Low-Latency Indoor Localization Using Bluetooth Beacons

Sudarshan S. Chawathe

Abstract— **Indoor localization refers to the task of determining the location of a traveler in spaces (such as large building complexes or airport terminals) using coordinates appropriate to those spaces (such as floor and room number or airport terminal and gate). Indoor localization using Bluetooth beacons is attractive because of the low cost and high spatial selectivity of Bluetooth devices. However, a significant drawback of the Bluetooth protocol for this application is the large delay incurred in the discovery phase of the protocol, which is the phase used for detecting beacons. These delays, of approximately 20 seconds, hamper the use of this localization method because typical walking speeds are likely to change the set of potentially visible beacons part-way through the discovery phase. We study the causes of these delays and propose methods for alleviating them for indoor localization applications. We formalize the key problem of finding a minimum-cost complete beacon-probing plan and present an algorithm for generating such plans.**

I. INTRODUCTION

INDOOR LOCALIZATION refers to the task of determining the location of a pedestrian in complex indoor environments, such as office buildings and airport terminals, by automated means. An important requirement is that the inferred location must be expressed using features of the relevant indoor map. It is usually not useful to determine the pedestrian's coordinates in some global or regional 3-dimensional coordinate system, such as latitude, longitude, and elevation, except as a potential intermediate step. For example, a traveler's location in a large airport may be expressed as "terminal C, gate 23"; expressing the location as "44.438938 -67.797402" in the WGS84 geodetic reference frame [1] is not useful, even if it were to be determined using GPS or related methods.

Satellite-based localization methods such as GPS, which are effective in most outdoor environments, are notoriously difficult to use in indoor environments due to both attenuation and reflection of signals by walls and other obstructions. This observation has led to much work in alternate methods for indoor localization, using ultrasound, infrared and radio transmissions, visible light, and other signals. Among methods that use radio, 802.11 (wifi) devices have been very popular due to their low cost and wide deployment. A common approach for 802.11 (and, to a lesser extent, Bluetooth) is to map received signal strength indicators (RSSIs) to distance from the device, to then use the distances to determine a mobile device's Euclidean coordinates using trilateration, and to finally map those coordinates to the appropriate map features. It has been widely reported, however, that mapping RSSI values to distance is both difficult and unreliable due to the intricacies of signal propagation in a typical indoor environment that includes diverse construction materials, furniture, equipment, duct-work, and other complex features.

Our approach to indoor localization is based on the use of Bluetooth signals. An important feature of our work, and one that distinguishes it from other efforts based on Bluetooth, is that it does not rely on RSSI or other indicators of signal strength. Instead, it uses only the visibility (a simple true-or-false test of signal reception) of Bluetooth beacons. Intuitively, since Bluetooth devices have a small range by design, the ability to communicate with a device may be used to infer close proximity. By using a large number of inexpensive devices (a few USD each) as beacons, we may determine a travelers location using a cell based method, as suggested by Figure 1.

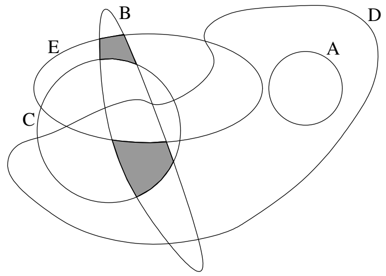


Fig. 1. Cell-based localization: A mobile receiver that is in the ranges of beacons B, C, and D but not in the range of A and E must be located in the lower shaded region.

Prior work has addressed some aspects of Bluetooth-based localization, such as the placement of beacons and the design and deployment of inexpensive beacons in practice [2]. That work has also identified a major hurdle to the effective use of Bluetooth beacons: Reliably determining the set of Bluetooth devices that are within communication range (i.e., the visible beacons) requires approximately 20 seconds in order to run the Bluetooth inquiry protocol, by specification [3]. The limited range (few meters) of beacons, which is advantageous for cell-based methods, also requires beacons to be placed at a comparable spatial resolution. Thus, at typical walking speeds, the set of visible beacons is likely to change every few seconds. A 20-second latency for each location fix therefore severely hampers the usability of Bluetooth-based localization methods.

Author's address: Department of Computer Science; University of Maine; Orono, ME 04469, USA. `chaw@cs.umaine.edu`. This work was supported in part by the U.S. National Science Foundation with grant CNS-0426683.

In this paper, we focus on the problem of reducing this localization latency, while remaining within the confines of the Bluetooth specification. The latter qualification is important because it enables the use of commodity devices, priced at a few dollars each, as beacons. For this purpose, we use another feature of the Bluetooth protocol that allows a device with a known identifier (hardware address) to be directly probed quickly (in at most 2.5 seconds, and often quicker, using the Bluetooth paging protocol). The mobile unit carried by the traveler maintains a map that describes the locations and identifiers of Bluetooth beacons in its environment. (This map may be conveniently transferred and updated using Bluetooth communication with some of the beacons serving double duty, but we do not focus on that aspect in this paper.) If the approximate location of the traveler is known (most likely based on earlier position fixes) then it should suffice to probe for a few Bluetooth beacons in the vicinity of that location. The details of how these beacons are determined and probed constitute the main topic of this paper. While the idea is intuitively simple, a naive implementation is unlikely to be effective because, for instance, probing for 12 beacons may take as long as $12 \times 2.5 = 30$ seconds, which is considerably worse than the inquiry protocol-based method on which we hope to improve. It is therefore crucial that the number of beacon probes required for localization be minimized.

Section II describes the model for indoor localization, Bluetooth beacons, related maps, and traversals in these maps. Section III describes a method for localization using beacon visibility and the modifications needed to use individual beacon probes instead of full discovery of all beacons visible from a location. It defines a key construct, the beacon-probing plan, and develops a formal version of the central problem: determining which beacons are to be probed, and in what order, so as to minimize the number of probes while retaining localization accuracy. Solutions to this problem, called the minimum-cost complete beacon-probing plan (MCBP), are presented in Section IV. We briefly discuss related work in Section V and conclude in Section VI.

II. MODEL OF BEACONS AND TRAVERSALS

A. Locations and Location Graphs

The central task of indoor localization is determining a traveler's location, which is expressed by reference to key features on the appropriate map. Examples of such key features on the map for a typical office building include hallway intersections, elevator access points, stairway landings, and entrances to rooms, identified by room numbers or other characteristics. We refer to the locations of these key features as *interesting locations*, or simply locations.

Indoor maps express the *geometric relationships* between locations in two- or three-dimensional Euclidean space, typically using a scaled representation. For example, a map of an office building may use scale or annotations to indicate the approximate distances between room entrances, elevator access points, and other locations. Localization applications such as navigation use these distances for route planning, arrival-time estimation, and other tasks.

In addition to these geometric relationships, indoor maps also express the *topological relationships* between locations, indicating the manner in which locations are connected to each other. For example, a map of an office building may indicate that travel from room 101 to room 201 requires traversing several hallways to a stairway even though the Euclidean distance between the two rooms is only a few meters.

We model the geometric and topological aspects of an indoor map using a *location graph* in which vertices represent interesting locations and edges represent the connectivity relationships among these locations. More precisely, we say a location a is a *neighbor* of location b if, in the physical environment being modeled, there is a shortest route from a to b that does not pass through any other interesting location. (Multiple shortest routes, with identical lengths, may exist.) The location graph contains edges (a, b) for precisely all such neighboring locations a and b. Each edge (a, b) is associated with a *label* that is a positive real number representing the distance between a and b.

B. Beacons and Beacon Maps

We refer to the set of locations from which a beacon is visible (detectable) as the *range* of that beacon. In order to enable an accurate delineation of the range of each beacon when deployed, we do not that assume beacon ranges conform to any predefined shapes (such as ellipsoid) or sizes (such as 10 meters). Since we model the range of a beacon as the set of locations from which it is detectable, without imposing any additional constraints, we can indirectly, but accurately, model the often complex artifacts of Bluetooth signal propagation due to walls, furniture, ductwork, etc. For example, a beacon located near the door of room 137 on the first floor may be detectable from some rooms on the first floor, and also from a few rooms on the third floor, but from no rooms on the second floor (perhaps due to channeling through ducts); its range may be represented by the set of locations comprised of rooms 135, 136, 137, 301 and 352. There are neither geometric nor topological constraints on the locations in a beacon's range. In our example, rooms 135–137 are separated by a few meters from each other, but by two floors from the others.

Since the locations in each beacon's range are represented by vertices in the location graph, the collection of beacon ranges may be viewed as a set of *hyperedges* on the same set of vertices, yielding a *beacon hypergraph*. Intuitively, a hypergraph generalizes a graph by permitting edges that are incident on multiple vertices: one, two (as in a graph), or several.

We refer to the combination of the location graph and the beacon hypergraph, on a common set of vertices, as the *beacon map*. In more detail, a beacon map consists of a 4-tuple (V, E, H, w) where

- each vertex in the set V represents an interesting location,
- each edge in the set $E \subseteq V \times V$ is incident on a pair of neighboring locations in the physical environment, in the sense defined above,
- each hyperedge in the collection of nonempty sets $H \subseteq 2^V \setminus \emptyset$ represents the range of a beacon as the set of locations from which it is detectable, and
- the weight function $w : E \rightarrow \mathbb{R}^+$ maps each edge in $(a, b) \in E$ to a positive real number denoting the distance between the a and b in the location graph.

C. Traversal

Consider a traveler at location u at time t. In the beacon map $M = (V, E, H, w)$, we may identify location u with the corresponding vertex $u \in V$. If the traveler moves with maximum speed s then the possible locations at time t' are only those whose distance from u is at most $s \cdot (t' - t)$. We model the path of a traveler as a sequence of locations at time points $t_0, t_1, t_2, \ldots$ that are separated by a fixed time-step $c_t = t_i - t_{i-1}$ for all $i \geq 1$. For ease of presentation, it is convenient to augment the set E of graph edges as follows: We add edges connecting each vertex $u \in V$ to all vertices $v \in V$ whose distance from u is at most $s \cdot c_t$. That is, we add to E all edges in the set $\{(u, v) \mid u, v \in V \wedge u \neq v \wedge d(u, v) \leq s \cdot c_t\}$, where $d(u, v)$ denotes the length of a shortest path from u to v in the graph (V, E).

In what follows, we assume that the location graph, and thus the beacon map, have been modified in this manner. With this modification, it is no longer necessary to carry the weights on the edges of the location graph, and we will henceforth skip them from the location graph and the beacon map $M = (V, E, H)$. If a traveler is at location $u \in V$ at some point t_i in time, then the the set of potential locations after one time step, at t_{i+1}, are precisely those locations that are neighbors of u in the graph (V, E), along with u itself; i.e., the *closed neighborhood* of u: $N[u] = \{v \in V \mid v = u \vee (u, v) \in E\}$.

III. LOCALIZATION

A. Localization by Beacon Visibility

In order to avoid the well-documented problems with the use of Bluetooth signal strength for localization, our *cell-based localization* method ignores signal strength and uses only the visibility (detectability) of beacons. Since the range of each Bluetooth-based beacon is typically small, a knowledge of the ranges of the available beacons (the map) and their detectability status from an unknown location permits the location to be determined to within a small region, as illustrated by the following example, adapted from earlier work [2].

Consider five beacons, A, B, C, D, and E, with ranges of varying shapes and sizes, as suggested by Figure 1. Now suppose the traveler is at some location from which beacons B, C, and D are visible, while beacons A and E are not visible. We may conclude that the traveler is located in the lower shaded region. Note that we use information on both visibility and non-visibility of beacons to determine the region of the traveler.

B. Visibility Patterns and Induced Regions

The *complement* $\bar{h}$ of a hyperedge h of a hypergraph (V, H) is the hyperedge containing the vertices in V that are not in h: $\bar{h} = V \setminus h$. It is convenient to associate the complement of a beacon's hyperedge with a virtual beacon that is visible from exactly those locations from which the real beacon is not visible. A *beacon visibility pattern* over the set H of hyperedges is a set that includes exactly one of h and $\bar{h}$ for each $h \in H$. Intuitively, a visibility pattern includes h if the beacon represented by h is visible, and $\bar{h}$ otherwise. In other terms, a beacon visibility pattern includes exactly one beacon from each pair of complementary real and virtual beacons. In the example of Figure 1, the illustrated beacon visibility pattern is $\{\bar{A}, B, C, D, \bar{E}\}$.

The *region induced by a beacon visibility pattern* Z is, intuitively, the set $R(Z)$ of locations at which beacon visibility matches Z. That is, $R(Z)$ contains a location u if the set of beacons visible from u is exactly the set of real beacons (uncomplemented hyperedges) in Z, with all other beacons (complemented hyperedges) not being visible. More concisely, $R(Z)$ is the intersection of all the real and virtual beacons in Z: $R(Z) = \bigcap_{g \in Z} g$. In the example of Figure 1, $R(\{\bar{A}, B, C, D, \bar{E}\})$ is the set of locations in the lower shaded region of the figure. (The vertices themselves are omitted from the figure.) Thus, localization by beacon visibility consists of mapping an observed beacon visibility pattern Z to its induced region $R(Z)$ of locations.

Every location belongs to the induced region of exactly one beacon visibility pattern because distinct patterns disagree on the visibility of at least one beacon, which must be either visible or not visible from the location. In general, an induced region may contain multiple locations. For example, the lower shaded region of Figure 1, which is the region induced by pattern $\{\bar{A}, B, C, D, \bar{E}\}$, may contain 10 locations. It is clear that multiple locations in an induced region cannot be distinguished using beacon visibility. Therefore, they may be treated as a single aggregate location. For ease of presentation, we henceforth assume, without loss of generality, that the beacon map has at most one location in each induced region.

C. Localization by Beacon Probes

Determining the set of Bluetooth-based beacons that are visible from a location requires the execution of the Bluetooth *inquiry* protocol which, by specification [3], requires approximately 20 seconds. The inquiry protocol cannot be terminated early without risking that some in-range beacons are not detected, leading to incorrect localization. However, the visibility of a single beacon may be determined by using the Bluetooth *paging* protocol, which requires approximately 2.5 seconds in the worst case, and is often faster as it can be safely terminated early on receipt of a suitable response.

These characteristics of the Bluetooth protocol suggest a *probe-based method* for localization. This method uses a beacon map and the last known location to determine potentially visible beacons in the vicinity of the current location and then determines their visibility using direct probes (Bluetooth paging protocol). In the example of Figure 1, we may probe for beacons B, C, and E. If B and E are visible, but C is not visible, then the current location must lie in the upper shaded region.

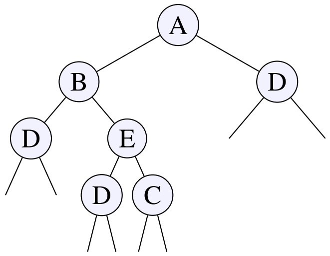
This probe-based method for localization leads us to an important question: How do we determine the set of beacons to be probed and, further, in what order should they be probed for the fastest localization? In the example of Figure 1, probing for B, C, and E, in that order, yields the earlier result. However, it is not clear that the first beacon to be probed should be B. If we begin by probing D and determine that it is not visible, we may follow up with probes for C (not visible), E (visible), and B (visible), to arrive at the earlier conclusion again, but this time requiring four probes instead of three. In order to minimize localization time, we should probe as few beacons as possible.

In general, the results of earlier probes may be used to guide the selection of beacons for later probes. In the example of Figure 1, if a probe for beacon A determines it is visible then there is no need for any additional probes, and no benefit from such probes because the current location cannot be refined to a region smaller than the disc-shaped range of A. However, if the initial probe determines that A is not visible, additional probes are necessary for localization. As before, the choice of the next beacon to be probed is not obvious.

D. Beacon-Probing Plans

A *beacon-probing plan*, or simply plan, is a description of an adaptive sequence of beacon probes, with later probes depending, in general, on the results of earlier probes. More precisely, a beacon-probing plan is either an empty plan ϵ (the base case, denoting no probes) or a triple (h, P_1, P_2) where h is a beacon to be probed and (recursively) P_1 is a beacon-probing subplan that is executed when the probe determines h is not visible while, similarly, P_2 is a beacon-probing subplan that is executed when the probe determines h is visible. We abbreviate (h, ϵ, ϵ) by (h). For example, $P_1 = (A, (B, (D), (E, (D), (C))), (D))$ is a plan that first probes beacon A. If A is visible then the plan probes D and terminates. Otherwise, the plan probes B. If B is not visible, the plan probes D and terminates; otherwise, the plan probes E, followed by either D or C, based on E's visibility.

It is natural to associate a binary tree $T(P)$ with a beacon-probing plan P. If P is empty, so is $T(P)$. Otherwise, $P = (h, P_1, P_2)$ and $T(P)$ has a root with label h; the left and right subtrees of the root are, respectively, $T(P_1)$ and $T(P_2)$ (recursively). The above plan P_1 is associated with the following binary tree $T(P_1)$:



We shall use the tree and linear representations of a plan interchangeably.

E. Plan Executions

Each path from the root of a plan to one of its leaves (terminal empty nodes) corresponds to a potential sequence of probe outcomes. We refer to each of these outcomes as an *execution* of the plan, and associate it with the corresponding leaf of the tree. We encode an execution that probes beacons $b_1, b_2, \ldots, b_k$ as a sequence $x_1, x_2, \ldots, x_k$ where x_i is b_i if the probe determines b_i is visible and $\bar{b}_i$ otherwise, for $i = 1, 2, \ldots, k$. For example, the execution associated with the third leaf (from the left) of the plan P_1 above is $\bar{A}B\bar{E}\bar{D}$. This execution probes beacons in the sequence A, B, E, D and determines that B is the only one of these four beacons that is visible.

Not all executions are *feasible* for a given beacon map. For example, the execution of P_1 that is associated with the second leaf from the right in $T(P_1)$ is $A\bar{D}$. This execution is not feasible because, in the beacon map of Figure 1, there is no region (hence, no location) where A is visible and D is not.

Executions are similar to the beacon visibility patterns of Section III-A, but there are two differences: (1) An execution may not indicate the visibility of all available beacons. (2) Executions list beacons in the order they are probed, while order is irrelevant in visibility patterns.

F. Complete Plans and Formal Problem Statement

The region $R(L)$ induced by an execution $L = x_1, x_2, \ldots, x_k$ of a beacon-probing plan is defined analogously to the region induced by a visibility pattern: It is the set of locations at which beacon visibility matches L, i.e., locations at which beacon b_i is visible if $x_i = b_i$ and not visible if $x_i = \bar{b}_i$. Recalling that beacons are identified with hyperedges that are sets of locations, we have $R(L) = \bigcap_{i=1}^{k} x_i$.

A *complete beacon-probing plan* is a beacon-probing plan for which each feasible execution induces a region that contains at most one location. Recall, from Section III-B, that each there is at most one location in each region of the beacon map. However, the regions induced by a beacon-probing plan may include aggregations of the regions of the beacon map. Since we wish localization to be as accurate as permitted by the deployed collection of beacons, as modeled by the beacon map, we require our plans to be complete in this sense.

We refer to the number of probes resulting from the execution of a beacon-probing plan as the *cost of that execution*. As illustrated earlier, this cost depends, in general, on the results of the probes. These results, in turn, depend on the actual movement of the traveler in the environment, modeled by edge traversals in the location graph. To model this dependence on traveler behavior, we associate each vertex u of the location graph with a *transition probability function* p_u that maps each neighboring vertex $v \in N[u]$ to the conditional probability of a traveler being at location v at time t_i given that the location at time t_{i-1} is u. The *cost of a beacon-probing plan* from location u is defined as the expected cost of an execution of that plan, given the transition probability function p_u. Recall that the closed neighborhood $N[u]$ includes u, so that the traveler remains at u with probability $p_u(u)$.

We may now formalize the problem of cell-based localization using a minimum number of beacon probes as follows:

> *Min-cost Complete Beacon-probing Plan (MCBP)*: Given a beacon map $M = (V, E, H)$, composed of location graph (V, E) and beacon hypergraph (V, H); a location $u \in V$ (the previous location); and the transition probability function p_u, find a minimum-cost complete beacon-probing plan P.

IV. Plan Generation

A. Smallest Regions First

Consider the instance of the MCBP problem suggested by the beacon map fragment depicted by Figure 2. The previous location is 1 and all transition probabilities are identical; that is, at the current time step, the traveler who was at location 1 previously is equally likely to be at each of the 6 locations in the closed neighborhood of 1. The hyperedges of the beacon hypergraph are: $A = \{1\}, B = \{2\}, C = \{3, 4, 5, 6\}, D = \{3, 4, 5\}, E = \{4, 5\},$ and $F = \{5\}$.

A simple method for generating a beacon-probing plan is to prioritize probing beacons by their likelihood of yielding executions that induce single-location regions. In our example instance, all transition probabilities are equal, so this guideline suggests prioritizing beacons that have a very small range, or a very large range (so that the complement is small). Applying this method to our example yields the plan $P_{11} = (A, (B, (D, \epsilon, (E, \epsilon, (F))), \epsilon), \epsilon)$ which is depicted in tree form in Figure 3. In that figure, and in similar ones that follow, we use round interior nodes to denote the probed beacons and square leaf nodes to denote the inferred locations. Each tree edge is labeled with the conditional probability of descending from parent to child along that edge (i.e., of continuing the plan execution along that path), conditioned on the plan execution having arrived at the parent.

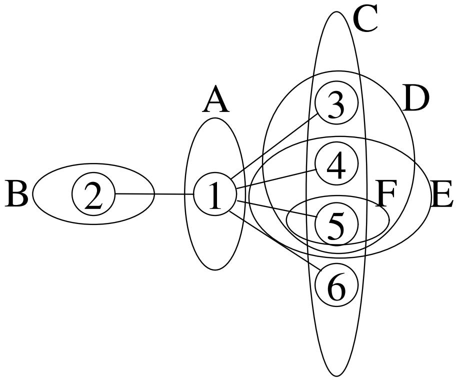


Fig. 2. A beacon map fragment, depicting the neighborhood of prior location 1. Numbered circular nodes $1, \ldots, 6$ denote locations, i.e., vertices of the location graph. Edges of the location graph are depicted as straight line segments. Hyperedges of the beacon hypergraph, i.e., the ranges of the six beacons $A, \ldots, F$, are depicted as regions enclosed by the corresponding ovals.

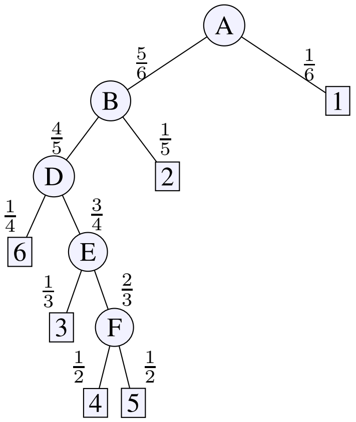


Fig. 3. A smallest-regions-first plan for the example of Figure 1.

The cost of a beacon-probing plan (expected number of probes) is easily computed from the above augmented tree representation. The cost of the plan P_{11} of Figure 3 is $c(P_{11}) = 1 + \frac{1}{6} \cdot 0 + \frac{5}{6} \cdot (1 + \frac{1}{5} \cdot 0 + \frac{4}{5} \cdot (1 + \frac{1}{4} \cdot 0 + \frac{3}{4} \cdot (1 + \frac{1}{3} \cdot 0 + \frac{2}{3} \cdot 1))) = 1 + \frac{5}{6} + \frac{4}{6} + \frac{3}{6} + \frac{2}{6} = 3\frac{1}{3}$.

While it is simple, the above strategy for plan generation does not produce minimum-cost plans in general. For our example, the plan $P_{12} = (D, (A, (B), \epsilon), (E, \epsilon, (F)))$ which is depicted in tree form in Figure 4, has cost $c(P_{12}) = 1 + \frac{1}{2} \cdot (1 + \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot 0) + \frac{1}{2} \cdot (1 + \frac{1}{3} \cdot 0 + \frac{2}{3} \cdot 1) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{2} + \frac{1}{3} = 2\frac{2}{3} < c(P_{11})$.

B. Bottom-Up Plans

An alternate strategy is to generate plans bottom-up, in a manner similar to Huffman coding and related methods [4]. The plan tree my be generated in a bottom-up manner by, at

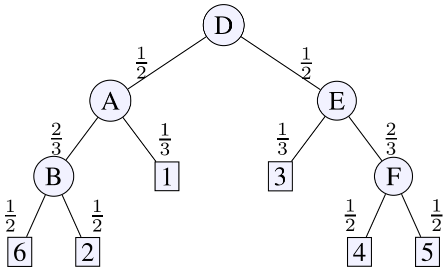


Fig. 4. An alternate for the example of Figure 1, with cost lower than that of the plan in Figure 3.

each step, forming an interior node whose two children are previously produced subtrees whose leaves have the lowest aggregate likelihoods. For our running example, this strategy produces the plan $P_{13} = (B_{56}, (B_{34}, (B_2), (B_4)), (B_5))$ which is depicted in tree form in Figure 5. The cost of this plan is $c_(P_{13}) = 1 + \frac{2}{3} \cdot (1 + \frac{1}{2} \cdot (1+1)) + \frac{1}{3} \cdot (1) = 1 + \frac{2}{3} + \frac{2}{3} + \frac{1}{3} = 2\frac{2}{3}$. However, this plan and, in general, other plans generated in this manner, are not feasible. In our example, for instance, there is no single beacon probe that can play the role of B_{56} in Figure 5, i.e., that can distinguish the set of locations $\{5, 6\}$ from the set $\{1, 2, 3, 4\}$.

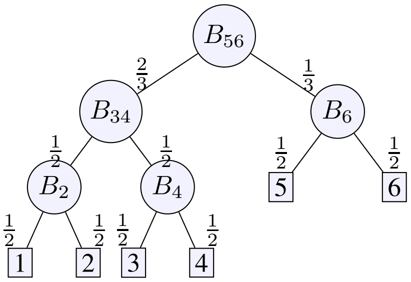


Fig. 5. A bottom-up beacon-oblivious and infeasible plan for the example of Figure 1.

For our example, another plan that may potentially be generated by a bottom-up strategy is in fact feasible, and has the same cost as P_{13}: Consider $P_{14} = (C, (A), (E, (D), (F)))$ which is depicted in tree form in Figure 6. Its cost is $c(P_{14}) = 1 + \frac{1}{3} \cdot (1) + \frac{2}{3} \cdot (1 + \frac{1}{2} \cdot (1+1)) = 1 + \frac{1}{3} + \frac{2}{3} + \frac{2}{3} = 2\frac{2}{3}$. However, it is not clear how to limit the generation of bottom up plans to such feasible ones while retaining optimality, in the general case. For instance, consider the simple beacon hypergraph suggested by Figure 7. It is easy to note that the plan $P_{21} = (E, (A), (B))$, which is depicted in tree form in Figure 8, is optimal and feasible, with cost $c_{21} = 1 + \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 = 2$. This plan is potentially generated by a bottom-up strategy that starts by distinguishing between 1 and 4 using A, then 2 and 3 using B, and then combining these two subtrees using E to distinguish $\{1, 4\}$ from $\{2, 3\}$. However, if we start by distinguishing 1 from 2 using A, then distinguishing 3 from 4 using C then there is no single beacon probe that can distinguish $\{1, 2\}$ from $\{3, 4\}$.

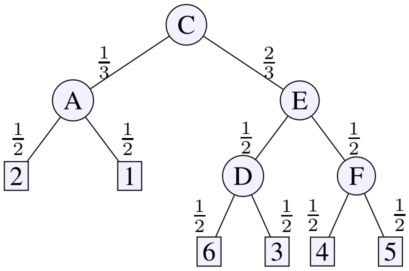


Fig. 6. An optimal plan for the example of Figure 1.

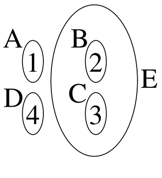


Fig. 7. A simple example illustrating difficulties with bottom-up plan generation. The ovals represent beacon ranges and the numbers denote locations.

C. Balanced Plans

Our method for generating plans is based on the idea of recursively dividing regions (sets of locations) into two partitions that are as equally probable as possible. Assume, without loss of generality, that $|N[u]| \geq 2$. (The degenerate case in which $N[u]$, the closed neighborhood of u, contains only one element is addressed separately, since in that case the traveler's location cannot change.) Set $S \leftarrow N[u]$ and invoke the following recursive function *bps*(M, u, p_u, S), which returns a beacon-probing plan.

1. Sort the locations $v \in N[u] \cap S$ in nonincreasing $p_u(v)$ order. Let $L = l_1, l_2, \ldots, l_n$ denote this sorted list.
2. Generate the set of pairs $D = (1, d_1), (2, d_2), \ldots, (n-1, d_{n-1})$, where $d_i = |\sum_{j=1}^{i} p_u(l_j) - \sum_{j=i+1}^{n} p_u(l_j)|$.
3. Sort the pairs in D in nondecreasing order of d_i; let D' denote the sorted list of pairs.
4. Let (m, d_m) be the leftmost element of D' such that there is a beacon (hyperedge) $h_m \in H$ such that either $\{l_j\}_{j=1}^{m} \cap S = h_m$ and $\{l_j\}_{j=m+1}^{n} \cap S \cap h_m = \emptyset$ or $\{l_j\}_{j=1}^{m} \cap S \cap h_m = \emptyset$ and $\{l_j\}_{j=m+1}^{n} \cap S = h_m$.
5. Recursively generate the subplans
   $p_1 =$ *bps*$(M, u, p_u, S - h_m)$ and
   $p_2 =$ *bps*$(M, u, p_u, S \cap h_m)$.
6. Return the plan (h_m, p_1, p_2).

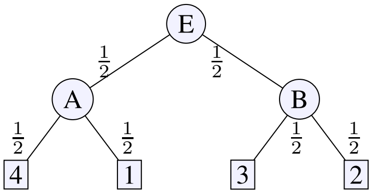


Fig. 8. An optimal plan for the example of Figure 7.

V. RELATED WORK

Our work on indoor localization is motivated by work on marker-based localization for pedestrians in indoor and outdoor settings, notably M-CubITS [5], [6], [7], [8], [9]. While less popular than 802.11-based localization, Bluetooth-based localization has been the subject of some recent work [10] [11] [12] [13] [14] [15] [16] [17] . However this work is based on the use of signal strengths for distance estimation, which has significant drawbacks as discussed earlier. In prior work, we introduced cell-based localization for Bluetooth [2] and, in particular, addressed the beacon placement problem along with some practical issues related to deployment. We have also presented a marker-based localization method that uses a short sequence of recently encountered markers, instead of only the currently visible markers [18]. That approach may be used in conjunction with the map-based methods to reduce the number of distinct beacon-identifiers required for localization.

Gelzayd's thesis provides a good overview of Bluetooth technology in general, and the protocol parameters in particular, such as the worst case paging delay of approximately 2.56 seconds [19]. Salonidis et al. [20] discuss methods for speeding up the Bluetooth protocol. An interesting, and very different, approach to reducing the latency associated with the Bluetooth inquiry protocol is to use infrared (IrDA) communications to bootstrap the Bluetooth protocol [21]. While the need for both kinds of receivers is a drawback, it may be interesting to incorporate some of those ideas for localization proposes. Finally, there are some interesting connections between data compression and coding [4] and the localization problem of this paper.

VI. CONCLUSION

Indoor localization using Bluetooth devices as beacons is attractive due to the very low cost of these devices, which allows them to be deployed in large numbers. Their limited range makes them particularly well suited to cell-based localization, so that the unreliable mapping of signal strength to distance is completely avoided. However, a key hurdle is the long time (20 seconds) required by the Bluetooth inquiry protocol to determine the devices within communication range, i.e., the visible beacons. We described a method to overcome this hurdle using judiciously selected probes of individual beacons. These probes are implemented using the Bluetooth paging protocol, which is much faster (2 seconds) than the inquiry protocol. We modeled and formalized the main problem in this regard as the problem of finding a minimum-cost complete beacon-probing plan, and presented an algorithm for creating such plans. In continuing work, we are extending the theoretical results to improve running time as well as optimality guarantees, as well as conducting additional field experiments.

REFERENCES

[1] B. Hofmann-Wellenhoff, H. Lichtenegger, and J. Collins, *GPS: Theory and Practice*. New York: Springer-Verlag, 1994.

[2] Sudarshan S. Chawathe, "Beacon placement for indoor localization using Bluetooth," in *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Beijing, China, Oct. 2008, pp. 980–985.

[3] Bluetooth Special Interest Group, "Specification of the Bluetooth system," http://www.bluetooth.com/, Apr. 2009, version 3.0 + HS.

[4] Debra A. Lelewer and Daniel S. Hirschberg, "Data compression," *ACM Computing Surveys*, vol. 19, no. 3, pp. 261–296, Sept. 1987.

[5] Tetsuya Manabe, Seiji Yamashita, and Takaaki Hasegawa, "On the M-CubITS pedestrian navigation system," in *Proceedings of the 9th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Toronto, Canada, Sept. 2006, pp. 793–798.

[6] Je-Yeon Kim and Takaaki Hasegawa, "An experimental study on the positioning by M-CubITS," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, D.C., Oct. 2004, pp. 977–981.

[7] Seiji Yamashita and Takaaki Hasegawa, "On the M-CubITS pedestrian navigation system by a camera-equipped mobile phone," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, D.C., Oct. 2004, pp. 714–717.

[8] ——, "On the M-CubITS pedestrian navigation system using textured paving blocks and its experiments," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Vienna, Austria, Sept. 2005, pp. 121–125.

[9] Je-Yeon Kim and Takaaki Hasegawa, "On repositioning of the M-sequence lane markers system in highways and applications in intersections," in *Proceedings of the 6th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Shanghai, China, Oct. 2003, pp. 520–523.

[10] Timothy M. Bielawa, "Position location of remote bluetooth devices," Master's thesis, Virginia Polytechnic Institute and State University, June 2005.

[11] Sheng Zhou and John K. Pollard, "Position measurement using Bluetooth," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, p. 555, May 2006.

[12] Kiran Thapa and Steven Case, "An indoor positioning service for Bluetooth ad hoc networks."

[13] Kenneth C. Cheung, Stephen S. Intille, and Kent Larson, "An inexpensive Bluetooth-based indoor positioning hack."

[14] Josef Hallberg, Marcus Nilsson, and Kare Synnes, "Positioning with Bluetooth," in *Proceedings of the 10th International Conference on Telecommunications*, 23 Feb–1 March 2003, pp. 954 – 958.

[15] G. Anastasi, R. Bandelloni, M. Conti, F. Delmastro, E. Gregori, and G. Mainetto, "Experimenting an indoor Bluetooth-based positioning service," *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops*, pp. 480 – 483, 19-22 May 2003.

[16] Rune Bentsen, Thomas Alfred Soby Dalsgaard, and Anderson Bo Pedersen, "findIT: Bluetooth positioning project."

[17] Silke Feldmann, Kyandoghere Kyamakya, Ana Zapater, and Zighuo Lue, "An indoor Bluetooth-based positioning system: Concept, implementation and experimental evaluation," in *Proceedings of the International Conference on Wireless Networks (ICWN)*, Las Vegas, Nevada, June 2003, pp. 109–113.

[18] Sudarshan S. Chawathe, "Marker-based localizing for indoor navigation," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Seattle, Washington, Oct. 2007, pp. 885–890.

[19] Yelena Gelzayd, "An alternate connection establishment scheme in the Bluetooth system," Master's thesis, Polytechnic University, Jan. 2002.

[20] Theodoros Salonidis, Pravin Bhagwat, and Leandros Tassiulas, "Proximity awareness and fast connection establishment in Bluetooth," in *Proceedings of the First Annual Workshop on Mobile Ad Hoc Networking Computing (MobiHOC)*, Boston, Massachusetts, Aug. 2000.

[21] Ryan W. Woodings, Derek D. Joos, Trevor Clifton, and Charles D. Knutson, "Rapid heterogeneous ad hoc connection establishment: Accelerating Bluetooth inquiry using IrDA," in *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, Orlando, Florida, Mar. 2002, pp. 342–349.