# Distributed Localization Using a Moving Beacon in Wireless Sensor Networks

Bin Xiao, *Member*, *IEEE*, Hekang Chen, *Member*, *IEEE*, and Shuigeng Zhou, *Member*, *IEEE*

**Abstract**—The localization of sensor nodes is a fundamental problem in sensor networks and can be implemented using powerful and expensive beacons. Beacons, the fewer the better, can acquire their position knowledge either from GPS devices or by virtue of being manually placed. In this paper, we propose a distributed method to localization of sensor nodes using a single moving beacon, where sensor nodes compute their position estimate based on the range-free technique. Two parameters are critical to the location accuracy of sensor nodes: the radio transmission range of the beacon and how often the beacon broadcasts its position. Theoretical analysis shows that these two parameters determine the upper bound of the estimation error when the traverse route of the beacon is a straight line. We extend the position estimate when the traverse route of the beacon is randomly chosen in a real-world situation, where the radio irregularity might cause a node to miss some crucial coordinate information from the beacon. We further point out that the movement pattern of the beacon plays a pivotal role in the localization task for sensors. To minimize estimation errors, sensor nodes can carry out a variety of algorithms in accordance with the movement of the beacon. Simulation results compare variants of the distributed method in a variety of testing environments. Real experiments show that the proposed method is feasible and can estimate the location of sensor nodes accurately, given a single moving beacon.

**Index Terms**—Distributed localization, range-free, moving beacon, radio irregularity.

## 1 Introduction

Wireless sensor networks (WSNs) are the current focus of research interest because of their broad applicability in areas such as environmental observation, military surveillance, building monitoring, and disaster relief. The performance of WSNs is crucially influenced by how accurately sensor nodes within the network are localized. Sensor localization information is used in the self organization and configuration of networks in deciding where events take place, tracking moving targets [5], [6], [19], assisting traffic routing [4], [12], and providing the network geographic coverage [1]. The techniques used to identify the position of each sensor node are central to such location-aware operations.

Constraints of cost and power consumption make it infeasible to equip each node in a network with a Global Positioning System (GPS). It is possible, however, to equip a small number of sensors. Such sensor nodes, referred to in this paper as *beacons*, may be static or mobile and may be used to identify the position of other nodes in a sensor network. This is done with either the *range-based* or *range-free* techniques. The range-based technique [2], [7], [8], [9], [15], [21] detects the position of a node by using either its signal propagation time or its signal strength degradation to estimate the distance or relative angle between two nodes. This technique has two drawbacks. First, the node that is being sought must contain expensive hardware. Second, the signal capture can be severely affected by the environmental interference, greatly degrading detection accuracy. These drawbacks make the range-based technique impractical for use in localization in a large-scale sensor network. In the range-free technique [13], [14], [24], each sensor node uses signals from a few beacons to calculate its approximate location. To directly receive signals, a node must fall inside the overlapping transmission area of several beacons. Nonetheless, the information that a node is outside the transmission area of a beacon can also be used to set a tighter bound on its location.

Where beacon nodes are powerful but expensive, as in a heterogeneous sensor network, it is desirable to minimize the number of beacons to be used to estimate the position of each sensor node. If a beacon can move and periodically broadcast its position to nodes in its vicinity, then a single beacon will suffice. Such a position-aware beacon can acquire its geographical position through GPS, or the position can be known, because it moves along a predefined route. Using a moving beacon that knows its position is broadly equivalent to using many stationary beacons, each broadcasting once. A sensor node can compute an area to confine its location if it receives the coordinate messages from the beacon in a couple of times. Once a sensor node has approximately determined its position, it can help localize its neighbor nodes or even distant nodes. Ultimately, the location estimate of each sensor node is determined by the moving beacon. Thus, the degree of accuracy with which a sensor node can be located is determined by the knowledge of the route that the beacon travels and how often it emits a signal.

In this paper, we propose a distributed method to the localization of sensor nodes using a moving beacon. The method allows a sensor node to locally compute its position

- *B. Xiao is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. E-mail: csbxiao@comp.polyu.edu.hk.*
- *H. Chen and S. Zhou are with the Department of Computer Science and Engineering, Fudan University, Shanghai 200433, P.R. China. E-mail: {hkchen, sgzhou}@fudan.edu.cn.*

*Manuscript received 20 July 2006; revised 18 May 2007; accepted 23 Aug. 2007; published online 11 Sept. 2007.*
*Recommended for acceptance by K. Nakano.*
*For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0198-0706.*
*Digital Object Identifier no. 10.1109/TPDS.2007.70773.*

estimate by using the arrival and departure information of the beacon. For present purposes, we conceive the beacon as having three movement patterns: the Sparse-Straight-Line (SSL) movement pattern, the Dense-Straight-Line (DSL) movement pattern, and the random movement pattern. SSL would travel across a proportionally smaller number of squares in a given grid, always turning at right angles. DSL would travel across a proportionally larger number of squares but would also always turn at right angles. The random movement pattern would travel across an indeterminate number of squares within a grid and may travel in any trajectory. For the straight-line movement patterns (that is, SSL and DSL), we provide an upper bound of the position estimation error of a sensor node. The upper bound is determined by two factors: the radio transmission range of the beacon and the regular interval distance that the beacon has traveled between two continuous broadcasts of its coordinate information. Based on the theoretical analysis given in this paper, we conclude that the position estimate of a sensor node is more accurate when the node is covered and distant from the line of movement of the beacon. In addition, to improve the sensor nodes' estimate accuracy and to reduce the computational overhead, we present different algorithms that the nodes can apply according to the movement patterns of the beacon.

This paper also investigates the impact on the location estimate accuracy of a circumstance, which can arise in a real sensor network scenario, specifically the circumstance in which a sensor node misses messages from the beacon due to radio irregularity, especially when the beacon displays the random movement pattern. We propose that in such cases, the sensor nodes can minimize estimation errors by applying the practical localization method taking care of message missing. We further show the simulation results of the proposed distributed method in a variety of testing environments, where radio irregularity exists. The results demonstrate that both the distance and route that a beacon travels directly affect the localization accuracy. The localization accuracy, when the random pattern is applied, can be improved by increasing the distance that the beacon travels. The DSL movement pattern is the best movement pattern among the three patterns, for it can not only localize all the nodes but is also accurate under any network topology. Finally, we implemented the proposed method in a real outdoor environment, where dozens of sensor nodes were randomly distributed in a flat ground, and a beacon moved in the SSL, DSL, or random movement patterns. Although the testing environment displays its impact on the localization accuracy, the estimate errors are small in the experiments. The computed localization errors from the experiments are larger than the simulation values, because the real outdoor environment could make the transmitted wireless signal even more unstable. Similar to the simulation results, the DSL movement pattern achieved the best location estimate of sensor nodes.

This paper is organized as follows: In Section 2, we describe the related work in the area of sensor node localization. Section 3 presents the estimation error analysis and its upper bound when a node locally localizes its position under the SSL or DSL movement patterns of a beacon. Section 4 provides algorithms that each node can use to compute its position within an area when the beacon is moving according to the SSL, DSL, or random pattern. Section 5 proposes a practical localization method under the radio irregularity model (RIM). Section 6 illustrates the simulation results of algorithms in both the ideal and real situations with radio irregularity. Section 7 shows the location estimate performance in the real outdoor experiments. In Section 8, we discuss some design issues and future work. Finally, we offer our conclusions in Section 9.

## 2 Related Work

Most localization methods are static. That is, both the sensor network and beacons are static. The static localization methods are either *range-based* or *range-free.* Range-based localization first uses AOA, RSSI, TOA, or TDOA [2], [7], [8], [9], [15], [21] to measure the distance or angle between an unknown node and a beacon and then uses trilateration, triangulation, or Maximum Likelihood to estimate the position of the unknown node. These methods have two significant drawbacks: they require costly additional hardware support, and fading and noise can cause distance measurement errors [20]. Range-free localization methods are meant to overcome these drawbacks, as they do not require the hardware support for measuring distances or angles. Instead, they exploit the communication and sensing ranges. He et al. [13] proposed an area-based APIT algorithm that, if nodes are in the triangular regions constructed by beacons, refines the area where nodes might be located. The satisfactory performance of this algorithm, however, requires a high ratio of beacons to nodes and a dense node connectivity.

Mobile localization [3], [16], [17], [18], [24] using moving beacons avoids the problems of static localization methods. Galstyan et al. [3] proposed a distributed online algorithm using a moving beacon to localize static sensor networks. This was extended to a general model using an unknown target, in which radio communication and sensing constraints minimize the area in which a node might be located. Hu and Evans [17] proposed the *Monte Carlo Localization* for sensor networks, in which both nodes and beacons can be mobile. The position of a node was iteratively refined using in-relationship or out-relationship information among nodes and beacons as a filter. Neither method, however, sought to account for how the movement of the beacon might influence the accurate estimate of the positions of sensor nodes. In fact, the movement pattern of the beacon directly links to the accuracy and accomplishment of the localization task.

In a real environment, the radio irregularity shows its impact on the location estimate accuracy of sensor nodes. How the radio irregularity can precisely be measured is defined by the Received Signal Strength (RSS) calculation model. The model [11] calculating RSS reflects the path loss by using parameters such as a referenced distance, path loss exponent, and a random variation. Recent work [10], [22], [23] proposed more accurate RSS models. In [10], Zhou et al. proposed the RIM. In this model, path loss is treated to be nonisotropic, and sending power should follow a normal distribution. Li and Martin [22] introduced a ray-sector model for indoor 802.11 networks, in which the RSS is calculated by adding signal bias associated with different sectors. Our proposed method differs significantly from the previous work in that it uses a single powerful moving

beacon, is robust and scalable, and can be applied to a real connected sensor network of any topology, because it incorporates the radio loss and irregularity model.

## 3 Distributed Localization Using a Moving Beacon

This section presents a *range-free* distributed localization method that uses a mobile beacon to estimate the positions of sensor nodes. The method is based on the weak assumption that if a node can receive a signal from the beacon, their distance is within the radio transmission range $r$ of the beacon. In this section, we first study how a node uses the broadcasting signal message from the beacon to confine its estimation area when the beacon is moving in a straight line. Then, we will show the upper bound of estimation error to the real position of a node.

### 3.1 Arrival and Departure of Beacon

Assume that a beacon moves in a straight line in a sensor network area, and at a certain distance interval, called the *broadcasting interval* $s$, it broadcasts a message containing its current position. The *position* of the beacon denotes the physical point at which it broadcasts a message. We also assume that a node can receive messages from the beacon only if it is inside the transmission range of the beacon. We define two states for every node:

1. *In.* The node is within the broadcasting range of the beacon.
2. *Out.* The node is out of the broadcasting range of the beacon.

We also define two dynamic transitions from these two states:

1. *Arrival.* A node receives the current scheduled signal of beacon but did not receive the scheduled signal from the beacon's previous position. The status of the node is reset from *out* to *in.* The beacon is now at the *arrival position.* Its previous position is called the *prearrival position*.
2. *Departure.* A node received the preceding scheduled signal from the beacon but does not receive the scheduled signal from the current position of the beacon. The status of the node is reset from *in* to *out.* The previous position of the beacon is called the *departure position*, and the current position of the beacon is called the *postdeparture position*.

These two transitions are useful for estimating the position of a sensor node. The fact that transitions always occur in a pair limit the possible position of a node into a small area, as shown in Fig. 1. Suppose that a beacon is moving from left to right along the $x$-axis. Initially, the state of node $G$ is set to be *out.* When the beacon arrives at position $B$, node $G$ hears the beacon for the first time, and its state resets to *in.* Node $G$ can calculate the prearrival position $A$ according to the arrival position $B$. Node $G$ must fall inside the area in the circle centered at $B$ with radius $r$ and out of the circle centered at $A$, forming an *arrival constraint area* as in the left-hand crescent shape in Fig. 1. When the beacon moves forward, it will walk out of the range to contact node $G$. Let $C$ be the departure position and $D$ be the postdeparture position. The state of node $G$ is reset to *out*, and it must fall inside the area in the

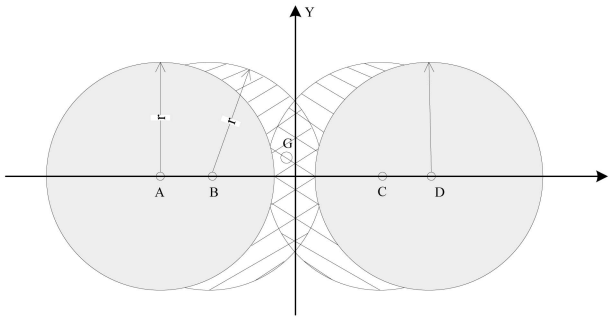


Fig. 1. Localization using the arrival and leaving information of the beacon.

circle centered at $C$ with radius $r$ and out of the circle centered at $D$, forming a *departure constraint area* as the right-hand crescent shape in Fig. 1. The *arrival constraint area* and *departure constraint area* will create an overlap called the *Arrival and Departure Overlap* (*ADO*). Thus, node $G$ must be in *ADO.* The intermediate broadcasting positions between the arrival and departure positions do not provide valuable information, because they cannot help narrow down the area of *ADO*.

To estimate its position, a node should obtain four critical positions of the moving beacon to compute its *ADO*:

1. prearrival position,
2. arrival position,
3. departure position, and
4. postdeparture position.

It is obvious that the *ADO* is symmetrical along the $x$-axis. The half *ADO* (*HADO*) above the $x$-axis is called the *upper HADO*, whereas the other *HADO* is called the *lower HADO.* Assume that we can determine whether a detected node is above or below the line of movement of the beacon (we show the way in Section 4.1). Without loss of generality, we analyze the *upper HADO* to show the location estimate.

### 3.2 Half Arrival and Departure Overlap Analysis

In this section, we provide the error upper bound of the position estimate of a node $G$ when the transmission range is $r$ and the broadcasting interval is $s$. Let the coordinates of node $G$ be $(x_G, y_G)$. Without loss of generality, we assume that the beacon starts moving along the $x$-axis from the coordinates (0, 0), and the node $G$ receives the first signal from the beacon when the beacon is at $(s, 0)$ and the last signal when the beacon is at $(ks, 0)$, where $k$ is the number of signals that the node hears between the arrival position and the departure position. The *arrival constraint area* of $G$ can be presented as

$$x^2 + y^2 > r^2, \tag{1}$$

$$(x - s)^2 + y^2 \leq r^2, \tag{2}$$

and the *departure constraint area* of $G$ can be presented as

$$(x - k \cdot s)^2 + y^2 \leq r^2, \tag{3}$$

$$(x - (k + 1) \cdot s)^2 + y^2 > r^2, \tag{4}$$

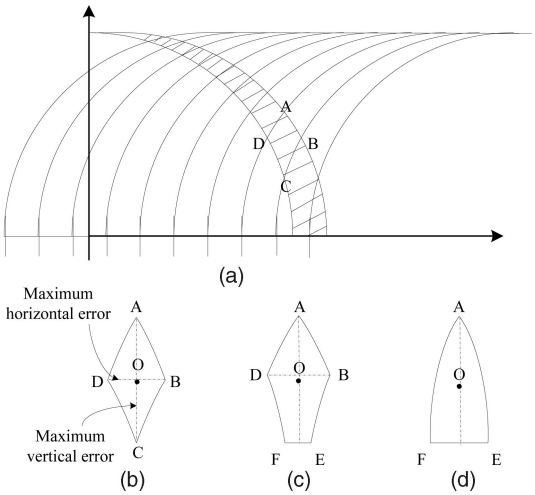


Fig. 2. (a) Formation of different ***HADOs***. (b) Four-point ***HADO***. (c) Five-point ***HADO***. (d) Three-point ***HADO***.

where $1 \leq k \leq \lfloor 2r/s \rfloor + 1$, guaranteeing that the *arrival constraint area* and the *departure constraint area* overlap. Let the position estimate of node $G$ be at point $O\ (x_O, y_O)$. As a result, the maximum horizontal error $MAX\{|x_O - x_G|\}$ and maximum vertical error $MAX\{|y_O - y_G|\}$ depict the upper bounds of estimation errors in the $x$-coordinate and $y$-coordinate, respectively.

The *HADO* area can shrink to a point or be shown as four-point *HADO*, five-point *HADO*, and three-point *HADO* in Figs. 2b, 2c, and 2d, respectively. The shape of the *HADO* is related to the value of $k$, which indicates the total number of times that node $G$ receives signals from the beacon. The extreme case in which the *HADO* shrinks to a point happens at most once when $ks = 2r + s$. In this extreme case, the coordinates of the point $(\frac{k+1}{2}s, \sqrt{r^2 - (\frac{k-1}{2}s)^2})$, which is $(r + s, 0)$, denote the real position of node $G$. The five-point *HADO* could exist when $2r - s < ks < 2r$, and the three-point *HADO* exists once when $2r \leq ks < 2r + s$. The four-point *HADO* reflects the most common situations when $s \leq ks \leq 2r - s$.

In the four-point *HADO*, the coordinates of four intersected points ($A$, $B$, $C$, and $D$) of circles calculated according to (1), (2), (3), and (4) are

$$A: \left(\frac{k+1}{2}s, \sqrt{r^2 - \left(\frac{k-1}{2}s\right)^2}\right),\ B: \left(\frac{k+2}{2}s, \sqrt{r^2 - \left(\frac{ks}{2}\right)^2}\right),$$
$$C: \left(\frac{k+1}{2}s, \sqrt{r^2 - \left(\frac{k+1}{2}s\right)^2}\right),\ D: \left(\frac{k}{2}s, \sqrt{r^2 - \left(\frac{ks}{2}\right)^2}\right).$$

If $k$ is small, the $y$-coordinate of these points are large, which means that the *HADO* is far from the $x$-axis. Point $O$ is defined as the midpoint of line $(A, C)$. Thus, the maximum vertical error is $|y_A - y_O|$ (which is the same as $|y_C - y_O|$), and the maximum horizontal error is $|x_B - x_O|$ (which is the same as $|x_D - x_O|$). Therefore, given different $k$, the maximum horizontal error remains the same (which is equal to $s/2$), and the maximum vertical error is $\frac{\sqrt{r^2 - (\frac{k-1}{2}s)^2} - \sqrt{r^2 - (\frac{k+1}{2}s)^2}}{2}$.

In the five-point *HADO*, we denote the intersection points as $A$, $B$, and $D$ (the intersected points among circles) and $E$ and $F$ (the intersected with the $x$-axis). The coordinates of $E$ and $F$ are $((k+1)s - r, 0)$ and $(r, 0)$. The coordinates of $O$ is defined as $(x_A, y_A/2)$. Thus, the maximum horizontal error is $s/2$, and the maximum vertical error is $\frac{\sqrt{r^2 - (\frac{k-1}{2}s)^2}}{2}$.

In the three-point *HADO*, we denote the intersection points as $A$, $E$, and $F$. The coordinates of $E$ and $F$ are $(r + s, 0)$ and $(ks - r, 0)$. The coordinates of $O$ is defined as $(x_A, y_A/2)$. The maximum horizontal error is $\frac{|EF|}{2}$, which is smaller than $s$ and is equal to $\frac{2r-(k-1)s}{2}$, and the maximum vertical error is the same as in the five-point *HADO* $\frac{\sqrt{r^2 - (\frac{k-1}{2}s)^2}}{2}$.

Based on this analysis of three *HADOs*, we conclude that the maximum horizontal error using node $O$ as the position estimate of node $G$ under all situations is

$$\begin{cases} s/2, & \text{if } s \leq ks \leq 2r, \\ \frac{2r-(k-1)s}{2}, & \text{if } 2r < ks \leq 2r + s, \end{cases} \tag{5}$$

and the maximum vertical error is

$$\begin{cases} f(k-1) - f(k+1), & \text{if } s \leq ks \leq 2r - s, \\ f(k-1), & \text{if } 2r - s < ks \leq 2r + s, \end{cases} \tag{6}$$

where $f(k) = \frac{\sqrt{r^2 - (\frac{k}{2}s)^2}}{2}$. From (5), it is obvious that the *upper bound* of the $x$-coordinate estimation error of node $G$ is equal to $s/2$. From (6), we know that the *upper bound* of the $y$-coordinate estimation error of node $G$ is equal to $\frac{\sqrt{2rs - s^2}}{2}$ when $k = \frac{2r}{s} - 1$ (in a real movement of a beacon, $k$ should be an integer, and this upper bound still holds). The upper bound of the position estimation error of node $G$ decreases when $s$ decreases, which implies that to obtain an accurate localization of sensor nodes, the moving beacon should shorten its message broadcasting interval.

## 4 Movement Patterns of the Beacon

The route that a moving beacon traverses seriously impacts the location accuracy of sensor nodes. In this section, we will study three different movement patterns of the beacon: the SSL, DSL, and random movement patterns. Before the beacon starts moving, it first informs all nodes of its movement pattern and broadcasting interval $s$ by flooding. In the two straight-line movement patterns, we virtually divide the whole area into grids.[1] Each square in the grid (cell) has an edge length of $r$, which is the radio transmission range of the beacon. The beacon travels along predefined straight lines and turns at right angles. In the random movement pattern, however, the beacon will, after a certain distance, randomly change its direction. To cope with these patterns, a node must use a different localization algorithm for each pattern. These distributed localization algorithms allow each node to estimate its position locally after receiving coordination messages from the beacon.

1. We assume that the sensor distribution area is known to the beacon.

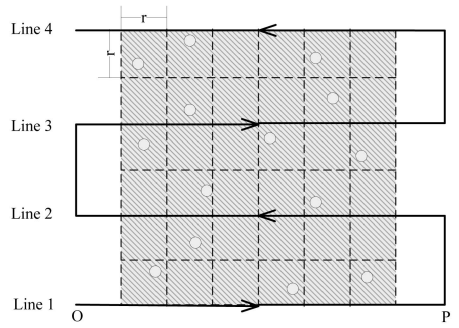


Fig. 3. Traverse route of the beacon in the SSL movement pattern.

### 4.1 Sparse-Straight-Line Movement Pattern

In the SSL movement pattern, a beacon will traverse a grid, as shown in Fig. 3. Each horizontal line that the beacon traverses is increasingly numbered in series from the bottom line up. The beacon starts at the bottom left point $O$ and traverses the grid until the whole area is covered. The line distance of movement space is $2r$. In order to guarantee that each node can obtain its *HADO* estimation area, the start and end points of each horizontal line are $r$ length wider to the boundary. For example, on line 1, the start point is $O$, and the end point is $P$. Using SSL, the beacon can "almost" cover the whole network. Only a small space, as shown in Fig. 6a, will not be covered by two successive circles.

If a sensor node receives signals from the beacon, its position can be constrained into two *HADOs* (either in the *upper HADO* or in the *lower HADO*), which is symmetric along the movement line of the beacon. When a node can decide its position above (below) a line, its real position should reside in the *upper (lower) HADO.* In order to improve the position estimate accuracy, we need to decide within which *HADO* the node resides. We denote that a node receives a signal from a line as the node receives a signal from the beacon moving along the line in this paper. Clearly, when a node receives a signal from the bottom line, a node can confine its location as being within the upper *HADO*, and when it receives a signal from the top line, it can confine its position as being in the lower *HADO.* However, if a node receives a signal from lines between the top and bottom, a node cannot classify itself in the upper or lower *HADO.* It is possible, however, to clearly identify the position of a node by using valuable location information obtained from neighbors whose positions are known.

To assist a node in estimating its position when it receives a signal from the $i$th line, we propose the following rules. Let $A$ be the node that must locate its position either above or below the $i$th line and let $B$ and $C$ be the nodes with determined positions. Suppose that node $A$ can contact node $B$ but not $C$. We denote the known *HADOs* of $B$ and $C$ as *HADO(B)* and *HADO(C)*, respectively. Let the possible *HADO* area of node $A$ above the $i$th line be $HADO_{upper}(A)$ and below as $HADO_{lower}(A)$. That an area can contact another area means that any line connecting them has a length shorter than $r$. If we cannot find such a single line, those two areas cannot contact each other:

*Rule 1.* If $B$ is in the row immediately above the $(i-1)$th line, the position of $A$ is below the $i$th line. If $B$ is in the row immediately below the $(i+1)$th line, the position of $A$ is above the $i$th line.

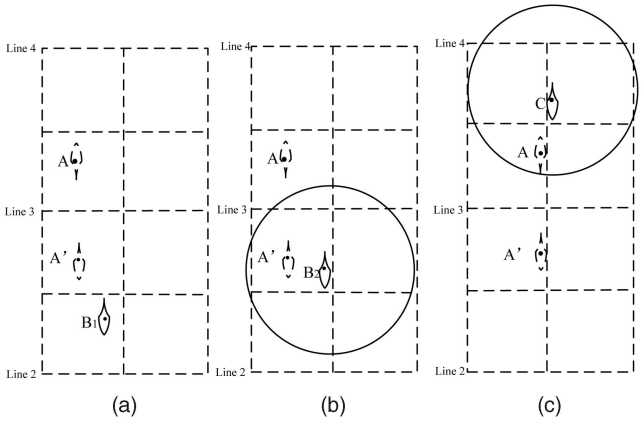


Fig. 4. Determination of a node position in a grid either above or below the moving line of beacon.

*Rule 2.* If *HADO(B)* cannot contact $HADO_{upper}(A)$, the position of $A$ is below the $i$th line. If *HADO(B)* cannot contact $HADO_{lower}(A)$, the position of $A$ is above the $i$th line.

*Rule 3.* If *HADO(C)* can contact $HADO_{upper}(A)$, the position of $A$ is below the $i$th line. If *HADO(C)* can contact $HADO_{lower}(A)$, the position of $A$ is above the $i$th line.

Fig. 4 presents examples to show how these three rules work. Node $B_1$ is immediately above the second line, whereas node $B_2$ is immediately below the third line. Suppose that node $A$ needs to determine whether it is above (shown as $A$) or below (shown as $A'$) the third line. Fig. 4a shows that $A$ has a localized neighbor $B_1$ in the row immediately above the second line. If so, according to Rule 1, $A$ is below the third line. The example in Fig. 4b shows a situation in which a localized neighbor $B_2$ or $HADO(B_2)$ cannot contact $HADO_{upper}(A)$. In this situation, according to Rule 2, $A$ is below the third line. If $A$ cannot find any neighbors that satisfy Rules 1 or 2, it must seek help from other nodes. Fig. 4c provides an example featuring node $C$, in which *HADO(C)* can contact $HADO_{upper}(A)$. In accordance to Rule 3, $A$ is below the third line.

Now, we study the localization algorithm for nodes that capture signals from the beacon in the SSL movement pattern. We denote nodes of the $i$th line as the nodes in the row either immediately above or immediately below the $i$th line. The localization process consists of two phases: individual localization and cooperative localization. In the first phase, as the beacon travels according to the SSL movement pattern, each node who receives a message from the beacon determines at which line it is located and calculates two possible *HADOs* symmetric along the line. Note that the nodes of the first line can determine its position, as it is definitely above the first line. The second cooperative phase allows each node to determine either the $HADO_{upper}$ or $HADO_{lower}$ in which it resides by using the localized neighbor information. The second phase proceeds as follows: Initially, any node in the row immediately above the first line can determine its position, as it is definitely above the first line. Then, it floods its *HADO* to nodes of the second line. If an unknown node of the second line can determine its estimate area by using any one of the above

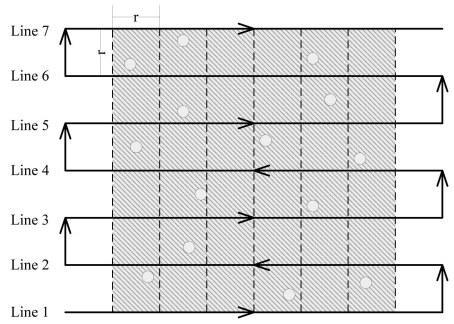


Fig. 5. Traverse route of the beacon in the DSL movement pattern.

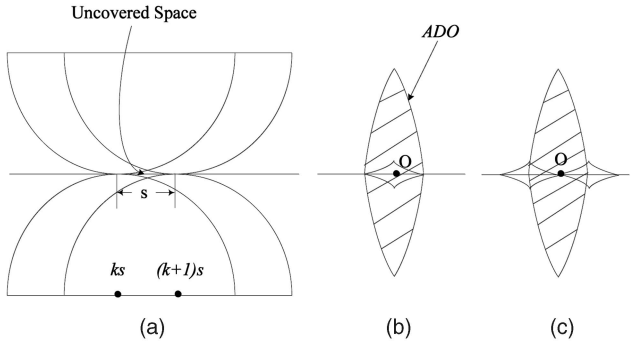


Fig. 6. Uncovered space when the beacon moves along upper or lower straight lines.

three Rules from received messages, it propagates its location to other lines. This localization wave continues till each node finds its location. According to above three Rules, the localized nodes of the $i$th line should flood a message containing its geographic *HADO* to nodes of the $(i-1)$th, $i$th, and $(i+1)$th lines. More specifically, if we divide the whole area into grids with length of transmission range $r$, a localized node only needs to flood geographic information to the nodes in its own and neighboring girds. As a result, we set a small Time-to-Live (TTL) value for the flooding messages to reduce the communication overhead. The flooding message generated from a single localized node will be broadcast to all nodes who are within a distance of $TTL \cdot r$. Obviously, this communication overhead is related to the network topology and the TTL value, as we will show it in the simulation. The localization algorithm for the SSL movement pattern is shown in Algorithm 1:

**Algorithm 1**: algorithm for the SSL movement pattern.
Input: $node_i$, *ADO*, *TTL*
Output: the position estimate $p_i$ of $node_i$

```
1: if node_i is of the first line then
2:    flood the upper HADO;
3:    compute p_i by using the upper HADO;
4:    output p_i;
5: else
6:   for each received HADO do
7:     if Rule 1 or Rule 2 or Rule 3 is satisfied then
8:       get the correct HADO;
9:       flood this HADO;
10:      compute p_i by using this HADO;
11:      output p_i;
12:    end if
13:  end for
14: end if
```

### 4.2 Dense-Straight-Line Movement Pattern

The beacon following the DSL movement pattern can cover and localize every sensor node. The dense movement pattern allows a more accurate position estimate of a node than does the sparse movement pattern. The dense straight movement is shown in Fig. 5, in which the distance between two adjacent lines is $r$.

In this pattern, a node can detect its position once or twice. The scenario in which a node can detect its position only once happens when the node stays in the uncovered space, as shown in Fig. 6a. The node does not receive any signal from lines above or below. However, it can obtain one *ADO* when the beacon moves along the line across the uncovered space. The position of the *ADO* can help identify the uncovered space in which the node resides. Note that the uncovered space is symmetrical along the line of movement, and we do not know whether the node is above or below the line. Because the uncovered space is relatively small, this space itself is a good position estimate, and it is not necessary to distinguish whether it is above or below. As analyzed in Section 3.2, the $x$-coordinate of the center point of the *ADO* is $\frac{k+1}{2}s$. Thus, there are two possible intersection situations of uncovered spaces with a verified *ADO*. The first situation is shown in Fig. 6b, where the *ADO* intersects with only one uncovered space when $k+1$ is an odd number. We choose the center node $O$ in the uncovered space as the estimate position. The second situation is shown in Fig. 6c, where the *ADO* intersects with two uncovered spaces when $k+1$ is an even number. To minimize the estimation error, we choose the connection point $O$ of two uncovered spaces as the estimate position. No matter what the situation is, the coordinates of node $O$ are $(\frac{k+1}{2}s, y_{line})$, where $y_{line}$ is the $y$-coordinate of the movement line.

The scenario in which a node detects its position twice happens for most of the nodes if the node stays outside the small uncovered area. Each detection creates an estimate area. This estimate area can be easily distinguished as the *upper* or *lower HADO*. A node hearing a signal from the $i$th line is below the line when it has already received a signal from the $(i-1)$th line. Otherwise, the node is above the $i$th line. Thus, a node in this scenario can be delimited into a more accurate area, which is the overlap of two *HADOs*. These two *HADOs* provide a complementary estimate of a node to narrow down the possible area, because if a node is close to one movement line of the beacon, it is far from the other line. Note that the farther a node is from the line of movement, the smaller the *HADO* area will be. In order to simplify the position estimate for nodes who obtain two *HADOs*, we first calculate the central point of each *HADO* separately and then set their midpoint as the estimate position. The localization algorithm for each sensor node in the DSL movement pattern is shown in Algorithm 2:

**Algorithm 2**: algorithm for the DSL movement pattern.
Input: $node_i$, *ADOs*
Output: the position estimate $p_i$ of $node_i$

```
1: if node_i has two ADOs then
2:    find two overlapping HADOs;
```

3: compute the central point of each overlapping *HADO*;
4: set $p_i$ to be the midpoint of the central points;
5: output $p_i$;
6: **else**
7: set $p_i$ to be $((k+1)s/2, y_{line})$;
8: output $p_i$;
9: **end if**

### 4.3 Random Movement Pattern

In this pattern, the beacon would travel across an indeterminate number of grids in a sensor network area and may travel in any trajectory. If it moves to the boundary of the network area, it bounces back. This means that it no longer necessarily travels in a straight line. The beacon should broadcast its previous position, current position, and next position (supposing that it has decided where to move), where it emits signals, because a node can no longer calculate the prearrival and postdeparture positions. A sensor node can compute an *ADO* area when it detects the prearrival, arrival, departure, and postdeparture positions of the beacon. These four positions could no longer be in a straight line, so the *ADO* area could not consist of two symmetrical parts. In order to localize the position of a node, we expect that at least two *ADOs* must be obtained.

That the beacon changes its direction randomly leads to the coverage density varied in different areas. If the distance of a route that a beacon has traveled across is short, there may exist a small portion of nodes incapable of being localized.

A sensor node $G$ can estimate its position from $k$ (at least two) calculated *ADOs*. Because the beacon moves randomly, the placements of *ADOs* are irregular. The overlap of *ADOs* creates a single *kernel overlap area* (KOA) in most cases, and the node should stay inside it. Thus, we can use the center position $O$ of the *KOA* as the position estimate. If more than one *KOA* exist, however, the node will not be localized. Suppose that the *KOA* consists of $n$ vertices. Let the coordinates of those $n$ vertices (from $v_1$ to $v_n$) be $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. The coordinates of node $O$ is defined as

$$\left(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n}\right),$$

which denote the estimate position of node $G$.

In the random movement pattern, it is critical to efficiently and correctly obtain the $n$ $(v_1, \ldots, v_n)$ vertices of a *KOA* to estimate the position of node $G$. We present the following procedure to attain this goal, even if node $G$ misses crucial messages from the moving beacon, which means that some *ADOs* may be wrongly represented. If node $G$ can acquire the information of $k$ *ADOs*, it gets the $k$ prearrival positions, $k$ arrival positions, $k$ departure positions, and $k$ postdeparture positions of the beacon. We can draw circles at these $4k$ positions with radius $r$. Let the point set $P$ contain all the intersection points of these $4k$ circles. Then, we have $\forall i$, $v_i \in P$ ($P$ may contain other points as well). We define the *OutPoints* as a set to encompass both the prearrival and postdeparture positions, whereas the *InPoints* is a set encompassing both the arrival and departure positions. Therefore, $2k$ *Inpoints* and $2k$ *OutPoints* form the center positions of beacon broadcasting messages. We calculate a degree for each point in $P$ in either the ideal or real situation. The points with the highest degree constitute the vertices $(v_1, \ldots, v_n)$ of *KOA*. Given a point $u$ in $P$, its initial degree is set to 0. If the distance between $u$ and an *InPoint* is not more than $r$, its degree is increased by 1. The degree can also be increased by 1 when the distance between itself and an *OutPoint* is not less than $r$. In the ideal situation without message missing, the degrees of each $v_i$ is $4k$. Other points in $P$ do not have this attribute, and their degrees are less than $4k$. Thus, we can find these $n$ vertices with degree $4k$ from the point set $P$ accordingly. In real situations where node $G$ may obtain wrong *ADOs* because of messages missing, $n$ vertices, which are close to $G$, still show a higher degree than others in $P$, although they may not reach the degree of $4k$. The localization algorithm in the random movement is shown in Algorithm 3, and such an algorithm is robust to handle the radio-irregular situation, as described in the next section:

**Algorithm 3**: algorithm for the random movement pattern.
Input: $node_i$, $InPoints$, $OutPoints$
Output: the position estimate $p_i$ of $node_i$
1: compute all the intersection points $P$;
2: **for** each point $u$ in $P$ **do**
3: $u.degree = 0$;
4: **for** each point $a$ in $OutPoints$ **do**
5: **if** $|au| \geq r$ **then**
6: $u.degree\,{+}{+}$;
7: **end if**
8: **end for**
9: **for** each point $b$ in $InPoints$ **do**
10: **if** $|bu| \leq r$ **then**
11: $u.degree\,{+}{+}$;
12: **end if**
13: **end for**
14: **end for**
15: fetch the points in $P$ with the highest degree;
16: compute $p_i$ to be the central point of these points;
17: output $p_i$;

## 5 Localization in the Real Environment

In a real environment, the radio irregularity shows its impact on the location estimate accuracy of sensor nodes. A sensor node might not capture a signal from the broadcasting beacon, even if it falls inside the normal transmission range, due to environmental interference. This phenomenon could confuse the real location of nodes and thus increase estimate errors. To obtain accurate location estimates of sensor nodes, we need to use a model to appropriately reflect the RSS for sensor nodes in a real environment. In this section, we first present a model that can accurately indicate the RSS in the real implementation, and then, we describe how the localization algorithms can be calibrated for the explored three movement patterns respectively using this model.

### 5.1 Model for Received Signal Strength Calculation

The RSS is dominated by the Radio Frequency (RF) sending power, path losses, frequency selective fading, and shadowing losses. Among these factors, path losses are deterministic to the power consumption. Frequency-selective fading is much correlated with frequency. Shadowing is caused by static or random-moving obstructions that impact the signal

transmission. To calculate RSS in a real environment, the log-normal shadowing model [11] is the most widely used, in which path losses, that is, the key factor of power consumption, are treated to be isotropic. However, based on the experiments conducted by Zhou et al. [10], path losses should be nonisotropic, which means that the path losses vary in directions. Meanwhile, they concluded that the RF sending power should not be invariable and it follows a normal distribution. According to these observations, the model for RSS calculation can be represented as follows based on [10]:

$$RSS = VSP\ Adjusted\ Sending\ Power - DOI\ Adjusted\ Path\ Loss + Fading. \quad (7)$$

Here, the *Degree of Irregularity* (*DOI*) is used to define the maximum variance of path loss at every direction, whereas the *Variance of Sending Power* (*VSP*) follows a normal distribution:

$$VSP\ Adjusted\ Sending\ Power = Sending\ Power * (1 + NormalRand * VSP), \quad (8)$$

$$DOI\ Adjusted\ Path\ Loss = Path\ Loss * K_i, \quad (9)$$

$$K_i = \begin{cases} 1, & i = 0, \\ K_{i-1} \pm Rand \times DOI, & 0 < i < 360 \wedge i \in N, \end{cases} \quad (10)$$

where $|K_0 - K_{359}| \leq DOI$.

Equation (10) produces a value for each of the 360 directions. The usage of this model is complex, because we should know the angle direction composed by two sensor nodes. We simplify this model by replacing (9) and (10) with the following:

$$DOI\ Adjusted\ Path\ Loss = Path\ Loss * (1 \pm\ Rand\ \times DOI). \quad (11)$$

Thus, we have a new way of calculating the RSS in response to radio irregularity through (7). We use a random value to present the variance of path loss, and this random value ($Rand$) follows a *Weibull* distribution. Parameter *DOI* determines the maximum variance. Through this new way of calculating the RSS, we not only maintain the property that the variance of path loss is distinct in every direction but also show that pass losses in the same direction may be varying due to the dynamic changing of environment.

### 5.2 Practical Localization Method

In a real environment, that a sensor node can receive a valid packet from another sensor node indicates that the received signal is stronger than the product of the Signal-to-Noise Ratio (SNR) value and background noise level around itself. Suppose that the strength of background noise is varying in a specific range according to the environment. We define that two nodes are *within range* if the RSS between them is larger than a threshold $P$. We also define the transmission range $r$ to be the distance at which the RSS is $P$ when both DOI and VSP are equal to zero. Thus, if the RSS is stronger than threshold $P$, the distance between two nodes falls within the transmission range $r$. The RSS denotes the real strength that a sensor node obtains as follows:

$$\begin{cases} distance \leq r, & \text{if } RSS \geq P, \\ distance > r, & \text{if } RSS < P. \end{cases} \quad (12)$$

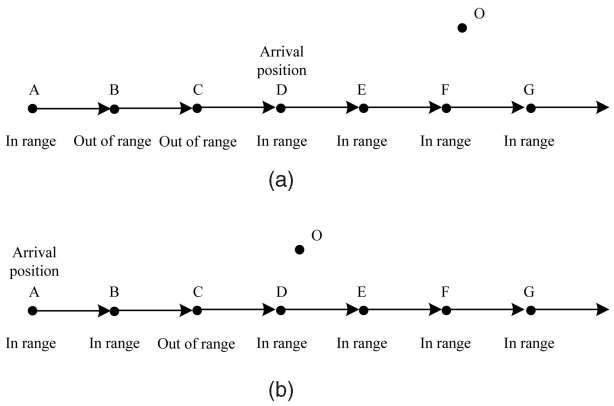


Fig. 7. Determination of arrival position in a real environment.

We first study the impact of the radio irregularity in the case of SSL and DSL movement patterns. In reality, the arrival position and departure position of a beacon will be different from the positions in the ideal situation, and we need to distinguish the actual arrival/departure position of the beacon to a sensor node. The captured arrival/departure position may be one or several broadcasting points before or after the ideal arrival/departure position due to the radio irregularity. In Fig. 7, *in range (out of range)* means that the sensor node receives (does not receive) signals from the beacon. Fig. 7 shows the possible scenarios that the state of *in range* interleaves the state of *out of range*, and vice versa. Without loss of generality, we take the arrival position into consideration (the departure position can be derived similarly) and suppose that the beacon is moving from left to right. A sensor node collects the signals captured locally when a beacon is passing. When a signal is captured, the state is denoted as *in range*. Without receiving a signal from the beacon at a predicted position, a sensor node conjectures the *out-of-range* state by knowing the moving direction and broadcasting interval $s$ from the beacon. We determine the actual arrival position of the beacon according to two observations. If the number of out-of-range points is larger than the one of in-range points in the arriving direction, we reset those in-range points to be out of range. Otherwise, those out-of-range points are reset to be in range. For example, in Fig. 7a, the fact that points $B$ and $C$ are out of range implies that point $A$ should be out of range and the arrival position of the moving beacon should be at point $D$. In Fig. 7b, point $C$ will be set to be in range, and the arrival position is at point $A$.

After the identification of the actual arrival and departure positions of the beacon, each node computes its position estimate by using the method presented in the ideal situation. It is possible, however, that we can no longer obtain an ADO area, as shown in Fig. 1, because of the deviation of the identified arrival (departure) position of the beacon from the one in an ideal environment. When such a scenario occurs, it is easy to conclude that the node is very near the moving line of the beacon. Thus, a good location estimate of the node could be obtained when we set the midpoint of arrival and departure positions as its $x$-coordinate and the latitude of the moving line as its $y$-coordinate.

We need to study the impact of radio irregularity in the random pattern as well, because a sensor node could obtain an outlier *ADO* unexpected in the real environment. Due to the random-moving route of the beacon, we cannot use the

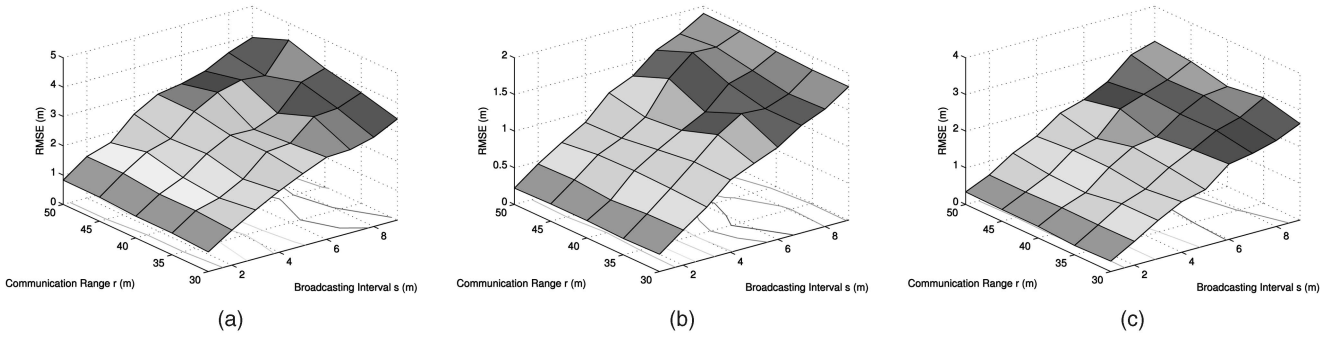


Fig. 8. RMSE comparisons among three movement patterns under different $r$ and $s$. (a) RMSE in the SSL movement pattern. (b) RMSE in the DSL movement pattern. (c) RMSE in the random movement pattern.

identification method proposed above for the SSL and DSL movement patterns to lower such an impact. However, if a node obtains more correct *ADOs* than wrong *ADOs*, the estimation error may limit to a small value. This is because correct *ADOs* will form a *KOA*, from which the node can attain its most likely location. If wrong *ADOs* have no overlap with the kernel area, they are detected as outliers. The algorithm shown in Algorithm 3 can still be applicable, and we request at least three *ADOs* to localize each node. A single wrong *ADO* is acceptable, since it shows little impact on the position estimation accuracy.

## 6 Performance Evaluation

In this section, we show the simulation results of the proposed localization algorithms in response to the three movement patterns of the beacon—the SSL, DSL, and random movement pattern—in both the ideal environment and the real environment, where the RSS calculation model is applied to reflect the impact of radio irregularity.

### 6.1 Performance in the Ideal Situation

In the ideal situation, suppose that 300 nodes are randomly distributed in an area of 500 m $\times$ 500 m. The total distance that the beacon travels, that is, the traveling distance, in the SSL or DSL movement pattern is determined by both $r$ and the sensor deployment area. This traveling distance is not fixed in the random movement pattern. In order to measure their performance on a comparative and fair basis, the default traveling distance in the random movement pattern is set to be the same as in the DSL movement pattern. The three movement patterns are measured in terms of the *Root Mean Square Error* (RMSE) of the localized nodes, the percentage of the localized nodes out of all the nodes, and the percentage of the accurately localized nodes whose estimation errors are below a threshold of 0.2 m. We tested the movement of the beacon 100 times over the same area, with a new random distribution of nodes each time. We also show the impact of the average node degree on the percentage of localized nodes in the SSL movement pattern and the impact of the traveling length of the beacon not only on the percentage of the localized nodes but also on the position estimate accuracy of each node.

#### 6.1.1 Comparisons of the Three Movement Patterns

We compare the performance of the SSL, DSL, and random movement patterns in two simulation scenarios. In our first simulation scenario, we compare the RMSE of successfully localized nodes, with $r$ (transmission range) varying from 30 to 50 m and $s$ (broadcasting interval) varying from 1 to 9 m. The performance results of the three movement patterns are shown in Fig. 8. No matter which movement pattern the beacon takes, the RMSE is proportional to $s$. A small $s$ leads to a small calculated *HADO*. When $r$ varies, the RMSE changes little, because the shape of *HADO* remains almost unchanged. This is because, first, the maximum horizontal error remains the same and, second, the maximum vertical error depends on the distance of the node from the movement line of the beacon, and in most situations, this error does not vary much. In the SSL movement pattern, the RMSE is comparatively worse than in the other two patterns. The RMSE is less than 1 m only when $s$ is set at 1 m. The DSL movement pattern can achieve the best results, because almost every node is localized by two complementary *HADOs*. When $s = 1$, the RMSE can be less than 30 cm. Even when $s = 4$, the RMSE is less than 1 m. The random movement pattern has its RMSE being less than that in the SSL movement pattern, for it takes longer moving length, and at least two *ADOs* are computed to guarantee accuracy. When $s$ is not longer than 3 m, the RMSE can achieve an accuracy of about 1 m or even less. In this simulation, the percentage of successfully localized nodes in the DSL movement pattern is kept at 100 percent, whereas the percentage in other two movement patterns are lower than 100 percent.

In the second simulation scenario, we study the percentage of accurately localized nodes whose estimate errors are under 0.2 m. As shown in the performance of RMSE, the position estimate error is not much related to the communication range $r$. Thus, we fix $r$ to be 40 m as a default. Fig. 9 displays the percentage of accurately localized nodes in three movement patterns. The percentage decreases with a larger $s$. SSL has the lowest percentage under any broadcasting interval. When $s$ is small, the percentage in the DSL movement pattern

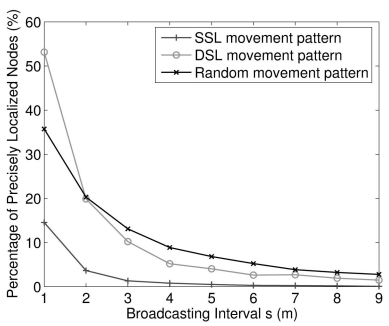


Fig. 9. Percentage of accurately localized nodes.

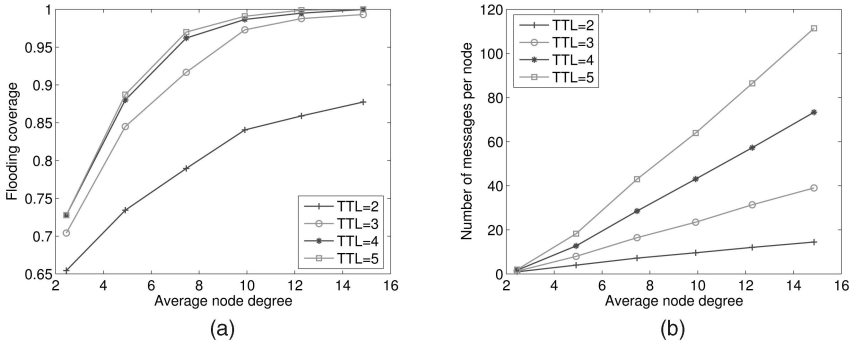


Fig. 10. Communication overhead in terms of the TTL and network average node degree in the SSL movement pattern. (a) Flooding coverage. (b) Flooding cost.

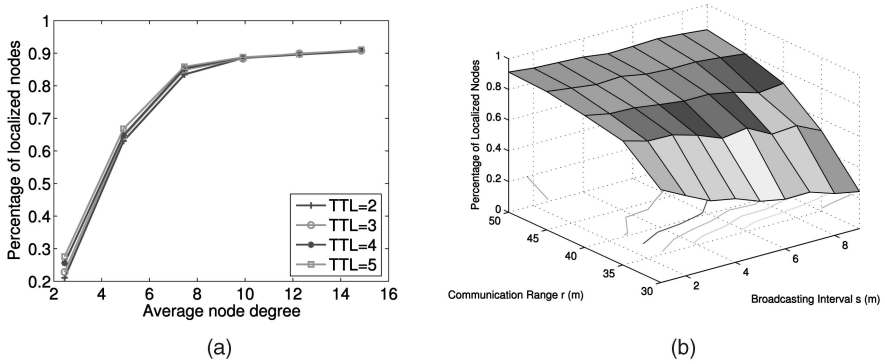


Fig. 11. Percentage of localized nodes in the SSL movement pattern. (a) Under different TTL values and average node degrees. (b) Under different $r$ and $s$.

is bigger than the random movement, which is mainly because of the obtained small *HADOs* and their complementary relationship. When $s$ is large, the percentage in the random movement pattern exceeds that in the DSL movement pattern a little. This is a consequence of the beacon passing more frequently through some areas and providing more *ADOs* to localize some sensor nodes.

### 6.1.2 Performance of the Sparse Straight Line Movement Pattern

In this section, we first analyze the communication overhead in the SSL movement pattern, which is related to the network average node degree and TTL value set in the flooding scheme. Then, we study the percentage of localized nodes under different parameters such as average node degree, TTL, $r$, and $s$.

As mentioned in Section 4.1, a localized node can provide valuable information to nodes who are in the same or neighboring grids. As a result, we define the *flooding coverage* of a node $u_i$ to be the percentage of nodes in the same or neighboring grids that can receive the flooding message from $u_i$. Fig. 10a shows the average flooding coverage under the setting average node degree and TTL values. More neighboring nodes can capture the valuable flooding message from $u_i$ when we increase the TTL value in a dense network (that is, bigger node degree). When the average node degree is 7.5 and TTL is set to 3, the flooding coverage is over 90 percent, which reflects that most nodes are able to receive the localization information from their neighbors. Fig. 10b demonstrates the corresponding flooding cost. We measure the flooding cost in terms of the number of messages broadcast by each node. The flooding overhead increases sharply, along with the increase in both the TTL and average node degree. When the average node degree is over 10, the broadcasted messages from each node could be a lot if the TTL value is over 3. In a default setting with 300 nodes distributed in an area of 500 m $\times$ 500 m, the average node degree is around 7.5, and the flooding cost is tolerable when TTL is equal to 3.

Fig. 11a shows the impact of the average node degree and TTL value on the percentage of localized nodes in the SSL movement pattern, where $r = 40$, and $s = 5$. The percentage increases, along with the increment of node degree, because of more nodes being able to use information from their neighbors (Rules 1, 2) to distinguish whether they are above or below the movement line of the beacon. When the average node degree of a node is more than 7.5, the percentage increases slowly. The slow increment comes from some nodes located in the uncovered space or adjacent upper and lower *HADOs* of some nodes unable to be identified. Although the increase in the TTL value can improve the percentage a little, TTL is not a major factor to impact the percentage of localized nodes. Thus, we only need to set TTL to 3 to reduce the communication overhead. Different $r$ and $s$ will also affect the percentage, which is shown in Fig. 11b. The percentage of localized nodes increases when $r$ increases and decreases when $s$ increases. If $r$ is large, more nodes can communicate with each other, which, in turn, increases the node connectivity. However, if $s$ is large, the area of *HADO* will be large, making it difficult to use Rules 2 and 3 to determine a node's position.

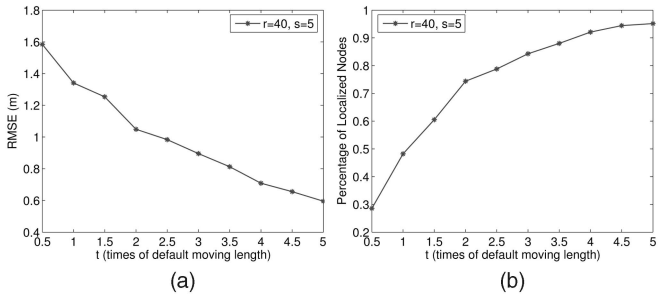


Fig. 12. Impact of the traveling distance of the beacon. (a) RMSE. (b) Percentage of localized nodes.

### 6.1.3 Impact of Traveling Distance in the Random Movement Pattern

As the traveling distance of the beacon increases, more nodes are covered, and those nodes are able to obtain more *ADOs*, which results in more accurate position estimates and higher percentage of modes to be localized. Assume that $r = 40$, $s = 5$, and the traveling distance of the beacon varies from 0.5 to 5 times a default metric. In the default traveling distance, which has the same length as in the DSL movement pattern, the RMSE is less than 1.4 m, and the percentage of localized nodes is slightly less than 50 percent. Fig. 12 displays that when the traveling distance increases, the RMSE decreases, and the percentage of localized nodes increases. In order to acquire the same RMSE as in the DSL movement pattern, the beacon needs to travel almost two times longer distance in the random movement pattern.

## 6.2 Performance in Real Situations with Radio Irregularity

In this section, we conduct two sets of simulations to evaluate localization algorithms in a real environment. The simulation encompasses 50 nodes randomly distributed in an area of 30 m $\times$ 30 m, with the default communication range 6 m. The first and second sets investigate the impact of the broadcasting interval and transmission range of the moving beacon, respectively.

### 6.2.1 Broadcasting Interval

In the first set of simulations, where $DOI = 0.05$, and $VSP = 0.5$, the broadcasting interval of the beacon is increasing from 0.25 to 1.5 m.

Fig. 13a shows that location algorithms under the respective movement patterns (the SSL, DSL, and random movement patterns) can yield accurate estimates where the

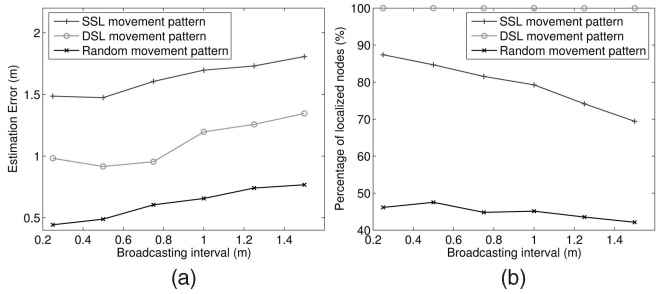


Fig. 13. Performance under various broadcasting intervals, with $\mathrm{DOI} = 0.05$, and $\mathrm{VSP} = 0.5$. (a) RMSE. (b) Percentage of localized nodes.

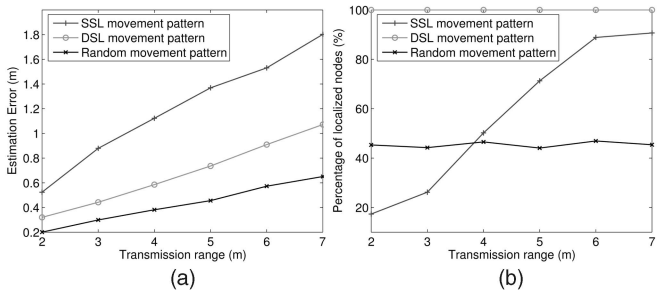


Fig. 14. Performance under various transmission ranges, with $\mathrm{DOI} = 0.05$, and $\mathrm{VSP} = 0.5$. (a) RMSE. (b) Percentage of localized nodes.

real location of each sensor node can be confined within a distance (less than 1.8 m) to the calculated one. The RMSE moves upward with a larger broadcasting interval. The DSL gives rise to a more accurate estimate with a smaller RMSE than the SSL does under the same conditions. The RMSE of the random movement pattern is the smallest, because we require at least three ADOs, which will further confine the possible area in which a node may reside. Fig. 13b shows the percentage of localized nodes. The DSL movement pattern can guarantee all nodes to be localized, whereas the random movement pattern localizes the fewest nodes. The percentage in the SSL and random movement patterns falls with larger broadcasting intervals. The reason for SSL, similar to the ideal situation, is that a larger broadcasting interval will generate a big HADO. In the random movement pattern, a larger broadcasting interval will slightly reduce the number of nodes that are able to capture three ADOs, and thus, the percentage falls accordingly.

### 6.2.2 Transmission Range

Note that the transmission range of the moving beacon can be adjusted using various RF powers. Given the broadcasting interval $s = 0.5$ m, $DOI = 0.05$, and $VSP = 0.5$, Figs. 14a and 14b show the RMSE and percentage of localized nodes for the three movement patterns, respectively. The RMSE is kept small when the transmission range is short, because a short transmission range causes a relatively small variance of DOI and VSP. The percentage is 100 percent for the DSL (which is capable of localizing all nodes), whereas it is less than 100 percent for both the SSL and random movement patterns. The percentage curves of SSL and random show distinct trends. The SSL curve moves upward, along with the transmission range increment, partly because a wider transmission range can increase the node connectivity and further help localize each other.

TABLE 1
General Performance Comparison, with a Transmission Range of 1.8 m

| | RMSE (m) | Maximum error (m) | Minimum error (m) | Percentage of localized nodes |
|---|---|---|---|---|
| DSL movement (0.3m) | 0.164 | 0.427 | 0.078 | 100% |
| DSL movement (0.6m) | 0.213 | 0.491 | 0.063 | 100% |
| SSL movement (0.3m) | 0.43 | 0.964 | 0.171 | 64.3% |
| SSL movement (0.6m) | 0.553 | 0.796 | 0.381 | 35.7% |

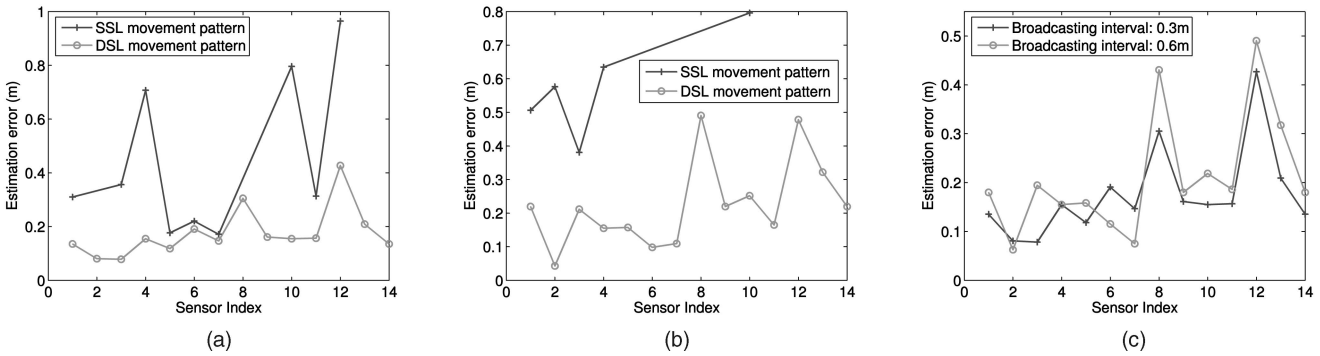


Fig. 15. Performance results in the first set of experiments, with a transmission range of 1.8 m. (a) The DSL movement pattern versus the SSL movement pattern, with a broadcasting interval of 0.3 m. (b) The DSL movement pattern versus the SSL movement pattern, with a broadcasting interval of 0.6 m. (c) Broadcasting interval of 0.3 m versus broadcasting interval of 0.6 m when using the DSL movement pattern.

However, the curve of the random movement pattern remains almost unchanged, since the transmission range increment will reduce the total traveling distance, which neutralizes its effect.

## 7 Experiments

In this section, we show the performance of our distributed localization method in a real outdoor environment. The method assists the sensor in computing its location locally and applying distinct algorithms according to the movement patterns of the mobile beacon. We conducted two sets of experiments using the Berkeley Micaz motes without equipping any sensor boards. In the first set of experiments, we used one sensor node as a moving beacon and randomly distributed 14 sensor nodes on a smooth ground of 8 m $*$ 8 m. The transmission range of the moving beacon is set to be 1.8 m by adjusting the transmission power. We evaluated the localization estimate errors of sensor nodes in both the DSL and SSL movement patterns with two different broadcasting intervals from the beacon: 0.3 and 0.6 m. In the second set of experiments, we used one sensor node as a moving beacon and randomly distributed 29 sensor nodes in an area of 15 m $*$ 15 m. The tested transmission range of the beacon was 3.3 m. We studied the DSL, SSL, and random[2] movement patterns of the beacon when it broadcast messages to sensor nodes within intervals: 1.2 and 2.4 m. In these two sets of experiments, the performance of the different algorithms applied in sensor nodes to estimate their positions according to movement patterns of the beacon is measured in terms of the RMSE, Maximum error, Minimum error, and percentage of localized nodes. The Maximum error denotes the largest deviation of the real position from the estimate position among all localized nodes, and the Minimum error denotes the smallest deviation. In the SSL movement, the flooding message from a sensor node contains an initial TTL value of 3.

Table 1 and Fig. 15 show the obtained results in the first set experiments. Table 1 illustrates the results of the RMSE, Maximum error, Minimum error, and percentage of localized nodes when the beacon broadcast signals at intervals of 0.3 and 0.6 m in both DSL and SSL movements. As expected, the performance of DSL is superior to the performance of SSL, that is, smaller values of the RMSE, Maximum, and Minimum errors and a higher percentage of localized nodes. A smaller broadcasting interval (0.3 m) can achieve better results in both DSL and SSL than a bigger broadcasting interval (0.6 m) does. In the circumstance of DSL, all nodes can estimate their positions within a distance of at most 0.491 m. In the circumstance of SSL, part of them (only 64.3 percent and 35.7 percent) can do the position estimate, whereas others lack messages from the beacon to make a final estimate. Figs. 15a and 15b show the detailed estimate error information of each localized node. When the broadcasting interval is 0.3 m, as shown in Fig. 15a, the estimate error of each localized node in DSL is smaller than that in SSL. We only connect nine points in the SSL movement pattern, which denotes that these nine nodes are successfully localized. The detailed estimate error results when the broadcasting interval $s$ is 0.6 m are shown in Fig. 15b. The major difference is that the estimate error becomes larger and fewer nodes (only five nodes) are localized in SSL. Fig. 15c presents the comparison of estimate errors between $s = 1.2\,\mathrm{m}$ and $s = 2.4\,\mathrm{m}$ when using the DSL movement pattern. Although the accuracy shows some variance, most of the nodes are localized more accurately when the broadcasting interval is set to 0.3 m.

Table 2 and Fig. 16 illustrate the performance results in the second set of experiments. The estimate errors in the second set experiment are normally larger than that in the first set, because the transmission range and broadcasting interval of the moving beacon are wider. Nevertheless, the results in the second set of experiments show a similar

TABLE 2
General Performance Comparison, with a Transmission Range of 3.3 m

| | RMSE (m) | Maximum error (m) | Minimum error (m) | Percentage of localized nodes |
|---|---|---|---|---|
| DSL movement (1.2m) | 0.524 | 1.022 | 0.047 | 100% |
| DSL movement (2.4m) | 0.861 | 1.55 | 0.384 | 100% |
| SSL movement (1.2m) | 0.938 | 1.860 | 0.192 | 76.9% |
| SSL movement (2.4m) | 1.164 | 1.90 | 0.368 | 73.1% |
| Random movement (1.2m) | 0.915 | 2.589 | 0.173 | 68.2% |
| Random movement (2.4m) | 1.022 | 2.639 | 0.132 | 59.1% |

2. The total traveling distance is set to be the same as the one in the DSL movement pattern.

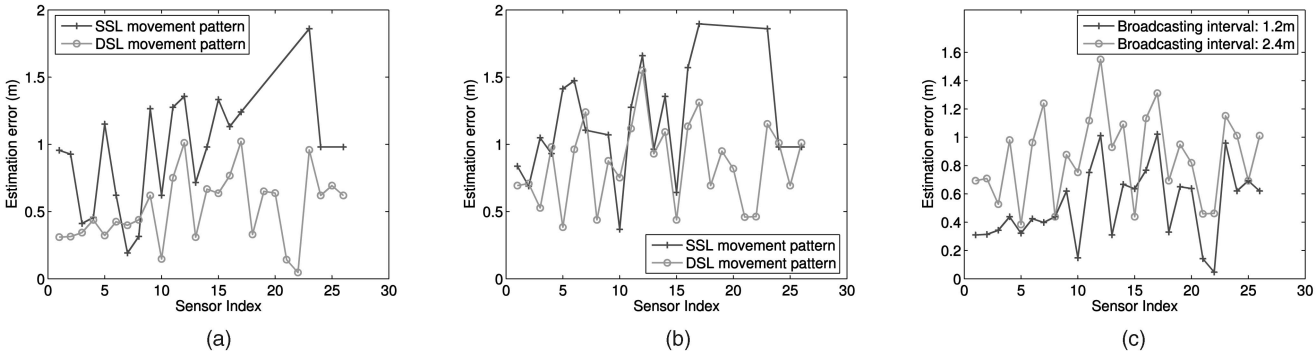


Fig. 16. Performance results in the second set of experiments, with a transmission range of 3.3 m. (a) The DSL movement pattern versus the SSL movement pattern, with a broadcasting interval of 1.2 m. (b) The DSL movement pattern versus the SSL movement pattern, with a broadcasting interval of 2.4 m. (c) Broadcasting interval of 1.2 m versus broadcasting interval of 2.4 m when using the DSL movement pattern.

conclusion as in the first set of experiments with regard to the movement patterns of the beacon. Among them, the performance of DSL is the best. The SSL and random movement patterns have close performance results with regard to the percentage of localized nodes. However, the traveling distance of the beacon in the random movement pattern is almost twice longer than that in the SSL movement pattern. Figs. 16a and 16b display the detailed estimate error of every localized node for both DSL and SSL. When the broadcasting interval of the beacon is 1.2 m, all nodes (29) can be successfully localized in DSL, and this number decreases to 20 in SSL. When the broadcasting interval increases to 2.4 m, the number of localized nodes in SSL further decreases to 19. The experimental results meet the previous theoretical analysis that the DSL movement can cover all sensor nodes no matter what the sensor node distribution topology is, whereas the SSL cannot. Thus, we only connect these localized nodes in Figs. 16a and 16b for comparison. We also observe that most of the nodes are localized more accurately in DSL. Fig. 16c compares the setting of $s = 1.2\,\mathrm{m}$ with the setting of $s = 2.4\,\mathrm{m}$ in DSL. Similar to the first experiment, a smaller broadcasting interval yields better results.

## 8 Discussion and Future Work

Although previous work has addressed the sensor position localization using mobile beacons, it lacks the crucial analysis of movement behavior from beacons. The movement pattern of a beacon could severely affect the accuracy and even the successful location of each sensor node. Thus, we proposed three distinct movement patterns for a single beacon, that is, the SSL, DSL, and random movement patterns, and present their performance in this paper. Given these patterns, the position estimate of each node is calculated using the broadcasting information of the beacon locally. Note that the area constraints of the nodes' selves, including the positive communication constraints among neighbors and the negative constraints among nodes that cannot communicate directly, can further increase the accuracy of position estimate or help localize nodes that are residing in uncovered spaces. However, using these constraints information to assist localization needs a large computation, and sensor nodes do not have such computation capability. One possible solution is to use the central computation on a powerful computer rather than on each sensor node. Finding a simple solution to use these constraints in a distributed low-cost way is our future work.

A range-based technique is unlikely feasible for static localization because of additional expensive hardware. In mobile localization, the powerful beacon can be equipped with such hardware support to measure the distance between itself and each node and then send back the distance information. This can further reduce the estimation errors on the basis of our mobile range-free localization. Combining the range-free and range-based localization in an effective and efficient way is also our future work.

## 9 Conclusion

We present a distributed range-free localization method that needs to use only one moving beacon within a sensor network. The basic idea is to narrow down the possible location of a node by using the arrival and departure constraint areas derived from the moving beacon. The moving behavior of the beacon is an important aspect for localization accuracy. We have described three beacon movement patterns, that is, SSL, DSL, and random, each allowing sensor nodes to be localized with a different degree of accuracy. According to the different movement patterns of the beacon, sensor nodes should apply the correspondingly proposed algorithms to efficiently compute their positions. In a real environment, where the RSS could be severely affected by the environment, a sensor may wrongly conjecture constraint areas to affect its location estimate precision. Thus, we adopt a practical RSS model to calibrate the localization method into a real environment to remove radio irregularity as much as possible. Both our simulation and experiment results show that the DSL movement pattern is invariably the most effective. This movement localizes all nodes with small degrees of error, even in real outdoor environments, that is, where transmission is unstable, and beacon messages are missing.

## Acknowledgments

Bin Xiao is supported by HK RGC PolyU 5307/07E and China National 973 Grant 2007CB307100. Hekang Chen and Shuigeng Zhou are supported by the National Basic Research Program of China under Grant 2007CB310806 and the NSFC Grants 60496327 and 90612007.

## REFERENCES


[1] T. Yan, T. He, and J.A. Stankovic, "Differentiated Surveillance Service for Sensor Networks," *Proc. First ACM Conf. Embedded Networked Sensor Systems (SenSys '03),* pp. 51-63, 2003.

[2] J. Hightower, G. Boriello, and R. Want, "SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength," CSE Report #2000-02-02, Univ. of Washington, Feb. 2000.

[3] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed Online Localization in Sensor Networks Using a Moving Target," *Proc. Third Int'l Symp. Information Processing in Sensor Networks (IPSN '04),* pp. 61-70, 2004.

[4] Y.-B. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Proc. ACM MobiCom '98,* pp. 66-75, 1998.

[5] H. Yang and B. Sikdar, "A Protocol for Tracking Mobile Targets Using Sensor Networks," *Proc. First IEEE Int'l Workshop Sensor Network Protocols and Applications (SNPA '03),* pp. 71-81, 2003.

[6] W. Zhang and G. Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," *IEEE Trans. Wireless Comm.,* pp. 70-84, 2004.

[7] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proc. IEEE INFOCOM '00,* pp. 775-784, Mar. 2000.

[8] N. Patwari and A.O. Hero III, "Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks," *Proc. Second ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '03),* Sept. 2003.

[9] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AoA," *Proc. IEEE INFOCOM '03,* pp. 1734-1743, 2003.

[10] G. Zhou, T. He, S. Krishnamurthy, and J. Stankovic, "Impact of Radio Irregularity on Wireless Sensor Networks," *Proc. Second Int'l Conf. Mobile Systems, Applications, and Services (MobiSys '04),* pp. 125-138, 2004.

[11] T.S. Rappaport, "Wireless Communications," *Principles Practice,* 1999.

[12] M. Mauve, J. Widmer, and H. Hartenstein, "A Survey on Position-Based Routing in Mobile Ad Hoc Networks," *IEEE Network Magazine,* pp. 30-39, 2001.

[13] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large-Scale Sensor Networks," *Proc. ACM MobiCom '03,* pp. 81-95, 2003.

[14] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low-Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm. Magazine,* vol. 7, no. 5, pp. 28-34, Oct. 2000.

[15] A. Savvides, C.-C. Han, and M.B. Strivastava, "Dynamic Fine-Grained Localization in Ad Hoc Networks of Sensors," *Proc. ACM MobiCom '01,* pp. 166-179, 2001.

[16] T. Parker and K. Langendoen, "Localisation in Mobile Anchor Networks," Technical Report PDS-2005-001, Delft Univ. of Technology, Feb. 2005.

[17] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," *Proc. ACM MobiCom '04,* pp. 45-57, 2004.

[18] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," *Proc. IEEE INFOCOM '05,* Mar. 2005.

[19] J.J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," *Proc. Second Int'l Workshop Information Processing in Sensor Networks (IPSN '03),* pp. 113-128, Apr. 2003.

[20] P. Bergamo and G. Mazzini, "Localization in Sensor Networks with Fading and Mobility," *Proc. 13th IEEE Int'l Symp. Personal, Indoor and Mobile Radio Comm. (PIMRC '02),* pp. 750-754, Sept. 2002.

[21] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: A Sequence-Based Technique for RF-Only Localization in Wireless Sensor Networks," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN '05),* pp. 285-292, Apr. 2005.

[22] X. Li and R.P. Martin, "A Simple Ray-Sector Signal Strength Model for Indoor 802.11 Networks," *Proc. Second IEEE Int'l Conf. Mobile Ad Hoc and Sensor Systems (MASS '05),* pp. 639-648, 2005.

[23] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin, "Statistical Model of Lossy Links in Wireless Sensor Networks," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN '05),* pp. 81-88, Apr. 2005.

[24] B. Xiao, H. Chen, and S. Zhou, "A Walking Beacon-Assisted Localization in Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '07),* pp. 3070-3075, June 2007.


**Bin Xiao** received the BSc and MSc degrees in electronics engineering from Fudan University, Shanghai, in 1997 and 2000, respectively, and the PhD degree in computer science from the University of Texas, Dallas, in 2003. He is currently an assistant professor in the Department of Computing, Hong Kong Polytechnic University, Hong Kong. His research interests include communication and security in computer networks, peer-to-peer networks, wireless mobile ad hoc and sensor networks. He is a member of the IEEE.

**Hekang Chen** received the bachelor's degree in computer science from Tongji University, Shanghai, in 2004. He is currently working toward the master's degree in the Department of Computer Science and Engineering, Fudan University, Shanghai. From August 2005 to February 2006, he was with the Department of Computing, Hong Kong Polytechnic University, as a research assistant. His research interests include sensor networks, database, and data mining. He is a member of the IEEE.

**Shuigeng Zhou** received the bachelor's degree in electronic engineering from Huazhong University of Science and Technology (HUST) in 1988, the master's degree in electronic engineering from the University of Electronic Science and Technology of China (UESTC) in 1991, and the PhD degree in computer science from Fudan University, Shanghai, in 2000. He is currently a professor in the Department of Computer Science and Engineering, Fudan University. He was with Shanghai Academy of Spaceflight Technology as an engineer from 1991 to 1997 and has been a senior engineer since August 1995. From 2000 to 2002, he was a postdoctoral researcher in the State Key Laboratory, Software Engineering, Wuhan University. His research interests include data management in P2P and sensor networks, data mining, and information retrieval. He has published more than 100 papers in domestic and international journals and conference proceedings. He is a member of the IEEE, the ACM, and the IEICE.