A Survey and Comparison on Localization Algorithms for Wireless Ad Hoc Networks

Saad Biaz

Department of Computer Science and Software Engineering
Auburn University
107 Dunstan Hall, Auburn University, AL, 36849-5347, USA
Email: sbiaz@eng.auburn.edu

Yiming Ji

Department of Computer Science and Software Engineering
Auburn University
107 Dunstan Hall, Auburn University, AL, 36849-5347, USA
Email: jiyimin@auburn.edu

Abstract: In wireless ad hoc networks, node's location information is useful for efficient routing and location-aware applications. This paper surveys this active area of research and presents a detailed comparison of most localization algorithms in literature. Hardware requirements, reliability, and accuracy are reviewed. The localization methods are evaluated and compared under the same network settings.

Keywords: Positioning, localization, precision, location discovery algorithm, wireless ad hoc networks.

Reference to this paper should be made as follows: Biaz S. and Ji Y. (2004) 'A Survey and Comparison on Localization Algorithms for Wireless Ad Hoc Networks', *Int. J. Mobile Communications*, Vol. X, Nos. X, pp.X-X.

Biographical notes: Saad Biaz received his PhD in Computer Science from Texas A&M University in 1999. He is now an Assistant Professor in Computer Science and Software Engineering, Department at Auburn University. His research interests include wireless networking, mobile computing, distributed systems, and computer architecture.

Yiming Ji is currently a PhD student at the Department of Computer Science and Software Engineering at Auburn University.

1 Introduction

An ad hoc wireless network is a self-organized and rapidly deployable network. It usually consists of a large number of unattended nodes that autonomously construct a network and communicate with each other over multi-hop paths. The unattended nature of the system necessitates a set of mechanisms to assure simple deployment while maintaining the self-organized nature.

Recently, location-based services have gained considerable attention from academia and industry. Current reported applications and services include coverage analysis [1, 2], location-aware applications [3-8], environmental monitoring [9,10], target tracking [11,12], battlefield surveillance [13], and intrusion detection [14]. In addition, location management and efficient routing protocols also require location knowledge [15-20].

To obtain the location information, the Global Positioning System (GPS) [21,22] could be used. However, it can not be used for in-door applications and is inappropriate for large scale ad hoc networks because of constraints on volume, power consumption, and cost.

Alternatively, distributed localization algorithms enable nodes to determine their *relative* positions automatically after deployment. If *absolute* positions are necessary, a limited number of nodes (called *anchors*) must have the capability of determining their absolute positions. Anchors obtain their absolute position using manual configuration or techniques such as GPS. Much localization research has been carried out using this approach [23-26].

Most of existing algorithms take advantage of mature micro-sensor technology. Time of arrival (TOA) measures the radio signals traveling time, and time difference of arrival (TDOA) measures the time difference of the radio signals arrived at various antennas. These are two common methods for range estimation and have been applied in many infrastructure based systems [27, 28] and infrastructure free sensor networks [29-31]. Additionally, angle of arrival (AOA) has been proposed to estimate relative angles between neighbors [32]. A good survey of these distance measurement systems can be found in [33, 34]. Still, a different method of measuring the range is through received signal strength indicator (RSSI). RSSI does not require extra equipment and is widely supported by current transceivers. Consequently, RSSI is commonly adopted by many localization systems [23, 25, 35, 36].

Based on the reliance on the hardware support, localization algorithms can be classified into two main categories: range-based algorithms and range-free algorithms. Range-based algorithms rely more on hardware support by applying either one or a combination of TOA, TDOA, AOA, or RSSI technologies. On the contrary, range-free algorithms require less or no hardware support [23, 24, 36-38].

Absolute positions are necessary for many applications. But some applications require only the knowledge of nodes' relative position (without actual physical or absolute position information). This means that, at least for some applications, the position information can be relative instead of absolute [39, 40]. Consequently, the localization algorithms can also be categorized as absolute and relative.

To initialize the positioning process, different methods are used by diverse localization algorithms. A common approach for absolute localization algorithms is to let anchors broadcast their position information to the whole network. On the contrary, relative localization algorithms usually have the interested nodes, who want to find an optimal routing path to the destination nodes, to first broadcast the hello packets to their neighbors. Through communication, neighboring nodes can measure the distance between themselves. Based on these distances, all nodes can estimate their relative positions with each other through distributed localization algorithms. In the presence of anchor nodes, absolute positions of all nodes can be derived accordingly.

The fraction of anchors within a network and their positions are critical to the precision and performance of localization algorithms. Bulusu *et al.* [41] proposed *self-configuring* nodes into anchors in order to optimize the performance of localization algorithms. The main purpose of the *self-configuring* is to select the most appropriate nodes that must act as anchors in order to achieve efficient and robust localization for unattended, scalable sensor networks under various system and environment. The method consists of two steps. First, it estimates all nodes' physical positions. Second, it analyzes the local node's density within the network. Based on the nodes' density, Bulusu uses either *HEAP* or *STROBE*: for sparse deployments, the *HEAP* algorithm is used to find candidate positions for additional anchor deployments; and for dense deployments, the

STROBE algorithm enables anchors to cooperatively switch on and off, with the aim of less interference and optimal energy conservation.

Most localization algorithms assume certain capabilities of the node in the network.

Range-based methods use micro-sensors to measure time, relative angle, or energy in order to get range estimation. Range-free methods assume anchor nodes can transmit radios with various power levels or with higher power compared with normal nodes. By default, all algorithms assume the sensors can measure signals without system bias, or at most with random errors that can be minimized by smoothing or averaging. Notice that calibration can be an exceptionally difficult task for large ad hoc networks, Bychkovskiy *et al.* [42] provided a collaborative calibration method for dense networks. The main idea is to minimize, by utilizing redundancy information, the systematic errors by calibrating one sensor's output against the output from others. Though it is a relative calibration method, we believe it can be useful for the distributed localization algorithms considered in this paper.

The main contribution of this paper is to present a survey of distributed localization algorithms with a focus on their constraints, advantages, and limits. In addition, we compare and improve a set of representative algorithms. We hope this paper will provide a better understanding of the activities in this research field.

The remainder of the paper is organized as follows: In Section 2, we introduce the range-based algorithms. In Section 3, we present range-free algorithms. Section 4 provides relative localization algorithms. Density analysis will be given in Section 5. In Section 6, we present the estimation precision from original research of all algorithms. Section 7 compares a set of representative algorithms under same network environment. Section 8 concludes the report and outlines future research.

2 Range-based algorithms

Range-based localization algorithms assume all nodes can measure relative distance (or angle) between directly connected nodes in the network. These algorithms mainly focus on static networks with a set of anchor nodes having a priori knowledge of their own positions.

Generally, the range-based localization algorithms contain three steps:

1. Through broadcast, each node measures the pair-wise distance (or angle) with each other, and furthermore it can estimate the distance to anchor nodes;
2. Each normal node derives a position based on the anchors' position information;
3. Iteratively refine the position estimation with the constraints of neighbors' information.

Various localization algorithms may use different communication methods, and apply special estimation approaches under unique constraints. Doherty *et al.* provided *Convex Position Estimation* method [43] and proposed a set of *convex constraint models*. The constraint models are general guidelines for all other algorithms [26,36].

Specially, the range-based algorithms can be classified into these categories:

1. Convex positioning by Doherty *et al.* [43];
2. Ad-hoc positioning system by Niculescu *et al.* [29, 32, 39];
3. N-hop multilateration or AHLoS system by Savvides *et al*. [26, 30, 31, 44, 45];
4. Robust Positioning by Savarese [44, 45];
5. Probabilistic approach by Ramadurai *et al*. [36, 46];
6. Multidimensional Scaling by Ji *et al*. [25, 47], and it advanced derivatives by Biaz *et al.* [48].

Note that the first approach is a centralized algorithm. However, the convex constraint models are general and useful for other algorithms. We will start from the constraint models and introduce all algorithms accordingly. Langendoen *et al.* made an excellent comparison in [49] for methods 2, 3, and 4, so we will not explore these methods in detail.

In the following sections, we will first introduce various range measurement techniques, and then present the localization algorithms in details.

2.1 Distance measurement technologies

This section will review existing distance measurement technologies. Based on the applied hardwares and the operating environment, the distance measurement technologies can be categorized into the following two main categories:

1. Satellite positioning technologies, which include Global Position System (GPS), Differential GPS (DGPS), Assisted GPS (A-GPS), Wide Area Augmentation System (WAAS), and Galileo system (European version of GPS);
2. Network-based technologies, which include Time of arrival (TOA), Time difference of arrival (TDOA), Angle of arrival (AOA), Enhanced observed time difference (E-OTD), and Received signal strength indicator (RSSI).

In the following sections, we will briefly review these measurement techniques.

2.1.1 Satellite positioning technologies

• **Global Positioning System (GPS)**, GPS consists of three components: satellites, control stations, and GPS receivers [22].

The GPS system consists of 24 orbiting operational satellites, which can transmit very low power radio signals that allow any GPS receiver to determine its position on the Earth. Among the 24 satellites, 21 of them are active at any time, and the rest 3 are backups.

GPS control system monitors the satellites and providing them with correct orbital and clock data. There are 5 control stations around the world.

In order to determine the position, the receiver needs to know at least four satellites' location and the distance to them. The approximate location of the satellites is obtained from the satellites directly. The approximation can be adjusted by using data from control stations.

The distance from the receiver to the satellite is calculated as the product of radio speed and the radio travel time between them. The travel time is estimated from the difference of the 'pseudo-random' code generated at the same time by the satellite and the receiver. With the information of four or more satellites, the position of the receiver can be determined.

There are two levels of service provided by GPS. The more-accurate one is Precise Positioning Service (PPS), which is only available to authorized users and is intended for military usage. The less-precise service is Standard Positioning Service (SPS), which is available for all civil users worldwide without charge or restrictions. According to [50], the position estimation error for PPS is between 5-10m. The reported SPS horizontal accuracy by U.S. Department of Defense is of ±100m at 99.5% confidence level and ±300m at 99.9%. In [51], an error of ±185m in latitude and ±216m in longitude was reported.

• **Differential GPS (DGPS)**, **Assisted GPS (A-GPS) and Wide area augmentation system (WAAS).**

DGPS is an extension of the GPS system intended to improve the accuracy of the GPS system. DGPS uses stations to broadcast position correction beacons. With these correction messages, GPS receivers can correlate them with received satellite signals. This technology effectively reduces the effect of selective availability (SA) and propagation delay. It is reported that DGPS provides estimation with 2 to 10 meters accuracy [52, 53].

A-GPS is a technique that improves the functionality and performance of GPS using only GPS satellite signals. It works by integrating the classic GPS information with sophisticated geographic software and mobile/cellular network information. A-GPS is currently widely used in many systems, and the accuracy in system *U-Map* is within 5 meters [53].

WAAS is similar to DGPS and it is initially intended to be used for precision flight control by the Federal Aviation Administration (FAA) and the Department of Transportation. WAAS system consists of a network of approximately 25 ground reference stations that provide a very large service area including inland and offshore.

The tested system in GPS LAB at Stanford university met all three safety metrics (accuracy, integrity and continuity) at an availability of 99.671%, with Horizontal and Vertical Alert Limits (HAL and VAL) of 30 meters and 12~20 meters separately [54,55].

- **Galileo system,** Galileo is the European version of satellite navigation system, it is a system that both competes and complements with the American GPS system [56].

The fully deployed Galileo system will consist of 30 satellites, with 27 of them operational and the other three 3 active in reserve. With dual frequencies, Galileo delivers real-time positioning accuracy of meter range.

2.1.2 Network-based technologies

- **Time of Arrival (TOA)**: This technology works by measuring the arrival time of a known signal sent from a (mobile) node received at three or more measurement units. Synchronization of the measurement units is essential. Therefore, this method requires additional measurement unit hardware in the network at the geographical vicinity of the (mobile) node so as to accurately measure the TOA of the signal bursts. It is reported that the accuracy of TOA is about 100~200m [57].
- **Time Difference of Arrival (TDOA)**: It works similar as TOA in that both technologies utilize signal propagation time. In TDOA, however, two types of signals are selected so that their propagation speeds are significantly different. When a transmitter sends two types of signals simultaneously, the receiver can easily detect the difference in the time of arrival between the two types of signals. The time difference can then be used to compute the distance between the communication pairs. To deal with multipath problems, TDOA systems historically have been based on wideband radio technology. The known accuracy is about 100~200m.
- **Enhanced Observed Time Difference (E-OTD)**: In this method, it is the (mobile) node that performs the time measurement of beacon signals from nearby base stations. This method does not require synchronization in the network. The reported accuracy for E-OTD is 50~200m.
- **Angle of Arrival (AOA)**: This method requires special antenna arrays at the base station to determine the angle of the arriving signal from the mobile node. The angles from two or more base stations can determine the position of the node by the intersection of the arrival directions. The accuracy is 100~200m.
- **Received Signal Strength Indicator (RSSI):** This method measures the strength of the received signal in order to deduce the possible range the signal has propagated from

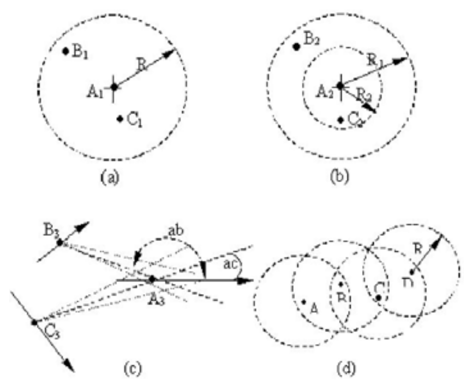
the sender to the receiver. It is applicable if the transmission power is constant or known in advance. It is a low accuracy localization method. However, it is simple and doesn't require extra equipment. Consequently, it is widely supported by most current wireless devices.

2.2 Radio communication constraint models

Radio communication constraint models are a set of geometry rules that can be used to bound position estimates. Generally, they may consist of a combination of the radial and angular constraints, and n-hop neighborhood constraints [23, 36, 43].

Radial constraint means that when a node *B* can hear a node *A*, the distance between them is less than *A*'s radio transmission range. When multiple transmission powers are used by an anchor node, other nodes that receive the radio signal from that anchor, may estimate their possible position by signal strength [23]. In Figure 1(a), nodes B_1 and C_1 are A_1's neighbors, the distance of A_1B_1 and A_1C_1 should be less than A_1's radio range *R*. In Figure 1(b), node A_2 can transmit radio in two different ranges R_1 and R_2. If node B_2 can be reached only with the larger power, then B_2 resides in the circular belt between R_1 and R_2.

Figure 1 Radio communication constraints



Angular constraint refers to the fact that when a node gets best reception at a certain angle, then it can estimate the relative angle to the source transmitter, which may be a cone [43]. A small group of neighbors may form a set of triangles with geometry constraints to further precise their position estimates [32]. In Figure 1(c), node A_3 can be determined with relative angles of *ac* and *ab*.

N-hop neighborhood constraint is a combination of individual constraints from neighbors. Satisfying this constraint will generally yield smaller feasible region for each node. The n-hop constraint is especially important for min-max boundary based and aggregation based methods like n-hop multilateration algorithm [26, 43] and APIT algorithm [24]. In Figure 1(d), nodes *A, B, C* and *D* are neighbors along one path from *A* to *D*. The distance between *A* and *D* will be less than *3R*. Node *B* resides in the radio intersection area of *A* and *C*. Node *C* is within intersection area of *B* and *D*.

2.3 Radio communication and distance estimation

Although some algorithms, like [43], do not explicitly describe the initialization communication method, most of them allow the anchor nodes to broadcast their position information first. When intermediate neighbor nodes receive the beacon message, they record the anchor nodes' position information, and then rebroadcast packets to their neighbors with updated hop counts. The process continues until all anchor nodes' position information is delivered to every node in the network.

To partially alleviate the expensive flooding, several optimization techniques are applied. Some of them are as follows:

- *Flood limit* method provided in [49]: This method assumes the normal nodes can derive good positions from a limited number of anchors. Consequently, when a normal node has recorded enough anchors, it will stop forwarding further information;

- *Hop-count* method in [45], in this method, if a node receives multiple packets from the same anchor node, it maintains and rebroadcasts the one with the least hop counts while ignoring all the rest. This optimization will eventually lead to a shortest path to the anchors.
- *Time stamp* method [29], in which a *time to live (TTL)* stamp is appended to each beacon packet. Out-dated packets are dropped silently.

To estimate the distance to anchor nodes, several different methods have been proposed. Based on the local neighborhood density, in [58] Kleinrock and Silvester provided a formula for the average distance between nodes. We present later this formula as *Kleinrock's formula*. Based on the information flooding, Niculescu [29] presented three methods: *DV-Hop propagation method*, *DV-Distance method* and *Euclidean method*. These methods are commonly used in many localization algorithms. For example, method *Hop-Terrain* by Savarese is similar to *DV-Hop* [29]; and method *Sum-dist* by Savvides is essentially the *DV-Distance* method [49]. In the next section, we will briefly introduce these radio propagation methods.

2.3.1 Propagation methods

In this section, we will introduce the propagation method.

- *DV-Hop* or *Hop-Terrain* propagation method: In this method, each node only communicates with its immediate neighbors as in distance vector routing. Starting from an anchor node, hop by hop information propagation allows a node to determine its distance, in hops, to that anchor.

Note that during the message propagation, each node only maintains and rebroadcasts packets with the smallest number of hop counts. The minimum hop counts that nodes retain will eventually be the length of the shortest path to the anchor. The distance to the anchor nodes is determined as the product of the hop counts and the average hop distance.

The average hop distance is determined by anchor nodes dynamically. When an anchor node receives a packet containing position information from other anchors, it can estimate the hop counts and the actual distance between them, and the average hop distance is obtained by averaging the distance with corresponding hop counts [29]. To avoid expensive flooding, a simple method of determining the average hop distance is to use the maximum radio range directly [45].

The advantage of the *DV-Hop* method is that it may allow fewer anchors for the initial estimation, and it work well in uniform networks.

- *DV-Distance* or *Sum-dist* propagation method: This method works similarly to *DV-Hop*, the difference is that each node now measures the pair-wise distance between neighboring nodes. The hop by hop information propagation transmits the cumulative distance instead of hop counts.

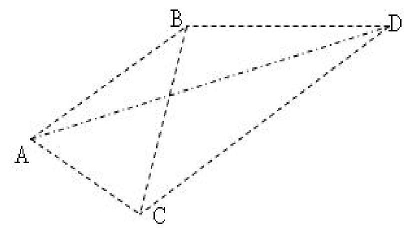
Same as *DV-Hop* method, each node only selects and rebroadcasts packets with minimum distance to the anchors. Compared with *DV-Hop*, to estimate nodes' position, this method is sensitive to measurement error because of the cumulative effect. However, this method is less coarse than the previous and may still perform well in randomly deployed networks.

- *Euclidean* propagation method: This method is based on the local geometry of nodes around the anchor. It assumes that the network is densely populated.

An anchor node initiates the propagation process by flooding its own position information in a beacon packet. Upon receiving the message, intermediate nodes can estimate their distance to the anchor and rebroadcast the message with added distance information of their own.

If node (A) receives messages from two neighbors (B and C) coming from the same anchor (D), and if the two neighbors (B and C) are also neighbors with each other (shown in Figure 2), then a quadrilateral ($ABDC$) may be formed with the knowledge of all sides and one diagonal. Note that node A can estimate its distance to B and C; and the diagonal is the distance from B to C, which can be determined since B and C are neighbors. The determination of the other diagonal (AD) is trivial, and the diagonal is the distance from node A to anchor D.

Figure 2 **Euclidean propagation model**



The algorithm allows all nodes to estimate their distance to the anchor node, and it works from the inner most circle to the faraway outside like a circular-wave form centered at that anchor.

The advantage of this method is that it obtains precise distance estimation to the anchor node even under random deployment networks. However, it is complex and only works in high-density networks. Furthermore, the error from pair-wise distance estimation is cumulative.

- *Kleinrock's formula*: Kleinrock and Silvester [58] showed that the average hop distance d_{hop} for a dense network depends only on local neighborhood density N:

$$d_{hop} = r(1 + e^{-N} - \int_{-1}^{1} e^{-\frac{N}{\pi}(\arccos t - t\sqrt{1-t^2})} dt \tag{1}$$

Nagpal *et al.* measured the average hop distance with variance N over several simulations from a random source [37]. Results validate Kleinrock's formula that it slightly underestimates the measured average hop distance.

Nagpal further noticed that N of 15 is a critical minimal threshold for achieving low errors in distance estimation.

2.4 Position estimation

After the anchor nodes' message propagation, each normal node obtains anchors' position, and the estimated distance to each of them. Based on the distance information, different localization algorithms can be applied.

2.4.1 Convex estimation

Convex position algorithm models the known peer-to-peer communication in the network as a set of geometric constraints on the node positions (refer section 2.2). Then a global convex optimization under these constraints yields feasible position estimation for all the normal nodes.

Based on connectivity and pair-wise angles between nodes, a linear program can be defined. A linear program (LP) is a problem of the form:

$$\text{Minimize } c^T x; \text{ Subject to: } Ax \le b \tag{2}$$

where x is the vector of variables to be solved for, A is a matrix of known coefficients, and c and b are vectors of known coefficients. The expression $c^T x$ is called the objective function, and the equations $Ax \le b$ are called the constraints. The matrix A is generally not square (usually more columns than rows), and $Ax \leq b$ is therefore quite likely to be under-determined, leaving great latitude in the choice of x with which to minimize $c^T x$.

In two dimension system, the position is represented with pair *(x,y)*, and a vector with all positions for the above equations can be formed as:

$$x = [x_1 y_1 \quad x_2 y_2 \quad \ldots x_m y_m \quad x_{m+1} y_{m+1} \ldots \quad x_n y_n]^T$$

where the first m entries represents anchor position and the remaining *n-m* are to be determined.

Geometrically, the connectivity of a network can be represented as a set of convex position constraints. By utilizing the radio communication constraint models discussed in section 2.2, it is possible to generate feasible positions for all the nodes in the network.

The Convex algorithm is a centralized algorithm. Through simulation, it is found that:

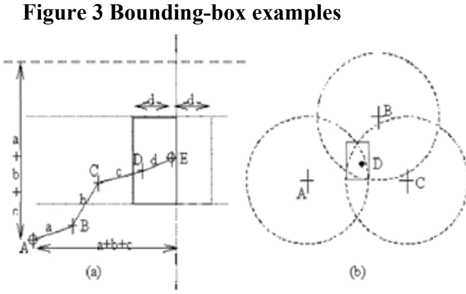
- For radically constrained connections, using a variable radius instead of a fixed radius improves overall estimation performance;
- For angle constrained connections, the angle estimation error affects the position prediction. If angle measurement can be combined with the distance information, the precision is improved.
- Results from radial and angular methods are not comparable.

2.4.2 Bounding-box or min-max method

The bounding-box defines a possible area that a node may reside in. There are essentially two basic methods. The first method is a distance-based approach proposed by Savvides [26], which uses the surrounding anchors' positions and the distance between them. The second method is a radio pattern based approach, which assumes regular circular radio pattern and constructs the bounding-box containing the overlapping or intersection area of neighboring nodes [43].

- **Distance based approach**: This is a general method that only assumes the knowledge of the distance from the estimating node to the anchors. Basically, we can construct a bounding-box for each surrounding anchor node. The possible area for the normal node is then determined as the intersection of those anchors' bounding-boxes.

Figure 3 Bounding-box examples



For example, suppose the distance from a normal node *x* to an anchor node *A* is *d*, and *A*'s coordinates are (x_A, y_A), then the bounding-box for *x* is defined as: $[x_A - d, y_A - d] \times [y_A + d, y_A + d]$.

If normal node *x* has multiple surrounding anchor nodes, then the bounding box for *x* is the intersection area of all anchor nodes' bounding boxes. The intersecting box is defined as: $[\max(x_i - d_i), \max(y_i - d_i)] \times [\min(y_i + d_i), \min(y_i + d_i)]$, where (x_i, y_i) is the anchor *I*'s position, and d_i is the distance between the normal node *X* and the anchor *I*. In Figure 3(a), node *A* and node *E* are anchors. Node *B, C*, and *D* are normal nodes. The bounding area for node *D* is: $[x_E - d, y_E - d] \times [x_A + (a + b + c), y_E + d]$.

The final estimated position for the normal node is set at the center of the bounding box.

- **Radio pattern based approach**: Instead of the distance from the estimating normal node to the surrounding anchors, this approach requires circular radio propagation pattern. That is, when one normal node receives a beacon packet from a neighboring anchor, it assumes it resides in the circular ring centered at that anchor. Accordingly, the bounding box for the normal node is the overlapping radio area from the neighboring (anchor) nodes. An example is shown in Figure 3(b), which shows a rectangular box for node *D* (anchor nodes *A, B*, and *C* are *D*'s neighbor).

Note that if the radio wave is circular, the second method is more precise.

2.4.3 Lateration algorithm

Lateration is a form of triangulation that uses least squares to estimate the position from a set of linearized equations in the form of $AX=b$.

Specially, from the radio propagation step, each normal node gets an estimated distance (d_i) to the anchor nodes with known position (x_i, y_i). If the normal node's position is (x,y), then a set of equations can be expressed as:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ \vdots \\ (x_n - x)^2 + (y_n - y)^2 = d_n^2 \end{cases} \tag{3}$$

Subtract the last equation from the first *n-1* equations, and reordering the equations, we get $AX=b$ with:

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_{n-1}) \end{bmatrix} \tag{4}$$

and $$b = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix} \tag{5}$$

then x can be estimated by least-squares with result: $x = (A^T A)^{-1} A^T b$.

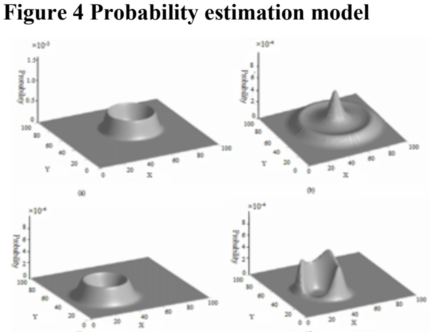
The least-squares is efficient because it minimizes possible range estimation errors accumulated along the propagation path.

2.4.4 Probability algorithm

For an outdoor environment, received signal strength is a function of distance: the longer distance from the transmitter, the smaller the signal strength. It is noticed in [46] that the probability distribution of signal strength follows a normal distribution.

The probability algorithm starts with a table that records the signal strength (mean and standard deviation) with the changing distance. In addition, it assumes the network is fully connected, and every node in the network is present in the entire space with equal probability.

Figure 4 Probability estimation model



When a normal node receives a beacon packet directly from an anchor, it estimates itself to be located on a circular surface centered at that anchor (see Figure 4(a) [46]). In

Figure 4, the x axis and y axis denote the position in meters, and z axis represents the position probability. Figure (a) gives the constraint model of one-hop beacon message. Figure (b) shows the constraints with two-hop beacon message. Figure (c) is the Old position estimation of a normal node. And figure (d) denotes the new estimates by intersecting the constraint with the old estimate. Each point on the circular surface represents a probability distribution at that location. A higher value means higher probability of the node's position at that position. Upon estimation, the normal node will add its ID as well as its estimated mean and standard deviation to a beacon packet, and rebroadcast the packet to its neighbors.

If the beacon packet comes from other normal nodes, the location constraint for the normal node is not necessarily Gaussian. Actually, the packet contains cascaded information of the estimated means and standard deviations of other normal nodes to the anchors. To estimate its position, the normal node has to process the cascaded distributions by *adding* all individual estimates (see Figure 4 (b)). The sum of these constraints is similar to the *convolution* of all the individual distributions.

After processing the beacon packet, the normal node updates its position by intersecting the new constraints with the current estimates. If the new estimates improve, it will wait for a specific period of time and advertise the new position to all neighbors. Figure 4 (c) and (d) shows an example of the procedure. Figure 4(c) gives the formal estimates of a normal node, and (d) shows the improved results to be advertised.

2.4.5 Multidimensional scaling method

Multidimensional scaling (MDS) is an exploratory technique used to analyze the dissimilarity of data on a set of objects. Multidimensional scaling takes roots in two important traditions within psychology: psychophysics and psychometrics [59,60], and it is called as "smallest space analysis" initially. Now MDS has already encompassed a collection of methods for general multivariate data analysis. Some excellent textbooks about MDS are [61-63].

Classical scaling technique is a metric multidimensional scaling technique [63], and it was originated in the 1930s by Young and Housholder [64]. Later, Schoenberg (1935) [65] and Young & Housholder (1938) provided the method for finding the original *Euclidean* coordinates from the *Euclidean* distances. In [25], the authors used classical multidimensional scaling algorithm in wireless ad hoc sensor networks for estimating the nodes' positions. And in [48], the authors extended the classical multidimensional scaling algorithms with more advanced features. Specifically two advanced methods, iterative multidimensional scaling (IT-MDS) and simulated annealing multidimensional scaling (SA-MDS), are provided in order to provide more accurate estimates.

Both [25] and [48] let anchors initiate the estimation procedure by broadcasting their position information to the whole network. Upon receiving a beacon packet, intermediate nodes will measure pair-wise distances to transmitting neighbors. If the intermediate node is an anchor, it will also estimate the average hop distance of the network. The average hop distance is determined by dividing the physical distance with the number of hop counts between the two anchors. If one routing path contains more than three anchors, and at least three anchors are not on a same line, classical multidimensional scaling can be used to estimate the coordinates of all normal nodes along the path.

Classical MDS works as follows [63, 25, 48]:

1. Compute the distance matrix $D = [d_{ij}]_{n \times n}$, where d_{ij} is the distance between node i and j, and n is the number of nodes in the path;

2. Compute a matrix J with $J = I - \frac{e \times e^T}{n}$, where I is the identity matrix and $e = (1,1,\ldots,1)$;
3. Double center matrix J with $H = -\frac{1}{2} J D^2 J$;
4. Eigen decomposition matrix H and let $H = UVU^T$, then sort descending the eigen matrix V, and change the matrix of U accordingly.
5. Choose an appropriate number of dimensions p. The coordinates of the estimated normal node in the p dimensional *Euclidean* space are given as the first p columns (two columns for 2-D case) in U.

The position information obtained through the above steps is the relative position with each other. To obtain the physical position of the nodes along the path, the estimated positions from MDS will, generally, be rotated and/or mirrored according to the reference anchor nodes along the path. Figure 5 shows an example.

Figure 5 MDS example

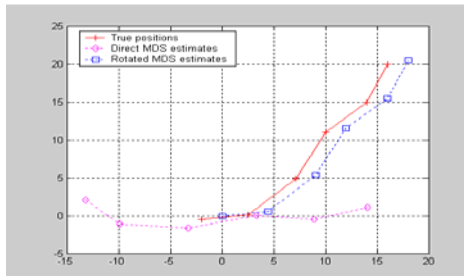


Figure 5 gives a routing path containing six nodes in 2-Dimensional space, and the x-axis and y-axis denote the coordinates in x and y direction respectively. The '+' presents the true positions; '◊' denotes direct estimates through MDS; and '□' represents the rotated positions from estimates, which will further shift to the true position.

2.4.6 Advanced multidimensional scaling method

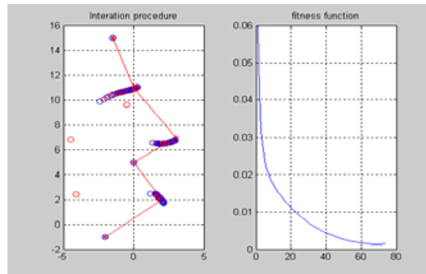
Classical MDS algorithm works if we can provide accurate pair-wise distance among all nodes along a path. Unfortunately, most general radio devices used today can only provide approximate distance measurements for neighboring nodes. In addition, the estimated pair-wise distance between nodes of multiple hops is error prone. Consequently, it is of upmost interest to evaluate the impact of the distance measurement errors on the location precision.

In [48], the authors consider the limits and communication constraints [43] of wireless ad hoc networks, and extend the classical multidimensional scaling algorithm with more advanced features. Specifically, [48] provides iterative multidimensional scaling (IT-MDS) and simulated annealing multidimensional scaling (SA-MDS). The IT-MDS algorithm considers the constraints from communicating neighbors, and embeds these constraints into the MDS algorithms in order to minimize the estimate errors iteratively. The main procedure can be described as follows:

```
while (estimation error > ε) {
    Run MDS algorithm
    Communication constraints check
    Update the anchors' position
    Update the distance among anchors
    Evaluate estimation errors
}
```

An example for IT-MDS is shown in Figure 6. The left chart in Figure 6 gives a route, in 2-dimension space, containing six nodes with three anchors and three normal nodes. The x-axis and y-axis denote the coordinates in two directions. During each iteration procedure, anchor nodes fix their positions while normal nodes progress to the true positions. The right chart shows the fitness function, which is denoted by y-axis. The x-axis represents the iterations. The fitness function is defined as:

Figure 6 IT-MDS example



$$fitness = \frac{(p_{estimate} - p_{true})^2}{p_{true}^2} \tag{6}$$

where $P_{estimate}$ and P_{true} are the sum of one-hop pair-wise distance of estimates and true positions separately. The lower the value of the fitness, the better estimates will be obtained.

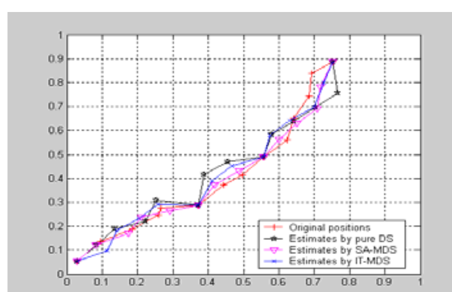
Different from IT-MDS, SA-MDS algorithm combines multidimensional scaling algorithm with Simulated Annealing (SA) method in order to achieve global optimization. This method simulates the process of a metal cooling down to the optimal state.

With SA, estimation procedure allows MDS algorithm to process a random search for each node's possible position, with the ability of accepting position updates that not only decrease estimation errors but also some updates that could increase the errors. The random search of the position information is carried out to simulate the annealing procedure in which possible optimization can be achieved. This optimization corresponds to minimum estimation errors of all possible positions. If only those updates with error-decreasing effects (downhill moves) are selected strictly, it is possible to only reach a *local minimum* solution. Consequently, in order to obtain *global optimization*, SA-MDS algorithm allows certain position searches, with probability, to also accept uphill moves that may increase the estimation errors.

Accordingly, the main algorithm can be detailed below:

```
while (not cooling down ) {
    Random search a node's position
    Run MDS algorithm
    Communication constraints check
    Evaluate estimation errors
    if (errors < previous estimates)
      keep change
    else
      keep change with probability
    end
}
```

Figure 7 Comparison of MDS-based algorithms



An example can be seen in figure 7. The figure shows a routing path in a network. The x-axis and y-axis represent the network coordinate in two directions. In the figure, symbol '∇' represents the estimated positions by SA-MDS method, '×' shows the estimates from IT-MDS, and the little star represents the estimates from MDS. The true position is denoted by '+'. The figure shows that the SA-MDS may achieve similar or even better estimates than the IT-MDS method.

2.5 Refinement

The purpose of the refinement process is to adjust the estimated position in the previous step. When one normal node estimates its own position, it informs its neighbors about its possible position. According to the radio communication constraints in section 2.2, the neighboring nodes can adjust their estimates accordingly. After a number of iterations, the position updating becomes small. Then the nodes can stop the refinement procedure and report the final position.

Convex positioning algorithm applies a central unit to process the position constraints for the whole network. Because of the global information, the refinement is obtained by simply executing the same algorithms multiple times. For each iteration procedure, a possible smaller rectangle bounding-box (see section 2.4.2) will be defined for each normal node. The narrowing procedure continues until predetermined precision is achieved.

Probability positioning algorithm measures the signal strength multiple times and uses a weighted formula to obtain the probability parameters (mean and standard deviation) corresponding to that position. During each advertisement phase, each node estimates and triggers an update only if the new prediction area is considerably smaller than the current.

Ad-hoc positioning algorithm extends the capabilities of GPS to non-GPS enabled nodes in a hop-by-hop fashion for an ad hoc network. The GPS triangulation method is executed iteratively upon each new advertised estimation, and the iteration process stops when the corrections are below a chosen threshold.

N-hop positioning algorithm groups all normal nodes into *computation subtree*, and executes multilateration process at each node in a consistent sequence. The *computation subtree* is a small set of collaborative nodes in which all nodes are over-constrained and each of them can only have one possible position. A simple example is one normal node connecting with three (or more) anchors (not collinear), in which the position of the normal node can be uniquely determined. The computation sequence does not need to be specified but it needs to be consistent for all iterations. This assures the order of the position updating sequence, and minimizes the *local oscillation* (local convergence without complying global optimization) and erroneous estimates [26].

Robust positioning algorithm includes a confidence metric with each node's position. The confidence weights are used to solve a weighted least-squares equation $wAx=wb$ instead of $Ax=b$, where w is the vector of confidence weights, with anchors have highest confidence of 1, and normal nodes have lower confidence associate with their observation conditions. Also, this algorithm detects *ill-connected groups*, i.e. the number of neighbors is less than 3 for a 2-dimensional space with the idea similar to the concept of *computation subtree* in N-hop algorithm. The refinement continues until desired precision is obtained.

Classical multidimensional scaling algorithm in [25] doesn't specifically address a refinement procedure. However, it also includes an iterative mechanism. To estimate all nodes position, it uses the newly updated positions from previous estimated nodes for the next iteration round.

The advanced MDS algorithms (SA-MDS and IT-MDS) presented in [48] improve the classical MDS. Both algorithms can provide better estimations even with approximate pair-wise distance between nodes along the routing path. As with many other algorithms, the refinement procedure of advanced MDS also relies on the neighbors' information. However, the authors found that the refinement may deviate a lot from the estimates by localization algorithms (IT/SA-MDS). The reason is that the refinement procedure is based on the imprecise estimates. Consequently, [48] associates a confidence factor with the refinement procedure, such that each refinement only updates, according to the confidence, specific values to the real positions.

2.6 Algorithm comparison

In this section, we will briefly compare all range-based localization algorithms in term of terrain dependence, radio pattern requirement, range estimation error, and others.

- Terrain effects on distance measurement methods: Different environment has different signal attenuation factor, so the radio transmission range and node's connectivity are dependent on the operating environment. For all radio propagation methods, only *Euclidean* method offers predictable performance across unpredictable conditions. Other methods (like *DV-based* methods and *Kleinrock's formula*) behave well only in isotropic topology, however they are less complex.

- Radio Pattern dependence: Convex algorithm and Probability localization algorithms assume perfect circular radio pattern, while others rely less on this assumption.

- Range error sensitivity: to estimate the distance between nodes, The *DV-hop* propagation scheme and the *Kleinrock's formula* do not require the range measurement. In other words, if distance between communicating neighbors can be measured accurately, *Euclidean* propagation method provides precise range estimates.

For the localization algorithms, Probability algorithm, *Convex algorithm* and *MDS* algorithms (including its advanced derivatives) are relatively less sensitive to the range measurement errors. Probability algorithm is based on the probability distribution of the signal strength, and it picks only those coordinates with highest probability as final estimates. *Convex algorithm* uses global constraints that can potentially minimize the measurement errors. *MDS* algorithms base on the anchors positions, and by using iteration or simulated annealing techniques, the position estimates can converge to the true position optimally.

- Density requirement: the density includes both anchors' density and overall nodes' connectivity. Most algorithms assume a high density and fully connected network. For simulation runs, most algorithms use connection degree of 7.5 (in term of neighbors per node) and above. For example, [49] found that a minimum connectivity of 9.0 is required for *DV-hop* and *Sum-dist* propagation methods. For *Kleinrock's formula* to work, the local density (in term of all nodes within unity area) of 10~15 is necessary [37].

More anchors will generally help the performance of localization algorithms; however, too many anchors will cause more interference. [25] found that about 10% anchor nodes is good enough to get minimum error estimation. Pure increment of anchors will not bring much improvement in position estimation.

- Communication cost: communication occurs mostly at the first step and the third step. In first step, anchor nodes broadcast their position information to the network, and at the third step, each normal node advertises its position estimates to their neighbors.

The Convex and MDS algorithms mostly rely on the first step communication, thus they are communication efficient. Different from these two methods, other algorithms have to rely on the extensive communication with neighbors in order to collect further constraints for refinement.

- Precision: The precision can be evaluated in two phases: phase 1 corresponds to the range estimation in radio propagation step, and the phase 2 corresponds to the position estimation and refinement steps.

[29] and [49] compared various propagation methods. The accuracy is measured against the anchor fraction (or anchor density). It is found that *Euclidean* method achieves higher accuracy when the anchor density increases. And *DV-based* methods

perform better when the anchor density is low. However, only *Euclidean* can work in anisotropic topology. The localization accuracy of the localization algorithms is dependent on the anchor density, the range measurement performance and the connectivity condition. Lower range estimation error, higher connectivity and higher anchor density will give better estimates for all algorithms.

It is further found that the MDS algorithm tends to generate more accurate position estimates at less dense networks, where small number of nodes are deployed.

We will provide the precision for all algorithms from their original research in section 6.1. In addition, we will compare the estimation performance of a set of representative algorithms in section 6.2.

3 Range-free algorithms

Range-free localization algorithms do not assume the range measurement capability so as to estimate distance between communicating nodes. However, in order to determine the node's location, each node still needs to use some mechanism to estimate the relative distance to anchors.

One approach is the use of the *Kleinrock's formula* (section 2.3.1}), which estimates the average hop distance only based on local neighborhood density in a uniformly deployed network. Consequently, if a prior knowledge of the local neighborhood density is given, the average hop distance can be calculated and then any range-based method in section 2.4 can be used. Nagpal *et al.* [37,38] use *multilateration* to estimate the nodes' position, and it is called *Amorphous localization algorithm*.

However, *Kleirock's formula* can only be used in densely deployed networks. For more general networks, special transmitters may be used to assist the localization procedure. For example, Premaratne *et al.* in [23] assumes that anchors can transmit radio signals with multiple level transmission powers. And in [24], He *et al.* uses high-powered transmitters. These two transmission methods will let anchors cover much larger areas. Consequently, with these special transmitters, an anchor can send beacon packets directly to normal nodes that are far away from the anchor itself. When a normal node obtains enough information about the surrounding anchor nodes, Premaratne's method allows the normal node to estimate its position by grid overlaying all possible regions it may reside in. We call this method *geometric grid overlaying (GGO)*. In contrast, He's method isolates the area into multiple triangular regions and use *approximate point-in-triangulation Test (APIT)* to obtain the position estimates. In the following sections, we will introduce these two methods.

3.1 Preliminaries

Same as range-based algorithms, range-free algorithms also assume a small percentage of nodes have knowledge of their positions (anchor nodes). In addition, *GGO* algorithm assumes that the network is fully connected, and anchor nodes can transmit radio signals in multiple radio transmission power levels, moreover, circular radio pattern is assumed. Similarly, *APIT* method assumes the usage of high-powered transmitters, in addition, each normal node can compare whether a neighboring node is closer to a given anchor node according to the signal strength (in the sense that the further away from a transmitter, the weaker the signal strength).

When initialization, each anchor node broadcasts its position information to the whole network. All normal nodes wait and collect the beacon information from anchors. This corresponds to the *radio propagation* step in range-based localization algorithms.

3.2 Position estimation

Because range-free algorithms apply different transmission technologies, the information that the normal nodes collect, and the detail localization mechanisms are different from those of range-based algorithms. For *GGO* algorithm, normal nodes still estimate the possible region (a ring-belt) they may reside in according to the specific signal power level they receive. And in *APIT* algorithm, normal nodes randomly choose three anchors from all audible anchors and test whether it is inside the triangle formed by connecting these three anchors.

3.2.1 *GGO* position estimation

GGO algorithm assumes that a normal node can transmit radio with a fixed range r. Further it supposes the anchor node can transmit radio signal in two power levels, which correspond with two radio range r_1 and r_2 respectively, with $r_1 < r_2$. To represent the possible region that a normal node could reside in, *GGO* denotes a ring-belt region as $B(C_j, r_m, r_n)$, where C_j is the position of an anchor, r_m and r_n are inner and outer radii of the radio range.

When a normal node receives a message from an anchor, it will select the one with minimum hop count. After recording the anchor's position information, it updates the hop count and repropagates the packet to its neighbors. With anchor's information, the node can estimate its possible location region according to the rules:

1). If a normal node is reached by an anchor C_j at both transmission power levels, its region is $B(C_j, 0, r_1)$;

2). If it can only be reached at second power level, its region is $B(C_j, r_1, r_2)$;

3). If the normal node is n hops away from the anchors, then its region is $B(C_j, r_2, n \times r)$.

For each received beacon packet, the normal node can estimate a region based on the above three rules. If a normal node is reached by multiple anchors, the intersection area of all single estimates from each anchor is denoted as final region for the normal node.

3.2.2 APIT position estimation

The *APIT* algorithm is based on the triangles formed by reached anchors. When a normal node is reached by a set of anchor nodes, it randomly chooses three of them and tests whether it is inside the triangle region formed by those three anchors. When all combinations have been used, the node can obtain a set of possible triangles it may reside in. By intersecting these triangles, a possible smaller area will represent the final estimation.

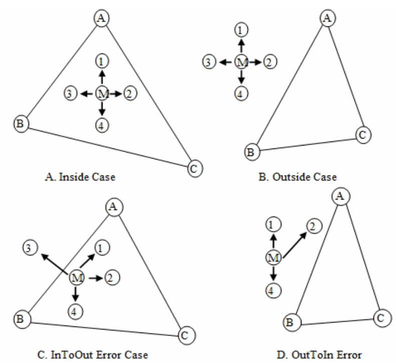
Though it seems straight in the beginning, one difficulty is the *Position in Triangle (PIT)* test because the node has no knowledge of its own position. In [24], the authors use neighborhood information to help the test. The algorithm assumes if two nodes receive signals from the same transmitter, the node that gets stronger signal (with higher energy) is closer to that transmitter. Based on this assumption, [24] addresses that if no neighbors of the normal node is further from or closer to all three anchor nodes simultaneously, the normal node can assume it is inside the triangle.

The *PIT* test works in most situations except in some special cases, it may suffer:

- The *edge effect*: when the normal node is near the edge of the triangle.
- The *collinear effect*: when the normal node and some of its neighbors reside on the same line.

An example [24] can be shown in Figure 8. Figure 8(a) and (b) show two normal situations where normal node M correctly estimates its position with anchor nodes A, B and C. (c) shows *edge effect* where M mistakenly thinks it is outside of the triangle because one neighbor (node 3) is further from all anchors. (d) shows the *collinear effect* where M will estimate that it is within the triangle because none of its neighbors is further from or closer to the anchors.

Figure 8 APIT examples



Fortunately, it is found that this error is relatively small, and it is less than 14% in worst cases [24].

3.3 Refinement

The *GGO* algorithm uses on-hop neighborhood information to reduce the estimated regions. This is achieved by advertising the region estimates to all the intermediate neighbors. Upon getting information from its neighbor, each normal node adjusts its position by intersecting the previous estimates with neighborhood constraints. This process is repeated until no further improvement can be achieved.

The *APIT* method does not specifically provide a refinement procedure. However, it gives one approach to obtain high redundancy anchor information. The mechanism uses a single moving anchor to send out location information at various positions in network.

By aggregating an additional set of triangles, a possible minimal region will be obtained. *APIT* uses the center of the gravity of the final region as the position for the node.

3.4 Performance evaluation

Both algorithms rely on the node density and the population of the anchor nodes. Higher are the nodes' density and population; the better is the estimation performance. The best performance reported in [23] for *GGO* algorithm is the error of 15% radio range, with a network of 5 anchors and 200 nodes in total. This means *GGO* algorithm is relative good in lower anchor density networks.

Compared with *GGO* method, *APIT* algorithm performs badly when the anchor number (heard by a normal node) is below 8. When anchor number increases, *APIT* works comparably as Nagpal *et al.*'s *Amorphous localization algorithm* in uniform deployment networks (with estimation error of less than 50% radio range).

Compared with *GGO*, *APIT* is insensitive to the terrain environment, and it makes no assumption of the radio transmission pattern. Also, it does not assume expensive refinement procedure.

4 Relative localization algorithms

When physical positioning is not available, or there is no need to obtain the physical positions for all nodes, relative localization algorithms seem to be good candidates to build relative position relation among connected nodes. In [40], Capkun *et al*. proposed *self-positioning algorithm (SPA)* algorithm. The *SPA* algorithm uses range capability to measure the relative distance among communicating nodes, and based on the distance information, *SPA* algorithm dynamically builds a set of local coordinate systems (*LCS*) for these neighboring nodes. By coordinating a set of *LCS*s, a relative stable network-wide coordinate system (*NCS*) can be constructed.

In [39], Niculescu and Nath extend the *SPA* only to construct position relation of those nodes that are active in communications. It is called *local positioning system* (*LPS*). *LPS* also assumes angle measurement capability of the system.

In the following sections, we will introduce the *SPA* algorithm, and then highlight the difference with the *LPS* system.

4.1 Self-positioning algorithm (SPA)

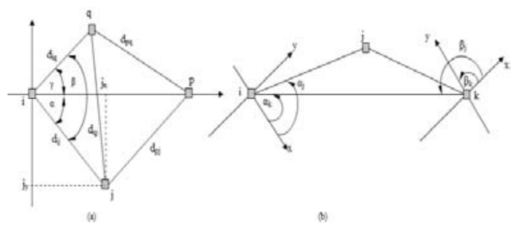
SPA assumes bidirectional links between communicating nodes. In addition, the algorithm assumes the range measurement capability for all nodes in the network. During initialization, each node will:

- Detect its neighbors;
- Measure the distance to the neighbors;
- Broadcast its ID and measured distance to all neighbors.

4.1.1 Local coordinate system (LCS)

Based on the local neighborhood information, a node can construct a *LCS*. Essentially, the node chooses, from its neighbors, another two nodes that are also neighbors with each other (the three nodes are not collinear). To form a coordinate system, the node sets itself as the origin. Further, it randomly picks one node on the positive x axis. Similarly y axis is determined such that the third node resides on the side with positive y coordinate value (See Figure 9(a) [40] for an example).

Figure 9 Relative coordinate system examples



In Figure 9(a), a *LCS* is defined for the normal node *i*. Nodes *p* and *q* are *i*'s direct neighbors. Nodes *p* and *q* are selected for the x-axis and y-axis, respectively.

Because local neighboring nodes can estimate the distance between them, by triangular theorem, most nodes can be uniquely determined within the local coordinate system. For example, in Figure 9(a), node *j* can be uniquely determined in the *LCS* of node *i*. The local neighboring nodes (node *i, p, q* and *j* in Figure 9) form a so called *local view set (LVS)* [40]. In the *LVS*, all nodes' positions can be distinctively determined according to a local coordinate system.

4.1.2 Network coordinate system (NCS)

To form a network coordinate, all *LCS*s have to coordinate with each other. First, the algorithm selects one *LCS* as a reference coordinate system, then all others can register to that reference system. Two steps are required for the coordination: i) Coordinate direction management, and ii) Position adjustment.

- Coordinate direction management: For two *LCS*s to coordinate with each other, at least one common node between the two *LCS* systems must exist. To register the direction of a *LCS* to the reference *LCS*, the *LCS* may have to rotate and/or to mirror one of its axes according the coordinates of the common node in the reference LCS. An example is shown in Figure 9(b), which depicts two *LCS*s of node *i* and node *k*. Node *j* is the common node shared by the two systems. In this case, *LCS* for node *k* need only rotate so as to register to *LCS* of node *i*.

- Position adjustment: When all *LCS*s have coordinated their directions, the position adjustment can be done by simply shift the *LCS*s with the relative distance to the reference coordinate system.

To take care of the mobility of the reference *LCS* that may be associated with a mobile node, a l*ocation reference group (LRG)* is used in [40]. The *LRG* is a group of nodes with highest local density in the network. Additionally, it redefines the origin of the reference *LCS* as the center of the topology for the *LRG*. Also, the direction of the reference *LCS* is computed as the mean value of all directions of the *LCS*s in the *LRG*.

The *LRG* is maintained periodically, and thus the *NCS* is relatively stable.

4.2 Local positioning system (LPS)

Instead of trying to construct a relative relationship for the whole network, Niculescu *et al.* found that locally positioning only the nodes involved in communication is good enough. The proposed *local positioning systems* (*LPS*) works similar as *SPA* algorithm with these differences:

1). *LPS* is dynamic and only involves a subset of nodes in the network;

2). *LPS* is coordinated at the packet originator instead of the center of the local reference group *LRG* as in *SPA*;

3). To improve the performance, *LPS* uses both angle and distance measurement to the construction of the positioning system.

4.3 Performance analysis

Since the relative localization system depends on the range and/or angle capabilities, they are sensitive to the measurement errors. However, if ranging and angle capabilities can be combined, better position estimation will be achieved. In addition, if digital compass can be used with AOA, less computation, and higher precision is achieved.

The communication cost and the computation cost for constructing and maintaining a network coordinate system is fairly high. This suggests that if the local coordinate system is enough for efficient routing, *local positioning system (LPS}* by Niculescu and Nath is preferred.

For both algorithms, higher connectivity is critical in order to build a local coordinate system. With poor connectivity, a node may not find two neighbor nodes for constructing a coordinate system. This recommends that high density or large transmission power is desired.

5 Anchor node density analyses

The anchor density of a network is an important parameter for absolute localization algorithms. However, few algorithms provide analysis for these questions: how many anchors are needed, and how should they be optimally deployed?

In [41], Bulusu *et al.* analyzed the impact of anchors density on the performance of the localization quality. The main idea is to use simple localization algorithms to estimate nodes' location, and further to analyze the anchor nodes' density locally. In addition, it describes the motivation, design, implementation and evaluation of a self-configuring localization system.

The proposed self-configuration algorithm is based on the anchor density.

- For sparse and medium density deployments, propose the *HEAP* algorithm to detect regions with poor localization, and select candidate points for new anchors.

- For dense anchor deployments, propose the *STROBE* algorithm. *STROBE* enables densely deployed anchors to coordinate with each other with the aim of minimizing self-interference and conserving energy.

5.1 Impact of anchor density

Bulusu *et al.* depicts the anchor density as two parameters, *anchor deployment density* ρ and *anchor per neighborhood* μ. *Anchor deployment density* ρ denotes the number of beacons per unit area: $\rho = \frac{N}{A}$. And *anchor per neighborhood* μ means the number of anchor nodes in a nominal radio transmission area. $\mu = \rho \cdot \pi \cdot Range^2$.

Consider a CSMA-like media access control (MAC) protocol for the ad hoc networks. If we assume T_x is the message transmission time over the beacon interval T, then the probability of an anchor node to transmit an advertisement packet is p, with $p = \frac{T_x}{T}$.

Let $p_{success}$ denote the probability of successful transmission in the wireless ad hoc network, and $p_{interference}$ represent the probability of interference. Then they can be expressed as follows [41, 58]:

$$\begin{aligned} p_{success} &= p \cdot (1-p)^{\mu} \\ p_{\text{int}\,erferece} &= 1 - p_{success} \end{aligned} \tag{7}$$

From Equations 7, we can see that the probability of interference increases exponentially with the anchor per neighborhood density μ. This means that we can not simply increase the number of anchor nodes in order to maintain higher localization performance. In other words, if the interference probability is to be maintained in high anchor density environment, the transmission probability p should be significantly small. However, if the transmission time T_x is relative fixed, then the beaconing interval time T must be increased. The result is that normal nodes will have to wait more time in order to receive beacon packets, thus the overall processing time for localization is increased correspondingly.

Thus, it can be claimed that there exists a threshold density μ_{thresh} for the anchor nodes, despite of the actual displacement. In [41], a threshold of *6* (in terms of beacons per nominal radio coverage area) is demonstrated for a 2-D case. This is corresponding to [58]'s conclusion of optimal average degree.

The self-configuration algorithms are proposed in order to maintain optimal anchor density. They are out of the scope of this paper. For further information, please refer to [41] and [66].

6 Precision analyses

This section will introduce experiment results of all localization algorithms reported in their original research. In addition, we will introduce some findings that could be useful for the localization algorithms. In the next section, a set of representative localization algorithms will be compared under same testing environment.

6.1 Reported precision

The best result is reported by Savvides *et al.* in [26, 31]. The testing setting is a network with 20% anchor nodes, more than six neighbors for each node, and the effective radio range is 15m. With multilateration localization method, the average error was 27.7mm with a standard deviation of 16mm. The simulation is based on the measurement noise parameters of a ultrasonic distance measurement system.

The probability algorithm by Ramadurai *et al.* [46] shows a test-bed of 60m×60m, with 5 anchors and 8 normal nodes. The radio range is 20m. The average precision error is about 9.4m, which is 47% radio range. If the GPS position estimation error is around 5m, this means the probability approach is pretty good.

The ad-hoc positioning system by Niculescu *et al.* [29, 39] considers both an isotropic and anisotropic topology with 100 nodes, and an average node degree of 7.6. The radio range is 10. The results show that the algorithm works better in isotropic topology. In isotropic environment, the error is about 35% radio range with 10% anchors density, and about 25% radio range under 20% anchors density. For anisotropic topology, the error is about 100% radio range for 10% anchors, and about 90% with 20% anchors.

Robust algorithm by Savarese *et al*. [45] assumes an average connectivity level of 7 or higher, with an anchor density of more than 5%, a relatively consistent error level of less than 33% (radio range) is achieved.

Convex positioning by Doherty *et al*. [43] does not provide similar precision estimation as the above algorithms, however, the research points out that the connectivity can improve precision dramatically, and the capability of variable radius radio transmission will help the localization precision.

Multidimensional scaling algorithm (MDS) by X. Ji [25] did not specifically provide a quantitative value; however, the research showed that the algorithm is less dependent on the radio pattern and the special terrain topology. Also, it concluded that the algorithm can achieve good estimation for a network with only 10% anchor nodes. The advanced MDS algorithms in [48] give a relative bounded estimation error of 40% (in terms of radio range) for most network settings.

Geometric grid overlaying (GGO) algorithm by Premaratne *et al.* [23] uses a test-bed of dense sensor system with very few reference nodes. For example, a 30×30 grid network contains 200 nodes (five anchor nodes), the localization accuracy of less than 15% of radio range can be achieved.

Approximate Point-In-Triangulation Test (APIT) algorithm by He *et al*. [24] relies more on the received anchor information, with optimal estimation error of about 45% radio range in uniform deployment environment, and about 50% radio range in random deployed networks. The advantage of the APIT algorithm is that it is less affected by connectivity as long as a degree of six can be maintained. Also, APIT is independent on the topology, radio pattern and the range measurement errors.

The Amorphous algorithm by Nagpal *et al*. [37] uses *Kleinrock's formula* [58] to estimate the average hop distance. For simulation run, four anchor nodes are placed at corner, and all others (total 200 sensors) are random placed in a square region of 6R×6R (R is the radio range). The simulation results show that position accuracy of 20% to 30% radio range can be achieved.

Relative localization algorithms by Capkum e*t al.* in [40], and by Niculescu *et al*. in [39] construct a local positioning system that can provide sufficient location information and support network functions. Because of the relative nature of the algorithm, both systems did not provide similar localization precision as the absolute positioning algorithms. However, the simulation results show that precision of range measurement is critical to the relative localization algorithms. With the aid of angular measurements (even with high errors), the overall performance is enhanced.

6.2 Algorithm optimization

Along the study, we also tested some of the introduced algorithms, including multidimensional algorithm (MDS) and multilateration algorithm. During simulation, we found that the following optimizations may be important:

First, the placement of the anchor nodes can heavily influence the algorithms' performance.

As stated in [45], most anchor nodes should be placed on the perimeter of the network. In [43], Doherty also shows that with as few as four known positions placed around the perimeter of the network, the estimate will result in smaller errors. Again, in [37], four anchor nodes are manually placed at the corner of the simulation area.

The perimeter deployment is important for the multidimensional scaling algorithm (MDS) because the algorithm only estimates those positions along a propagation path, with a constraint that the starting point and ending points of the path should both be anchor nodes. In most cases, the nodes around perimeter will never get their position predicted, or some special propagation methods will have to be used in order to involve them into the routing paths.

Though it is not discussed in [23, 24], we believe that the perimeter deployment should also be important for the APIT and GGO algorithms.

Second, the combination of some algorithms may produce better performance. The improvement includes higher precision, less processing time and more estimation coverage. For example, for MDS algorithm, even with perimeter deployment strategy, it may still not be able to predict all normal nodes' position. In this case, multilateration algorithm can simply be activated by those nodes that need their positions. After the initial MDS algorithm, those normal nodes, which did not get position predicted, may have already had information about anchor nodes and also their neighboring nodes. Consequently, the execution of multilateration algorithm will not cause much overhead in terms of communication.

Third, the radio constraints model is important for most localization algorithms. Applying the radio constraints to the iterative refinement phase increases the convergence speed in general.

7 Performance evaluations

In this section, we compare the following representative location estimation algorithms: *bounding-box, lateration, multidimensional scaling* and its derivatives from section 2.4, and the *APIT* algorithm from section 3.2.

We use *DV-Hop* radio propagation method to estimate the average hop distance, and we apply the *hop-count* method to minimize flooding (see section 2.3}).

Accordingly, we revise the *APIT* method and use the hop counts instead of the estimated distance in the *PIT* test. In addition, in this simulation, bounding-box method is *distance based* (see 2.4).

7.1 Experiment setting

We randomly deploy 100 nodes in a unit square area. The node deployment randomness is defined as the maximum displacement, in terms of radio transmission range, of each node that can swing from the cross-points of a coordinate grid. Accordingly, if all nodes strictly reside on the cross-points of a coordinate grid then the deployment randomness is null.

We evaluate the position estimation errors in terms of these changing network conditions:

1). Percentage of anchors
2). Randomness
3). Radio transmission range

Considering the ad hoc nature of the wireless networks, we run all algorithms extensively under each network settings (not necessary the same network). For each setting, we pick the average value as the final result in this report.

7.2 Results

The relation of the dependency of the estimation error to the anchor number is shown in Figure 10, where the x-axis denotes the anchor percentage, and the y-axis gives the estimation error in terms of radio propagation range. It shows more anchors will, generally, help the localization algorithms. However, when the anchors account reach 20%, adding anchors results in a marginal decrease in estimation error.

Figure 10 Estimation errors vs. anchor ratio

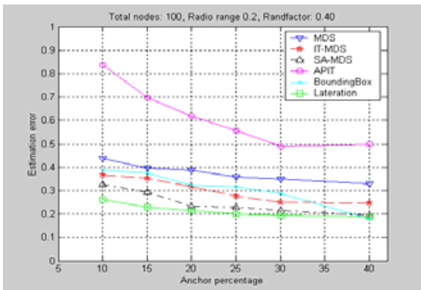


Figure 11 Estimation errors vs. randomness

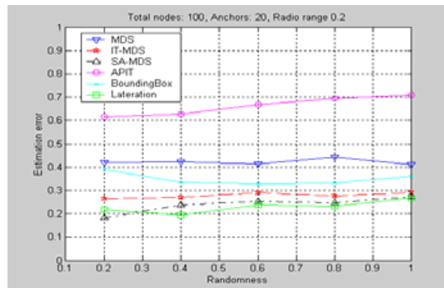
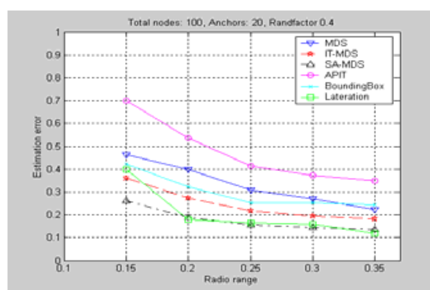


Figure 11 gives the results of the deployment randomness to the effect of the estimation errors. The x-axis represents deployment randomness and the y-axis gives the estimation error. These results show that most algorithms are marginally sensitive to deployment randomness. So the positioning of nodes has no impact on the estimation error, except for APIT.

Figure 12 plots the estimation error (in y-axis) *versus* the radio transmission range (in x-axis). Since the estimation error is reported as a fraction of the radio range, it is normal that we observe a slight decrease of estimation error.

Figure 12 Estimation errors vs. radio ranges



The comparison of all the implemented algorithms shows that lateration algorithm and SA-MDS yield better performance than all others. In all networking settings, APIT algorithm provides worst estimates. Though simple, the performance of bounding-box algorithm is fairly good (between general MDS algorithm and advanced MDS algorithms).

8 Conclusions

This study analyzes the localization algorithms in literature for wireless ad hoc networks. The surveyed algorithms include range-based algorithms and range-free algorithms. In addition, we also consider the relative localization algorithms that are useful when physical position information is not needed. Still, we analyze the anchor nodes' density on the performance of the localization algorithms,

Besides the algorithms, we introduce the simulation environment and the reported precision for each of them. Through the study, we found that certain optimization can be achieved by applying radio constraints to the iterative refinement phase for each algorithm, and by deploying the anchors around perimeter. Furthermore, a hybrid algorithm may be desired in order to achieve better performance.

From the study, we can find that most localization algorithms are sensitive to some network parameters, like node density, topology, radio pattern, and network connectivity. Besides, the precision of the algorithms also depends on the hardware performance such as range estimation.

For future research, a theoretical analysis of the impact of network parameters on the localization algorithms will be important. Also, we may include other science, like statistical analysis and mathematical theory in the study. At the same time, better range estimation methods are highly needed.

References

1 S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in wireless ad hoc sensor networks. In *International Conference on Mobile Computing and Networking (MobiCom'01)*, pages 139-150, Rome, Italy, July 2001.

2 T. Yan, T. He, and J.A. Stankovic. Differentiated surveillance service for sensor networks. In *Proceeding of First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, CA, 2003.

3 M. Ancona, S. Locati, and A. Romagnoli. Context and location aware textual data input. In *Proceedings of the 2001 ACM symposium on applied computing*, pages 425-428. ACM Press, 2001.

4 Varshney, U. (2003) 'Issues, requirements and support for location-intensive mobile commerce applications', *Int. J. Mobile Communications*, Vol. 1, No. 3, pp247-263.

5 Didier Chincholle, Mikael Eriksson, and Alex Burden. Location-sensitive services: it's now ready for prime time on cellular phones! In *Proceedings of the conference on Designing interactive systems*, pages 331-334. ACM Press, 2002.

6 Alan Dix, T. Rodden, N. Davies, J. Trevor, A. Friday, and K. Palfreyman. Exploiting space and location as a design framework for interactive mobile systems. *ACM Transactions on Computer-Human Interaction (TOCHI), in press*, 7(3):285-321, Sep.2000.

7 Eija Kaasinen. User needs for location-aware mobile services. *Personal Ubiquitous Comput.*, 7(1):70-79, 2003.

8 Olla, P. and Patel, N.V. (2003) 'A framework for delivering secure mobile location information', *Int. J. Mobile Communicaitons*, Vol. 1, No. 3,pp.289-300.

9 D.C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole. Research challenges in environmental observation and forecasting systems. In *Proceedings of the sixth annual international conference on Mobile computing and neting*, pages 292-299. ACM Press, 2000.

10 A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: application driver for wireless communications technology. In *Proceedings of the ACM SIGCOMM Workshop on Data Communications*, pages 20-41, 2001.

11 H. Wang, J. Elson, L.Girod, D. Estrin, and K. Yao. Target classification and localization in habitat monitoring. In *Proceedings of the IEEE ICASSP 2003*, Hong Kong, April 2003.

12 P. Boettcher, J. A. Sherman, and G. A. Shaw. Target localization using acoustic time-difference of arrival in distributed sensor networks. In *Proceedings of SPIE 47th Annual Meeting*, 2002.

13 E. Howden. Networked sensors for the objective force. In *Proceedings of SPIE 47th Annual Meeting*, 2002.

14 R.L.Moses, D. Krishnamurthy, and R. Patterson. An auto-calibration method for unattended ground sensors. In *Proc. ICASSP*, pages 2941-2944, May 2002.

15 Luo, Y., Li, J., Xiao, Y., and Lin X. (2004) 'A simulation study of overflow replacement policies for location management in mobile networks', *Int. J. Communications*, Vol. 2, No. 2, pp103-127.

16 Varshney, U. (2003) 'Location management for wireless networks: issues and directions', *Int. J. Mobile Communications*, Vol. 1, Nos. ½, pp. 91-118.

17 B. Karp and H. T. Kung. GPSR:greedy perimeter stateless routing for wireless networks. *Proceedings of MOBICOM'00*, August 2000.

18 Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. *Proceedings of MOBICOM'01*, July 2001.

19 K. Chen and K. Nahrstedt. Effective location-guided tree construction algorithms for small group multicast in MANET. *Proceedings of IEEE INFOCOM 2002*, Jun. 2002.

20 X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network magazine*, 16(4), 2002.

21 Kumar, S. and Stokkeland, J. (2003) 'Evolution of GPS technology and its subsequent use in commercial markets', *Int. J. Mobile Communications*, Vol. 1, Nos. ½, pp. 180-193.

22 B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer-Verlag, 4th edition edition, 1997.

23 Kamal Premaratne, Jinsong Zhang, and Murat Doguel. Location information-aided task-oriented self-organization of ad-hoc sensor systems. *IEEE sensors Journal*, 4(1), Feb. 2004.

24 Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free location schemes for large scale sensor networks. *MobiCom'03*, Sep. 2003.

25 Xiang Ji and Hongyuan Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *IEEE INFOCOM*, 2004.

26 A. Savvides, H. Park, and M. B. Srivastava. The n-hop multilateration primitive for node localization problems. *Journal of Mobile Networks and Applications (MONET)*, 8:443-451, 2003.

27 Y. Oshman and P. Davidson. Optimization of observer trajectories for bearings-only target localization. *IEEE Trans. Aerosp. Electron.*, 35:892-992,1999.

28 N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. *Proceedings of MOBICOM'00*, Aug. 2000.

29 Dragos Niculescu and Badri Nath. Ad-hoc positioning system (APS). In *GLOBECOM 2001*, San Antonio, Nov. 2001.

30 Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *ACM MOBICOM*, Rome, Italy, 2001.

31 A. Savvides, H. Park, and M. Srivastava. The bits and °ops of the n-hop multilateration primitive for node location problems. *First ACM International Workshop on Wireless Sensor Networks and Application (WSNA)*, pages 112-121, 2002.

32 Dragos Niculescu and Badri Nath, editors. *Ad Hoc Positioning System (APS) using AoA*, San Francisco, CA, 2003. IEEE INFOCOM.

33 Jeffrey Hightower and Gaatano Boriello. Localization systems for ubiquitous computing. *IEEE Computer Magazine*, 34(8):57-66, August 2001.

34 Joshua A. Tauber. Indoor location systems for pervasive computing. Technical report, MIT, August 2002.

35 P. Bahl and V. Padmanabhan. RADAR: An in-building rf-based user location and tracking system. *Proc. IEEE Infocom 200*, pages 775-784, 2000.

36 M.L. Sichitiu, V. Ramadurai, and P.Peddabachagari. Simple algorithm for out-door localization of wireless sensor networks with inaccurate range measurements. pp. 300-305, Las Vegas, NV, June 2003. in Proc. Int. Conf. Wireless Networks.

37 Radhika Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. *In the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003.

38 Radhika Nagpal. Organizing a global coordinate system from local information on an amorphous computer. *A.I. Memo 1666*, Aug. 1999.

39 Dragos Niculescu and Badri Nath. Localized positioning in ad hoc networks. *in IEEE SNPA 2003*, May 2003.

40 S. Capkun, M. Hamdi, and J.P. Hubaux. GPS-free positioning in mobile ad-hoc networks. *Cluster Computing*, 5(2), April 2002.

41 Nirupama Bulusu, John Heidemann, Deborah Estrin, and Tommy Tran. Self-configuring localization systems: Design and experimental evaluation. In *ACM Transactions on Embedded Computing Systems (ACM TECS), Special issue on networked embedded systems*, 2003.

42 Vladimir Bychkovskiy, Seapahn Megerian, Deborah Estrin, and Miodrag Potkonjak. A collaborative approach to in-place sensor calibration. *2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, Dec 2003.

43 Lance Doherty, Kristofer S. J. Pister, and Laurent E. Ghaoui. Convex position estimation in wireless sensor netowrks. *Proc. of IEEE Inforcom*, 3:1655-1663, 2001.

44 A. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad-hoc wireless sensor networks. *In IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2037-2040, May 2001.

45 Chris Savarese, K. Langendoen, and J. Rabaey. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*, pages 317-327, 2002.

46 Vaidyanathan Ramadurai and M. L. Sichitiu. Localization in wireless sensor networks: A probabilistic approach. In *Proc. of the 2003 International Conference on Wireless Networks (ICWN 2003)*, pages 275-281, Las Vegas, NV, June 2003.

47 X. Ji and H. Zha. Multidimensional scaling based sensor positioning algorithms in wireless sensor networks. In *Proceedings of the 1st Annual ACM Conference on Embedded Networked Sensor Systems*, pages 328-329, Nov. 2003.

48 Saad Biaz and Yiming Ji. Precise distributed localization algorithms for wireless networks. S*ubmitted to IEEE Infocom 2005*, July, 2004.

49 K. Langendoen and N. Reijers. Distributed location in wireless sensor networks: A quantitative comparison. *Computer Networks*, 43:499-518, 2003.

50 W. Kumm. Gps global positioning system. *Bielefeld,Germany:Klasing*, 1995.

51 Jacek Czajewski. The accuracy of the global positioning systems. *IEEE Instrumentation & Measurement Magazine*, 2004.

52 Robin Strahan. Location sensing technologies. Report No. e-mc2.2.1.1.2002, Department of Computer Science, University College Dublin (UCD), 2002.

53 http://www.tycoelectronics.com/gps.

54 http://waas.stanford.edu/ak ft.html.

55 R. Eric Phelts, Todd Walter, Per Enge, Dennis M. Akos, Karl Shallberg, and Tom Morrissey. Range biases on the WAAS geostationary satellites. San Diego CA, January 2004. Presented at the Institute of Navigation's National Technical Meeting.

56 http://europa.eu.int/comm/dgs/energy transport/galileo.

57 http://www.gps-practice-and-fun.com/a-gps.html.

58 K. Kleinrock and J. Silvester. Optimum tranmission radii for packet radio networks or why six is a magic number. In *National Telecommunications Conference*, pages 4.3.1-4.3.5, Birmingham, Alabama: IEEE, Dec. 1978.

59 G. Young and A.S. Householder. A note on multidimensional psychophysical analysis. In *sychometrika, 6*, pages 331-33, 1941.

60 L. Guttman. A new approach to factor analysis: the radex. in p. lazarsfeld (ed.). In *Mathematical thinking in the behavioral sciences*, New York:Free Press, 1954.

61 T.F. Cox and M.A.A. Cox. *Multidimensional scaling*. London: Chapman Hall, 1994.

62 I. Brog and P. groenen. *Modern multidimensional scaling: Theory and applications*. New York: Springer, 1997.

63 Trevor F. Coox and Michael A.A. Cox. *Monographs on Statistics and Applied Probability 88 Multidimensional Scaling*. Chapman and HallCRC, 2 edition, 2001.

64 G. Young and A.S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19-22, 1938.

65 I.J. Schoenberg. Remarks to maurice frechet's article "sur la definition axiomatique d'une classe d'espaces vectoriels distancies applicables vectoriellement surl'espaace de hilbert". *Ann. Math.*, 36:724-732, 1935.

66 Nojeong Heo and P.K. Varshney. A distributed self spreading algorithm for mobile wireless sensor networks. *Wireless Communications and Networking, 2003. WCNC, EEE*, 3:1597-1602, March 2003.