Personal and Ubiquitous Computing manuscript No.
(will be inserted by the editor)

Efficient and Accurate Sensor Network Localization

Md. Tareq Adnan · Suprakash Datta · Stuart MacLean

Received: date / Accepted: date

Abstract Wireless Sensor Networks have great potential in Ubiquitous computing. However, the severe resource constraints of wireless sensor networks rule out the use of many existing networking protocols and require careful design of systems that prioritizes energy conservation over performance optimization. A key infrastructural problem in wireless sensor networks is localization – the problem of determining the geographical locations of nodes. Wireless Sensor Networks typically have some nodes called seeds that know their locations using Global Positioning Systems (GPS) or other means. Non-seed nodes compute their locations by exchanging messages with nodes within their radio range.

Several algorithms have been proposed for localization in different scenarios. Algorithms have been designed for networks in which each node has ranging capabilities, i.e., can estimate distances to its neighbours. Other algorithms have been proposed for networks in which no node has such capabilities. Some algorithms only work when nodes are static. Some other algorithms are designed specifically for networks in which all nodes are mobile. We propose a very general, fully distributed localization algorithm RMCB (Range-based Monte Carlo Boxed) for wireless sensor networks. RMCB allows nodes to be static or mobile and that can work with nodes that can perform ranging as well as nodes that lack ranging capabilities. RMCB uses a small fraction of seeds. It makes use of the received signal strength measurements that are available from the sensor hardware. We use RMCB to investigate the question: "When does range-based localization work better than range-free localization?" We demonstrate using empirical signal strength data from sensor hardware (Texas Instruments EZ430-RF2500) and simulations that RMCB outperforms a very good range-free algorithm WMCL (Weighted Monte Carlo localization) in terms of localization error in a number of scenarios and has a similar computational complexity to WMCL. We also implement WMCL and RMCB on sensor hardware and demonstrate that it outperforms WMCL. The performance of RMCB depends critically on the quality of range estimation. We describe the limitations of our range estimation approach and provide guidelines on when range-based localization is preferable.

Keywords Wireless Sensor Networks · Localization · Ranging · Monte Carlo Sampling

Md. Tareq Adnan
Computer Science and Engg. Dept.
York University, Toronto, Canada
E-mail: adnan@cse.yorku.ca

Suprakash Datta
Computer Science and Engg. Dept.
York University, Toronto, Canada
E-mail: datta@cse.yorku.ca

Stuart MacLean
Computer Science and Engg. Dept.
York University, Toronto, Canada
E-mail: smaclean@cse.yorku.ca

1 Introduction

Wireless sensor networks (WSNs) are an emerging technology that have numerous possible applications including many areas of personal and ubiquitous computing. Sensor nodes can be used to build body area networks [7], smart homes [5] and intelligent environments [14]. They are also used for a variety of monitoring applications [31,13]. WSNs are formed by small autonomous nodes that contain a CPU, memory, battery, a wireless transceiver and some sensors that measure physical attributes, e.g. temperature, velocity, light etc. These nodes are expected to self-organize into a coherent network that can respond to queries from external users and report about sensed phenomena. The attractive aspects of WSNs include their low cost, great flexibility and the robustness arising from the distributed nature of the network. The proposed practical uses of WSNs include areas such as environmental sensing [19, 32, 20], industrial monitoring, military applications [18], on-site tracking of materials [23], and security monitoring [18]. The design challenges for these networks include the crippling limitations on processing, memory, battery capacity and also the likelihood of sensor node failure. Indeed, these challenges require that several problems that have been solved for wired and wireless networks be addressed again because resource conservation rather than performance optimization is the primary objective for WSNs.

A key infrastructural problem in the design and deployment of WSNs is *localization*: the estimation of the geographical locations of sensor nodes. Location information is essential for many applications like location aware services [29, 11] and enhanced security protection mechanisms [34]. In addition, basic middleware services such as routing often rely on location information like geographic routing [17, 6, 8, 16] and context-based routing protocols [9, 12]. Recently, location aware services have received much attention. For example, a WSN deployed in hospitals can be used to keep track of patients, doctors, nurses, and medical equipment. WSNs can be used to create responsive environments. For example, a WSN in a museum or exhibition can guide the visitors by providing them explanations based on their locations. WSNs can also be used in enterprises for locating assets and employees in and around the workplace as well as reducing theft and loss of valuable assets. Animal tracking and home automation for the elderly and disabled are also examples of applications of WSNs.

Manual localization is feasible in small networks of static nodes. When sensor networks are large, mobile or deployed in hostile environments, automated node localization is required. Localization algorithms can be centralized or distributed. In this work, we focus on designing fully distributed algorithms that can be implemented on real sensor platforms.

There are several design choices that have to be made in building sensor networks that allow accurate localization. The first choice pertains to whether nodes can measure distances between each other. Given the low cost and small size of the nodes and the need to preserve power, many sensor nodes do not provide any means of obtaining distance information. Therefore a lot of research has gone into *range-free* localization, or localization without any distance information. The only information available to nodes is whether another node is within radio range of it, and this essentially provides an upper bound on the distance between nodes. This idea can be extended to use *hop distances* – or the minimum number of hops required for a message to travel from one node to another. In contrast, *range-based* localization makes use of distance information obtained in some manner, which includes using radio signal strength measurements, use of time delays of signals and other means. In this paper, we focus on localization using distances estimated from radio signal strengths of received packets.

Yet another design choice is the use of local or global coordinates. We choose the latter since it is more general. In order to localize nodes in a global coordinate system, a subset of the nodes (called *seeds*) must have the capability to determine their locations at all times. Non-seed nodes compute their locations by communicating with other nodes. Finally, localizing static nodes is simpler and earlier work on localization assumed that sensors were static. Later papers proposed algorithms that worked in the presence of node mobility.

1.1 Range-free or range-based?

There is a key question faced by WSN designers that has not been addressed much in the literature: which is desirable – range-free or range-based localization? This is not a simple question, and the answer depends on many factors. It would be natural to assume that if ranging can be done, range-based algorithms should yield higher localization accuracy since they use more accurate location information. However, we describe several factors that prevent this.

In this paper, we propose a general localization algorithm RMCB that can accommodate mobile nodes and works when some (possibly all or none) of the nodes have ranging capability. No algorithms have been proposed to work in all these scenarios to our knowledge. RMCB is designed to make use of a unified framework for node localization in both ad hoc and sensor networks. We evaluate the algorithm under a variety of scenarios. In each scenario, a node receives location information from some or all of its neighbours (seeds and non-seeds) and uses this information to compute its location. We show that in each such scenario, the information available to a node can be distilled down to a set of constraints that the location of the node must satisfy. These constraints would depend on whether the nodes are mobile or not, and whether ranging is used or not. However, the process of estimating the location is not dependent on the specific constraints used. Seen from this point of view, a good localization algorithm must make good use of the constraints available to it.

In order to answer the question raised in this subsection, we built a simulator for RMCB and also implemented it on sensor hardware. We chose very good range-free localization algorithm called WMCL [35] for comparison (See Section 5 for reasons for this choice) and also implemented WMCL on our hardware. We conducted simulation studies and experiments with sensor hardware to compare the performance of RMCB and WMCL. Our results demonstrate that RMCB outperforms WMCL in many scenarios. However, the performance of RMCB depends critically on the quality of range estimation. We describe the limitations of our range estimation approach and provide guidelines on when range-based localization is preferable.

1.2 Related Work

A comprehensive survey of the WSN literature on localization is beyond the scope of this paper. We refer the reader to the surveys by Mao *et al.* [22], Bachrach and Taylor [1] and by Savvides *et al.* [28] for a more comprehensive overview of the literature. We only deal with fully distributed algorithms in this paper, and refer the reader to the survey by Mao *et al.* [22] for details on centralized algorithms.

1.2.1 Range-based localization

Various techniques have been proposed for range-based localization algorithms. Ward *et al.* [33] used *Time of Arrival* of signals and Priyantha *et al.* [24] and Savvides *et al.* [27] used *Time Difference of Arrival* of messages to estimate distances. Sugano *et al.* [30], Bahl *et al.* [3] and Bischoff *et al.* [4] used received signal strength (RSS) to estimate distances. In Bahl *et al.* [3], distances from 3 fixed beacons are input to a triangulation-based algorithm for computing positions within a building. In [4] the distance estimates are used for topology maintenance but localization in a global coordinate system is not attempted. Sugano *et al.* [30] propose a localization algorithm for mobile nodes using distance estimates from static sensor nodes. Havinga *et al.* [10] proposed a range-based algorithm based on a Monte Carlo sampling approach. They assume that range measurements are available and do not discuss where the measurements come from. In addition they assume range measurements to be single values rather than upper and lower bounds.

1.2.2 Range-free localization

Many range-free localization algorithms have been proposed in the literature. We survey only the work relevant to this paper. Hu and Evans [15] presented a Monte Carlo Sampling-based algorithm called MCL (Monte Carlo Localization). Rudafshani and Datta [26] proposed algorithms MSL and MSL*; both these algorithms improved on MCL by using more location information. The MCB (Monte Carlo Localization Boxed) algorithm proposed by Baggio and Langendoen [2] improved on the computational complexity of the algorithms mentioned above by using bounding boxes. Zhang *et al.* [35] further improved the computational complexity and localization accuracy in algorithm WMCL (Weighted MCL). More recently, Maclean and Datta proposed algorithm Orbit [21]. Orbit uses properties of unit disk graphs to derive stronger constraints on node locations and this yields lower localization errors than existing algorithms. Although Orbit is able to use both range-free and range-based location information, its performance was evaluated in [21] only for range-free localization.

1.2.3 Distance from received signal strength

There have been several attempts at constructing models that allow computation of distances from measured signal strength, including [3,30]. We found that measurements taken with our hardware did not fit any of these models and therefore we chose a simple, intuitive algorithm that is explained in Section 4.2.

2 Issues in range-based localization

There are two primary computational challenges in the design of localization algorithms. The first challenge is how to use location information of the neighbours of a node, particularly the imprecise location estimates of non-seed neighbours. The second challenge is the maintenance of one or more feasible locations of a node at each time step. We address the first problem by deriving constraints on the set of possible locations of a node from all available information. The second problem is addressed by sampling points from a region generated using the set of available constraints and filtering out points that do not satisfy the constraints. The generation of samples and their filtering are done in a manner very similar to WMCL [35].

2.1 Location information as constraints

Let us first assume nodes are static. For simplicity, let us assume that radio range is a perfect circle with radius

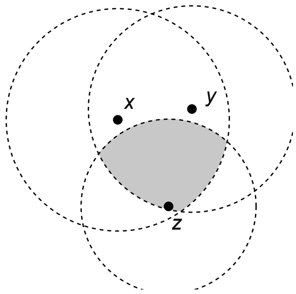

Fig. 1 Constraints defined by mobility

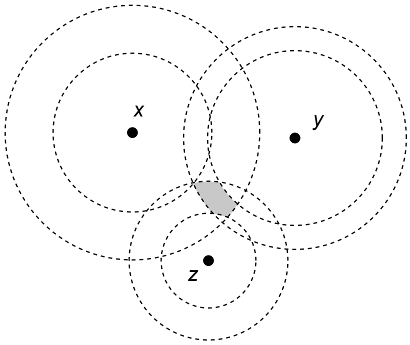

Fig. 2 Constraints defined by ranges

r. Each node within radio range of a node is called its *neighbour*; if a node y is a neighbour of a neighbour of x, but y is not a neighbour of x, then y is said to be a *second neighbour* of x.

The key observation in the design of our algorithm is that each piece of location and ranging information from neighbours is a (*positive*) constraint on the set of possible locations of a node. In the absence of ranging, a node is within a circle of radius r centred at the location of each seed neighbour. Each non-seed neighbour m has an approximate location with a position p_m and an upper bound on the positional error ϵ_m. Therefore a node must be within a circle centred at p_m and radius $r + \epsilon_m$. The location of non-neighbour seeds serve as *negative* information - a node cannot be in a circle of radius r centred at the location of a non-neighbour. In this paper we only consider first and second neighbours of nodes to generate constraints.

Mobility can be handled by incorporating $v_{\max}$ (as defined in Section 3) into the constraints. For example, if a node knows the location of a seed neighbour at the previous time step, it must be within a circle centred at that location and radius $r + v_{\max}$ (Figure 1).

Range information is also easily incorporated. Since distance measurements are approximate in practice, particularly when made from RSS, we assume that we are given upper and lower bounds on each measurement. Thus a node x is constrained to lie in an annular region defined by the location of a neighbour j and the upper and lower bounds on the distance, i.e. if $d(x, j)$ is the distance between the locations of x, j, and l_j, u_j are the lower and upper bounds on the distance between x, j then $l_j \leq d(x, j) \leq u_j$. The upper bound ($d(x, j) \leq u_j$) works as a positive constraint and the lower bound ($l_j \leq d(x, j)$) as a negative constraint (see Figure 2). Uncertainties in the locations and mobility are handled exactly as in the case of range-free information. This way of interpreting constraints allows us to utilize ideas used to design range-free localization algorithms in a range-based algorithm.

It is crucial that the constraints must be *consistent* – i.e., the set of points satisfying the constraints must be non-empty, otherwise the localization algorithm fails and a node does not find any estimates for its location.

Constraints are used to generate sample locations that satisfy them. Since it is hard to generate only those points that satisfy all constraints, samples are drawn from some larger area and then filtered out if they do not satisfy one or more constraints. An example is shown in Figure 3.

2.2 Negative Information: Advantages and Risks

A constraint of the form "node i is *not* within distance ρ of node j" is called negative information in the literature. This has been used in range-free localization ("node i is not within radio range of j") to increase localization accuracy. Negative information has the potential to greatly improve the accuracy of range-based localization but it is not free of pitfalls. If negative information is too pessimistic (the distance between two nodes is *underestimated*) the constraints on locations it translates to are loose and may result in less accurate localization. In contrast, if negative information is too optimistic (the distance between two nodes is *overestimated*) then the constraints may be inconsistent and no feasible locations may be found.

3 Model and assumptions

We assume all nodes to be identical, except for seeds having self-localization ability. We assume that nodes are deployed in some manner (perhaps randomly) on a two-dimensional sensor field that is free of obstacles.

We assume that a (small) fraction of the nodes, called seeds, can determine their locations at all times, perhaps by using GPS or similar means, in a global co-ordinate system. The other nodes do not have this capability and must compute their locations by communicating with other nodes. We assume that nodes may move at any speed in the range $[0, v_{\max}]$, and $v_{\max}$ is

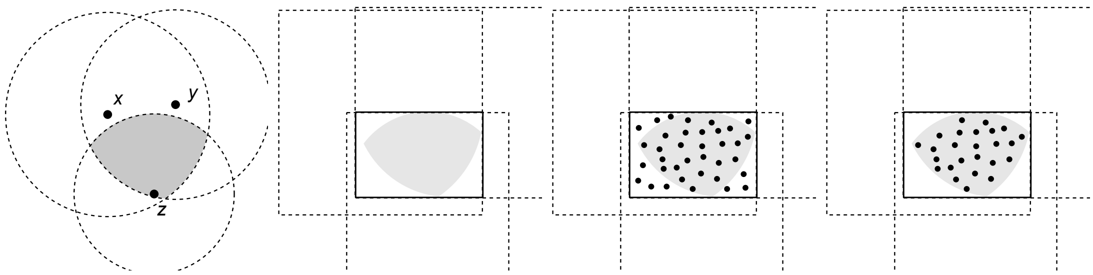

Fig. 3 (a) some constraints (b) the corresponding bounding boxes, (c) Sampled Points and (d) Sampled points after filtering

known to all nodes. We assume that nodes do not know the direction or actual speed of their movement. This assumption is made to conform with realistic networks, in which direction and speed measurements require extra hardware and may not be very accurate. If such information is available then our model and algorithm can incorporate them and improve accuracy.

We assume that time is discrete and, although not required by our algorithm, there is reasonable clock synchrony among nodes. We assume for the sake of simplicity that a reliable medium access control (MAC) layer is available to us. We do not assume the presence of a routing infrastructure for our algorithm.

3.1 Performance metrics

The primary metrics of interest to us are *average localization error* and *sampling efficiency*. We define the average localization error to be the expected value (over all nodes) of the Euclidean distance between the true location of a node and the location output by an algorithm. We exclude from consideration *isolated* nodes – nodes that are not within radio range of any other nodes. Sampling efficiency measures the fraction of samples generated that are not filtered out. It is an important measure of performance because low sampling efficiency results in wasted CPU cycles and battery power.

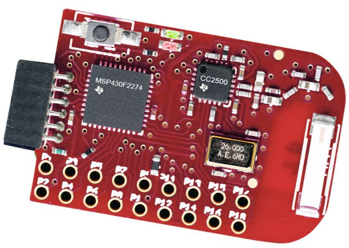

Fig. 4 Sensor hardware

3.2 Hardware

RMCB can be used with most off-the-shelf sensor hardware. We use the Texas Instruments sensor device EZ430-RF2500 for all our research. This resource constrained device is composed of the MSP430 CPU and the CC2500 radio (See Figure 4). Note that these nodes allow us to record received signal strengths (as integer values) for each packet received.

4 Algorithm RMCB

We first describe the range-free localization algorithm WMCL [35]. RMCB has the same overall structure as WMCL, and uses similar ideas in utilizing bounding boxes, filtering and weighting of samples and location computation. However, since RMCB is designed to function with both range-based and range-free nodes, it extends the use of bounding boxes for this more general scenario (Section 4.3), and makes improvements to the sampling algorithm (Section 4.4).

4.1 Algorithm WMCL

WMCL is a fully distributed algorithm. It divides each time step into two phases. In the first phase, all non-seed nodes broadcast the the set of possible locations, an estimated location (the weighted average of the set of possible locations), and the error of the estimate to its neighbours (the error is a measure of the quality of the location estimate). All seed nodes broadcast their locations to their neighbours. During the second phase of each time step, nodes receive data from their neighbours and use the received data to update their sets of possible locations, the new estimated location and the error of the new estimate. The steps in this algorithm are below. Assume that x is the node that is being localized.

1. Construct bounding box B that encloses the intersection of the positive constraints imposed by neighbour and second neighbour nodes.

2. Trim the box B using negative constraints.
3. Sample M points uniformly at random from B.
4. Eliminate (filter out) the points that are not within distance r from neighbouring seeds, or do not have distance between r and $2r$ from second-neighbour seeds, or are more than distance $e + v_{\max}$ from the estimated previous location of x, where e is the error in the estimate of the previous location of x.
5. For each remaining sample s and each non-seed neighbour k whose error is less than the error of the samples of the node x in the previous time step, compute the partial weight $w(s, k)$ as the fraction of samples of k that are within distance $r + v_{\max}$ of s. The weight of the sample s is computed as $w(s) = \prod_k w(s, k)$. The weights of all the samples s are normalized so that they sum to 1.
6. Compute the predicted location of a node as p = weighted average of filtered points (using weights computed in the previous step); also compute the error e = max distance from p to any filtered sample point.
7. Return location p and error estimate e.

While RMCB has a similar overall structure to that of WMCL, there are key differences between WMCL and RMCB are in the constraints used as well as the bounding box refinement to improve sampling efficiency. In WMCL, a neighbouring seed j of a node x gives a positive constraint $(d(x, j) \leq r)$, in RMCB , we get a positive and a negative constraint from each seed neighbour, as pointed out before. The new parts of RMCB are described next.

4.2 Distance estimation from received signal strength

Since obstacles and multipath effects greatly influence the accuracy of range estimation from received signal strength, obtaining reliable upper and lower bounds on the distance from RSS is not easy. Extensive experiments with our sensor nodes in various indoor settings (including large open areas) have revealed that standard path loss models [25] fit our data very poorly (see Figure 5). Therefore we use empirical measurements for obtaining upper and lower bounds on the distance between nodes from RSS. We measured RSS (in dBm) between nodes using the average of 4 packets. This experiment was repeated 100 times for each distance, and for 44 distances in the range 0.5 - 170 feet. From this data we compute a distribution $D(n)$ of distances for each RSS value n and another distribution $I(x)$ of measured RSS values for actual distances x. We experimented with taking the upper and lower bound of the distribution D as well as leaving out 10% of the values at each end. The latter resulted in much lower accuracy and therefore was not used in our experiments. In order to make the simulations realistic we sampled the distribution I to generate the RSS value for each actual distance x and this value was seen by the receiving node in our simulations.

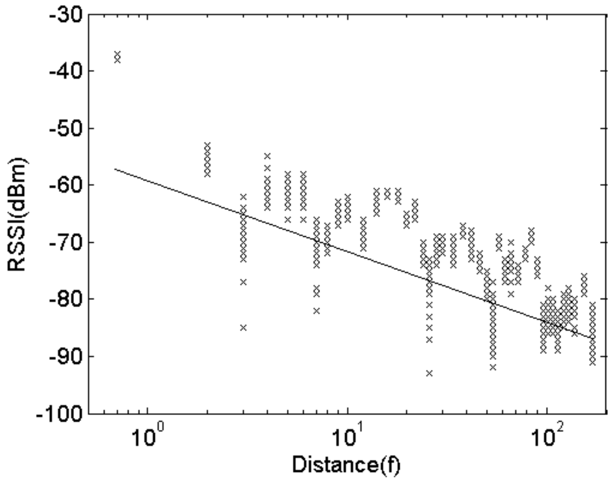

Fig. 5 Received signal strength data in an indoor hallway fitted to the path loss model Path loss $= 10n \log_{10}$ distance $+ c$ to our empirical data. The line fit corresponds to $n = -1.509, c = -63.2778$

The results here are based on indoor measurements in a long narrow hallway. In 4.7 we describe the adaptation of RMCB to other environments. We note that the environment used for these results is the one where the distance bounds are the least tight. In other scenarios we get higher accuracy than the results presented here because of more accurate distance estimation.

4.3 Details of RMCB

We assume a WSN deployed in a rectangle planar region where all nodes (seeds and non-seeds) can move. RMCB follows a generic framework of Sequential Monte Carlo sampling based localization algorithm which consists of three parts: initialization, sampling, and filtering.

In the initialization part, candidate samples for each non-seed are drawn randomly from the deployment rectangle. We assume that each time step has two phases. In the first phase, all seeds broadcast their exact locations and each non-seed broadcasts a set of possible locations, an estimated single location computed from the possible locations, and a measure of the quality of the estimate (computed at previous time step) to its neighbours. During the second phase of each time step, each non-seed node builds a sample area using the received information and its own set of locations (computed dur-

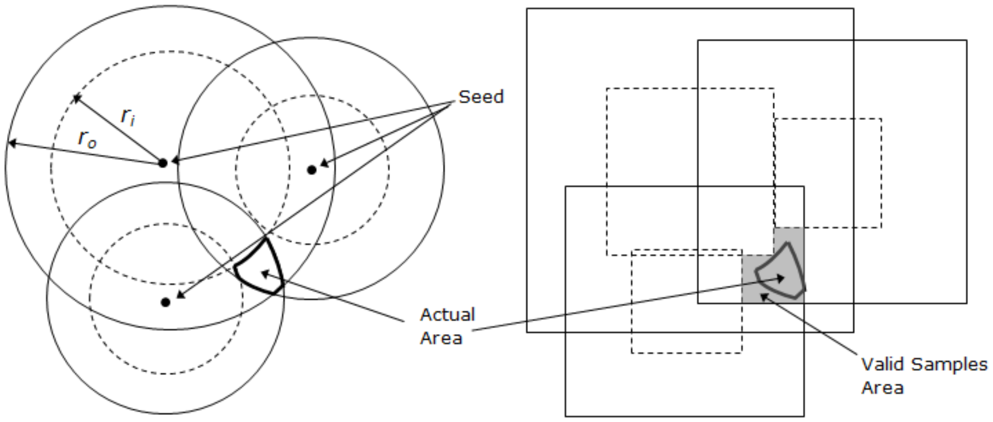

Fig. 6 Bounding box distance estimation

ing the previous time step). Then candidate samples are drawn using the sample area and each candidate sample is given weight based on the received information. Samples with weight 0 are discarded during filtering. The weighted average of the filtered samples is computed and broadcast the next time step as the estimated current location of the non-seed node. The filtered samples, their weighted average (assumed to be the current estimated location) and a quality measure associated to the current location estimate are also broadcast in the next time step.

Bounding box creation: Sequential Monte Carlo sampling based localization algorithms generate candidate samples by selecting points randomly from a candidate samples area and filter these samples by eliminating those that do not satisfy some location constraint to get the valid samples. The candidate sample area should be small to increase sampling efficiency but should not be hard to compute. Since our constraints are circular, their intersection often results in a non-convex areas or even disconnected pieces. Accurately modelling the intersection is computationally challenging. Therefore, we first build a bounding box that contains each positive constraint. The intersection of these bounding boxes forms the candidate sample area. We draw the candidate samples uniformly at random over this area.

The presence of negative constraints decreases the sampling efficiency because large portions of the candidate sample area are eliminated from the set of possible locations of a node. In RMCB , upper and lower annular bounds on each distance from the one-hop seed give a positive and a negative constraint as shown in Figure 6. The smaller the gap between the upper and lower bound, the more accurate the distance estimate. In order to use the negative information, we choose squares that properly fit inner circles (formed by lower bounds as shown by dashed squares in Figure 6). Upon excluding the negative region, the valid sampling area (the shaded/coloured area) becomes much smaller.

Two-hop seed neighbours are also used to reduce the bounding box by replacing r with $2r$. Following WMCL, *negative information* from two-hop seeds, nonseed neighbour's and own previous location estimates, and maximum error in the x-axis and y-axis are used to shrink the bounding box. We found that extra constraints from range information improve the accuracy of RMCB but at the cost of lower sampling efficiency.

4.4 Sampling efficiency improvement

If we use all available negative constraints, we potentially get a very complex area which is hard to model. We use a heuristic that involves choosing a seed whose annulus has the minimum intersection area with the outer bounding box to improve sampling efficiency and reduce computation. In order to reduce computation, we approximate the area of intersection by using squares to approximate the circles and the area between the squares to approximate the annulus. Thus we get a modified area within the bounding box with a rectangle removed as shown in the left of Figure 7.

RMCB uses a larger number of constraints compared to WMCL because of the introduction of negative information from the one-hop seed neighbours and thus its sampling efficiency may be less than that of WMCL. In order to improve the sampling efficiency, we use a heuristic that eliminates the weak constraints in the presence of strong constraints. While strong constraints includes the positive and negative information from the one-hop and two-hop seeds, weak constraints include the previous time step information of the neighbours

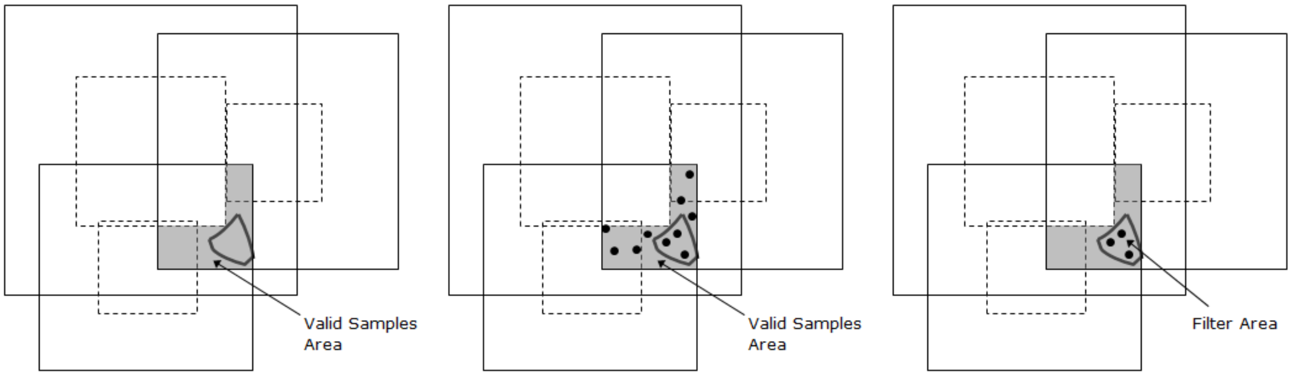

Fig. 7 Sampling and filtering from the modified region

that are used to construct the bounding box. A non-seed also uses its own previous estimated location and valid samples to construct the bounding box from where candidate samples will be drawn. We observe that when a non-seed uses the information of neighbours and its previous samples, then in some cases the bounding box does not contain its true location. This leads to the generation of more candidate samples and hence a reduction in sampling efficiency. Our heuristic is: whenever a non-seed finds a seed in its one-hop neighbourhood, the weak information from its neighbours and its previous samples are not considered for the bounding box construction. Our simulation results show that RMCB outperforms WMCL using this heuristic in terms of sampling efficiency without compromising the localization accuracy.

4.5 Filtering and weighting the samples

A candidate sample is filtered out if it does not satisfy one or more constraints. The remaining samples are the possible locations of a unlocalized non-seed as shown in the right of Figure 7.

In order to compute error in x and y direction, the smallest axis-aligned rectangle that encloses all the filtered samples: $(X_{\min}, X_{\max}, Y_{\min}, Y_{\max})$ is chosen. If location estimation is (X_e, Y_e), the error in the X-direction will be $\max(X_e - X_{\min}, X_{\max} - X_e)$ and the error in the Y-direction will be $\max(Y_e - Y_{\min}, Y_{\max} - Y_e)$ as shown in Figure 8.

The filtered samples are provided weights as similar to WMCL. Note that each localized non-seed broadcasts its location estimation and the set of filtered locations, and the error. An unlocalized non-seed node uses neighbour locations and error estimates, and the set of filtered samples of the node itself in the previous time step to compute weight for each of the samples. For each filtered sample s, a non-seed node n computes weight based on the received information from each one-hop non-seed node k whose error is less than the error of the node n in the previous time step. The sample s gets a partial weight $w(s,k)$ as the fraction of samples of k that are within distance $r + v_{\max}$ of s. Using the information from all the one-hop non-seeds, the final weight of the sample s is computed as $w(s) = \prod_k w(s,k)$.

4.6 Location computation

The final step of the algorithm is to compute a location q_x for an unlocalized node x and get an estimate on the positional uncertainty. Like many papers in the literature, we choose q_x to be the weighted average of the samples to be the estimate for node x's location. Following [35], we choose the error in the estimate of the location of x to be the maximum distance from q_x to any sample used to compute q_x.

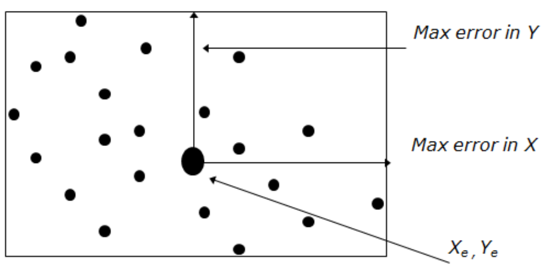

Fig. 8 Computing maximum error

RMCB (i)
// runs at each non-seed node i
if i has no neighbours
  then Cannot localize isolated node.
  else create outer bounding boxes B_j for each constraint j
    Create a bounding box B from B_j following WMCL
    Trim the box B using negative constraints
    find the seed neighbour m whose annulus has the minimum intersection area with B
    $B \leftarrow B \setminus B_m$
    Sample M points from B
    Filter the points using neighbourhood and non-neighbourhood constraints
    Using neighbour error information and previous location samples, compute weight of each sample
    $p_i \leftarrow$ weighted average of filtered points (using weights computed in the previous step)
    $e_i \leftarrow$ max distance from p_i to any filtered sample point
    Return location p_i and error estimate e_i

Fig. 9 Steps of algorithm RMCB

Figure 9 summarizes the steps of the computation at each node in our algorithm.

4.7 Reducing ranging error through calibration

It should be intuitively clear that the accuracy of any range-based algorithm depends on the accuracy of the distance estimates used by the algorithm. RMCB uses RSSI measurements for ranging, and accurate estimation of distances from RSSI is known to be difficult. Our experiments show that we can obtain usable range estimates only by *calibrating* our range computation function beforehand by estimating the standard path loss model parameters in Figure 5 empirically.

We have performed this calibration phase in a variety of indoor and outdoor settings. The distributions $D(), I()$ described in Section 4.2 change somewhat based on the scenario. For example, Figure 10 shows the RSSI data of the receiving packets as a function of distance in an open field. The path loss model exponent was about 1.7, i.e., very close to the value of free space loss. Here the best-fit line has parameters $n = -1.7, C = -50.11$.

Figure 11 shows RSSI versus distance in a typical computer lab. Due to lots of reflectors and obstacles, path loss exponent is relatively higher. The best fit line for path loss corresponds to $n = -2.0$, $C = -43.92$.

We noticed that the path loss exponent in the hallway is less than that in open field and computer lab. The reason behind this is that the hallway acts as a waveguide and results in the signal strength decaying less rapidly than in free space. However, it is also notable that the variance in RSSI is higher in the hallway, and therefore we get lower localization accuracy in the other environments that in the hallways because of more accurate distance estimation.

We also experimented in regions with obstacles, walls and moving traffic. Unfortunately the range estimation was unusable in these cases. For example, in the presence of walls and obstacles the signal loss often is so high that the resulting distance estimate is very high and produces distance constraints that are inconsistent with other constraints. Based on our experience, range-based localization can outperform range-free localization in environments that do not change after calibration and where there are few obstacles.

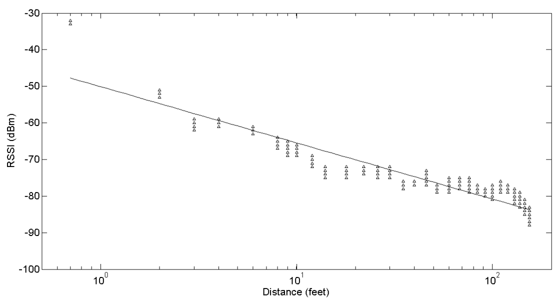

Fig. 10 RSSI as a function of distance, Open Field

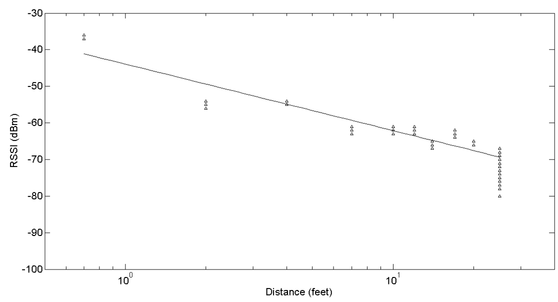

Fig. 11 RSSI as a function of distance, Computer Lab

5 Performance evaluation

We implemented RMCB in the Java-based simulator supplied to us by the authors of WMCL [35]. We looked for range-based localization algorithms for comparing the performance of RMCB . Unfortunately, none of the existing RSS-based algorithms were appropriate. [3] uses 3 fixed beacons; [4] does not perform localization in a global coordinate system; [30] localizes nodes using static beacons whose coordinates are precisely known. We measured the performance of the algorithm in [10] with range measurements made with our hardware; our experiments yielded very high localization error, and very low sampling efficiency. Therefore we chose WMCL [35] for comparison. Despite being a range-free algorithm, WMCL performs very well in terms of both localization error and sampling efficiency.

Unless otherwise stated, our simulations were done with a 1000 units $\times 1000$ units obstacle-free field. While many applications will not have an obstacle-free environment, this choice was made for simplicity. We set the radio range $r = 170$ units. This value is used in WMCL and to normalize velocity and localization error. The actual RSSI values in our simulations are generated using empirical data and the range computation is done by RMCB using the method described in Section 4.2. We note that our empirical model, while more realistic than simple models used in the literature, is still a coarse model of real radios because radio range is not circular and therefore connectivity changes with antenna orientation.

We used 40 to 200 sensor nodes. The number of seeds was varied between 4 and 30. All nodes (including seeds) were deployed uniformly randomly over the rectangular field. We allowed all nodes (including seeds) to be mobile. Like many previous papers (including [35]), we modelled node mobility using the modified random waypoint mobility model. In the random waypoint model, each node chooses a destination at random and moves in a straight line toward it using a velocity chosen uniformly at random from $[0, v_{\max}]$. Upon reaching the node, it pauses a fixed amount of time and then repeats the process. The modified random waypoint model eliminates the fixed pause.

We assumed a reliable MAC layer for our simulations, in keeping with [35] and most other papers in the literature. This is reasonable for at least for low data rate networks, because reliability can be implemented using retransmissions.

The parameter $v_{\max}$ was varied between 0 and 340. Following [35], the simulator runs for a warm-up period of 600 steps, and then takes measurements for 400 steps. This is repeated 30 times and the readings are averaged. We use the term *seed degree* to denote the average number of seeds a non-seed node identifies as its first neighbours. This is often (mistakenly) called seed density in the literature. In keeping with the literature, average localization error is expressed as a fraction of radio range r.

5.1 Localization error

We measured the variation of localization error with seed degree first. We used $v_{\max} = 10$ and the number of nodes $N = 60$. Figure 12 shows the error produced by RMCB and WMCL. RMCB consistently produces a localization error of roughly 5% (of radio range) less than that of WMCL. Figure 13 shows the effect of varying $v_{\max}$ on the localization error. The number of seeds was set at 10 and the other parameters were unchanged. For most applications, $v_{\max}$ should not exceed $0.5r$, but we show the results for much higher values as well. As seen in Figure 13, RMCB produces 8-10% lower localization error as compared to WMCL at low speeds.

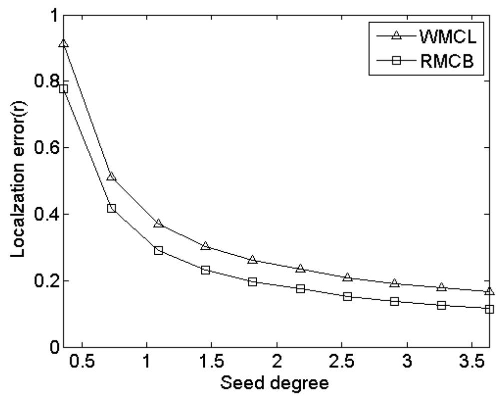

Fig. 12 Localization error vs seed degree

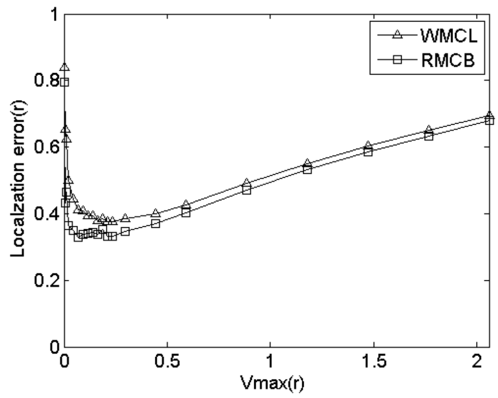

Fig. 13 Localization error vs speed

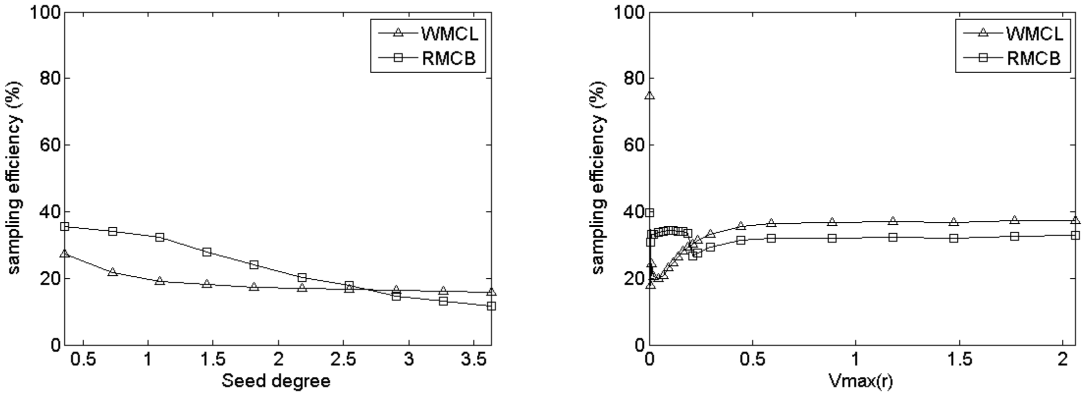

Fig. 14 (a) Sampling efficiency vs seed degree, and (b) Sampling efficiency vs speed

5.2 Sampling Efficiency

Since RMCB has extra constraints, its sampling efficiency could be expected to be lower than WMCL. Figure 14(a) and (b) show the variation of sampling efficiency with seed degree and speed (respectively). RMCB has better sampling efficiency than WMCL for low to medium seed degree and low speeds. At higher speeds and seed densities, the sampling efficiency of RMCB is higher than that of WMCL, but remains within 5% of that of WMCL.

5.3 Computational Overhead

We compared the computational load of RMCB with WMCL using the metrics used in [35] – viz., number of comparisons and number of distance computations. Figure 15 shows that RMCB uses less comparisons and distance computations than WMCL for low and medium seed degrees. At high seed degrees, the decrease in sampling efficiency seen in Figure 14(a) results in an increased number of comparisons. We infer that the heuristics we used to limit the sampling area in Section 4.4, and shown in Figure 7, need improvement for high seed degrees.

5.4 Communication cost

Nodes in RMCB broadcast location information similar to WMCL. Each seed broadcasts its location and each node broadcasts its location estimate and errors, as well as the location information it received from neighbours, so that each node knows of its two-hop neighbourhood. If we assume that the location estimate and maximum error of the estimate of a node are k bytes long, then the communication cost of each node is $O(S_d+1+k)$ where S_d is the seed degree. Although RMCB uses range information, unlike WMCL, the RSSI values do not need any extra communication. Thus the communication cost for these both algorithms are the same.

6 Hardware implementation and experiments

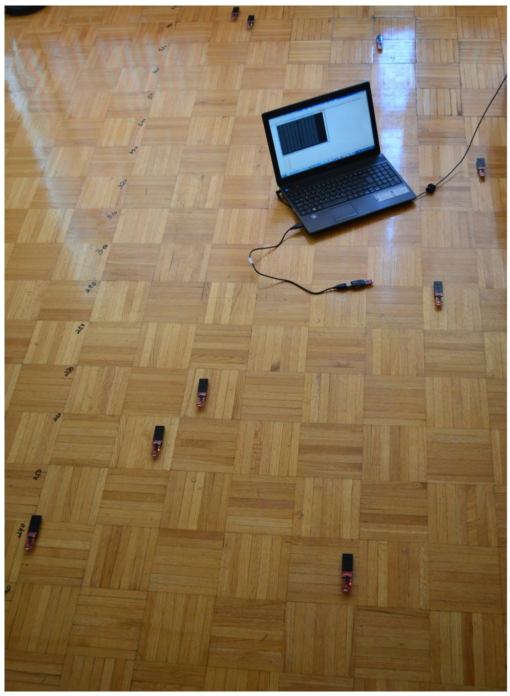

Fig. 16 EZ430-RF2500 Testbed

We implemented RMCB and WMCL on our sensor nodes. In the hardware implementation, we allow nodes to use only the information from the one-hop neighbours, although both RMCB and WMCL allow nodes the option to use location information from two-hop

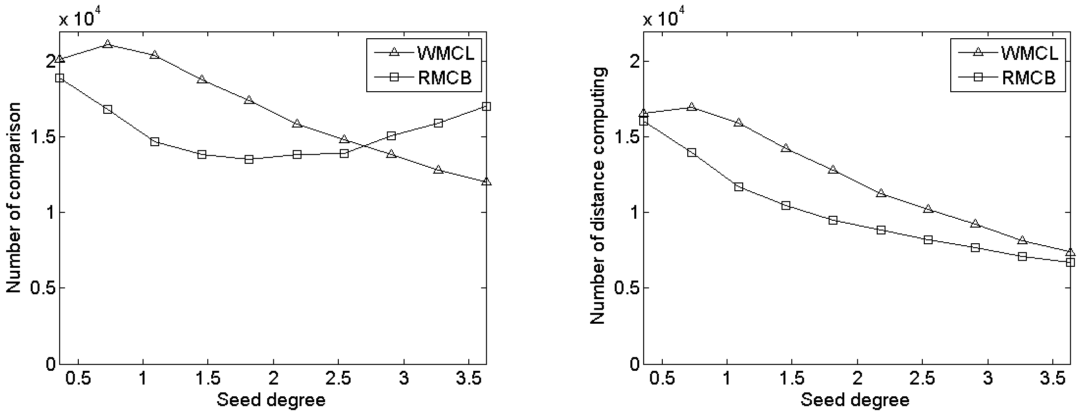

Fig. 15 (a) No. of comparisons vs seed degree, and (b) No. of distance computations vs seed degree

Table 1 Comparison of localization accuracy of WMCL, RMCB from hardware implementations

No. of nodes	No of non-seeds	Error (RMCB)	Error Variance (RMCB)	Error (WMCL)	Error Variance (WMCL)
number	number	number	number	number	number
15	3	9.6	2.0	38.3	2.2
12	5	36.6	33.5	45.3	24.2
10	5	25.9	20.0	36.8	14.1
8	5	19.1	13.1	44.5	22.6

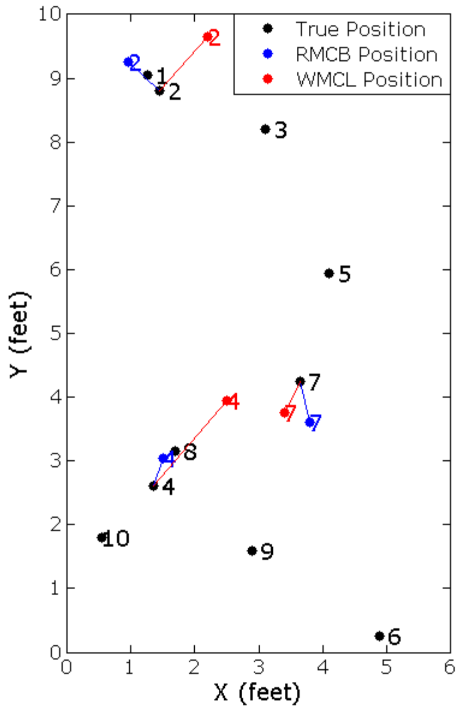

Fig. 17 True and estimated Locations in WMCL, RMCB

neighbours. Thus we do not need any routing infrastructure. A simple statically scheduled TDMA MAC protocol is also programmed in all the nodes. Specifically, each node is allowed to broadcast in the assigned TDMA slot, so that collision-free communication is guaranteed. We do not consider any mobility. We implemented a multi-hop environment using a network of 10 TI EZ430-RF2500 nodes placed in 5 feet × 10 feet rectangular obstacle-free area. Figure 16 shows our experimental testbed. We set the receiver sensitivity at -61 dBm, and that resulted in a radio range of about 1.5 feet. The hardware implementation demonstrates that the computational complexity of RMCB is low enough for it to be implemented on existing sensor hardware.

Figure 17 shows our experimental results. Black dots represent the true positions of the nodes. 3 non-seed nodes compute their locations using the information from the seeds and localized non-seed nodes. Blue dots show where the non-seed compute their locations in our testbed using RMCB. Similarly red dots are the WMCL output of locations of the non-seeds. It is seen that in two of the three cases, RMCB localized nodes more accurately than WMCL.

Table 1 shows the results of more systematic hardware experiments comparing the localization accuracy of WMCL and RMCB. For these experiments, nodes were static again and radio range was normalized to 100 units. The error shown is in these units. Thus an average error of 9.6 units indicates a 0.096r error. The table shows that the mean and variance of localization error are better for RMCB that WMCL. This implies that RMCB is not only better on average, but it also produces low localization errors more consistently than WMCL.

However it is worth pointing out that when the node density was increased beyond those shown in the table, RMCB sometimes produced inconsistent constraints. We hypothesize that he presence of many nodes in the physical vicinity of a node caused signal decay, perhaps due to interference and this negatively impacts range estimation as indicated in Section 2.2.

7 Conclusion

In this paper we propose a very general, fully distributed localization algorithm RMCB for WSNs that allows nodes to be static or mobile and that can work with nodes that can perform ranging as well as nodes that lack ranging capabilities. RMCB makes use of the received signal strength measurements that are available from the sensor hardware. We use RMCB to address the question: "When does range-based localization work better than range-free algorithms?" We demonstrate using simulations and hardware implementations that RMCB outperforms a very good range-free algorithm WMCL in terms of localization error in a number of scenarios. We describe the limitations of our range estimation approach and provide guidelines on when range-based localization is preferable. More research is needed to design accurate ranging algorithms in regions with lots of obstacles.

Acknowledgements

The authors gratefully acknowledge the financial support provided for this work by the Natural Sciences and Engineering Research Council of Canada.

References

1. Bachrach, J., Taylor, C.: Handbook of Sensor Networks, chap. Localization in Sensor Networks. Wiley (2005)
2. Baggio, A., Langendoen, K.: Monte carlo localization for mobile wireless sensor networks. Ad Hoc Networks **6**(5), 718–733 (2008)
3. Bahl, P., Padmanabhan, V.: RADAR: An in-building RF-based user location and tracking system. In: Proceedings of IEEE Infocom, pp. 775–784 (2000)
4. Bischoff, U., Strohbach, M., Hazas, M., Kortuem, G.: Constraint-based distance estimation in ad-hoc wireless sensor networks. In: Proceedings of the Third European Workshop on Wireless Sensor Networks (EWSN), pp. 54–68 (2006)
5. Chan, M., Estève, D., Escriba, C., Campo, E.: A review of smart homes-present state and future challenges. Comput. Methods Prog. Biomed. **91**(1), 55–81 (2008)
6. Chawla, M., Goel, N., Kalaichelvan, K., Nayak, A., Stojmenovic, I.: Beaconless position based routing with guaranteed delivery for wireless ad-hoc and sensor networks. In: Proceedings of 19th International Federation for Information Processing World Computer Congress, pp. 61–70 (2006)
7. Chen, M., Gonzalez, S., Vasilakos, A., Cao, H., Leung, V.C.: Body area networks: A survey. Mob. Netw. Appl. **16**(2), 171–193 (2011)
8. Clouser, T., Miyashita, M., , Nesterenko, M.: Fast geometric routing with concurrent face traversal. In: Proceedings of 12th International Conference on Principles of Distributed Systems (OPODIS08), pp. 346–362 (2008)
9. Cugola, G., Migliavacca, M.: A context and content-based routing protocol for mobile sensor networks. In: Proceedings of the 6th European Conference on Wireless Sensor Networks., pp. 69–85 (2009)
10. Dil, B., Dulman, S., Havinga, P.J.M.: Range-based localization in mobile sensor networks. In: Proccedings of the Third European Workshop on Wireless Sensor Networks (EWSN), pp. 164–179 (2006)
11. Friedman, R., Kliot, G.: Location services in wireless ad hoc and hybrid networks: A survey. Tech. Rep. CS-2006-10, Technion, Haifa, Israel (2006)
12. Haque, M., Matsumoto, N., Yoshida, N.: Context-aware cluster-based hierarchical protocol for wireless sensor networks. International Journal of Ad Hoc and Ubiquitous Computing pp. 379–386 (2009)
13. Hayes, G.R., Patterson, D.J., Singh, M., Gravem, D., Rich, J., Cooper, D.: Supporting the transition from hospital to home for premature infants using integrated mobile computing and sensor support. Personal and Ubiquitous Computing **15**(8), 871–885 (2011)
14. Hong, X., Nugent, C.D.: Segmenting sensor data for activity monitoring in smart environments. Personal and Ubiquitous Computing **17**(3), 545–559 (2013)
15. Hu, L., Evans, D.: Localization for mobile sensor networks. In: Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom), pp. 45–57 (2004)
16. Kalosha, H., Nayak, A., Rhrup, S., Stojmenovic, I.: Select-and-protest-based beaconless georouting with guaranteed delivery in wireless sensor networks. In: Proceedings of 27th Annual IEEE Conference on Computer Communications (INFOCOM), pp. 346–350 (2008)
17. Karp, B., Kung, H.: Greedy perimeter stateless routing for wireless networks. In: Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pp. 243–254 (2000)
18. Lawlor, M.: Small systems, big business. Signal Magazine (2005)
19. Lorincz, K., Welsh, M., Marcillo, O., Johnson, J., Ruiz, M., Lees, J.: Deploying a wireless sensor network on an active volcano. IEEE Internet Computing pp. 18–25 (2006)
20. Ma, Y., Richards, M., Ghanem, M., Guo, Y., Hassard, J.: Air pollution monitoring and mining based on sensor grid in London. Sensors **8**(6), 3601–3623 (2008)
21. MacLean, S., Datta, S.: Improving the accuracy of connectivity-based positioning for mobile sensor networks. In: Proceedings of the 22nd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (2011)
22. Mao, G., Fidan, B., Anderson, B.D.O.: Wireless sensor network localization techniques. Computer Networks **51**(10), 2529–2553 (2007)

23. Meyer, G., Främling, K.: Intelligent products: A survey. In: Computers in Industry, vol. 60, pp. 154–165 (2009)
24. Priyantha, N., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM), pp. 32–43 (2000)
25. Rappaport, T.S.: Wireless communications principles and practices. Prentice-Hall (2002)
26. Rudafshani, M., Datta, S.: Localization in wireless sensor networks. In: IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks, pp. 51–60 (2007)
27. Savvides, A., Han, C.C., Strivastava, M.B.: Dynamic fine-grained localization in ad-hoc networks of sensors. In: Proceedings of the 7th annual international conference on Mobile computing and networking (MobiCom), pp. 166–179 (2001)
28. Savvides, A., Srivastava, M., Girod, L., Estrin, D.: Wireless sensor networks, chap. Localization in sensor networks, pp. 327–349. Kluwer Academic Publishers, Norwell, MA, USA (2004)
29. Sayed, A.H., Tarighat, A., Khajehnouri, N.: Network based wireless location. In: IEEE Signal Processing Magazine, vol. 22, no. 4, pp. 24–40 (2005)
30. Sugano, M., Kawazoe, T., Ohta, Y., Murata, M.: Indoor localization system using RSSI measurement of wireless sensor network based on zigbee standard. In: Proceedings of the IASTED International Conference on Wireless Sensor Networks (WSN 2006) Wireless Sensor Networks (2006)
31. Varkey, J.P., Pompili, D., Walls, T.A.: Human motion recognition using a wireless sensor-based wearable system. Personal and Ubiquitous Computing **16**(7), 897–910 (2012)
32. Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., Corke, P.: Data collection, storage, and retrieval with an underwater sensor network. In: Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys), pp. 154–165. ACM (2005)
33. Ward, A., Jones, A., Hopper, A.: A new location technique for the active office. IEEE Personal Communications **4**(5), 42–47 (1997)
34. Xiao, Y., Sethi, S., Chen, H.H., Sun, B.: Security services and enhancements in the ieee 802.15.4 wireless sensor networks. In: Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), pp. 1–5 (2005)
35. Zhang, S., Cao, J., Chen, L., Chen, D.: Accurate and energy-efficient range-free localization for mobile sensor networks. IEEE Transactions on Mobile Computing **9**, 897–910 (2010)