Fig. 12. Location Error for Moving Users (Galaxy S5, N_B =9, $f_D = 50\%$)

Figure 12 plots the CDF of the location error for $N_B = \{9\}$, using the Galaxy S5, for the case of a continually moving user. (The results represent the aggregated CDF across a variety of random movement trajectories across the deployment area.) Even with T_s =5 secs, the median location error has now increased to over 6 meters, with almost 100% of the location estimates off by at least one landmark. We also repeated the experiments with $N_B = \{3,6\}$ and confirmed that the location accuracy degrades with a decrease in the number of beacons simultaneously heard by the phone.

D. Implications & Comparisons with Wi-Fi

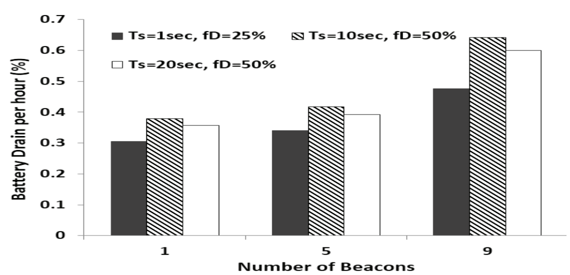
Our results and studies reveal three key insights.

- First, due to the relatively low duty cycle and high-values of scan intervals configured on the smartphone, the location system can exhibit a lag of 5 or more secs, compared to the user's true location. This has implications for many of the ubiquitous proximity-aware applications (e.g., instantaneous product displays in the aisles of retail stores) envisaged for BLE beacons: users may need to linger at specific in-store retail shelves for 10-20 secs before their phone infers their updated location.
- Second, BLE-based location systems are very likely to provide fine-grained location support if the scan interval is low enough, in places such as stores and food courts. From our experience with the Estimote beacons, accuracy levels of 2-3 meters or below are achievable, but only with a dense deployment density (e.g., with 9 beacons within range at every location coordinate).
- Finally, it is instructive to compare the BLE-based approach against more "conventional" Wi-Fi based solutions. Client-side Wi-Fi solutions offer more rapid scanning (once every 2.5s on the Galaxy devices) but its power overhead (≈220 mW [6]) is too high for continuous operation. Conversely, our deployed server-side RSSI-based solution incurs no additional energy drain but experiences an update latency of 2-3 minutes (primarily due to limitations on the Wi-Fi controller).

VIII. Discussion

There are two limitations of our study that we hope to extend in the future:

- *Studying iOS devices:* For this paper, we do not study the comparative performance of iPhones, as (i) the battery cannot be removed (at least officially) and hence, accurate fine-grained power measurements are difficult, and (ii) we need to jailbreak phones to bypass iOS' usual restriction on letting normal applications run continually in the background. Early results from ongoing work suggest differences in the behavior of the Android vs. iOS BLE stacks. For example, Figure 13 plots the estimated battery drain percentage per hour (obtained by measuring the time taken to drain the battery from 100% to 95%) for iPhone 5s for 3 different (T_s, f_D) combinations, as the number of beacons is varied from $\{1,5,9\}$. The figure shows that the battery drain rate of iPhone 5s appears to increase by an average 65%, in contrast to the Samsung devices where the power drain was largely insensitive (Figure 8) to the number of beacons.



Each bar is a different (T_s, f_D) combination (as shown in legend)

Fig. 13. Battery Drain Rate (iPhone 5s) vs. No. of Beacons

- *Investigating the internal Android duty-cycling modes:* As mentioned in Section II, more recent specifications indicate that Android internally allows 3 different modes of *fine-grained* duty cycling within a single scan period. As our results use the default LOW_POWER mode, we believe that alternative modes will likely result in even higher power drain.

–

IX. Related Work

BLE-based Location Tracking: Bluetooth-based indoor positioning has been extensively researched in the past–e.g., [7] used Bayesian fusion over RSSI probability distributions to estimate indoor location at a reported accuracy of around

4.7m. An early investigation of BLE-based indoor location tracking [11] showed that while the problem of fast fading was more acute in BLE (due to its narrower 2MHz channel width), the use of RF fingerprint based approaches resulted in a 95-percentile accuracy bound of 2.5m or less for BLE (compared to almost 8.5m for a Wi-Fi based system). The promise of BLE as a ubiquitous, low-energy beacon technology has driven research efforts in exploring its use for indoor location tracking. The BlueSentinel system [9] uses a modified iBeacon protocol (where each beacon cyclically advertises different virtual "regions", so as to have the iOS mobile device provide more continuous beacon readings) to demonstrate room-level location accuracy for spot measurements (non-moving users), using a k-NN classifier.

Bluetooth-based Context Recognition: Bluetooth-based traces have been extensively used to infer various human activity contexts. The Reality Mining effort [10] was a pioneer in using Bluetooth-based contact and proximity logs obtained during regular lifestyle activities to infer social patterns and relationships. Bluetooth contact traces from multiple phones have been recently used [17] to estimate the density of crowds at various public events. More recently, the ContextSense system [8] uses longitudinal Bluetooth contact traces among individuals and objects to continuously identify new social contexts of an individual.

X. Conclusion

Based on a detailed performance analysis of the interactions between a smartphone's BLE interface and commodity BLE beacons, we obtained the following key insights:

- Continuous scanning (or scan intervals ≤ 1 sec, with duty cycles of 50% or higher) is untenable from the standpoint of continuous operations, as the resulting energy overheads (arising from the wake lock and the transient ("tail energy") energy spikes) impose more than 30% additional overhead, over the average energy drain experienced by Android users under regular daily use.
- Accordingly, for continuous operations, an operating choice of (scan interval=5sec, duty cycle=50%) appears to be the most aggressive "sustainable" operating point, but still results in beacon reception loss rates of over 60%, on both the Samsung S3 and S5 smartphones.
- The indoor location accuracy (using the RADAR fingerprinting-based algorithm) was seen to be pretty high (median error values of only 1–2 meters) for stationary devices, only if the BLE beacon density was high (an average beacon density of 0.14/m^2 in our experiments). However, under (large T_s, low f_D) settings, the location accuracy degraded appreciably (the median location error becoming more than 6 meters in our experiments) for *moving users*.

While the absolute performance metrics will naturally evolve over time, our results show that: **to satisfy energy, constraints practical systems must adopt a more *intermittently-operated* BLE paradigm, where additional cheaper context triggers are used to activate BLE scanning only during *relevant* time periods**.

Acknowledgment

This work was supported partially by Singapore Ministry of Education Academic Research Fund Tier 2 under research grant MOE2011-T2-1001 and partially by the National Research Foundation, Prime Ministers Office, Singapore, under the IDM Futures Funding Initiative and administered by the Interactive and Digital Media Program Office, Media Development Authority. All findings and recommendations are those of the authors and do not necessarily reflect the views of the granting agency, or Singapore Management University.

References

[1] Estimote beacons. http://estimote.com.

[2] Aislelabs. ibeacon and battery drain on phones: A technical report. http://www.aislelabs.com/reports/ibeacon-battery-phones.

[3] Bahl, P. and Padmanabhan, V. Radar: An in-building rf-based user location and tracking system. *Proceedings of IEEE INFOCOM 2000*, 3 2000.

[4] Balan, R. K., Lee, Y., Tan, K. W., and Misra, A. The challenge of continuous mobile context sensing. *Proc. of the 6th International Conference on Communication Systems and Networks (COMSNETS)*, Jan. 2014.

[5] Balan, R. K., Misra, A., and Lee, Y. LiveLabs: Building an in-situ real-time mobile experimentation testbed. *Proc. of HotMobile*, Santa Barbara, CA, 2014.

[6] Brouwers, N., Zuniga, M., and Langendoen, K. Incremental wi-fi scanning for energy-efficient localization. *Proc. of Pervasive Computing and Communications (PerCom)*, March 2014.

[7] Chen, L., Pei, L., Kuusniemi, H., Chen, Y., Kröger, T., and Chen, R. Bayesian fusion for indoor positioning using bluetooth fingerprints. *Wireless Personal Communications*, 70(4):1735–1745, 2013.

[8] Chen, Z., Chen, Y., Hu, L., Wang, S., Jiang, X., Ma, X., Lane, N. D., and Campbell, A. T. Contextsense: Unobtrusive discovery of incremental social context using dynamic bluetooth data. *Proceedings of Ubicomp '14, Adjunct*, 2014.

[9] Conte, G., De Marchi, M., Nacci, A. A., Rana, V., and Sciuto, D. Bluesentinel: A first approach using ibeacon for an energy efficient occupancy detection system. *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, BuildSys '14, 2014.

[10] Eagle, N. and (Sandy) Pentland, A. Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.*

[11] Faragher, R. and Harle, R. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation*, ION GNSS+ 2014, 2014.

[12] LitePoint. Bluetooth low energy. http://www.litepoint.com/wp-content/uploads/2014/02/Bluetooth-Low-Energy_WhitePaper.pdf.

[13] Ravi, N., Scott, J., Han, L., and Iftode, L. Context-aware battery management for mobile phones. *6th Annual IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2008.

[14] Rolleston, J. Indoor wifi location and beacons: Better together, July 2014. http://blogs.cisco.com/wireless/indoor-wifi-location-and-beacons-better-together.

[15] Sen, R., Lee, Y., Jayarajah, K., Misra, A., and Balan, R. K. Grumon: Fast and accurate group monitoring for heterogeneous urban spaces. *Proceedings of SenSys '14*.

[16] Stollmann Entwicklungs und Vertriebs GmbH. Whitepaper bluetooth low energy release r03. http://www.stollmann.de/fileadmin/01_Content/pdf/Whitepaper_BLE_r03.pdf.

[17] Weppner, J. and Lukowicz, P. Bluetooth based collaborative crowd density estimation with mobile phones. *Pervasive computing and communications (PerCom), 2013 IEEE international conference on.*