Mobility-Driven BLE Transmit-Power Adaptation for Participatory Data Muling

Chung-Kyun Han, Archan Misra, Shih-Fen Cheng
School of Information Systems
Singapore Management University
Singapore
Email: ckhan.2015@phdis.smu.edu.sg, {archanm, sfcheng}@smu.edu.sg

***Abstract*—This paper analyzes a human-centric framework, called SmartABLE, for easy retrieval of the sensor values from pervasively deployed smart objects in a campus-like environment. In this framework, smartphones carried by campus occupants act as data mules, opportunistically retrieving data from nearby BLE (Bluetooth Low Energy) equipped smart object sensors and relaying them to a backend repository. We focus specifically on dynamically varying the transmission power of the deployed BLE beacons, so as to extend their operational lifetime without sacrificing the frequency of sensor data retrieval. We propose a memetic algorithm-based power adaptation strategy that can handle deployments of thousands of beacons and tackles two distinct objectives: (1) maximizing BLE beacon lifetime, and (2) reducing the BLE scanning energy of the mules. Using real-world movement traces on the Singapore Management University campus, we show that the benefit of such mule movement-aware power adaptation: it provides reliably frequent retrieval of BLE sensor data, while achieving a significant (5-fold) increase in the sensor lifetime, compared to a traditional fixed-power approach.**

***Keywords*-BLE beacon, Data muling, Transmission power adaptation**

I. INTRODUCTION

With the emergence of the Internet-of-Things (IoT) paradigm, there is an increased interest in digitally interfacing with everyday objects, such as garbage bins, coffee makers and cafeteria seats. Ensuring networked connectivity to such devices, however, remains an open and challenging problem: traditional wireless protocols such as WiFi are too energy-intensive and have limited range, while more recent protocols such as LoRa provide long-range connectivity, but support very low bandwidth. As an alternative and simple connectivity solution, we have recently proposed the human-centric, BLE-based *SmartABLE* framework [1], where smartphones carried by users effectively act as *data mules*, collecting and transferring data from nearby BLE-equipped objects to a backend infrastructure. In this approach, each IoT object periodically broadcasts relevant data (e.g., the remaining space in a garbage bin, or the number of coffee pods dispensed by a coffee maker) using one-hop BLE *Advertisements*, which are picked up by a nearby smartphone via periodic Bluetooth *scans*.

SmartABLE has the following attractive features: (i) the networking interface is very straightforward, involving one-hop short range data transfer between the BLE-equipped object and a smartphone, with a subsequent transfer from the smartphone to the backend via a conventional LTE or WiFi interface, and (ii) the BLE-equipped objects have low energy overhead, as the energy-intensive task of BLE scanning is delegated to the smartphones carried by different individuals. In [1], we have shown the feasibility of building a *SmartABLE*-based smart campus solution, leveraging on the predictive movement patterns of long-term campus residents. The downside, of course, is that the reporting gap–i.e., the time between successive updates from a BLE beacon–is no longer deterministic, but depends on the ad-hoc, collective mobility pattern of these users. In extreme cases, for infrequently visited areas of the campus, our analyses showed that the gap between successive updates can exceed an hour, which would be inadequate for latency-sensitive monitoring solutions (e.g., tracking seat-level occupancy in a cafeteria, which should be refreshed at least once every 5 minutes).

One way to mitigate such high inter-report gaps would be to increase the transmission range of the BLE beacons. Clearly, with a higher range, the Advertisements broadcast by a smart object have a higher likelihood of being picked up by a more distant mule–e.g., a smartphone located on an adjacent floor. Of course, this higher transmission range comes with a higher energy cost, resulting in a more rapid drain of the beacon's battery, thereby reducing its operational lifetime. To ensure the practical viability of our vision, this reduction should be as modest as possible. Most commercial beacons currently promise battery lifetimes of *9 months–1 year* or longer, and the lifetime of such BLE devices is significantly affected by the *transmission power* and *advertising interval* of the beacons [2].

In this paper, we tackle this problem by developing an adaptive mechanism that adjusts the power level (or transmission range) of each BLE beacon smartly, to provide the best balance between *responsiveness* (bounds on the inter-report gap) and *energy efficiency*. Fig. 1 illustrates the intuitive idea that different beacons have different power levels: in particular, a beacon's power level should be high enough, but no higher, to ensure that its advertisements reach at least one *actively-scanning* mule within a designated interval (the inter-report gap). We must, however, tackle three challenges:

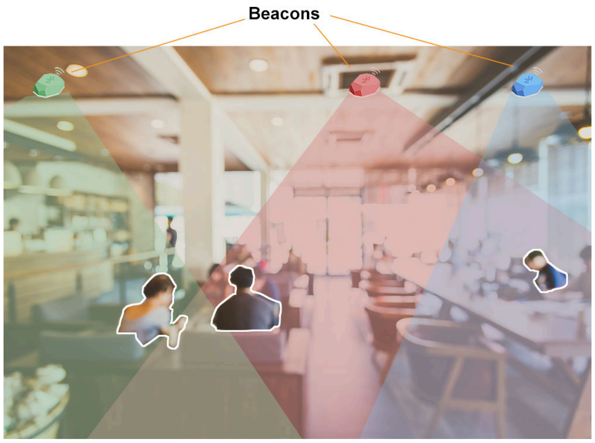


Figure 1. Variable BLE Transmit Power in a Smart Campus: Beacons adjust their transmission power to minimally ensure that they are within range of *at-least-one* mule

- *Uncertainty in Mule Movement:* In realistic scenarios, the trajectory of each of the data mules (i.e., the users' smartphones) is not known deterministically, but can only be probabilistically predicted based on past historical data. This implies that the power level of each beacon should be adjusted *periodically* (for reasons we discuss in Section III) to maximize, in some probabilistic sense, the *likelihood* that the transmission is read by one or more mules within earshot.
- *Minimize the Scanning Overhead:* Given the significantly higher energy expended in Bluetooth scanning (see [3]), the goal of reducing the transmission power of beacons must be balanced with a desire for a schedule that minimizes the number of *actively-scanning* smartphones. In other words, the adaptive strategies must consider the coupling between reduced beacon transmission power levels and the additional set of scanning smartphones needed to compensate for the lower transmission range.
- *Low Computational Complexity:* In practical scenarios, a *SmartABLE* deployment will involve potentially thousands (or even tens of thousands) of beacons, and several hundred mules. To rapidly adapt to changes in movement dynamics, the determination of the right choices for both the beacon power *and* set of scanning smartphones should impose low computational overhead.

Key Contributions: We tackle these challenges by developing a computationally-scalable optimization framework that simultaneously addresses the two objectives of (i) maximizing the BLE beacon operational lifetime, and (ii) minimizing the scanning-related energy overhead of the data mules. We make the following key contributions:

- *Problem Formulation:* We tackle this problem of smart & adaptive BLE power adaptation as an epoch-based dual-objective optimization problem: in each epoch, minimize the energy drain of the most critical BLE beacon (the one with the lowest residual battery), while also minimizing the number of smartphones that scan actively. The optimization framework captures the inherent uncertainty of mule movement via a reliability constraint, which ensures that the each mule's advertisements are captured, with high probability, by at least one actively-scanning mule.
- *Low-Complexity Heuristic:* Our objectives can be formally modeled as a non-linear integer program, with combinatorial complexity. To provide a low-latency solution for practical, large-scale deployments, we propose a heuristic based on a *memetic* algorithm [4]. In this approach, a proposed combination of per-beacon power levels and the subset of scanning smartphones is represented as a *chromosome*, with genetic evolution techniques being used to iteratively derive improved solutions. Trace-driven studies quantify this approach for a pervasive per-floor deployment (16 beacons, 57 smartphones).
- *Significant Real-world Benefit of Power Adaptation:* We empirically show the algorithm's impact, using real-world movement traces of thousands of users on our university campus. We show that our memetic heuristic can take advantage of dense user populations, to ensure adequate coverage of the deployed beacons, with significantly longer operational lifetimes. Very specifically, for our simulated deployment, we can retrieve fresh readings from each beacon within 15 minutes 99% of the time, with a beacon operational lifetime of more than 19 months, which is five times longer than an equivalent constant-power alternative baseline (a lifetime of less than four months).

II. Related work

Related work lies principally in the areas of (a) data muling for sensor networks, (b) Delay Tolerant Networks, and (c) transmission power adaptation in Wireless Sensor Networks (WSNs).

Data Muling: The idea of using mobile 'data mules' as collectors of data from sparsely deployed sensor nodes is first introduced in [5]. Since then, a variety of work (e.g., [6]–[8]) have explored the concept of using mobile collectors for data retrieval in delay-tolerant network settings. In most past studies, the collectors are assumed to be controllable by the network operator; hence, the focus is principally on defining better-coordinated movement schedules to maximize some measure of the information retrieved, while minimizing travel overhead. Moreover, in contrast to our paradigm where the mules upload the data using either WiFi or cellular networks, the collectors in such sensor networks usually transfer the data to a sink node by moving to its proximity. The use of human-carried smartphones as possible data mules is first explored in [9], which showed that both intentional and opportunistic mobility can be used for data muling in various indoor & outdoor scenarios. The use of mobile sinks (similar to mules) to maximize the lifetime of a WSN has been investigated in [10]; in this approach, the movement of multiple mobile sinks are coordinated to reduce the packet forwarding overhead on battery-constrained nodes. More recently, Qu et al. [11] have explored the use of data mules to save the transmission energy overhead on sparsely deployed sensor nodes. The key idea is to intelligently duty cycle their sleep/transmission cycles to match the projected trajectories of the mules. *SmartABLE* has similarity to [12], which used BLE advertisements as a way to transmit residential power sensing data to a smartphone. Our work differs in our use of mules whose actual movement is outside our control, and in our use of multiple potential mules with varying levels of movement uncertainty.

Delay Tolerant Networks: Several researchers have investigated the use of delay tolerant networks (DTNs) in campus environments. In this mode of operation, individual nodes form a multi-hop mobile network, helping transfer data from a source to sink via multiple intermediate data exchanges [13]. The focus here is on developing effective packet forwarding/routing strategies, taking into account mobility-driven characteristics such as the inter-node contact time [14] and the predicted trajectory of individual nodes. Su et al. [15] used empirical movement traces to show that it is indeed possible to form a campus-based DTN based on human mobility, while Zhu et al. [16] specifically investigated the possibility of deriving better behaviorally-inspired mobility models for DTNs. In contrast to such work, we focus on (i) empirically deriving predictions of individual movement on campus, and (ii) using such predictions to adjust the transmission power of beacons.

Transmission Power Control in WSNs: There is a rich corpus of research on the adaptive control of transmission power levels in WSNs. Kubisch et al. [17] proposed two schemes that seek to reduce the transmit power level (and thereby increase the sensor node lifetime) while ensuring that the underlying network remains connected. Researchers have also investigated the use of transmission control to shape the underlying WSN topology, thereby improving performance in terms of metrics such as robustness to failure [18].

III. Design Assumptions & Choices

The *SmartABLE* framework, introduced in [1], assumes the existence of a centralized controller, which adjusts relevant parameters, in both the scanning smartphones (the mules) and the BLE-equipped smart object sensors. We envision the use of this approach in a densely-occupied campus setting, where most beacons usually have *multiple users* within their transmission range. In this approach, each BLE beacon transmits beacons periodically and frequently (typical beacon advertising intervals are $100 - 200$ms).

The adaptive power setting approach presented here makes the following design assumptions:

1) Each mule is aware of its current location, using some operationally deployed location tracking system. In particular, in our exemplar of on-campus deployment, all smartphones retrieve their location that is computed by a WiFi-based server-side location tracking system that has been operational on our campus for the past 3 years. This system computes the smartphone's location (using purely passive WiFi measurements) once every 5 seconds, and achieves a median accuracy of 6-8 meters.
2) The location and communication identifier of each BLE-equipped sensor is available in a centralized repository, and this repository is shared with all participating mules. This repository will be used by each mule to identify the set of BLE sensors that are near its current location.
3) The BLE sensors, embedded on various smart objects, support *only* Bluetooth-based communication. More specifically, in *SmartABLE*, each beacon piggybacks sensor-related information on the periodic BLE Advertisements that it broadcasts, while mules can engage in unicast communication with an individual beacon to occasionally update its power-level settings.

Adjusting the Beacon Transmission Power: The most important design of the *SmartABLE* approach is its extensive use of an extremely simple, opportunistic, one-hop short-range wireless link between BLE-equipped smart objects and nearby user smartphones. Accordingly, we propose to use a separate downlink channel (from mule to beacon) to periodically issue *configuration commands* and adjust the power levels of individual beacons. In this approach, the central controller periodically shares the (sensor ID, power level) tuples, for all beacons, for the *upcoming epoch* with the set of participating mules. Each mule then performs a lookup, using the knowledge of its current location, on the centralized repository to ascertain the set of nearby BLE sensors, and then initiates unicast communication with each such sensor to provide its designated "power level setting for the next epoch".

Periodic Updates of Power Levels: *SmartABLE* utilizes the predicted movement trajectory of individual mules to compute the optimal transmission power levels for each beacon for each *epoch*. In contrast to continuous re-optimization, based on latest mule locations and states, our predictive approach is more desirable in two aspects: (i) *SmartABLE* requires no additional communications on frequent location updates, directly leading to conservation of energy; on the other hand, re-optimization approach would require constant updates and consume significantly more energy, and (ii)

SmartABLE utilizes predictive information on future mule movement patterns in planning–this is in contrast to the re-optimization approach, where only current locations are being considered.

IV. Problem Definition

We now present our formal modeling of the power adaptation problem, which seeks to periodically adjust the beacon's broadcasting power to lower levels (if possible) and also identify the subset of available smartphones (mules) tasked to actively scan during this period. Fig. 2 shows an idealized scenario illustrating this interaction between beacon power levels and mule trajectories. In this example, beacon b_2's power level is less than that of beacon b_4, implying that b_2 has a smaller transmission range (and lower energy drain) than b_4. The figure also illustrates the different predicted *possible* trajectories for three different mules: m_1, m_2, m_3. We represent each mule's trajectory with a line, such that lines' thickness is proportional to the likelihood of the mule following that specific trajectory. Thus, in the figure, mule m_1 has 3 trajectories, with its most likely path traversing the transmission range of b_1, but not b_2, b_3 or b_4.

The key is to note that single mule might potentially *cover* multiple beacons. Thus, one possible trajectory for mule m_1 will cover both b_1 and b_2, whereas another alternative path will cover b_1 and b_4. Such coverage shows that intelligent selection of the mules might avoid redundant scanning. For example, if m_1's path already covers b_4, then mule m_2 offers no additional scanning benefit.

We formally define the goal of any such power adaptation technique via dual objectives:

- The first objective is to maximize the operational "system lifetime" of the deployed beacons. Similar to past work on network lifetime maximization, we assume that this lifetime is driven by the *most critical* beacon–i.e., the one that has the least battery power. We thus define lifetime by the earliest time-to-depletion of any beacon.
- The second objective is to minimize the number of mules who collect sensing data. As in past work, we assume that each actively scanning mule operates with a predefined scanning frequency and duty cycle, and that minimizing the number of such mules indirectly minimizes the total scanning energy. Note that there may be alternative legitimate variants, such as maximizing fairness (i.e., minimizing the variance in the energy drain across all mules) or bounding the maximum permitted battery depletion rate. These represent minor changes in our basic optimization model, and are thus not discussed explicitly in this paper.

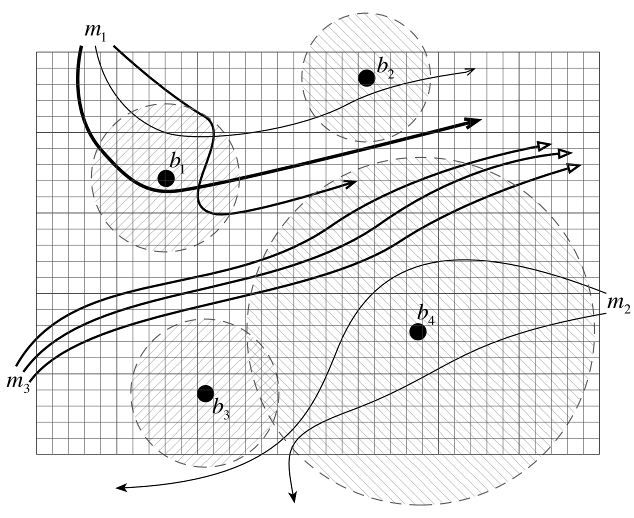


Figure 2. An illustrative example

We can formalize these objectives as follows:

$$\max \min_{b \in B} \big(c_b - \sum_{l \in L} e_l \cdot x_{b,l}\big). \tag{1}$$

$$\min \sum_{m \in M} y^m. \tag{2}$$

where B is the set of beacons, L is the set of power level and M is the set of mules. Moreover, c_b is the remaining battery power of beacon $b \in B$ and e_l is the amount of energy consumed when a beacon's power level is set to a value $l \in L$. We consider two types of binary variables $x_{b,l}$ and y^m. $x_{b,l}$ is set to 1 if the power level of beacon b is l. y^m indicates whether mule m is designated as a scanning mule or not.

There is a natural tradeoff between two objectives. Continuing with the example of Fig. 2, we see that no mule can collect beacon b_3's sensing data, given the current power level configuration. However, if beacon b_3 increases its transmission range, thereby expending more energy, mules m_2 and m_3 have a higher likelihood of collecting its data. Among the mules, m_3 could, in fact, become the single scanning device.

The optimization problem has two distinct constraints:

- The first one captures the fact that each beacon can have one, and only one, setting for the power level in one epoch:

$$\sum_{l \in L} x_{b,l} = 1, \qquad \forall b \in B. \tag{3}$$

- The second one describes the need to ensure that each beacon's data is retrieved, with sufficiently high probability and sufficiently high frequency. As noted earlier, data freshness is one of the essential factors that the *SmartABLE* framework must support. To simplify this issue, we split time into a series of epochs (with the duration of each epoch being equivalent to the tolerable gap between successive beacon updates) and then require each beacon to be read *at least once* in each epoch. (Very specifically, if the maximum permissible interval between successive readings is T_R, setting the epoch period to $\frac{T_R}{2}$ ensures adherence to the freshness constraint.) Given the inherent movement uncertainty, we specify that the probability of data collection must be greater than or equal to a system reliability level, R. Formally:

$$1 - \prod_{m \in M} \prod_{l \in L} (1 - p_{b,l}^{k,m} \cdot y^m \cdot x_{b,l}) \geq R, \quad \forall k \in K,\ \forall b \in B. \tag{4}$$

where $p_{b,l}^{k,m}$ is the probability that mule $m \in M$ transits through beacon b's transmission range when the power level is l at epoch $k \in K$. This probability is calculated as the weighted sum, over each of the different likely paths of the mule, with an indicator function that is 1 *iff* a specific path traverses through the beacon's transmission range.

Our optimization problem has combinatorial complexity, and is a nonlinear integer program. Accordingly, we shall now develop an efficient heuristics-based approach.

V. Solution Approach

To tackle this non-linear, combinatorial optimization problem, we combine a genetic algorithm (GA) with local search. This combined approach is named as a *memetic algorithm*, and its combined use of population-based global search and local hill-climbing is attractive for many combinatorial problems [4]. Algorithm 1 shows the pseudo code for our memetic algorithm. Intuitively speaking, the algorithm starts off with a set of (possibly, random) solutions, called *chromosomes*. In our specific instance, a chromosome is represented by an array of size $|B|$ (no. of distinct beacons) + $|M|$ (no. of distinct mules), with each of the B elements assuming a value from the discrete set of transmission power levels, and the M elements assuming a binary value (0 indicating that the mule is not selected for active scanning). We now explain in detail the optimization procedure, which consists of the repeated application of three processes—*local search*, *crossover* and *mutation* The algorithm is characterized by the following parameters:

- N_p: Size of population–i.e., no. of alternative chromosomes that survive in each generation
- N_o: Number of offsprings to be generated–i.e., cardinality of chromosomes subject to crossover and mutation in each generation
- N_G: Number of generations–i.e., total number of iterations
- N_s: Number of sampled neighbors (used to define the complexity of local search)
- δ: Points (specific indices in the chromosome array) where the crossover operation occurs

Algorithm 1: Memetic algorithm

$P \leftarrow Init(N_p)$
$t = 0$
while $t \leq N_G$ **do**
  $P \leftarrow LocalSearch(P, N_s)$
  $O \leftarrow Sample(P, N_o)$
  $O_1, O_2 \leftarrow Split(O)$
  for *each* $i \in \{1, ... \frac{|O|}{2}\}$ **do**
    $o_1 = O_1[i]$
    $o_2 = O_2[i]$
    $o_1, o_2 \leftarrow CrossOver(o_1, o_2, \delta, p_c)$
  for *each* $o \in O$ **do**
    $o \leftarrow Mutation(o, p_m)$
    $Evaluation(o)$
  $P \leftarrow Selection(P, O)$
  $t = t + 1$

- p_c: Crossover probability
- p_m: Mutation probability

The algorithm starts off with N_p random chromosomes. In any chromosome, the first B elements represent the currently selected broadcasting power of the beacons–if $x_{b,l} = 1$, $A[b] = l$. The subsequent M elements represent the selection of each of the M distinct mules–if $A[|B| + i] = 1$, this implies that the i^{th} mule has been selected for scanning. Each chromosome, representing a possible solution to the optimization problem, has a two-tuple, dual-objective *fitness* value, computed using Equations 1 and 2. (If a solution is infeasible–i.e., it violates Equation 4, the fitness value is set to the tuple $(-\infty, \infty)$.

In each iteration, we start with the current batch of N_p chromosomes and try to improve each solution's *fitness* with a local search. More specifically, a chromosome c has $((|L| - 1) \times |B|) + |M|$ neighbors–i.e., other arrays that differ from δ in just 1 element. The local search process randomly selects N_s of these neighbors, and checks if these alternatives *dominate* the current (incumbent) solution–i.e., if the alternative has a better value (higher residual power in the critical beacon & lower number of scanning mules) on *both* objectives. If so, the chromosome is replaced by this dominant neighbor. After this search, we move to the GA phase. In this phase, the algorithm first randomly selects N_o of the chromosomes as "offspring". These offspring vectors are then divided into two groups. One child from each group is then selected for "crossover" (swapping of vector's elements segmented by δ, the designated crossover points) with a corresponding child in the other group. For example, if $\delta = (2, 4)$, then the segment from the second element to the fourth element is swapped between the selected pair of child chromosomes.

Finally, each chromosome undergoes a *mutation*, whereby one element in each chromosome is "randomly" modified (with probability p_m). From the resulting $N_p + N_c$ chromosomes, the Selection process picks N_p chromosomes, as "survivors" for the next iteration. The Selection process first picks all the non-dominated vectors first, with the remaing members being selected randomly. This entire process is repeated N_G times. Eventually, once the iterations have finished, the "optimal solution" is chosen to be one of the non-dominated chromosomes (if it exists); else, it is picked at random.

VI. Experiments

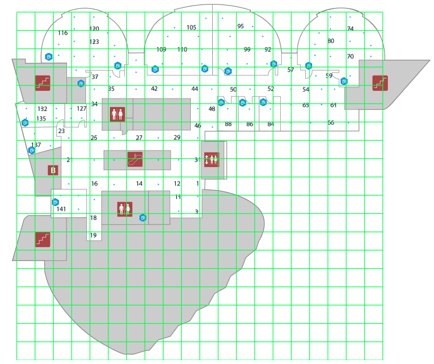
We now experimentally study the proposed heuristic, using real-world data collected from our university campus, and compare its performance against the current, fixed-power alternatives.

A. *Dataset & Data Filtering*

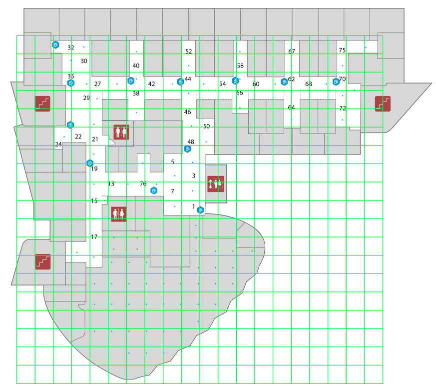
Our analysis utilizes longitudinal traces of movement data obtained using the LiveLabs location service [19], which utilizes a server-side WiFi-based localization technique to capture the location history of all WiFi-enabled devices on campus. The indoor location service operates across five separate academic buildings and a connecting public concourse. To accommodate the $6-8$ meter median localization error, we capture the movement traces at *section-level* granularity, rather than at individual landmarks (which are spaced roughly 3 meters apart). The campus is divided into *247* logical sections, with section sizes varying from $18-10$ m^2.

For our studies, we utilize location data for the period of two months. We utilize 4 weeks of location history (from February 2017) to build a movement predictor (i.e., compute $p_{b,l}^{k,m}$) which predicts trajectory as a series of *stay-points*, and then evaluate our approach by evaluating such prediction-based power adaptation on the actual movement traces for the first 5 working days of March 2017. Selected buildings also have a deployed infrastructure consisting of Estimote™ beacons, deployed roughly 6-10 meters apart. While our overall framework applies to the entire campus-wide deployment, we realize that the optimization can be partitioned as the typical range of each beacon is limited to an individual floor. Accordingly, we specifically focus on two floors of an academic building as exemplars. Fig. 3 show the layout of both levels, Level 2 and Level 4. Each black dot represents a location landmark (selected landmark IDs are shown as well).

The second floor consists primarily of lecture halls and group study rooms, and is principally used by undergraduate students, whereas the fourth floor consists of a mix of graduate research and faculty offices, and is thus dominantly occupied by graduate researchers and faculty. The blue hexagons in Fig. 3 represents the position of the deployed beacons. Most of the beacons on the fourth floor are deployed along the corridors, while those on the second floor



(a) Level 2



(b) Level 4

Figure 3. Layout and deployment of beacons

are primarily deployed inside the publicly accessible lecture & meeting rooms. To simplify our computations, we snap all locations to a grid-based coordinate system, with each grid representing a rectangular 5×5 m^2 area (see Fig. 3).

Beacon Power vs. Range Model: Our investigations are based on the Estimote Proximity beacon that has been deployed on our campus. It provides six power level settings ranging -30 dBm $\sim$ +4 dBm. Considering the granularity of the grid and the overall system's lifetime, we only consider three of the most popular power settings {-12 dBm, -4 dBm, 0 dBm}, which provide transmission ranges varying from $15-50$ meter. (We do not consider higher transmission power levels, as the battery depletion rate becomes unacceptably high.) Table I summarizes the different settings for the broadcast transmission power and the corresponding range. The first two columns show absolute values, while the remaining two columns translate those readings to our grid-based coordinate system. (For simplicity, we normalize power consumption to that for the -12dBm setting–i.e., the -12dBm setting is assumed to consumed 1 unit of power.)

Mule Selection & Trajectory Prediction: Because each mule's trajectory, especially for university students, will vary with the day of week and hour, we compute each mule's trajectory separate for each day of week (DW) and

Table I
BROADCASTING POWER AND COVERING RANGE

Real world		Our settings	
Power (dBm)	Range (meter)	Power	Range
-12	15	1.00	3
-4	35	6.31	7
0	50	15.85	10

hour of day (HD), combination. Moreover, we restrict our analysis to only regular working hours (9 am - 6 pm). Fig. 4 shows the variation in the number of candidate mules for each such (DW, HD) value, after the filtering process. As mentioned earlier, many undergraduate students use the second floor (Lv2), and their presence on that floor is thus highly influenced by their class schedules, thereby explaining the higher variability across days on the second floor.

After the filtering process, we build two different mobility models for comparison:

- The zeroth-order model: uses the staypoint history to derive the probability that a mule would appear in each location (percentage of time the mule appear in each location).
- The first-order Markovian model: similar to the zeroth-order model, but derive the probability that a mule would appear in each location based on his current location (thus the Markovian property).

Fig. 5 summarizes the accuracy of these two predictors. The figure shows the CDF of the models' error, with the x-axis representing the 'error'. The error is calculated, for any section S, in any epoch, by calculating the difference between the predicted probability $p_{b,S}^{k,m}$ and a binary variable Z ($Z = 1$ iff the mule actually visited section S). In other words, the error is high if a user visits a section that was predicted to be highly unlikely, or fails to visit a highly likely location. The figure shows that both models do reasonably well (80% of error values are 0), with the first-

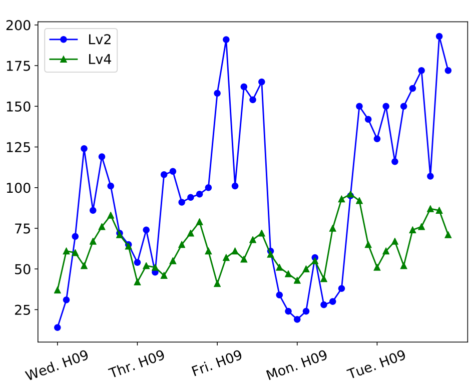


Figure 4. Number of mules for each day and hour

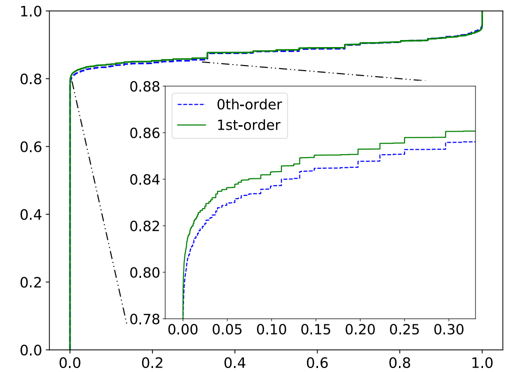


Figure 5. CDF of Markov models' error

order model being slightly better. Of course, the mobility is not completely predictable (note the long tail of the CDF).

B. Algorithm Implementation

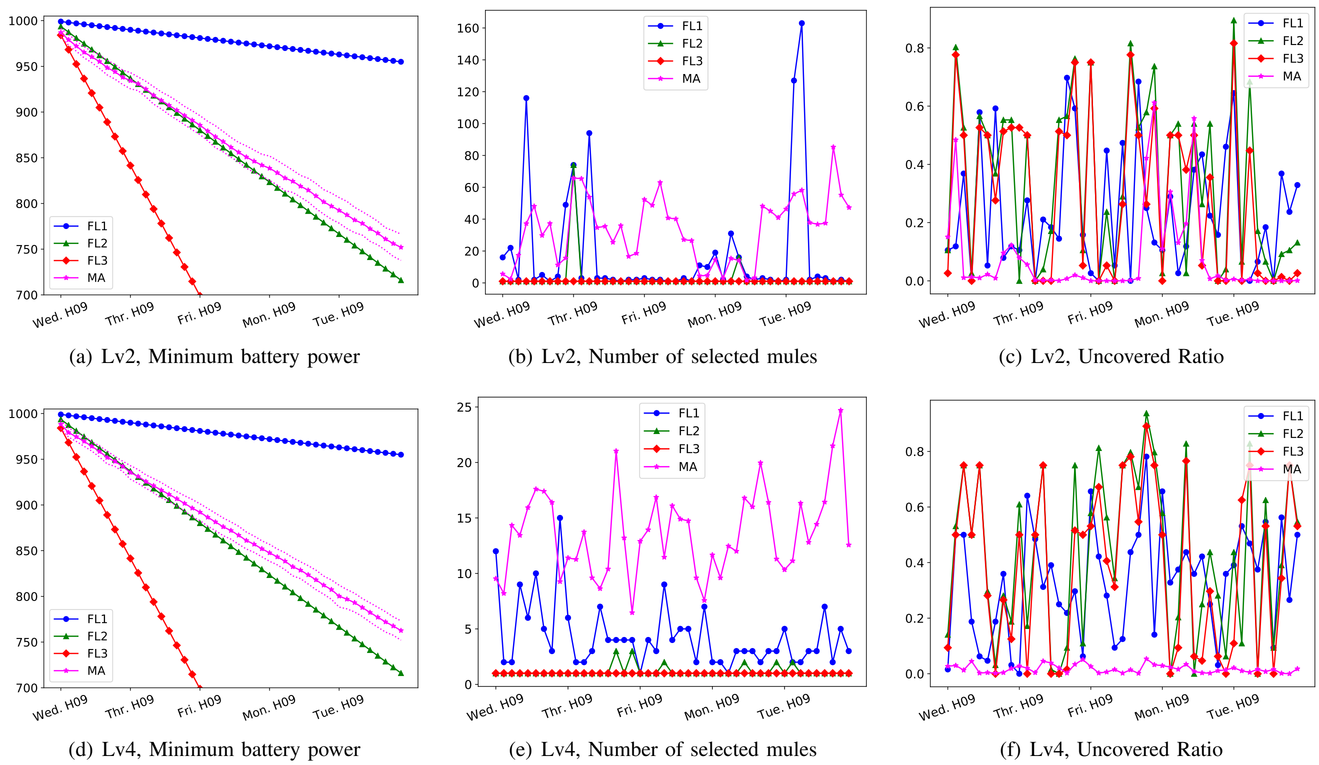
We implement our memetic algorithm using Python 3.6 and execute our code on a dedicated Intel Xeon E5-2620 (2.00GHz) machine running Linux. We set reliability level (R) as 90%, and the number of epochs per hour (K) to be four–i.e., we compute movement trajectories and then derive beacon power levels in 15-minute chunks. By default, we use the following parameter settings:

- Number of generation (N_G): 50
- Size of population (N_p): 100
- Number of samples for the local search (N_s): 10
- Number of offsprings (N_p): 80
- Crossover points (c): ($|B|/2$, $|B|$, $|B| + |M|/2$))
- Crossover probability (p_c): 0.5
- Mutation probability (p_m): 0.5

We validate the performance of our memetic algorithm with empirical datasets through data-driven simulations. As mentioned before, we first calculate $p_{b,l}^{k,m}$ for each (DW, HW) combination, then determine each beacon's power level ($x_{b,l}$) and the set of selected scanning mules (y^m) for the next epoch. The results show the summarized statistics over 30 randomized runs, with the initial battery energy level (for all beacons) being 1000 units.

C. Performance Comparison vs. Fixed-Power Alternatives

We now compare the performance of our proposed power adaptation approach against a baseline approach, where the transmission power is kept constant at all times. (For our numerical comparisons, we assume that all beacons use a common, fixed power level.) To determine the set of mules in the baseline approach, we order the set of eligible mules first and iteratively increase the size of mules until we achieve the desired reliability level. We experimented with three different Fixed Power Level (FL) choices–**FL1** represents cases where the beacons use the lowest broadcasting power



(a) Lv2, Minimum battery power (b) Lv2, Number of selected mules (c) Lv2, Uncovered Ratio

(d) Lv4, Minimum battery power (e) Lv4, Number of selected mules (f) Lv4, Uncovered Ratio

Figure 6. Objectives and simulation results

(value=1.00 in Table I), whereas **FL3** represents the setting with highest transmission power (value=15.85 in Table I).

To compare the algorithm's ability to reliably retrieve the BLE updates, we compute a metric called the *Uncovered Ratio* (UR), which counts the fraction of beacons from which we are unable to retrieve *at least* one reading within an epoch, as follows:

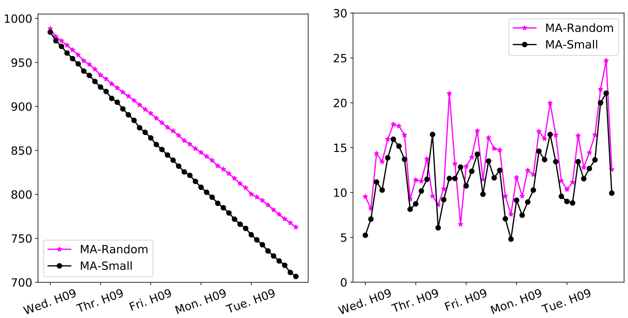
1) WBK represents the number of all beacon-epoch pairs–i.e., one beacon, observed over one epoch, $= 1$ WBK.
2) UBK represents the number of beacon-epoch pairs where no mule is able to retrieve the corresponding sensor data.
3) UR is then given by: UR $=$ UBK$/$WBK

Fig. 6 summarizes our results. The x-axis in all graphs represents the office hours during the first five weekdays of March. The top and bottom three graphs correspond to the second and fourth floor cases, respectively. We represent our memetic algorithm's results by **MA**. The normal line shows the average value for a measure and the two dotted lines around **MA** in Fig. 6(a) and 6(d) show $\pm\sigma$ (one standard deviation) from the mean. Note that we have ignored the negligible additional energy consumption for infrequently receiving commands for changing power levels, as a beacon's lifetime is driven mostly by its frequent advertisement transmissions.

Fig. 6 illustrates the main benefits of our approach. First, we see that our **MA** approach drains the critical beacon's battery level more slowly (see Fig. 6(a) and 6(d)) than the high/moderate-power fixed settings, FL3 and FL2. In particular, the **MA** algorithm drains the battery capacity at a rate that is merely 30% of the rate that is observed under FL3. Although a low fixed power setting (FL1) will result in a longer lifetime, it will suffer from a much higher UR value. In particular, for the fourth floor, the least battery power observed from **MA** is 6% higher than FL2 and 62% higher than FL3.

When it comes to the number of scanning mules, the second floor requires a larger number of scanning mules (compared to the fourth floor). This is an artifact of beacon deployment: on the fourth floor, the beacons are deployed along the public, relatively narrow, corridors, implying that a small number of mules are capable of covering the beacons.

Also, our algorithm tends to select more mules than baselines, as it picks a solution among non-dominated alternatives *randomly*. In contrast, the baseline techniques choose mules *greedily*, resulting in a smaller number of mules. Accordingly, we also test an alternative approach that favors solutions (among non-dominated solutions) utilizing fewer mules. We label the alternative approach **MA-Small**. Fig. 7 compares the energy drain and no. of mules needed for this approach against **MA-Random**, our original approach for



(a) Minimum battery power (b) Number of selected mules

Figure 7. **MA-Random** vs. **MA-Small** strategies

random selection of non-dominated alternatives. For **MA-Random**, numbers are identical to that of Fig. 6(d) & 6(e). We see that **MA-Small** has a slightly shorter lifetime, but as expected, requires a smaller number of mules (compared to **MA-Random**).

Fig. 6(c) and 6(f) show an interesting tradeoff between the number of selected mules and the UR. Because selected mules sometimes do not follow its predicted trajectory, the greedy mule selection approach leads to poorer performance (higher UR), as compared to MA's supplemental mule selection approach. Comparing Fig. 6(c) and 6(f), we can see that the mule trajectories on floor 4 are more predictable (typically employees working in their office rooms), with UR almost always being below 3%.

D. Sensitivity Studies

We also carried out experiments to understand the performance sensitivity to different *epoch* values. Table II summarizes results for 3 different values=$\{60,30,15\}$mins (the smaller the epoch, the more frequent the need to obtain fresh data from each sensor). The first column indicates reading intervals, the second column shows the *most critical* residual battery energy, while the last two columns show the average value of 'no. of mules' and UR, over the whole simulation period. In general, somewhat counter-intuitively, we see that larger epoch values lead to (marginally) higher UR. This is because a mule's path is more unpredictable over longer periods of time: selected mules often do not follow the expected trajectory and end up 'missing' certain beacons. Interestingly, the residual battery energy values do not show a monotonic trend, but seem to be largely insensitive to the epoch length.

Overall, our results demonstrate the attractiveness of the centralized MA approach: by employing a mobility-driven, adaptive power adjustment framework, it significantly improves the operational lifetime over higher-powered baselines, while guaranteeing much more reliable coverage compared to lower-powered baselines.

VII. Discussion

Several aspects of our power-adaptive *SmartABLE* framework are open for future exploration.

Variable Update Latencies: Our current epoch-based memetic approach does not consider that the *freshness* requirement might vary considerably across different sensors. For example, a 30-minute reporting gap might be okay for monitoring the state of a dustbin, whereas seat availability in a food court may need to be refreshed every 2 minutes. One approach for capturing such freshness diversity may be to assign different reliability bounds for different sensors, implicitly translating into a higher level of coverage (and more frequent retrievals) by multiple mules.

Alternative Wireless Protocols: In recent years, protocols, such as LoRa, offer the possibility of low-power, wide-area connectivity. Such protocols may offer a competitive alternative to our approach. However, initial analyses suggest that a LoRa deployment of $\sim$1000 nodes (typical of a single campus building) would be limited to 20 messages/day/device, implying an inter-report interval of over 1 hour, making it nonviable for low-latency sensing (e.g., available seats in the cafeteria). However, a hybrid LoRa-*SmartABLE* approach may be worthy of investigation.

Variable Priorities, Battery Capacity & Mobility: In practice, a deployment may have a heterogeneous set of beacon–some of which have higher battery capacity, whereas others have smaller form factors (e.g., BLE stickers) and lower battery capacity. Moreover, we currently assume that the BLE beacons are affixed on static objects–in practice, some objects, such as chairs or coffee-makers may be mobile. In such cases, our centralized approach may need to be augmented by additional decentralized decision-making, where a BLE beacon independently decides to modify its suggested transmission power settings.

Privacy Concerns & Issues: Our current model assumes that the centralized engine is aware of the location trajectories of each potential mule. Individual users, concerned about privacy, may selectively de-activate their BLE scanning at specific 'sensitive' locations, resulting in unanticipated drops in the coverage reliability. In future work, we shall investigate whether learning-based approaches may be used to augment the current memetic algorithmic logic to incorporate such 'beacon non-conformance'.

Table II
Impact of different reading intervals

Interval	Lowest Power	Mule	UR
60 min.	760.62	15.49	0.025
30 min.	759.17	14.42	0.023
15 min.	762.63	13.61	0.017

VIII. Conclusion

In this paper, we have demonstrated the benefits of dynamic power adaptation by BLE beacons, as part of a human-centric sensing framework that utilizes participant smartphones as opportunistic data mules in a densely-occupied, "smart campus" setting. We showed how to combine Markovian mobility prediction with a reliability constraint to ensure the periodic retrieval of updates from all the deployed BLE beacons, even though the movement of humans is not pre-determined. Using a memetic programming approach, we showed how we can efficiently develop solutions that both reduce the battery drain of the BLE beacons and minimize the scanning energy consumed by the participating data mules. Real-world studies on the Singapore Management University campus demonstrate the overall impact: we can retrieve readings every 15 minutes from the deployed beacons 99% of the time, while requiring less than a third of the available mules in any given epoch, and while achieving a 5-fold increase in the battery life of the beacons. Our results provide further support for the proposed *SmartABLE* framework, even though additional architectural innovations may be needed to achieve very high-frequency updates (e.g., if we require updates from each BLE-equipped object once every 30 secs). We anticipate that this approach would be applicable to other smart buildings with a stable pool of occupants with predictable movement–e.g., office buildings and college campuses.

Acknowledgment

This material is supported partially by the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centers in Singapore Funding Initiative.

References

[1] M. Radhakrishnan, S. Sen, A. Misra, R. Balan, and Y. Lee, "Smart monitoring via participatory ble relaying," in *Proc. of 10^{th} International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2018.

[2] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer, "A study of bluetooth low energy performance for human proximity detection in the workplace," in *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017, Kona, Big Island, HI, USA, March 13-17, 2017*, 2017, pp. 90–99.

[3] M. Radhakrishnan, A. Misra, R. Balan, and Y. Lee, "Smartphones and ble services: Empirical insights," in *Proc. of IEEE Mass*. IEEE, 2015.

[4] P. Moscato, "On evolutionary, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," California Institute of Technology, Tech. Rep., 1989.

[5] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2, pp. 215–233, 2003.

[6] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," in *Proc. of IPSN'03*, 2003.

[7] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mobile Networks and Applications*, 2006.

[8] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard computing: Sensor networks in agricultural production," *IEEE Pervasive computing*, 2004.

[9] U. Park and J. Heidemann, "Data muling with mobile phones for sensornets," in *Proc. of SenSys'11*, 2011, pp. 162–175.

[10] Y. Yun and Y. Xia, "Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications," *IEEE Transactions on Mobile Computing*, vol. 9, no. 9, Sep. 2010.

[11] Y. Qu, K. Xu, J. Liu, and W. Chen, "Toward a practical energy conservation mechanism with assistance of resourceful mules," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 145–158, 2015.

[12] M. Choi, J. Han, and I. Lee, "An efficient energy monitoring method based on bluetooth low energy," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2016.

[13] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay-tolerant networks," in *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, ser. WDTN '05, 2005.

[14] P. U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. de Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on dtn routing," in *IEEE INFOCOM 2009*, 2009.

[15] J. Su, A. Goel, and E. de Lara, "An empirical evaluation of the student-net delay tolerant networks," in *3rd International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS)*, San Jose, CA, July 2006.

[16] X. Zhu, Y. Bai, W. Yang, Y. Peng, and C. Bi, "SAME: A students' daily activity mobility model for campus delay-tolerant networks," in *APCC*. IEEE, 2012, pp. 528–533.

[17] M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong, and J. Rabaey, "Distributed algorithms for transmission power control in wireless sensor networks," in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, March 2003.

[18] G. Hackmann, O. Chipara, and C. Lu, "Robust topology control for indoor wireless sensor networks," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '08. ACM, 2008.

[19] K. Jayarajah, R. K. Balan, M. Radhakrishnan, A. Misra, and Y. Lee, " LiveLabs: Building in-situ mobile sensing and behavioural experimentation testbeds," in *the 14th Annual International Conference*. New York, New York, USA: ACM Press, 2016, pp. 1–15.