Range-Free Localization

Radu Stoleru[1], Tian He[2] and John A. Stankovic[3]

[1] Department of Computer Science, University of Virginia,
stoleru@cs.virginia.edu
[2] Department of Computer Science and Engineering, University of Minnesota
tianhe@cs.umn.edu
[3] Department of Computer Science, University of Virginia,
stankovic@cs.virginia.edu

1 Introduction

Advances in micro-electro-mechanical systems have triggered an enormous interest in wireless sensor networks (WSN). WSN are formed by large numbers of densely deployed nodes enabled with sensing and actuating capabilities. These nodes have very limited processing and memory capabilities, limited energy resources and it is envisioned that they will be mass produced, to reduce costs.

Several challenging problems exist in wireless sensor networks. Among these is how to obtain location information for sensor nodes and events present in the network. From this perspective, we categorize the localization problem as: node localization, target localization and location service. Node localization is the process of determining the coordinates of the sensor nodes in the WSN. Target localization is the process of obtaining the coordinates of an event or a target present in the sensor network. The location of a target can be obtained either passively (the nodes sense the target) or actively, when the target cooperates and communicates with the sensor network. A location service acts as a repository that can be used to answer questions like "where is entity X?". In the remaining part of this chapter we focus on the node localization problem in WSN.

Node localization is a complicated and important problem for wireless sensor networks (WSN). The aspects of this problem that have challenged the research community can be summarized as follows:

- **Assumptions** - The node localization problem remains a difficult challenge to be solved practically. To make the problem practically tractable, its complexity had to be reduced, by making simplifying assumptions. As a result, many localization schemes proposed solutions that are based on assumptions that do not always hold or are not practical. Examples of such assumptions are: circular radio range, symmetric radio connectivity, additional hardware (e.g., ultrasonic), lack of obstructions, lack of line-of-sight, no multipath and flat terrain.

- **Localization Protocol Design** - The problem of localization in WSN is further complicated by the large number of parameters that need to be considered when designing a localization system for a particular WSN deployment. Among these parameters are: the deployment method for the sensor network; the existence of a line-of-sight between sensor nodes and a remote, central point; the time required by the localization scheme; the presence of reference points (anchors) in the network, and the density; the cost for localization, represented by additional hardware (form factor) and energy expenditure (messages exchanged or time necessary for localization).
- **Cost/Accuracy trade-off** - Due to the mostly static nature of many WSN, obtaining the location information by each sensor node is often a one time or rare event. Adding hardware to each sensor node, to assist in the localization, is a costly solution, and, so far, has been ruled out from real system deployments. For example, GPS is a typical high-end solution, which requires sophisticated hardware to achieve high resolution time synchronization with satellites. The constraints on power and cost for tiny sensor nodes and the need for a line of sight from a sensor node to four or more satellites preclude this as a viable solution. Other solutions require per node devices that can perform ranging among neighboring nodes. The difficulties of these approaches are two-fold. First, under constraints of form factor and power supply, the effective ranges of such devices are very limited. For example the effective range of an ultrasonic transducer is on the order of a few meters, when the sender and receiver are not facing each other. Second, since most sensor nodes are static, i.e., the location is not expected to change, it is not cost-effective to equip these sensors with special circuitry just for a one-time localization.
- **Performance Evaluation** - The problem of localization in wireless sensor networks has been studied and evaluated predominantly in simulators. Due to the severe hardware constraints imposed on wireless sensor nodes, real system implementations of the proposed simulated solutions have not produced encouraging results. Solutions that use the most tempting means of evaluating relative distances between sensor nodes - RF signal strength, have largely failed in practice, due to the unreliable nature and irregular pattern of the radio communication. Localization schemes that are based on the receive signal strength indicator (RSSI) have been, however, intensively studied in simulators.
- **Security** - Since localization is a critical factor in WSN, attacks on it can render the sensor network ineffective. To date, very little work has been done on creating robust and secure localization schemes. A few notable exceptions are [15] [14] [17] [16] [5].

For wireless sensor networks ranging is a difficult option. The hardware cost (hardware used only for localization), the energy expenditure, the form factor, the small range, all are difficult compromises, and it is hard to envision cheap, unreliable and resource-constraint devices make use of range-based localization solutions. Their high accuracy in localization is very desirable, however.

To overcome the limitations of the range-based localization schemes, many range-free solutions have been proposed. These solutions estimate the location of sensor nodes by, either, exploiting the radio connectivity information among neighboring nodes, or exploiting the sensing capabilities that each sensor node possesses. Due to the distinct characteristics of these two approaches, we categorize the range-free localization schemes into: anchor-based schemes (which assume the presence of sensor nodes in the network that have knowledge about their location) and anchor-free schemes, which require no special sensor nodes for localization. The range-free localization schemes eliminate the need of high-cost specialized hardware on each sensor node. The fact that the radio propagation characteristics vary over time and are environment dependent, imposes higher calibration costs for the anchor-based localization schemes.

In this chapter we review a *representative* set of range-free localization schemes, from the perspective of the above proposed taxonomy: anchor-based and anchor-free solutions. We point out that hybrid solutions exist and, sometimes, one solution does not neatly fit in either one of the categories. Also, in addition to the localization schemes described below, many more have been proposed. To name a few: the ELA [32], Thunder [35], Hop-TERRAIN [26], KPS [7], RIPS [18], Resilient LSS [13], Robust Quadrilaterals [19] and MAL [23] . In the remaining part of this chapter, we use R to denote the radio range of a sensor node.

2 Anchor-Based Solutions

The location of a sensor node has to be expressed in a coordinate system. In a 2D space, three anchor nodes (three fixed points in the space) uniquely determine a coordinate system. In a 3D space, four anchor nodes are required. In this section, to demonstrate a wide range of possible solutions, we present several range-free localization schemes that use radio connectivity to infer proximity to a set of anchor nodes.

2.1 Centroid

The Centroid scheme was proposed by Bulusu et al. in [2]. This localization scheme assumes that a set of anchor nodes (A_i, $1 \leq i \leq n$), with overlapping regions of coverage, exist in the deployment area of the WSN. The main idea is to treat the anchor nodes, located at (X_i, Y_i), as point masses m_i and to find the center of gravity (centroid) of all these masses. In the most general form, the coordinates of the centroid of n point masses m_i are given by:

$$(X_G, Y_G) = \left(\frac{\sum_{i=1}^{n} m_i X_i}{\sum_{i=1}^{n} m_i}, \frac{\sum_{i=1}^{n} m_i Y_i}{\sum_{i=1}^{n} m_i} \right)$$

which, for equal masses m_i simplifies to:

$$(X_G, Y_G) = \left(\frac{\sum_{i=1}^{n} X_i}{n}, \frac{\sum_{i=1}^{n} Y_i}{n} \right)$$

An example of how the Centroid scheme works is shown in Figure 1, where a sensor node N_k is within communication range to four anchor nodes, $A_1 ... A_4$. The node N_k localizes itself to the centroid of the quadrilateral $A_1A_2A_3A_4$ (for the case of a quadrilateral, the centroid is at the point of intersection of the bimedians - the lines connecting the middle points of opposite sides).

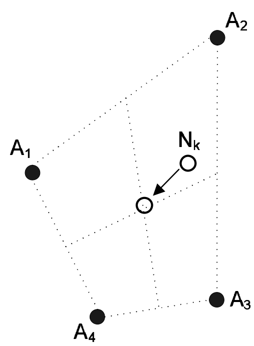

Fig. 1. Centroid localization - node N_k localizes to the centroid of the $A_1A_2A_3A_4$ quadrilateral.

The steps of the localization scheme are the following:

- Each anchor node A_i broadcasts its position.
- Each sensor node N_k listens for beacons from anchors and computes a connectivity metric, for each anchor node A_i it has received beacons from. This metric is defined as follows:

$$CM_{k,A_i} = \frac{N_{recv}(A_i, t)}{N_{sent}(A_i, t)}$$

  where $N_{recv}(A_i, t)$ and $N_{sent}(A_i, t)$ are the numbers of beacons received from anchor A_i and sent by anchor A_i, respectively, in the time interval t.
- A node N_k computes its location as the average of all the anchor nodes A_i it has heard from with a connectivity higher than a threshold, e.g., $CM_{k,A_i} > 90\%$, as follows:

$$(X_k, Y_k) = \left(\frac{X_{A_{i1}} + ... + X_{A_{ij}}}{j}, \frac{Y_{A_{i1}} + ... + Y_{A_{ij}}}{j} \right)$$

where j is the number of anchors with a higher connectivity than the threshold.

In subsequent work [3], the authors explored adaptive mechanisms for placing additional anchor nodes in a WSN, in order to reduce the average localization error.

2.2 APIT

APIT [8] is an area-based range-free localization scheme. It assumes that a small number of nodes, called anchors, are equipped with high-powered transmitters and know their location, obtained via GPS or some other mechanism. Using beacons from these anchors, APIT employs a novel area-based approach to perform location estimation by isolating the environment into triangular regions between anchor nodes as shown in Figure 2. A node's presence inside or outside of these triangular regions allows a node to narrow down the area in which it can potentially reside. By utilizing different combinations of anchors, the size of the estimated area in which a node resides can be reduced, to provide a good location estimate.

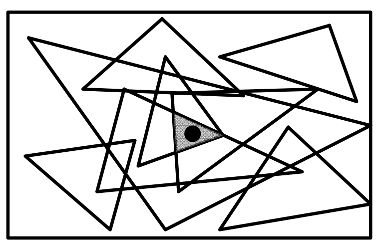

Fig. 2. Area-based APIT Algorithm Overview

The theoretical method used to narrow down the possible area in which a target node resides is called the *Point-In-Triangulation Test* (PIT). For three given anchors: $A(a_x, a_y), B(b_x, b_y), C(c_x, c_y)$, the Point-In-Triangulation test determines whether a point M with an unknown position is inside triangle $\triangle$ABC or not. APIT repeats this PIT test with different anchor combinations until all combinations are exhausted or the required accuracy is achieved. At this point, APIT calculates the center of gravity (COG) of the intersection of all of the triangles in which a node resides to determine its estimated position. These steps are shown in Algorithm 1.

In [8], the authors provide a perfect, albeit theoretical, solution for perfect Point-In-Triangulation test as follows:

Algorithm 1 APIT

1: Receive location beacons (X_i, Y_i) from N anchors;
2: $InsideSet = \emptyset$;
3: **for** each triangle $T_i \in \binom{N}{3}$ triangles **do**
4: **if** Point-In-Triangle-Test (T_i) == TRUE **then**
5: $InsideSet = InsideSet \bigcup T_i$;
6: **end if**
7: **end for**
8: Estimated Position = CenterOfGravity($\bigcap T_i \in InsideSet$);

Perfect P.I.T. Test Theory: If there exists a direction such that a point adjacent to M is further/closer to points A, B, and C simultaneously, then M is outside of △ABC. Otherwise, M is inside △ABC (Figure 3).

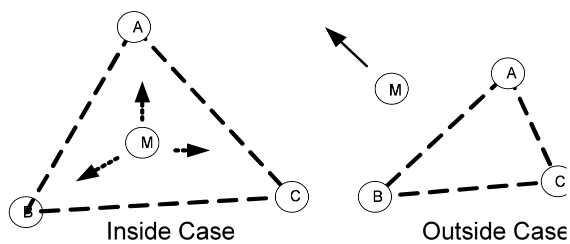

Fig. 3. Cases for Point-In-Triangulation Test

The perfect P.I.T. test can correctly decide whether a point M is inside triangle △ABC (the formal proofs can be found in [8]). However, there are two major issues to apply this theory practically in wireless sensor networks: First, how does a node recognize directions of departure from an anchor without moving? Second, how to exhaustively test all possible directions in which node M might depart/approach vertexes A, B, C simultaneously? The answer to the first question is to use RSSI comparisons. Through experiments, the authors confirm that in a narrow direction, the further away a node is from the anchor, the weaker the received signal strength (RSSI) will be. Through signal strength comparisons, a node can determine whether the direction towards a neighboring node is closer to a given anchor or not. To address the second issue, the authors propose an approximation (APIT) for the perfect PIT test, which uses neighbor information, through RSSI comparisons, to emulate the node movement in the Perfect PIT test. With a finite number of neighbors, APIT can only evaluate a limited number of directions. Consequently, APIT could make an incorrect decision. Fortunately, experiments indicate that the percentage of APIT tests exhibiting such an error is relatively small (14% in the worst case). Figure 4 demonstrates this error percentage as a function of node density. When node density increases, APIT can evaluate more directions, considerably reducing false positive, i.e. APIT returns true, while a node is outside of triangle (OutToInError). On the other hand, false negative (InToOutError) will slightly increase due to the increased chance of edge effects.

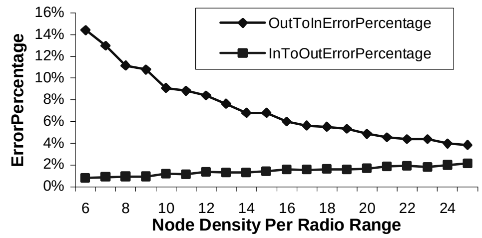

Fig. 4. APIT Error under Varying Node Densities

Once the individual APIT tests finish, APIT aggregates the results (inside/outside decisions among which some may be incorrect) through a grid SCAN algorithm (Figure 5). In this algorithm, a grid array is used to represent the maximum area in which a node likely resides. In the experiments, the length of a grid side is set to 0.1R, to guarantee that estimation accuracy is not noticeably compromised.

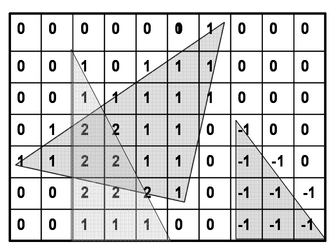

Fig. 5. APIT Error under Varying Node Densities

For each APIT inside decision (a decision where the APIT test determines the node is inside a particular region (Figure 5) the values of the grid regions over which the corresponding triangle resides are incremented. For an outside decision, the grid area is similarly decremented. Once all triangular regions are computed, the resulting information is used to find the maximum overlapping area (e.g., the grid area with value 2 in Figure 5). Since the majority (more than 85% in the worst case shown in Figure 4) of APIT tests are correct, the correct decisions build up on the grid and the small number of errors only serves as a slight disturbance to the final estimation. Evaluation in [8] indicates APIT works better than other range-free solutions under irregular radio patterns and random node placement. However, it should be pointed out that APIT has a more demanding requirement on the number of anchors used in localization.

2.3 SeRLoc

SeRLoc [15] is another area-based range-free localization. The authors assume two types of nodes: normal nodes and locators (i.e., anchors). Normal nodes are equipped with omnidirectional antennas, while locators are equipped with directional sectored antennas and their locations of locators are known a priori. In SeRLoc, a sensor estimates it location based on the information transmitted by the locators. Figure 6 shows the main idea, with node N_k within radio range to locators A_1, A_2 and A_3:

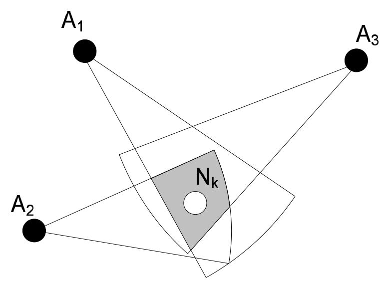

Fig. 6. SerLoc Localization

SeRLoc localizes the sensor nodes in four steps. First, a locator transmits directional beacons within a sector. Each beacon contains the locator's position and the angles of the sector boundary lines. A normal node collects the beacons from all locators it hears. Second, it determines an approximate search area within which it is located based on the coordinates of the locators heard. Third, it computes the overlapping sector region using a majority vote scheme. Finally, SeRLoc determines a node location as the center of gravity of the overlapping region. These steps are shown in Algorithm 2.

We note that SeRLoc is unique in its secure design. It can deal with various kinds of attacks including wormhole and Sybil attacks. We do not describe its security features here except to note that the authors prove in [15] that their approach is more secure, robust and accurate in the presence of attacks, compared with other state-of-the-art solutions that largely ignore this issue.

2.4 Multidimensional Scaling

The MDS-MAP algorithm proposed by Shang et al. in [28] is based on a data analysis technique, called MultiDimensional Scaling (MDS), extensively used

Algorithm 2 SeRLoc

1: Receive beacons from locators; each beacon contains the position of locator and the angles of sector boundary.
2: Find four values: $(X_{min}, Y_{min}, X_{max}, Y_{max})$ among all the locator positions heard.
3: Set the search area as the rectangle $(X_{min} - R, Y_{min} - R, X_{max} + R, Y_{max} + R)$, where R is the radio range.
4: Partition the search area into grids.
5: **for** each beacon received **do**
6: Increase the value of a grid point by one if this grid point is within the sector defined in this beacon.
7: **end for**
8: Estimated Position = CenterOfGravity(the grid points with the largest values)

in psychometrics. MDS attempts to provide a visualization (2D or 3D) of the original data, and preserving the essential information present in the data set (i.e., similarities in a multidimensional space). The MDS-MAP algorithm uses the classical metric scaling, the simplest case of the MDS technique, which has a closed form solution, enabling a relatively efficient computation (requires no iterations).

An important concept for MDS is how to compute the distance between two points. If we denote by $\mathbf{X}$ the matrix of coordinates of points ($n \times m$ matrix of n points in m dimensions), and by $\mathbf{D} = [d_{ij}]$ the matrix of distances between points, it can be shown that the matrix of squared distances between points, i.e., $\mathbf{D}^{(2)}$, can be written as follows:

$$\mathbf{D^{(2)}} = \mathbf{c1'} + \mathbf{1c'} - 2\mathbf{XX'} = \mathbf{c1'} + \mathbf{1c'} - 2\mathbf{B}$$

where $\mathbf{c}$ is a vector with elements the diagonal elements of $\mathbf{XX'}$. The matrix $\mathbf{B} = \mathbf{XX'}$ is the scalar product matrix. So the questions becomes, given the matrix of squared distances $\mathbf{D}^{(2)}$ how can one obtain $\mathbf{B}$, and implicitly $\mathbf{X}$? It can be shown [1] that by double-centering $\mathbf{D^{(2)}}$, one can obtain $\mathbf{B}$:

$$-\frac{\mathbf{1}}{\mathbf{2}}\mathbf{JD^{(2)}J} = \mathbf{B}$$

where $\mathbf{J} = \mathbf{I} - \mathbf{11'}/n$ (called the centering matrix), $\mathbf{I}$ the identity matrix and $\mathbf{1}$ the n-dimensional column vector with all elements one. Once $\mathbf{B}$ is obtained, the coordinates $\mathbf{X}$ can are computed by eigendecomposition.

The steps of classical scaling are summarized as follows:

1. Compute the squared distances matrix $\mathbf{D^{(2)}} = [d_{ij}^2]$
2. Double-center the $\mathbf{D^2}$ matrix:

$$\mathbf{B} = -\frac{1}{2}\mathbf{JD^{(2)}J}$$

3. Compute the singular value decomposition of $\mathbf{B} = \mathbf{VAV^T}$
4. Compute the coordinate matrix:

$$\mathbf{X} = \mathbf{V}_+ \mathbf{A}_+^{\mathbf{1/2}}$$

   where $\mathbf{A}_+$ is the matrix of the first m eigenvalues greater than zero and $\mathbf{V}_+$ the first m columns of V.

The MDS-Map algorithm that uses the classical metric scaling technique is shown in Algorithm 3.

Algorithm 3 MDS-MAP

1: Compute the shortest paths d_{ij}, $1 \leq i, j \leq n$. This gives the distances matrix $\mathbf{D}$.
2: Compute the relative positions (map), by applying classical MDS to the distance matrix $\mathbf{D}$, and retain the largest 2 (for a 2D space) or 3 (for a 3D space) eigenvalues and eigenvectors.
3: Transform the relative map, into an absolute map, based on the absolute positions of anchor nodes.

The main drawbacks of the MDS-MAP algorithm, the need for global information and centralized computation are addressed in subsequent work by the authors [27].

2.5 Gradient

In the Gradient algorithm, proposed by Nagpal et al. in [20], the anchor nodes initiate a gradient that self-propagates and allows a sensor node to infer its distance from the anchor. After estimating distances to three anchors a sensor node infers its own location through multilateration.

The steps of the algorithm are as follows:

- Each anchor node A_i initiates a flood of the network by broadcasting a packet containing its position and a counter with the initial value set to one.
- Each sensor node N_j keeps track of the shortest path (in terms of radio hop counts, h_{j,A_i}) to an anchor A_i from which it has received a beacon. A distance estimate, between the sensor node and anchor is obtained by:

$$d_{ji} = h_{j,A_i} d_{hop}$$

  where d_{hop} is the estimated Euclidian distance covered by one radio hop, and it is given by the Kleinrock-Silvester formula [11]:

$$d_{hop} = r \left(1 + e^{-n_{local}} - \int_{-1}^{1} e^{-\frac{n_{local}}{\pi}\left(\arccos t - t\sqrt{1-t^2}\right)} \right)$$

- Each node N_j computes its coordinates such that the total error is minimized:

$$E_j = \sum_{i=1}^{n} (d_{ji} - \hat{d_{ji}})$$

where $d_{ji} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ and $\hat{d_{ji}}$ is the estimated distance computed through gradient propagation, as shown above.
The coordinate are incrementally updated in the following way:

$$\Delta x = -\alpha \frac{\partial E_j}{\partial x_j} \text{ and } \Delta y = -\alpha \frac{\partial E_j}{\partial y_j}$$

where:

$$\frac{\partial E_j}{\partial x_j} = \sum_{i=1}^{n} (x_j - x_i) \left(1 - \frac{d_{ji}}{\hat{d_{ji}}}\right) \text{ and } \frac{\partial E_j}{\partial x_j} = \sum_{i=1}^{n} (x_j - x_i) \left(1 - \frac{d_{ji}}{\hat{d_{ji}}}\right)$$

Sources for errors in location estimation arise in the Gradient scheme from: incorrect estimation of the one-hop distance (d_{hop}), and the multilateration procedure.

2.6 Ad-Hoc Positioning System

In a similar manner with the Gradient method, the Ad-Hoc Positioning System (APS) proposed by Niculescu and Nath [22], uses the hop-by-hop propagation of distances to known anchors (a set of anchors is assumed to be present in the WSN). After obtaining distance estimates to three or more anchors, a sensor node employs a multilateration (similar with that of GPS) for iteratively improving its location estimation. The main difference resides in how a sensor node N_j estimates its distance to an anchor A_i (d_{ji} from the Gradient method, presented before).

The steps of the APS localization scheme algorithm are the following:

- Each anchor node A_i initiates a flood of the network by broadcasting a packet containing its position and a counter with the initial value set to one.
- Each sensor node N_j keeps track of the shortest path (in terms of radio hop counts, h_{j,A_i}) to an anchor A_i from which it has received a beacon. In [22] the authors propose four methods for propagating the distances from anchors to sensor nodes: DV-Hop, DV-Distance, Euclidian and Coordinate. The method that does not assume ranging, DV-Hop is described below. An example of the DV-Hop scheme is shown in Figure 7. At the end of this phase, node N_j knows that it is 3 hops, 2 hops and 1 hop from A_1, A_2 and A_3, respectively.

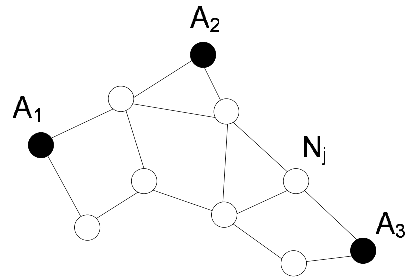

Fig. 7. The DV-Hop localization scheme

- Once an anchor node A_i obtains distances to other anchors, it computes a correction factor c_i (the estimated 1 radio hop Euclidian distance), which it propagates in the network. Corrections are propagated through controlled flooding, i.e., after a node receives and forwards the first correction, it will stop forwarding subsequent corrections. The correction factor is computed as follows:

$$c_i = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_i}$$

  for all anchors $A_j \neq A_i$ from which it has received a beacon (anchor A_j is positioned at (x_j, y_j) and h_i is the number of hops between the sensor node and anchor A_i).
  For the example shown in Figure 7, if the distances $\overline{A_1A_2}$, $\overline{A_2A_3}$ and $\overline{A_3A_1}$ are 30m, 40m and 50m, respectively, the correction factor for anchor A_3 is: c_3=(50+40)/(4+3)=12.9m/hop.
- A least square method (the authors used the Householder method) is employed for solving the non-linear system of equations:

$$\begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \Delta\rho_3 \\ ... \\ \Delta\rho_n \end{bmatrix} = \begin{bmatrix} \hat{1}_{1x} & \hat{1}_{1y} \\ \hat{1}_{2x} & \hat{1}_{2y} \\ \hat{1}_{3x} & \hat{1}_{3y} \\ ... & ... \\ \hat{1}_{nx} & \hat{1}_{ny} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

  where $\Delta\rho_i = \hat{\rho}_i - \rho_i$, $\hat{\rho}_i$ and ρ_i are the estimated and the real distances between a sensor node and an anchor A_i, $\hat{1}_{ix}$ is the unit vector of $\hat{\rho}_i$ in the x direction and Δx and Δy are the corrections in the position estimate for the node N_j.
  For the example shown in Figure 7, the estimated distances between node N_j and anchors A_1, A_2 and A_3 are $\hat{\rho}_1$=4*12.9=51.6m, $\hat{\rho}_2$=38.7m and $\hat{\rho}_3$=12.9m.

2.7 Probability Grid

In a similar manner with the DV-Hop, a localization scheme that can be used in scenarios where the topology of deployment is known a priori to be a grid, is proposed in [31].

The steps of the localization scheme are the following:

- Each anchor node A_m initiates a flood of the network by broadcasting a packet containing its position and a counter with the initial value set to one.
- Each sensor node N_k keeps track of the shortest path (in terms of radio hop counts) to each of the anchors A_l from which it has received beacons.
- Once an anchor node A_m obtains distances to other anchors, it computes a correction factor c_m (the estimated radio range), and it propagates it in the network.
- After receiving hop-count estimates to three or more anchor nodes, and a correction factor c_m a sensor node N_k evaluates the probability of being located at any position in the grid (labeled (i, j)). For this, it computes an expected hop count:

$$\lambda = d_{(i,j),l}/c_m$$

where $d_{(i,j),l}$ is the Euclidian distance between anchor A_l and the point (i, j) being evaluated. It then computes the probability of it (node N_k) to be positioned at (i, j):

$$p_{k,(i,j)} = \prod_{l=1}^{|A|} P_{(i,j)}^{h_{k,l}}$$

where $P_{(i,j)}^{h_{k,l}}$ is the probability of node N_k, positioned at (i, j), to be $h_{k,l}$ hops from anchor A_l.
- A node N_k chooses as its location, the position in the grid (i, j) with the maximum probability $p_{k,(i,j)}$.

The authors make the observation that $h_{k,l}$ is a discrete random variable that represents the number of radio hops between one anchor and the point of interest, i.e. (i, j). The main features that the distribution function for $h_{k,l}$ needs to exhibit are: to have one parameter λ (defined above) , to be narrow and skewed positively for small values of λ and become broader and relatively symmetric for larger values of λ. This is illustrated in Figure 8:

These requirements follow the intuition that for smaller values of the parameter λ (i.e., grid points closer to the anchor) the number of hops (call it τ) has a limited range of possible values with higher and higher values being less and less probable (positively skewed). As the distance between the anchor and the node increases (λ increases), the number of possibilities for the hop count (τ) increases and the distribution becomes bell-shaped, i.e., smaller and larger

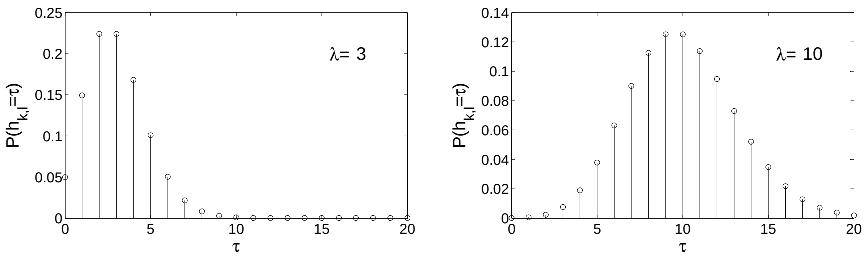

Fig. 8. The intuition behind the PMF of $h_{k,l}$

hop counts are equally probable. The authors found through simulations that a Poisson distribution is a good approximation for the $h_{k,l}$ discrete random variable. The distribution is given by:

$$P_{(h_{(k,l)}=\tau)} = \frac{\lambda^{\tau-1}e^{-\lambda}}{(\tau-1)!}$$

where $\tau = 1, 2, ...$

3 Anchor-Free Solutions

Anchor-free localization schemes exploit the proximity to an event with a known location: a light event in [30] [24] or a nearby radio packet in [29]. One common characteristic for these schemes is the moving of the complexity (hardware and computational, associated with an accurate localization) from the sensor node to a central, more sophisticated device. By controlling well the spatio-temporal properties of the events (light and radio packets), a much higher accuracy in localization (when compared with the anchor-based schemes) can be obtained. While anchor nodes are not required for any of the following schemes, anchor nodes can be beneficial for extensions of the proposed schemes.

3.1 Spotlight

The main idea of the Spotlight localization system [30] is to generate controlled events in the field where the sensor nodes are deployed. An event could be, for example, the presence of light in an area. Using the time when an event is perceived by a sensor node and the spatio-temporal properties of the generated events, spatial information (i.e. location) regarding the sensor node can be inferred. The system architecture for the Spotlight localization system is shown in Figure 9.

With the support of these three functions, the localization process goes as follows:

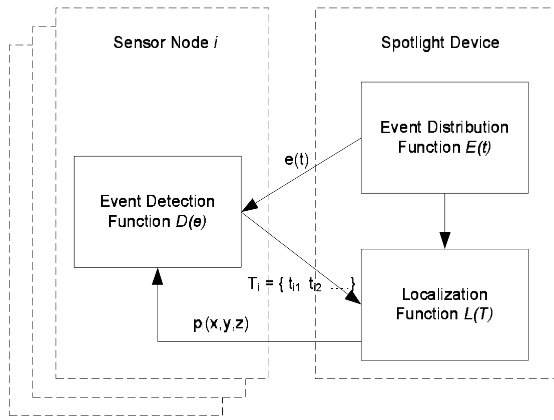

Fig. 9. Spotlight system architecture

- A Spotlight device distributes events $e(t)$ in the space A over a period of time.
- During the event distribution, sensor nodes record the time sequence $T_i = \{t_{i1}, t_{i2}, ..., t_{in}\}$ at which they detect the events.
- After the event distribution, each sensor node sends the detection time sequence back to the Spotlight device.
- The Spotlight device estimates the location of a sensor node i, using the time sequence T_i and the known $E(t)$ function.

The Event Distribution Function $E(t)$ is the core technique used in the Spotlight system and the authors propose three designs for it, with different tradeoffs/costs. These designs are presented below.

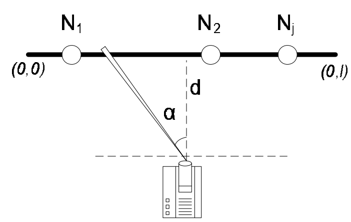

Fig. 10. The implementation of the Point Scan EDF

Point Scan

The Point Scan EDF is applicable to a simple sensor system where a set of nodes are placed along a straight line ($A = [0, l] \subset R$). The Spotlight device generates point events (e.g., light spots) along this line with constant speed s, as shown in Figure 10. The set of timestamps of events detected by a node i is $T_i = \{t_{i1}\}$. The Event Distribution Function $E(t)$ is:

$$E(t) = \{p \mid p \in A, p = t * s\}$$

where $t \in [0, l/s]$. The resulting localization function is:

$$L(T_i) = E(t_{i1}) = \{t_{i1} * s\}$$

Line Scan

Some devices, e.g. lasers, can generate an entire line of events simultaneously. With these devices, the Line Scan Event Distributed Function can be supported. Assuming that the sensor nodes are placed in a two dimensional plane ($A = [l \times l] \subset R^2$) and that the scanning speed is s. The set of timestamps of events detected by a node i is $T_i = \{t_{i1}, t_{i2}\}$.

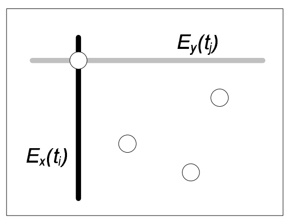

Fig. 11. The implementation of the Line Scan EDF

The Line Scan EDF, depicted in Figure 11, is defined as follows:

$$E_x(t) = \{p_k \mid k \in [0, l], p_k = (t * s, k)\} \texttt{ for } t \in 0, l/s$$

$$E_y(t) = \{p_k \mid k \in [0, l], p_k = (k, t * s - l)\} \texttt{ for } t \in l/s, 2l/s$$

and $E(t) = E_x(t) \cup E_y(t)$.

The location of a node can be calculated from the intersection of the two event lines, as shown in Figure 11. More formally:

$$L(T_i) = E(t_{i1}) \cup E(t_{i2})$$

Area Cover

Other devices, such as light projectors, can generate events that cover an area. This allows the implementation of the Area Cover EDF. The idea of Area Cover EDF is to partition the space A, where the sensor nodes are deployed, into multiple sections and assign a unique binary identifier, called code, to each section. Let's suppose that the localization is done within a plane ($A \in R^2$). Each section S_k within A has a unique code k.

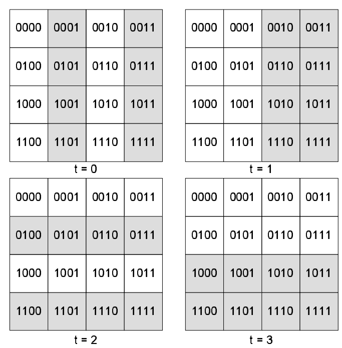

Fig. 12. The implementation of the Area Cover EDF

The Area Cover EDF, with its steps shown in Figure 12 is then defined as follows:

$$BIT(k,j) = \begin{cases} \text{true if } j^{th} \text{ bit of } k \text{ is } 1 \\ \text{false if } j^{th} \text{ bit of } k \text{ is } 0 \end{cases}$$

$$E(t) = \{p \mid p \in S_k, BIT(k,t) = true\}$$

and the corresponding localization algorithm is:

$$\begin{aligned} L(T_i) = \{p \mid p = COG(S_k), BIT(k,t) = true \ \texttt{if} \ t \in T_i, \\ BIT(k,t) = false \ \texttt{if} \ t \in T - T_i\} \end{aligned}$$

where $COG(S_k)$ denotes the center of gravity of S_k.

Cost Comparison

Although all three proposed techniques are able to localize the sensor nodes, they differ in the localization time, communication overhead and energy consumed by the Event Distribution Function (call it Event Overhead). Assume that all sensor nodes are located in a square with edge size D, and that the Spotlight device can generate N events (e.g. Point, Line and Area Cover events) every second and that the maximum tolerable localization error is r. Table 1 presents the execution cost comparison of the three different Spotlight techniques.

Criterion	**Point Scan**	**Line Scan**	**Area Cover**
Localization Time	$D^2/r^2)/N$	$D^2/r^2)/N$	$\log_r(D/N)$
# Detections	1	2	$\log_r D$
# Time Stamps	1	2	$\log_r D$
Event Overhead	D^2	$2D^2$	$D^2 \log_r(D/2)$

Table 1. Execution Cost Comparison

Table 1 indicates that the Event Overhead for the Point Scan method is the smallest - it requires a one-time coverage of the area, hence the D^2. However the Point Scan takes a much longer time than the Area Cover technique, which finishes in $\log_r D$ seconds. The Line Scan method trades the Event Overhead well with the localization time. By doubling the Event Overhead, the Line Scan method takes only $r/2D$ percentage of time to complete, when compared with the Point Scan method. From Table 1, it can be observed that the execution costs do not depend on the number of sensor nodes to be localized. It is important to remark the ratio "Event Overhead"/"Localization Time", which is indicative of the power requirement for the Spotlight device. This ratio is constant for the Point Scan (r^2N) while it grows linearly with area, for the Area Cover ($D^2N/2$). If the deployment area is very large, the use of the Area Cover EDF is prohibitively expensive, if not impossible. For practical purposes, the Area Cover is a viable solution for small to medium size networks, while the Line Scan works well for large networks.

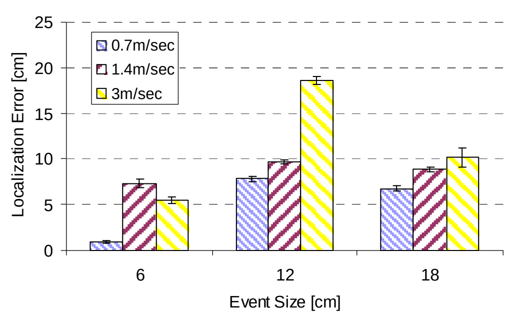

Fig. 13. Localization Error vs. Event Size for Spotlight system.

For the Spotlight system evaluation, the authors deployed 10 XSM [6] motes in a football field. The Spotlight device consisted of diode lasers, a computerized telescope mount, connected to a laptop. The Event Distribution Function investigated was the Point Scan. The range between the Spotlight device and the sensor nodes was approximately 170m.

Figure 13 shows the average localization errors versus the size of the event (diameter of the laser beam, on the ground), for different scanning speeds s. Localization errors of 10-20cm are reported.

3.2 Lighthouse

In a similar way to the Spotlight localization system, the Lighthouse scheme, proposed by Römer [24], makes use of the free-space optical channel between a device (called Lighthouse in this case) and sensor nodes.

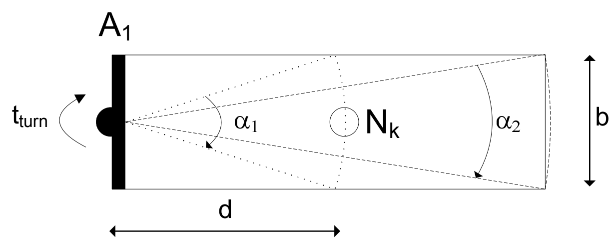

Fig. 14. Lighthouse Localization

The main idea of the Lighthouse system is exemplified in Figure 14. A parallel light beam of width b, emitted by anchor A_1 rotates with a certain period t_{turn}. A sensor node N_k detects this light beam for a period of time t_{beam}, which is dependent on the distance d between the Lighthouse device and the sensor node, in the following way:

$$d = \frac{b}{2\sin(\alpha_1/2)} = \frac{b}{2\sin(\pi t_{beam}/t_{turn})}$$

From measuring t_{beam} and knowing b and t_{turn}, one can compute the distance between the sensor node and the lighthouse device d. By constructing a device with three mutually perpendicular light emitting Lighthouses, a 3D location can be obtained.

The main difficulty encountered by the authors in the implementation of the Lighthouse prototype is ensuring that the light beam is perfectly parallel (zero divergence), having a width b. Instead, two laser beams of widths b_i and angle orientations β_i, γ_i and δ_i $i = 1, 2$, are used. To account for the misalignments, the authors develop a better approximation for the resulting beam width b:

$$b \approx C^b + \sqrt{d^2 + h^2}C^\beta + hC^\gamma + dC^\delta$$

where $C^b = b_1 + b_2$, $C^\beta = \sin\beta_1 + \sin\beta_2$, $C^\gamma = \tan\gamma_1 + \tan\gamma_2$ and $C^\delta = \sin\delta_1 + \sin\delta_2$. These parameters are constant for a particular Lighthouse system, and they are obtained through calibration, by localizing four points with known locations.

The experiments use 22 nodes placed in a 5x5m^2 area, with the Lighthouse device positioned at the coordinate (0,0). The accuracy of the localization algorithm is presented relative to the distance between the Lighthouse device and the sensor node (i.e., $|\hat{x} - x|/x$). The mean relative error (difference between the computed location and ground truth) in localization is 1.1% in one direction and 2.8% in the second direction (the difference is attributed to the calibration). This translates in localization errors of a few centimeters.

3.3 Walking-GPS

In many applications it is envisioned that WSN will be deployed from Unmanned Aerial Vehicles. In the meantime, manual deployments have been prevalent and the employed localization solutions have used some variant of associating the sensor node ID with prior knowledge of that ID's position in the field.

In [29] the authors propose a solution, called Walking GPS, in which the deployer (either person or vehicle) carries a GPS device that periodically broadcasts its location. The sensor nodes being deployed, infer their position from the location broadcast by the GPS device. The proposed solution is simple, cost effective and has very little overhead.

In the Walking GPS architecture the system is decoupled into two software components: the GPS Mote and the Sensor Mote. The GPS Mote runs on a Mica2 mote. The mote is connected to a GPS device, and outputs its location information at periodic intervals. The Sensor Mote component runs on all sensor nodes in the network. This component receives the location information broadcast by the GPS Mote and infers its position from the packets received. The proposed architecture pushes all complexity derived from the interaction with the GPS device to a single node, the GPS Mote, and to significantly reduce the size of the code and data memory used on the sensor node. Through this decoupling, a single GPS Mote is sufficient for the localization of an entire sensor network, and the costs are thus reduced.

A relatively simple design for the GPS Mote would have been to periodically broadcast the actual GPS location received from the GPS device. In order to reduce the overhead incurred when exchanging data containing global GPS coordinates, the Walking GPS system uses a local, Cartesian, coordinate system. The conversion between coordinate systems is performed by the GPS mote. A local coordinate system of reference is better suited for WSN, than a global coordinate system.

The localization scheme that makes use of the Walking GPS solution has two distinct phases:

1. The first phase is during the deployment of the sensor nodes. This is when the Walking GPS solution takes place. The deployer has a GPS-enabled mote attached to it; the GPS-enabled mote periodically beacons its location; the sensor nodes that receive this beacon infer their location based on the information present in this beacon.
2. The second phase is during the system initialization. If at that time, a sensor node does not have a location, it asks its neighbors for their location information. The location information received from neighbors is used in a triangulation procedure by the requester, to infer its position. This second phase enhances the robustness of the scheme.

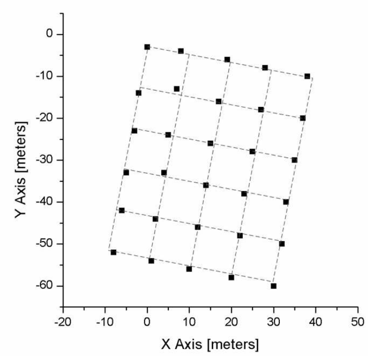

Fig. 15. Walking GPS system evaluation. Nodes deployed in a grid.

The experimental evaluation of the entire system, consisted of 30 MICA2 motes that were deployed in a 5x6 grid (for ease of measuring the localization error). The experimental results are shown in Figure 15. The average localization error obtained from fitting a grid to the experimental data is 0.8m with a standard deviation of 0.5m.

4 Open Problems

4.1 Security

Recently, several research groups have started to address robust and secure localization. For example, SeRLoc [15] demonstrates robustness against wormhole, Sybil and compromise of network nodes attacks. However, this work assumes a particular two-tier architecture and special hardware they call locators. In addition, the jamming of the wireless medium is not considered. It is an excellent start, but a lot more needs to be done especially for military domains and to meet various reality assumptions.

For securing localization, robust statistical methods (e.g. least median square) have been proposed [16]. The assumption is that an attacker selectively alters distance estimates to known anchor locations. The highest contamination ratio (i.e., affected readings) that the mathematical model supports is 50%, with significant degradation even at 35%. If an attacker possesses the capability of affecting distance estimates easily, then it is very likely that all distance estimates will be affected, making this set of solutions, less effective. The idea, however, of using robust statistical models, is a very good one.

In a similar approach, in [17], two solutions are proposed for secure localization: an attack resistant minimum mean square error (MMSE) which suffers from an unbounded localization error (the attack can result in an arbitrarily large localization error), and a voting-based scheme, which corrects the unbounded localization error, at a higher computational and storage cost.

Distance bounding has also recently been proposed as a technique for secure verification of localization. In [25] a combination of ultra-sound and radio communication is used for bounding the location claimed by a node, to a region. In this region, called region of interest, a set of trusted verifiers has to exist. This scheme is robust against attackers that can not be physically present in the region of interest. Similarly, [4] proposes a Verifiable Multilateration, that also relies on the distance bounding technique. The basic idea is to use the Time of Flight (ToF) of radio communication. Since the speed of light can not be exceeded, the location to be verified can not be closer than it actually is. It can only be further. However, claiming a longer distance would require a shorter distance to an even further positioned verifier. The main drawback is the hardware requirements (with nanosecond accuracy) imposed on the sensor nodes. In addition, [4] requires a relatively large number of anchor nodes.

In order to address some of the deficiencies of SeRLoc (e.g. jamming is not considered) and the Verifiable Multilateration (e.g., relatively high number of anchor nodes), a new scheme is proposed in [14]. This scheme can be used for both, location determination and location verification. The main idea is to fully utilize the strengths of both solutions: SeRLoc's use of sectored antennas and the distance bounding properties of Verifiable Multilateration. The deficiencies of both schemes, are still present.

The most recent effort on secure localization [5], attempts to depart from the aforementioned, "traditional", approaches, which require high speed hardware, sectored antennas or statistics, with a limited robustness. The idea is to use covert, hidden base stations (their position is known only to an authority), in addition to the "public", known base stations. The role of covert base stations is to perform TDoA (between radio and ultra-sound) ranging and verify the location computed and claimed by a node. For the effectiveness of this solution, the covert base stations communicate with a central location verification authority either in a wired manner or infra-red, to reduce the risk of being detected by the attacker. The authors also propose mobile base station assisting with the verification of location. While this direction for secure localization is novel, in its current form, has demanding requirements for the infrastructure of covert base stations.

4.2 Impact of Localization on Protocols

In localization for WSN, achieving better results (usually with regard to location accuracy) requires increasing the relative cost of the localization scheme via additional hardware, communication overhead, or the imposition of constraints and system requirements. Although more accurate location information is preferable, the desired level of granularity should depend on a cost/benefit analysis of the protocols that utilize this information. In this section, we investigate the impact of localization error on other communication protocols and proposed sensor network applications. Designers of sensor network systems with certain performance requirements can use this analysis to aid in their architectural design and in setting system parameters. Although requirements are expected to vary between deployments, we found that in the general case for the protocols studied, performance degradation is moderate and tolerable when the average localization error is less than 0.4R.

Routing Performance

A localization service is critical for location-based routing protocols such as geographic forwarding (GF) [21], [10], [12] and [34]. In these protocols, individual nodes make routing decisions based on knowledge of their geographic location. While most work in location-based routing assumes perfect location information, the fact is that erroneous location estimates are virtually impossible to avoid. Problems arise as error in the location service can influence location-based routing to choose the best next hop (the neighbor closest to the destination), or can make a node inadvertently think that the packet could not be routed because no neighbors are closer to the final destination.

To investigate the impact of localization error on routing, the authors of [8] studied the GF [21] routing protocol under the low traffic network conditions so that network congestion does not influence the results. The baseline was the perfect localization, the protocol where every sensor node knows its correct physical location.

Figure 16 shows the delivery ratio (the percentages of packets that reach destination over all packets sent) with regard to node density for various levels of location error. From this graph, we see that for average localization errors of 0.2R and 0.4R, the delivery ratios of GF are very close to the baseline (no error). Beyond these numbers, the results diminish with increased error; a trend that could be problematic and costly depending on the implemented architecture, reliability semantics, tolerance of message loss, and application requirements. For example, when localization error is the same as the node radio range, even with high node density (20 nodes per radio range), the delivery ratio still falls below 60%.

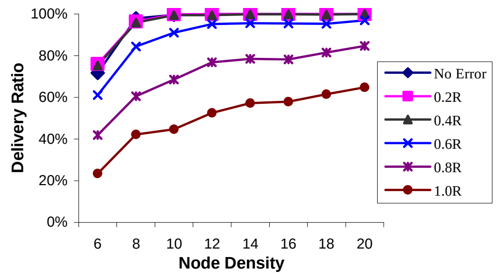

Fig. 16. Delivery ratio with different localization errors, changing node density

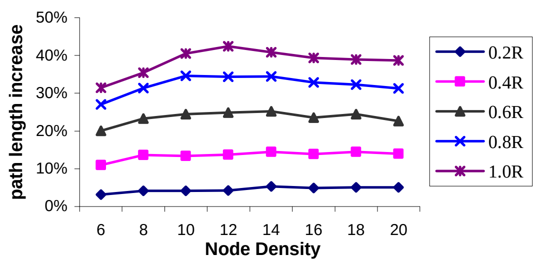

Fig. 17. Path length overhead with different localization errors under varying node density

Another metric affected by localization error is the route path length. Figure 17 measures the hop count increase (in percentage) due to location error to assess the cost in communication overhead of this error. We see from this graph that for low localization error (less than 0.4R), this routing overhead remains moderate (less than 15%). However, as was the case for the delivery ratio metric, when localization error grows beyond 0.4R, the routing overhead increases to as high as 45%. We also note that this trend occurs regardless of the network node density, a fact that was not true for our previous metric. We acknowledge here that GF was chosen as a representative protocol, and an in depth study about localization's impact on various routing protocols and its implications on the design of location-dependent systems is future work.

Target Estimation Performance

Many of the most frequently proposed applications for WSN utilize target position estimations for tracking, search and rescue, or other means. In these proposed applications, when a target is identified, some combination of the nodes that sensed that target report their location to a centralized node (leader or base station). This node then performs aggregation on the received data to estimate the actual location of the target. Because target information could be used for locating survivors during a disaster, or identifying an enemy's position for strategic planning, the accuracy of this estimation is crucial to the application that uses it.

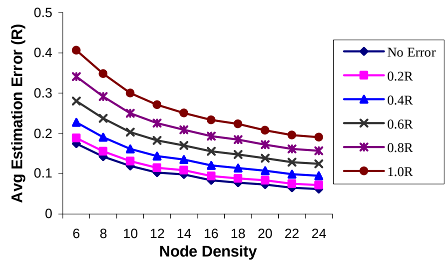

Fig. 18. Target estimation error with different localization errors under varying node density

Intuitively an increase in localization error directly leads to target estimation error. To better understand the degree to which this error propagates to other protocols, the authors of [8] investigate the average estimation error under different node densities for varying degrees of location error. For these experiments, a simple and widely used target estimation algorithm is used: the average X and Y coordinates of all reporting nodes are taken as the target location estimation. The sensing range is set to be equal to the node radio range so that the node density is equivalent to the average number of sensors involved in target estimation. The results of various experiments are depicted in Figure 18. This graph shows that target estimation error due to location error is dampened during the aggregation process. As before, the baseline occurs when no localization error exists. Aside from showing varying degrees of estimation error with respect to node location error, Figure 18 also shows that the absolute target estimation error decreases with increased node density. For example, when localization error is equal to 1.0R, and node density reaches 12 nodes per radio range, the estimation error is only about 67% as large as when the node density is 6 nodes per radio range. From this chart we see that more nodes participating in estimation results in more random estimation error being ameliorated through aggregation.

Object Tracking Performance

In [8], the authors further evaluate the performance of target estimation by simulating a tracking application that uses estimation in context. In this experiment, a mobile evader randomly walks around the specified terrain while a pursuer attempts to catch it. In this simple experiment, the pursuer is informed of the current location of the evader periodically via sensing nodes in the terrain that detect the evader, coordinate to estimate the targets position with regard to their own positions, and periodically report this result to the mobile pursuer. When receiving a report, the pursuer readjusts its direction in an attempt to intercept the evader. When the pursuer comes within the node communication radius of the evader, the evader is considered caught and the simulation ends. For this experiment, the average tracking time (the time from pursuer take-off to when the evader is caught) under different localization errors is compared to the tracking time in the case of no localization error. Figure 19 shows normalized tracking time in relation to the pursuer speed for various degrees of localization error.

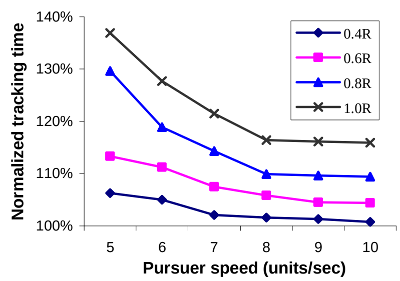

Fig. 19. Normalized tracking time with different localization errors varying pursuer speed. Terrain size 1000x1000m, Radio range = 40m, density = 8 nodes/radio range. Evader speed = 5m/s

From Figure 19 we see that the tracking time overhead decreases with increased pursuer speeds. More importantly, Figure 19 shows that the tracking time increases as localization error increases. This result implies that it is important for tracking applications with real-time requirements to take localization error into consideration. For example, when the average localization error is known to be 0.8R, and the pursuer speed is 5 units per second, the pursuer requires 30% more time in comparison to the ideal situation in which no localization error exists. To reduce this overhead to 10%, either the pursuers speed must be increased to 10 units per second, or the estimation error must be reduce to 0.4R. Again, Figure 19 shows that 0.4R is a tolerable bound for estimation error since tracking time only increases by 7% in the worst case.

4.3 Impact of Environment on Localization

The problem of range-free localization is further complicated by the diverse types of environments, where a WSN system can be deployed. Outdoor, real deployment environments very little resemble typical lab environments. Hence, issues like calibration, mobility (if nodes are "moved" by the environment, or the WSN is designed to be mobile), the lack of line-of-sight, the existence of obstructions and multipath effects often arise in realistic, outdoor environments.

Some preliminary work on the aforementioned issues are the following:

Calibration

Whitehouse formulates the calibration problem in WSN as a parameter estimation problem [33]. Each device in the WSN is parameterized and the values of the parameters are chosen such that the system performance is maximized (higher accuracy in location estimation). The author propose a macro-calibration procedure, called joint-calibration, that calibrates each device, by optimizing the overall system performance, instead of individual nodes. The steps of the joint calibration are the following:

1. Model the overall system, by using individual, device specific parameters.
2. Collection data.
3. Tune the parameters of individual devices, such that the overall system performance is improved.

The key insight into how to choose parameters to be tuned, such that the overall system performance improves is to look at trends in the transmitter/receiver pairs, and identify individual nodes for which the chosen parameters are problematic.

The proposed joint calibration is a good solution where manual calibration is possible. Obviously, in rugged, remote outdoor environments, auto-calibration (i.e., no manual intervention) is highly desirable.

Mobility

Hu and Evans [9] propose a sequential Monte-Carlo (SMC) localization algorithm for WSN in which sensor nodes and anchors are all mobile. The authors show that mobility can be used to enhance localization accuracy, a rather counterintuitive result - one would expect to be a significant impediment for an accurate positioning.

The proposed algorithm is an adaptation of the Sequential Monte Carlo localization scheme, frequently used in robot localization, target tracking and computer vision, to the domain of WSN. The main idea of the SMC localization algorithm is to represent the posterior distribution of possible locations using a set of weighted samples and to update them recursively in time.

From simulations of a 10Rx10R WSN, with an average number of nodes per transmission range of 10, the authors report localization errors of approximately 0.5R, when both sensor nodes and anchors move at a speed of R meters/sec. The localization error starts from high values (1.9R) and decreases rapidly, with the accumulation of new observations (nodes entering the ranges of new anchor nodes).

Line-of-Sight and Multipath

Real, outdoor environments pose significant challenges for range-free localization. Localization schemes designed and evaluated in "friendly" environments frequently fail to produce encouraging results in real deployments. When line-of-sight is a main assumption of the scheme [30], and does not always hold, or when obstructions and multipath for acoustic and radio waves are not considered [35], the performance of the localization scheme is degraded.

In order to address this, a potential direction to pursue is multimodal localization. In a multimodal localization system, more than one localization scheme is executed, in an attempt to reduce the impact the assumptions of a single localization scheme could have on the overall localization accuracy. By using Bayesian inference, and the knowledge (even if partial) obtained during the execution of one localization scheme, a finer, more accurate positioning can be obtained from the execution of subsequent localization schemes. For example, if a WSN is localized using the Line Scan scheme of the Spotlight system, described before, and due to some environmental conditions one of the two events created in the network is not detected (the Spotlight localization scheme fails to produce a location in this case), the knowledge gained from the detection of the other event can be used to initialize a subsequently executed localization scheme.

5 Conclusions

In this chapter we presented a suite of range-free localization schemes for WSN. We define ranging, in the context of sensor networks, as the ability of a sensor node to infer distances to its neighbor sensor nodes, either through localization specific hardware (e.g., ultrasound transceivers) or the strength of the received radio signal. Hence, the localization schemes presented here (i.e., range-free schemes) do not posses sophisticated hardware and do not rely on the received signal strength for inter-node ranging. The sensor nodes we consider have simple radio communication and sensing capabilities.

The taxonomy that we adopt for categorizing the range-free localization schemes is based on the (non)existence of an infrastructure of anchor nodes (i.e., at least three nodes, for a 2D localization, with known locations) in the WSN. An anchor-free localization scheme exploits the proximity to an event with a known location: a light event in Spotlight [30] and Lighthouse [24] and a nearby radio packet in Walking GPS [29].

One main observation is the high accuracy in localization of the anchor-free, event based, localization schemes, at a reduced, per node, cost. It is remarkable to obtain location accuracies of tens of centimeters, at zero dollar cost (if the sensor node is equipped with a photo sensor for the mission it was deployed for) and relatively low communication overhead (reduced energy cost). Characteristic to the anchor-free localization schemes, is the moving of the complexities associated with the localization from the sensor node to a capable, sophisticated device. While the cost of such device is not negligible, the possibility of its reuse make the event-based, anchor-free solutions very attractive. The anchor-free, event based, class of localization schemes seems a very promising direction for high accuracy, low cost localization in WSN.

Despite the extensive attention the range-free localization has received, several open problems remain. Among these are how to secure the radio communication and sensing channels that sensor nodes posses, how to make range-free localization more robust against attacks, node or protocol failures (possibly due to its strict assumptions), understand the impact of localization schemes on other protocols and how to design more robust, cost efficient, calibration techniques. The breadth and depth of all these issues present interesting opportunities for future research in the domain of range-free node localization in WSN.

References

1. Borg, I., and Groenen, P. *Modern Multidimensional Scaling.* Series in Statistics. Springer, 1997.
2. Bulusu, N., Heidemann, J., and Estrin, D. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine 7*, 5 (October 2000), 28–34.
3. Bulusu, N., Heidemann, J., and Estrin, D. Adaptive beacon placement. In *International Conference on Distributed Computing Systems (ICDCS)* (2001).
4. Capkun, S., and Hubaux, J. Secure positioning of wireless devices with application to sensor networks. In *IEEE Conference on Computer Communications (Infocom)* (2005).
5. Capkun, S., Srivastava, M., and Cagalj, M. Securing localization with hidden and mobile base stations. Tech. rep., NESL-UCLA, 2005.
6. Dutta, P., Grimmer, M., Arora, A., Bibyk, S., and Culler, D. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *International Symposium on Information Processing in Sensor Networks (IPSN)* (2005).
7. Fang, L., Du, W., and Ning, P. A beacon-less location discovery scheme for wireless sensor networks. In *IEEE Conference on Computer Communications (Infocom)* (2005).
8. He, T., Huang, C., Blum, B., Stankovic, J. A., and Abdelzaher, T. Range-Free localization schemes in large scale sensor networks. In *ACM International Conference on Mobile Computing and Networking (Mobicom)* (2003).
9. Hu, L., and Evans, D. Localization for mobile sensor networks. In *ACM International Conference on Mobile Computing and Networking (Mobicom)* (2004).
10. Karp, B., and Kung, H. T. Gpsr: Greedy perimeter stateless routing for wireless networks. In *ACM International Conference on Mobile Computing and Networking (Mobicom)* (2000).
11. Kleinrock, L., and Silvester, J. Optimum tranmission radii for packet radio networks or why six is a magic number. In *IEEE National Telecommunication Conference* (1978).

12. Ko, Y. B., and Vaidya, N. H. Location-aided routing (lar) in mobile ad hoc networks. In *ACM International Conference on Mobile Computing and Networking (Mobicom)* (1998).
13. Kwon, Y., Mechitov, K., Sundresh, S., Kim, W., and Agha, G. Resilient localization for sensor networks in outdoor environments. In *IEEE International Conference on Distributed Computing Systems (ICDCS)* (2005).
14. Lazos, L., Capkun, S., and Poovendran, R. Rope: Robust position estimation in wireless sensor networks. In *International Workshop on Information Processing in Sensor Networks (IPSN)* (2005).
15. Lazos, L., and Poovendran, R. SeRLoc: Secure range-independent localization for wireless sensor networks. In *ACM Workshop on Wireless Security (WiSe)* (2004).
16. Li, Z., Trappe, W., Zhang, Y., and Nath, B. Robust statistical methods for securing wireless localization in sensor networks. In *International Workshop on Information Processing in Sensor Networks (IPSN)* (2005).
17. Liu, D., Ning, P., and Du, W. Attack-resistant location estimation in sensor networks. In *International Workshop on Information Processing in Sensor Networks (IPSN)* (2005).
18. Maróti, M., Kusý, B., Balogh, G., Völgyesi, P., Nádas, A., Molnár, K., Dóra, S., and Lédeczi, Á. Radio interferometric geolocation. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)* (2005).
19. Moore, D., Leonard, J., Rus, D., and Teller, S. Robust distributed network localization with noisy range measurements. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)* (2004).
20. Nagpal, R., Shrobe, H., and Bachrach, J. Organizing a global coordinate system from local information on an ad hoc sensor network. In *International Workshop on Information Processing in Sensor Networks (IPSN)* (2003).
21. Navas, J. C., and Imielinski, T. Geographic addressing and routing. In *ACM International Conference on Mobile Computing and Networking (Mobicom)* (1997).
22. Niculescu, D., and Nath, B. Ad-hoc positioning system. In *IEEE Global Communications Conference (GLOBECOM)* (2001).
23. Priyantha, N. B., Balakrishnan, H., Demaine, E., and Teller, S. Mobile-assisted localization in wireless sensor networks. In *IEEE Conference on Computer Communications (Infocom)* (2005).
24. Römer, K. The lighthouse location system for smart dust. In *ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys)* (2003).
25. Sastry, N., Shankar, U., and Wagner, D. Secure verification of location claims. In *ACM Workshop on Wireless Security (WiSe)* (2003).
26. Savarese, C., Rabaey, J., and Langendoen, K. Positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX Annual Technical Conference* (2002).
27. Shang, Y., and Ruml, W. Improved mds-based localization. In *IEEE Conference on Computer Communications (Infocom)* (2004).
28. Shang, Y., Ruml, W., Zhang, Y., and Fromherz, M. P. J. Localization from mere connectivity. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)* (2003).

29. Stoleru, R., He, T., and Stankovic, J. A. WalkingGPS: A practical localization system for manually deployed wireless sensor networks. In *IEEE Workshop on Embedded Networked Sensors (EmNetS)* (2004).
30. Stoleru, R., He, T., Stankovic, J. A., and Luebke, D. A high-accuracy low-cost localization system for wireless sensor networks. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)* (2005).
31. Stoleru, R., and Stankovic, J. A. Probability grid: A location estimation scheme for wireless sensor networks. In *IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON)* (2004).
32. Vicaire, P., and Stankovic, J. A. Elastic localization. Tech. Rep. CS-2004-35, University of Virginia, 2004.
33. Whitehouse, K., and Culler, D. Calibration as parameter estimation in sensor networks. In *ACM Intenational Workshop on Sensor Networks and Applications(WSNA)* (2002).
34. Xu, Y., Heidemann, J., and Estrin, D. Geography-informed energy conservation for ad hoc routing. In *ACM International Conference on Mobile Computing and Networking (Mobicom)* (2001).
35. Zhang, J., Yan, T., Stankovic, J. A., and Son, S. H. Thunder: A practical acoustic localization scheme for outdoor wireless sensor networks. Tech. Rep. CS-2005-13, University of Virginia, 2005.