

# Chapter 11

## Range-Free Localization Techniques

Christian Poellabauer

**Abstract** In wireless sensor networks, sensors are often deployed without a priori knowledge of their locations or sensor node locations can change during the lifetime of a network. However, location information is essential for a variety of reasons. Sensors monitor phenomena in the physical world and given the location of the sensors, it is then possible to estimate the location of the observed phenomenon. For example, chemical and humidity sensors deployed on a farm can provide information about soil moisture, crop health, and animal movement if the sensor locations are known. Accurate location information is also needed for various sensor network management tasks such as routing based on geographic information, object tracking, and providing location-aware services. Frequently, sensor node localization is performed using ranging techniques, where the distances between a sensor device and several known reference points are determined to derive the position of a sensor. However, the cost and limitations of the hardware needed for range-based localization schemes often make them poor choices for WSNs. Therefore, a variety of localization protocols have been proposed that attempt to avoid the use of ranging techniques with the goal to provide more cost-effective and simpler alternatives. These range-free localization techniques estimate a node's position using either neighborhood information, hop counts from well-known anchor points, or information derived from the area a node is believed to reside in. This chapter introduces the basic concepts of range-free localization, surveys a variety of state-of-the-art localization techniques, compares qualitatively the characteristics of these protocols, and discusses current research directions in range-free localization.

---

C. Poellabauer (✉)  
University of Notre Dame, Notre Dame, USA  
e-mail: cpoellab@nd.edu

## 1 Overview

Wireless sensor networks are built and deployed to collect real-time information on the spatial-temporal characteristics of the physical environment. Therefore, the knowledge of the location of each sensor node in a network is critical to many WSN applications and services to appropriately interpret sensors readings. For example, environmental sensor networks measure air, water, and soil quality to detect the presence and sources of pollutants [16], but they require accurate spatial information to support precise modeling and simulation of the dispersion of the pollutants. Further, in battlefield scenarios, sensor locations are essential to accurately determining or predicting enemy movements [19] and in emergency response systems, sensor locations are needed to guide first responders towards survivors and away from harm [17]. Sensor node localization is also needed to support a variety of network management services, such as routing protocols (e.g., many routing protocols are based on the principle of geographic forwarding [34]), coverage area and topology control [33], energy management techniques (e.g., protocols that adjust in-network data management to preserve energy [13]), clustering [38], boundary detection [10], and various network security mechanisms and protocols [37].

As a consequence, localization has been the focus of a significant number of research efforts, ranging from manual configurations to distributed positioning algorithms. For example, Global Positioning System (GPS) has been a popular choice for localization in mobile devices, but it is often a poor choice for WSNs because of its high hardware cost (compared to the low cost of the miniaturized devices often used for sensing) and its inability to provide location services in indoor settings and other scenarios where no clear view of the sky is available. The type of location information provided by GPS can be expressed as *global* metric, i.e., a position within a general global reference frame (e.g., longitudes and latitudes). Another example of such a system is the Universal Transverse Mercator (UTM) coordinate system, which provides positions within zones and latitude bands. In contrast, *relative* metrics are based on arbitrary coordinate systems and reference frames, e.g., a sensor's location expressed as distances to other sensors without any relationship to global coordinates.

When systems such as GPS are unsuitable for a WSN, a network can also rely on a subset of nodes that know their global positions for localization. Other nodes in the network can then use these *anchor nodes* (or *reference nodes*) to estimate their own positions. Techniques that rely on such anchors are called *anchor-based localization* (as opposed to *anchor-free localization*). Many localization techniques, particularly in the category of anchor-based techniques, are based on range measurements, i.e., estimations of distances between several sensor nodes. These *range-based localization* techniques require sensors to monitor measurable characteristics such as received signal strengths of wireless communications or time difference of arrival of ultrasound pulses.

In contrast, *range-free localization* techniques do not rely on distance estimates, but instead rely on approaches such as estimating their relative positions to other

nodes using connectivity-based algorithms. These range-free localization techniques are typically more cost-effective, because they do not require the nodes to have special hardware functionalities. On the downside, range-free localizations typically lead to coarser estimates. However, these estimates are sufficient for many types of WSN applications where limited localization accuracy is acceptable.

The remainder of this chapter discusses the fundamentals and state-of-the-art solutions for range-free localization techniques. When comparing localization techniques, two important qualities of localization information are the *accuracy* and *precision* of a position. For example, a GPS sensor indicating a position that is true within 10 m for 90% of all measurements, the accuracy of the GPS reading is then 10 m (i.e., how close is the reading to the ground truth?) and the precision is 90% (i.e., how consistent are the readings?). Apart from *physical* positions (such as the longitudes and latitudes provided by GPS), many applications (e.g., indoor tracking systems) may only require *symbolic* locations [7], such as “office building A14” or “mile marker 12 on Highway 50.” Other metrics that determine the suitability of a localization technique for a specific WSN scenario include energy efficiency, support for Quality-of-Service, and localization overheads and costs.

## 2 Range-Free Versus Range-Based Localization

Most WSN localization techniques rely on variations of the same principle, i.e., they establish sensor node locations based on information exchange between neighboring nodes. When certain characteristics of the exchanged messages or signals are used to determine distances between nodes, a variety of localization techniques can be applied to determine a sensor’s position, where the positioning accuracy depends on the quality of distance measurements. These types of localization techniques belong to the category of range-based localization. In contrast, if information exchange is only used to establish connectivity information (i.e., learn about the topology of a portion of the WSN), then a sensor node can learn its position in a network relative to other sensor nodes. While the accuracy of these range-free localization techniques is typically lower than the accuracy of range-based techniques, range-free approaches can provide more cost-effective solutions for large, low-cost wireless sensor networks. This section provides an overview of these two main categories of sensor node localization.

### 2.1 Overview of Range-Based Localization Techniques

In range-based localization, sensor nodes obtain distance estimates between themselves and other sensor nodes, e.g., using measurements of certain characteristics of radio signals, including signal propagation times, signal strengths, and angle of arrival. When the concept of *Time of Arrival (ToA)* is applied [3], the distance between the sender and receiver of a signal is determined by combining the signal propagation

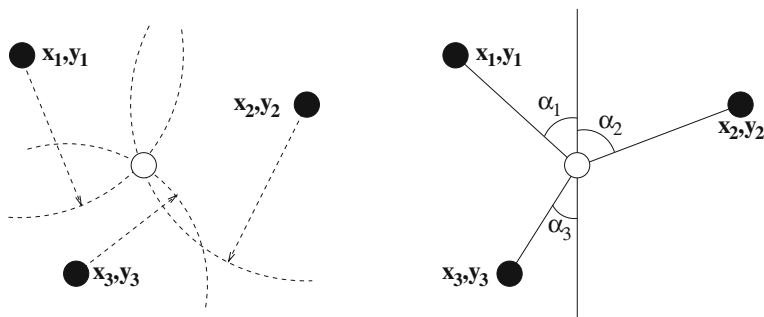
time with the known signal velocity. For example, an acoustic signal requires approximately 30 ms to travel a distance of 10 m (assuming a velocity of 343 m/s), while a radio signal requires only about 30 ns for the same distance (assuming a velocity of 300 km/s). Radio-based distance measurements require highly accurate clocks to measure such short propagation times, thereby adding to the cost and complexity of range-based localization techniques. Time of arrival methods can further be categorized as *one-way* methods and *two-way* methods. With one-way methods, the signal transmission time and reception time are measured at the sender and receiver, respectively, thereby requiring accurate time synchronization between sender and receiver. With the two-way method, the round-trip time of a signal is measured at the signal sender, therefore removing the need for time synchronization.

A second class of range-based localization techniques uses the *Time Difference of Arrival (TDoA)* approach [4], where two signals travel from the sender to the receiver with different velocities. For example, the first signal could be a radio signal, followed by an acoustic signal. The receiver is then able to determine its location similar to the ToA approach. TDoA-based approaches do not require the clocks of the sender and receiver to be synchronized and can obtain very accurate measurements. However, a disadvantage of the TDoA approach is the need for additional hardware, e.g., a microphone and speaker to transmit and receive acoustic signals.

Another popular technique is to determine the direction of signal propagation, typically using an antenna or microphone arrays. The *Angle of Arrival (AoA)* is then the angle between the propagation direction and some reference direction known as orientation [25]. For example, for acoustic measurements, several spatially separated microphones can be used to receive a single signal and the differences in arrival time, amplitude, or phase can then be used to determine an estimate of the arrival angle.

Finally, the *Received Signal Strength (RSS)* method determines distance based on signal attenuation. While RSS measurements may not lead to as accurate distance measurements as the ToA method (e.g., signal attenuation depends not only on distance, but also multi-path propagation effects, reflections, and noise), these measurements are often readily available in many wireless network cards.

Given a distance measurement between a sensor node and a reference node, the sensor node knows that its position must be along the circumference of a circle (in two-dimensional space) or sphere (in three-dimensional space) centered at the reference node, with the radius representing the distance between the reference node and the sensor node. To obtain a unique location in two dimensions, distance measurements from at least three non-collinear reference nodes are required (or distance measurements from at least four non-coplanar reference nodes in three dimensions). This process of determining a sensor node location is called *trilateration*. In contrast, a similar process called *triangulation* uses the geometric properties of triangles to estimate sensor locations. Triangulation requires at least two angles (or *bearings*) and the locations of the reference nodes or the distance between them to determine the sensor node location in two-dimensional space. The processes of trilateration and triangulation are illustrated in Fig. 1, where three reference nodes are used (identified by their known  $(x_i, y_i)$  coordinates).



**Fig. 1** Example of trilateration (shown on the *left*) and triangulation (shown on the *right*)

Triangulation and trilateration rely on the presence of at least three reference nodes. However, in many sensor networks, it is infeasible to assume that every sensor node can directly communicate with at least three reference nodes. Therefore, these techniques can further be extended, i.e., once a sensor has determined its own position (using either trilateration or triangulation with three or more reference nodes), this sensor can itself become a reference node for other sensors. This iterative process is called *iterative multilateration* [28]. A downside of this process is that every iteration contributes to the localization error.

A variation of this approach is called *collaborative multilateration*, where the goal is to construct a graph of *participating* nodes, i.e., nodes that are reference nodes or have at least three participating neighbors. A node can then estimate its position by solving the corresponding system of over-constrained quadratic equations relating the distances between the node and its neighbors.

### Case Study: GPS-Based Localization

As an example of a trilateration system, consider the Global Positioning System (GPS) [8], which is the most widely publicized location-sensing system, providing an excellent lateration framework for determining geographic positions. GPS is the only fully operational global navigation satellite system (GNSS), consisting of at least 24 satellites orbiting the earth at altitudes of approximately 11000 miles. The satellites are uniformly distributed in a total of six orbits (i.e., there are four satellites per orbit) and they circle the earth twice a day at approximately 7000 miles per hour. Each satellite constantly broadcasts coded radio waves that contain information on the identity of the particular satellite, the location of the satellite, the satellite's status, and the date and time a signal has been sent. In addition to the satellites, GPS further relies on infrastructure on the ground to monitor satellite health, signal integrity, and orbital configuration, e.g., at least six monitor stations located around the world constantly receive the data sent by the satellites and forward the information to a master control station (MCS). This MCS uses the data from the monitor stations to compute corrections to the satellites' orbital and clock information, which are then sent back to the satellites via ground antennas.

Satellites and receivers use very accurate and synchronized clocks so that they generate the same code at exactly the same time. A GPS receiver compares its own generated code with the code received from the satellite, thereby determining the actual generation time of the code at the satellite and the time difference between the code generation time and the current time. This time difference expresses the travel time of the code from the satellite to the receiver. Since radio waves travel at the speed of light, given the time difference, the distance between receiver and satellite can be determined. Given this distance, the receiver knows that it must be positioned somewhere on the sphere centered on the satellite. With two more satellites, a receiver can determine two points where the three spheres intersect and typically one of these two intersections can be eliminated, e.g., because it would position the receiver far out in space. While three satellites appear to be sufficient for localization, a fourth satellite is required to obtain an accurate position.

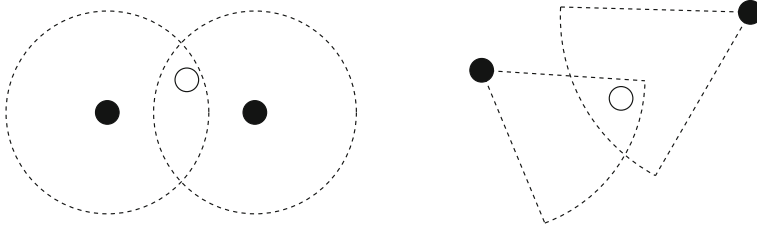
While most GPS receivers available today are able to provide position measurements with accuracies of 10 m or less, advanced techniques to further increase the accuracy are also being used. For example, Differential GPS (DGPS) [20] uses land-based receivers with exactly known locations to receive GPS signals and compute correction factors, which are then broadcast to regular GPS receivers. Further, Real Time Kinematic (RTK) navigation is another technique used to provide centimeter-level accuracy by using a stationary GPS receiver together with one or more mobile units, where the stationary receiver re-broadcasts the signals it receives from the satellites as correction signals to the mobile units.

## 2.2 Overview of Range-Free Localization Techniques

The localization approaches discussed so far rely on ranging techniques such as RSS, Time of Arrival, Time Difference of Arrival, and Angle of Arrival. In contrast, *range-free localization techniques* do not require additional hardware and are therefore a cost effective alternative to range-based techniques. With range-free techniques, instead of estimating distances between sensor nodes, other approaches are used to determine a sensor node's location at a coarser granularity. These approaches can be grouped into techniques based on *area*, *hop count*, and *neighborhood* information [16].

In area-based localization, a network can be divided into areas or regions, and localization is then concerned with determining the region a node occupies. For example, a sensor node that is able to hear radio signals from two reference nodes can determine that it must reside in the overlapping region of the radio coverage areas of these reference nodes. The more reference nodes are available, the more accurately a node can determine its position (or the region it occupies becomes smaller).

Figure 2 shows two examples of area-based localization. On the left, a node (white node) determines that its position is in the region created by the overlapping radio coverage regions of its two neighboring nodes (black nodes). The figure on the right shows a similar situation with two nodes with directional antennas.



**Fig. 2** Area-based localization using two reference nodes with omni-directional antennas (*left*) and directional antennas (*right*)

When directional antennas are used, the size of the overlapping region can be reduced, thereby increasing the positioning accuracy.

In localization techniques based on hop count, hop distances between two nodes A and B are used to estimate node distances and node positions. The hop count is the minimum number of hops  $h_{\min}$  (i.e., the shortest route) that separates nodes A and B. The maximum distance between A and B is then  $R * h_{\min}$ , where R is the radio coverage radius (assuming that all nodes have the same radio range).

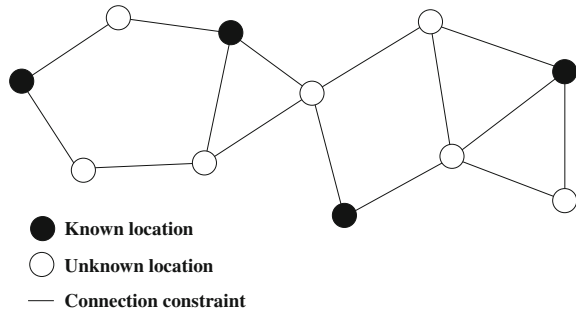
Further, in neighborhood-based localization, reference nodes placed throughout the sensor network periodically issue beacons that include the emitting node's location. If a sensor node receives beacons from only one reference node, the sensor node simply assumes its own location to be identical to the reference node's location. However, if beacons from multiple reference nodes can be received, more sophisticated methods (e.g., the centroid technique) can be used to determine a node's position in the network.

Finally, recently a number of event-based localization techniques have emerged, where event emitters transmit signals that are being received by the sensors and these signals (or certain timing aspects of these signals) can be used to determine a sensor's location. In these techniques, localization is based on the presence of (often costly) event generator devices, therefore replacing the need for reference nodes. In general, in anchor-based techniques, the placement of reference nodes (anchors) needs to be carefully planned to ensure that sensor nodes can communicate with a sufficient number of reference nodes. In contrast, in event-based protocols (as an example for anchor-free techniques), it has to be ensured that all sensors can be in range (or line-of-sight) of one or more signal generator, which may limit the scale of the sensor network.

### 3 Case Studies of Range-Free Localization Techniques

In the remainder of this chapter, we discuss a variety of range-free localization solutions that utilize area, hop count, or neighborhood information to position sensors, but also more recent event-based techniques that rely on event detection by sensor nodes to determine sensors positions.

**Fig. 3** Illustration of a sample network topology for the convex position estimation approach



### 3.1 Convex Position Estimation

One of the first area-based localization techniques determining positions exclusively on connectivity-induced constraints was introduced in [5]. In this approach, a network is represented as a graph with  $n$  nodes, with a subset of  $m$  nodes serving as reference nodes, i.e., their Cartesian positions are known (see Fig. 3, which shows reference nodes as black nodes, nodes with unknown positions as white nodes, and nodes that are able to communicate are connected via lines).

More formally expressed, the network is a graph with  $n$  nodes, where the positions of the first  $m$  nodes are known and expressed as  $(x_1, y_1, \dots, x_m, y_m)$ . The challenge is then to find the unknown positions  $(x_{m+1}, y_{m+1}, \dots, x_n, y_n)$  of the remaining  $n-m$  nodes such that all proximity constraints are satisfied. In [5], a centralized solution to this challenge is proposed, where nodes communicate their connectivity information to a single computer that solves this optimization problem.

This approach is based on finding solutions to linear and semidefinite programs that can be used to generate feasible positions for the nodes in a network. The semidefinite problem (SDP) is a generalization of the linear program (LP) [21], with the objective function  $c^T x$  and the following constraints:

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \mathbf{x}_1 \mathbf{F}_1 + \dots + \mathbf{x}_n \mathbf{F}_n < 0$$

$$\mathbf{A}\mathbf{x} < b$$

$$\mathbf{F}_i = \mathbf{F}_i^T.$$

The first constraint represents a matrix inequality on the cone of positive semidefinite matrices, i.e., the eigenvalues of  $\mathbf{F}(\mathbf{x})$  must be nonpositive, which is known as linear matrix inequality (LMI). Each node has a position  $(x, y)$ , allowing us to form a single vector containing all positions as

$$\mathbf{x} = [x_1 y_1 \dots x_m y_m x_{m+1} y_{m+1} \dots x_n y_n]^T.$$



Given this mathematical basis and assuming that the connectivity in a network can be represented as a set of convex position constraints (i.e., a convex set is one for which any two points in the set can be connected with a line entirely contained in the set), the work in [5] continues to propose solutions for convex constraint models for RF and optical communication systems. In the symmetric model where the communication range of each node can be represented as a circle, a connection between nodes can be expressed as a 2-norm constraint on the node positions. For example, the LMI can be determined as:

$$\|a - b\|_2 \leq R \Rightarrow \begin{bmatrix} I_2 R & a - b \\ (a - b)^T & R \end{bmatrix} \geq 0,$$

where  $R$  is the maximum range and node positions are expressed as  $a$  and  $b$ . Using Schur complements [2] transforms the quadratic inequality into an LMI with a  $3 \times 3$  matrix, where  $I_2$  represents the two-dimensional identity. Multiple LMIs can be stacked in diagonal blocks to form a single large SDP for the entire network.

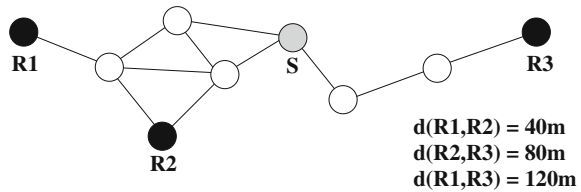
This approach can further be extended, e.g., instead of maximum radio ranges  $R$ , smallest ranges  $r_{ab}$  are assigned to each constraint (i.e., replace  $R$  in the above formula with  $r_{ab}$ ). Values for  $r_{ab}$  can be obtained during the initialization phase by varying the transmission power of the radios. Once a connection is first detected at a power  $P_0$ , the receiver node can calculate the maximum possible separation for reception at power  $P_0$ . This maximum separation  $r_{ab}$  can then be used to determine a tighter upper bound for each connection in the network. Another extension to this approach considers sensors with directional antennas or laser transmitters and receivers (optical communication), where antennas and detectors can be rotated until a signal is detected. This leads to cones representing the feasible sets and this can be used to reduce the size of the area where a sensor is believed to be, thereby reducing the localization error.

### 3.2 Ad Hoc Positioning System

As an example of a hop count based localization technique, the *ad hoc positioning system* (or *APS*) [22] provides a distributed connectivity-based localization approach that estimates node locations using a set of at least three reference nodes (the more reference nodes, the higher the accuracy obtainable in APS). An important aspect of APS is that it is based on the concept of *distance vector (DV)* exchange [18], where nodes periodically exchange their routing tables with their one-hop neighbors to obtain accurate network-wide connectivity (and routing) information.

APS supports various schemes, the simplest of them being the *DV-hop* approach. Here, each node maintains a table  $\{X_i, Y_i, h_i\}$ , where  $\{X_i, Y_i\}$  represents the location of node  $i$  and  $h_i$  is the hop distance between this node and node  $i$ . Using the routing table updates exchanged in DV, reference nodes will learn about the presence (and location) of other reference nodes and can then calculate an average hop size, called

**Fig. 4** Example of DV-hop localization



the *correction factor*  $c_i$ :

$$c_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_i}$$

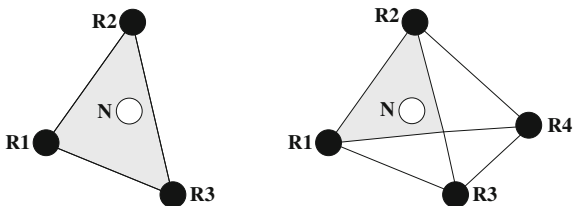
for all other reference nodes  $j$  ( $i \neq j$ ). The computed correction factors are also propagated throughout the network and given the locations and correction factors of the reference nodes, a sensor node is then able to estimate its own location. Figure 4 illustrates an example of DV-hop localization using three reference nodes (R1, R2, R3) and six sensor nodes, where sensor node S is attempting to estimate its position.

Each reference node knows the Euclidean distance and the minimum hop count between itself and all other reference nodes. Given this information, reference node R1 is able to compute its correction factor as  $(40 + 120)/(2 + 6) = 20$ , which indicates the estimated hop distance in meters. Similarly, reference node R2 determines its correction factor as  $(40 + 80)/(2 + 5) = 17.1$  and reference node R3 determines its correction factor as  $(80 + 120)/(5 + 6) = 18.2$ .

To ensure that each node will only use one correction factor (typically the one from the closest reference node), correction factors are propagated using controlled flooding, i.e., once a node has received a correction factor from one of its neighboring reference nodes, subsequent correction factors are ignored. For example, in Fig. 4, sensor node S may have received the correction factor from reference node R2 first (since it is closest in distance) and uses R2’s correction factor to determine its distances to all reference nodes. That is, S computes its distance to R1 as  $17.1 \cdot 3$ , its distance to R2 as  $17.1 \cdot 2$ , and its distance to R3 as  $17.1 \cdot 3$ .

A variation of this approach is the *DV-distance* method, where distances between neighboring nodes are determined using radio signal strength measurements. As before, these distances are propagated throughout the network (but this times in meters instead of hops). While this approach provides finer granularity (not all hops are estimated to be the same size), it is also more sensitive to measurement errors. Another variation of this technique is to use true Euclidean distances. Here, a node must have at least two neighbors that have distance measurements to a reference node, where the distance between the two neighbors is known. Based on this information, simple trigonometric relationships can be used to determine the distance of a node to a reference node.

**Fig. 5** Location estimation based on the intersection of the triangles formed by reference nodes

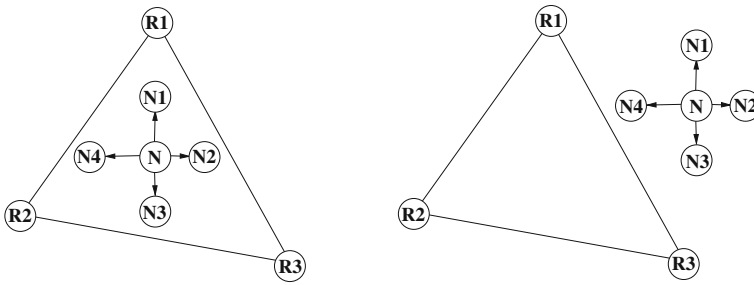


### 3.3 Approximate Point In Triangulation

A well-known example of an area-based range-free localization technique is *APIT* (or *Approximate Point In Triangulation*) [6], which, similar to APS, also requires the presence of reference nodes with well-known locations. The main idea behind APIT is to consider the triangles formed by different sets of three reference nodes and to determine whether a node resides within or outside of these triangles. By identifying the triangles a node resides in, it is possible to narrow down the node’s potential locations.

The key procedure in this approach is the *Point In Triangulation (PIT)* test, which allows a node to determine these triangles. In this test, once a node has determined the locations of a set of reference nodes (via location updates similar to the APS protocol), it tests whether it resides within or outside of each triangle formed by each set of three reference nodes. Consider a set of three reference nodes  $R1$ ,  $R2$ , and  $R3$ . A node  $N$  is then situated outside the triangle formed by these reference nodes, if there exists a direction such that a point adjacent to  $N$  is either further or closer to  $R1$ ,  $R2$ , and  $R3$  simultaneously. If no such direction exists, node  $N$  is inside the triangle and the triangle formed by the three reference nodes can be added to the set of triangles in which  $N$  resides. This technique is illustrated in Fig. 5. In this example, node  $N$  resides within the triangle formed by reference nodes  $R1$ ,  $R2$ , and  $R3$ , shown in the left graph. On the right of Fig. 5, a fourth node,  $R4$  has been added (i.e.,  $N$  has learnt about the presence and location of the fourth reference node), which leads to additional triangles formed the sets  $\{R1, R2, R4\}$ ,  $\{R1, R3, R4\}$ , and  $\{R2, R3, R4\}$ . Using this additional information, node  $N$  is able to reduce the size of the area it resides in.

However, this test as described above is not feasible in practice since it would require that nodes could be moved in any direction. Therefore, instead of this *perfect PIT* test, an *Approximate PIT (APIT)* test can be used as long as network density is sufficiently large. In APIT, node movement is emulated using neighbor information that is exchanged between nodes using beacon messages, e.g., using such beacon messages, a ranking among nodes in reach of a reference node can be established based on their signal strengths (and therefore distances). For example, if no neighbor of node  $N$  is further from or closer to all three reference nodes  $R1$ ,  $R2$ , and  $R3$  simultaneously, node  $N$  assumes that it resides inside the triangle formed by the three reference nodes; otherwise node  $N$  assumes that it is outside the triangle.



**Fig. 6** Examples of APIT test scenarios

Figure 6 illustrates this approach, where the left graph shows an example where node  $N$  is within the triangle formed by the three reference nodes and the right graph shows an example where node  $N$  is outside. In the left example, node  $N$  has four neighbors  $N[1..4]$ , none of which is simultaneously closer to or further away from all three reference nodes. Therefore node  $N$  determines that it must reside within the triangle formed by the reference nodes. In contrast, the graph on the right shows an example where node  $N$  determines that it must reside outside the same triangle since node  $N4$  is closer to all three reference nodes than node  $N$ , while node  $N2$  is further away from all reference nodes, compared to node  $N$ .

In this approach, it is possible that a node's determination is wrong, simply because only a finite number of neighboring nodes (and therefore directions) can be used to evaluate whether the node is inside or outside the reference node triangle and small errors in signal strength measurements can falsify the result. For example, in the left graph of Fig. 6, if the signal strength measurements from node  $N4$  indicate that it must be further away from reference node  $R2$  than node  $N$ , node  $N$  would incorrectly conclude that it resides outside the triangle. Such measurement errors are very likely to occur due to the signal strength easily being affected by obstacles, multi-path propagation, etc.

Once the APIT test completes, a position estimate can be computed as the center of gravity of the intersection of all triangles in which  $M$  resides in. While the APIT approach provides a simple way of determining locations without the need for additional hardware on the sensor nodes, its main disadvantages are the effect of distance measurement errors on localization accuracy and the need for large network density (i.e., sensor nodes must have several reference nodes that can be reached and they must have several neighboring nodes to perform the APIT test).

### 3.4 Ring Overlapping Localization Techniques

Another area-based ranging technique that has been proposed recently [15, 39] is based on forming rings around reference nodes, where a sensor node estimates its position to be within the intersection of multiple such rings. The goal of these

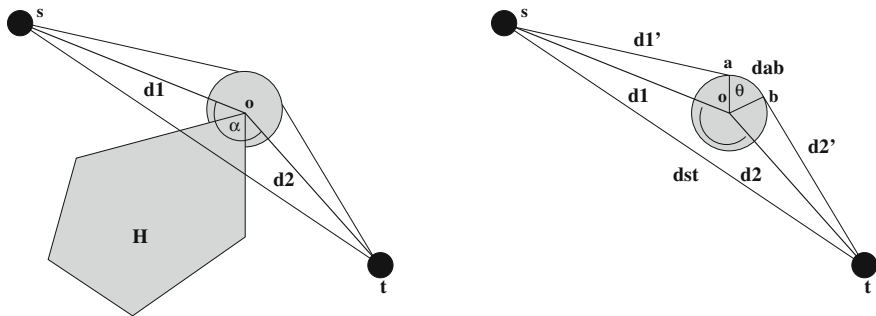


Fig. 7 Principle of the overlapping circles technique

techniques is to provide increased localization accuracy (e.g., compared to hop-based techniques), while keeping the cost of communication and computation low.

For example, in [15], the authors propose a technique called “Annulus Intersection and Grid Scan” or AIGS. In this technique, a reference node X is considered as the center of a circle, where the radius of the circle is the distance to another reference node Y. If the reference node density is sufficient, each reference can thereby form multiple such rings. In AIGS, a sensor node A with an unknown location tries to determine two specific rings around each neighboring reference node: (i) the circle with the shortest radius among all circles that contain node A and (ii) the circle with the largest radius among all circles that do not contain it. To select such rings, node A must be within the annulus of these circles and therefore, the annulus of the circles can be used to estimate the location of node A. In AIGS, the center of the intersections of the overlapping rings is taken as the estimated coordinate of the unknown node A.

This basic principle is shown in Fig. 7. This figure shows several reference nodes, with the circles indicating the radius of the closest and furthest neighbor beacon nodes. The shaded area indicates the intersection of the circles and the unknown node estimates its position to be in the center of this region. In detail, the algorithm performs the following steps. Note that the algorithm assumes that all nodes use omnidirectional antennas, all nodes are randomly deployed in 2-dimensional space, all nodes have the same transmit power, and there is no node mobility.

- *Step 1:* Localization is based on the transmission of two beacon messages by the reference nodes. The first one contains the coordinates of the reference node (x, y) and each receiver records the reference node’s location and signal strength. In the second beacon, each reference node broadcasts its view of the network, i.e., the information previously received from other reference nodes. All sensor nodes receiving these beacons also record this information.
- *Step 2:* A sensor A with unknown location successively selects a reference node that is within its one-hop range and uses this node as the center of a number of circles, each with a radius that equals the distance between the reference node and one of its neighboring reference nodes. Then node A attempts to find the smallest annulus that exactly contains node A. If there is such an annulus, the algorithm continues

in Step 3, otherwise the process is repeated until all neighboring reference nodes have been investigated.

- *Step 3:* Node A now attempts to find the intersection area of the annulus; if it is the first annulus, then the intersection area will be itself, otherwise it will be the intersection of the newly discovered annulus the previously determined intersection area.
- *Step 4:* Node A repeats Step 2 until all neighboring reference nodes have been investigated.
- *Step 5:* If there is an intersection area, node A now computes the centroid area and takes the resulting location as its coordinate. Otherwise, node A computes its coordinates using a centroid localization algorithm.

In comparison to DV-Hop, AIGS performs well in both overhead and accuracy. The results presented in [15] indicate that in a network with 30 % reference nodes (out of 200 nodes in total), the localization error can be significantly reduced, particularly for large communication ranges. For example, while DV-Hop had an average localization error of 30 % (which stayed constant with varying communication ranges), the localization error for AIGS was less than 20 % for low communication ranges and less than 5 % for high communication ranges. The computational complexity of AIGS is similar to DV-Hop, while the communication overhead is lower (beacons are exchanged only within the one-hop neighborhood).

### 3.5 Multidimensional Scaling

A popular technique to achieve range-free localization (and an example for localization based on hop count) is multidimensional scaling (MDS) [1, 31], which has its roots in psychometric and psychophysics, and is a set of data analysis techniques that display the structure of distance-like data as a geometrical picture. MDS requires a powerful centralized device (e.g., a base station connecting a WSN to the rest of the world), which collects topological information from the network, determines the nodes' locations, and propagates this information back into the network.

With MDS, the network is represented as an undirected graph of  $n$  nodes, where the edges in the graph represent connectivity information. Further, a subset of  $m$  nodes represents reference nodes (i.e., nodes with well-known locations). The goal of MDS is to preserve the distance information, assuming that the distances between all pairs of nodes are known, such that the network can be recreated in the multidimensional space. The result is then an arbitrarily rotated and flipped version of the original network layout.

#### Classical MDS and MDS-MAP

One of the simplest versions of MDS, called classical MDS, has a closed form solution allowing for efficient implementations. In this version, the process is as follows. First, assume a matrix representing the squared distances between nodes:

$$D^2 = c1' + 1c' - 2SS',$$

where  $I$  is an  $n \times n$  vector of ones,  $S$  is the similarity matrix for  $n$  points, where each row represents the coordinates of point  $i$  along  $m$  coordinates,  $SS'$  is called the scalar product matrix, and  $c$  is a vector consisting of the diagonal elements of the scalar product matrix. Next, by multiplying both sides of this equation by the centering matrix

$$T = I - \frac{11'}{n},$$

where  $I$  is the identity matrix and  $1$  is again a vector of ones, leads to

$$TD^2T = T(c1' + 1c' - 2SS')T = Tc1'T + T1c'T - T(2B)T,$$

where  $B = SS'$ . Since centering a vector of ones yields a vector of zeros, this can be simplified to

$$TD^2T = -T(2B)T.$$

Next, multiplying both sides with  $-1/2$  results in

$$B = -\frac{1}{2}TD^2T.$$

$B$  is a symmetric matrix and can therefore be decomposed into

$$B = Q\Lambda Q' = Q'\Lambda^{\frac{1}{2}}(Q\Lambda^{\frac{1}{2}})' = SS'.$$

This can then be decomposed into

$$B = Q\Lambda Q' = \left(Q'\Lambda^{\frac{1}{2}}\right)\left(Q\Lambda^{\frac{1}{2}}\right)' = SS'.$$

Finally, given  $B$ , the coordinates of  $S$  can now be determined by eigendecomposition:

$$S = Q\Lambda^{1/2}.$$

This approach is used in [31], where a localization method called MDS-MAP is proposed. In MDS-MAP, a distance matrix  $D$  is constructed using an all pairs shortest path algorithm (e.g., Dijkstra's), with  $d_{ij}$  representing the distance (i.e., the number of hops) between two nodes  $i$  and  $j$ . In the next step, the classical MDS approach as described in this section is used to arrive at an approximate value of the relative coordinate of each node. These relative coordinates can then be transformed into absolute coordinates by aligning the relative coordinates obtained for the reference nodes with their absolute coordinates. Using least-squares minimization, even more

refined location estimates could be obtained. Another extension to this approach is to divide a WSN into overlapping regions, where localization is performed, as described above, separately for each of these regions. The resulting local *maps* can then be stitched together to arrive at a global network topology. This is achieved using nodes that appear in multiple maps, i.e., nodes that are shared between adjacent regions. The outcome is an improved performance for networks that are shaped irregularly, by avoiding the use of distance information of nodes that are far away from each other. Another modification of this approach is to implement it as a distributed solution (instead of the centralized solution used so far, which relies on collecting all nodes' information at a central location), which has been proposed in [29].

### 3.6 Rendered Path Localization

A challenge in range-free localization is to ensure that large numbers of reference nodes are uniformly distributed across the network. If this is not given, many solutions will fail or deliver poor results, particularly in *anisotropic* WSNs with possible holes. In anisotropic networks, the Euclidean distances between pairs of nodes may not closely correlate with the hop counts, because holes in the network force paths between the nodes to have curves. This is particularly common when nodes are deployed randomly, with denser areas of nodes alternating with sparser areas, leading to such holes.

As a consequence, the Rendered Path (REP) [14] protocol addresses this problem by path rendering and virtual hole construction operations in a distributed manner. The basic idea behind REP is to identify the boundaries of holes in a network and to label the boundary nodes of different holes with different colors. When the shortest path between two nodes passes the holes, it is rendered with the colors of the boundary nodes, i.e., a path can be rendered by multiple colors. By passing holes, the path is segmented according to the intermediate colorful boundary nodes. In addition, REP also generates virtual holes to augment and render the shortest path by calculating the Euclidean distance between two nodes based on the distance and the angle information along the rendered path.

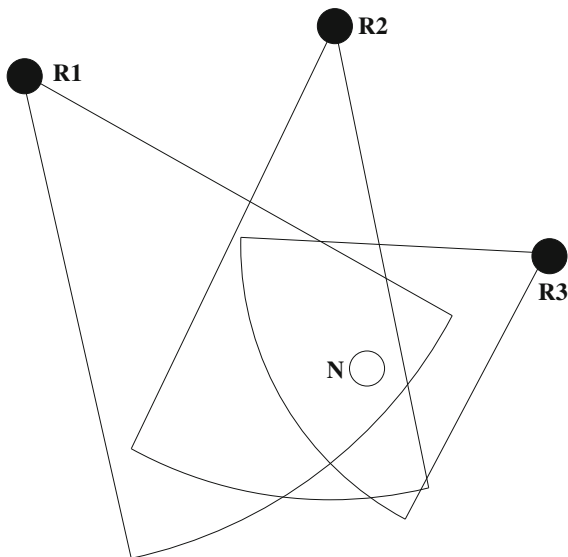
Consider two nodes  $s$  and  $t$ , separated by a hole  $H$  (see Fig. 8), where REP renders a shortest path  $P_G(s, t)$  between these two nodes.

Every point on the boundary of hole  $H$  is colored with the color of  $H$  and these points are said to be  $H$ -colored. When a hole between nodes  $s$  and  $t$  exists, the shortest path must intersect with the hole boundaries and REP knows (from the colored points and their colors) how many different holes a path has passed, i.e., the number of passed holes is equal to the number of different rendered colors.

Figure 8 illustrates a simple basic scenario for REP. The figure shows a convex hole  $H$  at point  $o$ , which is  $H$ -colored. The shortest path between  $s$  and  $t$  ( $P_{st}$ ) is segmented into  $so$  and  $ot$ , where we assume that  $|so| = d_1$  and  $|ot| = d_2$ . Based on the law of cosines, the triangle formed by  $s$ ,  $t$ , and  $o$  has the following mathematical relationship:



Fig. 8 Basic scenario of REP



$$|st|^2 = |so|^2 + |ot|^2 - 2|so| \cdot |ot| \cdot \cos \angle sot$$

and consequently:

$$d_{st} = \sqrt{d_1^2 + d_2^2 - 2d_1d_2\cos\alpha}.$$

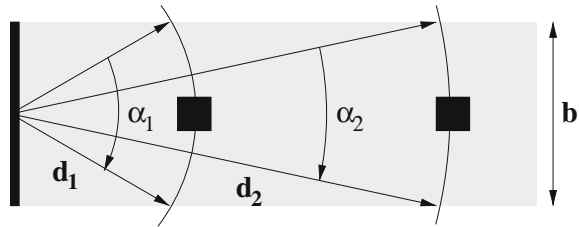
In order to obtain the angle between  $so$  and  $ot$ , REP creates a virtual hole (approximately round-shaped) around  $o$ , with radius  $r$ , such that the former shortest path  $s-o-t$  is blocked. The center  $o$  of this virtual hole is then called *focal point*. The color of the virtual hole is the color of  $o$  and the new shortest path between  $s$  and  $t$  is therefore augmented to bypass the enlarged hole. The right graph in Fig. 8 illustrates that the new shortest path  $P_{st}^*$  can be segmented into three pieces: an uncolored line  $sa$  (of length  $d_1'$ ), an  $o$ -colored arc (of length  $d_{ab}$ ), and another uncolored line  $bt$  (of length  $d_2'$ ). The length  $d_{ab}$  reflects angle and can be determined as

$$\alpha = 2\pi - \frac{d_{ab}}{r} - \arccos \frac{r}{d_1} - \arccos \frac{r}{d_2}.$$

The distance  $d_{st}$  between nodes  $s$  and  $t$  can then be determined from the length information on the two rendered paths  $P_{st}$  and  $P_{st}^*$ .

The focus in this discussion and illustration has been on a simple scenario with a single hole separating two nodes. However, in real settings, it may often occur that paths interact with hole boundaries at more than one point, e.g., when a path intersects along a segment of a boundary with a convex hole or at several discrete points with a concave hole. However, solutions to these scenarios can be similarly derived as for the situation with a single boundary point.

**Fig. 9** Principle of SeRLoC localization



### 3.7 Secure Range-Independent Localization

The Secure Range-Independent Localization (or SeRLoC) protocol [12] is concerned with providing range-free localization in untrusted environments. To achieve secure localization, SeRLoC is primarily concerned with providing a decentralized solution that is resource-efficient, but also robust against security threats.

Toward this end, SeRLoC assumes that some nodes in the network know their location and orientation, e.g., the positions of these *locators* can be obtained using GPS receivers. Another assumption is that locators are equipped with sectorized antennas with  $M$  sectors (while regular sensor nodes are equipped with omnidirectional antennas). A certain known directivity gain  $G(M)$  and an idealized angular reception are also assumed. Sensors then determine their positions using beacons that are transmitted by the locators; this principle is illustrated in Fig. 9. Each locator transmits different beacons at each antenna sector, where each beacon contains the coordinates of the locator and the angles of the antenna boundary lines. Sensor nodes collect these beacons and use them to determine whether they are within a specific antenna sector for the transmitting locator. Each sensor also knows the maximum transmission range for each locator, which can further be used to limit the size of the sector. Once beacons from all locators have been collected, a sensor node can determine the overlap between the sectors to identify the region within which they must reside.

Once this region has been determined, the sensor node then computes the center of gravity and assumes that this is the sensor's location.

As previously mentioned, a primary goal of SeRLoC is to provide localization in untrusted environments and therefore, SeRLoC includes a variety of security mechanisms. First, encryption is used to protect localization information, i.e., all beacon messages are encrypted using a shared global symmetric key (pre-loaded on sensor before deployment). Every sensor also shares a symmetric pairwise key with every locator.

Next, the use of a shared symmetric key does not identify the source of the messages a node receives. Therefore, a malicious node could impersonate multiple locators. To prevent malicious nodes from injecting false localization information into the network, sensors must authenticate the source of the beacons using collision-resistant hash functions. Each locator  $L_i$  has a unique password  $PW_i$ , which is blinded with the use of a collision-resistant hash function such as MD5 [26]. Due to

the collision resistance property, it is computationally infeasible to find a value  $PW_j$ , such that  $H(PW_i) = H(PW_j)$ , with  $PW_i \neq PW_j$ . The hash sequence is generated using the following approach:

$$H^0 = PW_i, H^i = H(H^{i-1}), i = 1, \dots, n,$$

with  $n$  being a large number and  $H^0$  never revealed to any sensor. Each sensor has a pre-loaded table containing the identifier of each locator and the corresponding hash value  $H^n(PW_i)$ . Now, assume that locator  $L_i$  wants to transmit its first beacon and sensors initially only know the hash value  $H^n(PW_i)$ . The beacon message contains  $(H^{n-1}(PW_i), j)$  with the index  $j = 1$  (i.e., the first hash value published). Every sensor receiving this beacon can now authenticate the locator only if  $H(H^{n-1}(PW_i)) = H^n(PW_i)$ . Once verified, the sensor replaces  $H^n(PW_i)$  with  $H^{n-1}(PW_i)$  in its table and increases the hash counter by one.

### 3.8 Event-Driven Localization

Recent work on localization has resulted in another type of range-free solutions, called *event-driven* localization. In these techniques, certain types of events, such as radio signals, light or laser beams, sound blasts, etc., are issued and nodes in the network perform event detection to determine their positions. This section provides an overview of solutions that belong to this category of localization algorithms.

#### The Lighthouse Approach

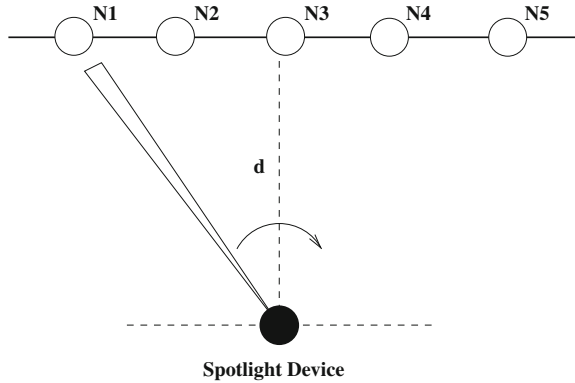
The idea behind event-based localization is to determine distances, angles, and positions using concrete events such as radio waves, beams of light, or acoustic signals transmitted by a reference node and received by a sensor node. In the Lighthouse location system [27], sensor nodes can determine their location with high accuracy without the need for additional infrastructure components besides a base station equipped with a light emitter.

The concept behind the Lighthouse approach is illustrated in Fig. 10. It uses an idealistic light source, which has the property that the emitted beam of light is parallel, i.e., the width  $b$  remains constant. The light source rotates and when the parallel beam passes by a sensor, it will see the flash of light for a certain period of time  $t_{\text{beam}}$ , which varies with the distance between the sensor and the light source (since the beam is parallel). The distance  $d$  between the sensor and the light source can then be expressed as

$$d = \frac{b}{2 \sin \frac{\alpha}{2}},$$

where  $\alpha$  expresses the angle under which the sensor sees the beam of light as follows:

**Fig. 10** The lighthouse localization approach (*top view*)



$$\alpha = 2\pi \frac{t_{beam}}{t_{turn}}$$

Here,  $t_{turn}$  is the time the light source takes to perform a complete rotation. The beam width  $b$  is known and a constant, therefore, a sensor can then calculate the value of  $t_{beam}$  as

$$t_{beam} = t_2 - t_1$$

and

$$t_{turn} = t_3 - t_1,$$

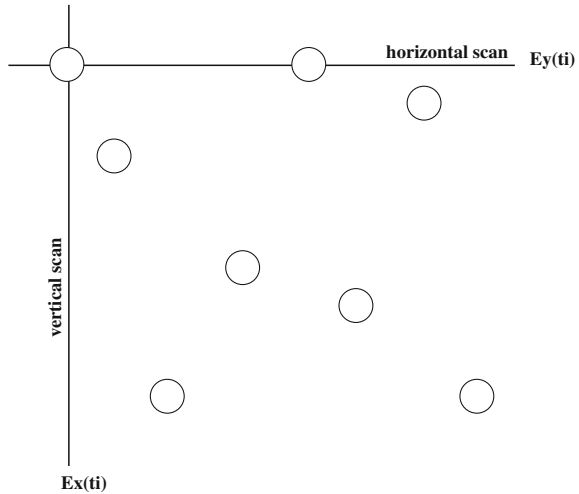
where  $t_1$  is the time the sensor sees the light for the first time,  $t_2$  is the time the sensor no longer sees the light, and  $t_3$  is the time when the sensor sees the light again.

So far, we assumed that the beam width  $b$  is constant, independent from the distance between light source and receiver node. However, perfectly parallel light beams are difficult to realize in practice and even very small beam spreads can result in significant localization errors. For example, a beam width of 10 cm and a beam spread of leads to a beam width of 18.7 cm at a distance of 5 m. An additional requirement is that the beam width should be as large as possible to keep inaccuracies small. To achieve this, two laser beams can be used to create the outline of a “virtual” parallel beam since the sensor nodes are only interested in detecting the edges of the virtual beam represented by the two laser beams.

**The Spotlight Approach**

Similar to the Lighthouse approach, another example for a range-free localization method that does not require reference nodes and uses events for localizations is Spotlight [35, 36]. The key concept behind this approach is to generate events (e.g., light beams), where the time when an event is detected, together with certain spatio-temporal properties of the generated events, can be used to derive sensor node locations. Key assumptions for this approach are that a Spotlight device can

**Fig. 11** Point scan event distribution function



generate events that can be detected by sensor nodes, the Spotlight device knows its location and the time when the event was created, the sensors in the network are time-synchronized, and the sensors have line-of-sight (LOS) to the Spotlight device.

The Spotlight protocol performs the following steps:

1. A Spotlight device distributes events  $e(t)$  in the space  $A$  over a certain period of time.
2. Sensor nodes record the time sequence  $T_i = \{t_{i1}, t_{i2}, \dots, t_{in}\}$  at which events are detected.
3. After the event distribution, these collected times are transmitted to the Spotlight device.
4. The Spotlight device determines the locations of the sensor nodes using the collected time sequences and the known Event Distribution Function  $E(t)$ .

The Event Distribution Function  $E(t)$  is the core technique and an essential component of the Spotlight approach and multiple solutions for  $E(t)$  have been proposed:

- In the **Point Scan** function, it is assumed that the sensors nodes are placed along a straight line, as shown in Fig. 11. The Spotlight device generates events, such as light spots, along this straight line with a constant speed. This leads to a series of event detections at the sensor nodes, where the set of timestamps when such events were detected by node  $i$  is described as  $T_i = \{t_{i1}\}$ . The event distribution function is described as:

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

where  $A$  describes the straight line going from coordinates  $(0,0)$  to  $(0,l)$ , i.e.,  $A = [0, l]$  and The localization function can then be derived as

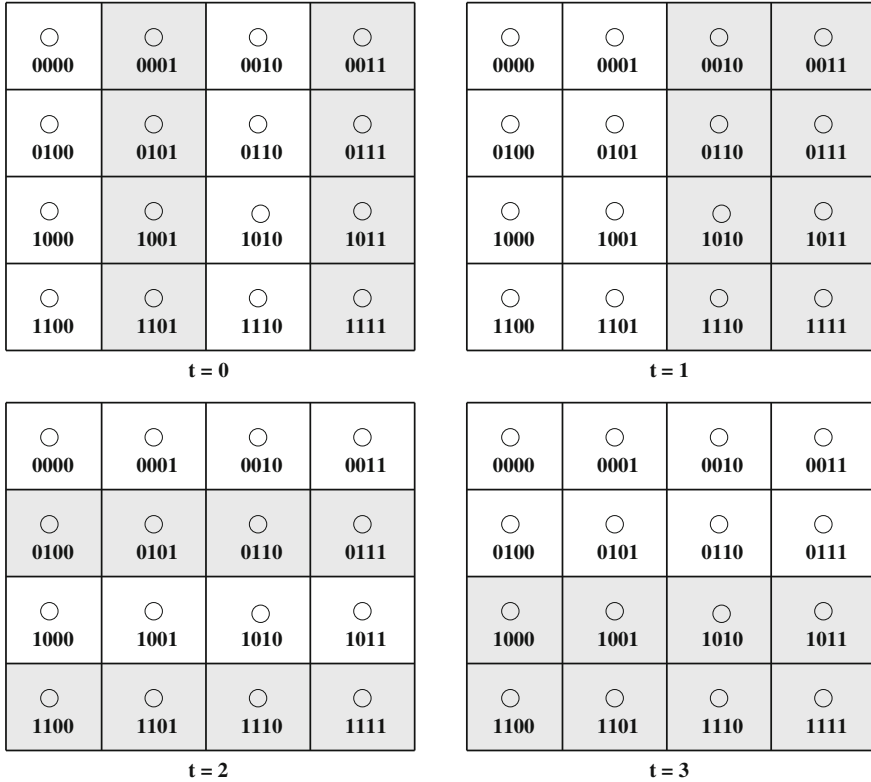


Fig. 12 Line scan event distribution function

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

- In the **Line Scan** function (Fig. 12), nodes are distributed in a two dimensional plane ( $A = [l \times l] \subset R^2$ ) and it is assumed that the Spotlight device can generate an entire line of events simultaneously (e.g., such as a laser). Given a scanning speed of  $s$  and a set of event detection timestamps (for node  $i$ ), the line scan function is defined as

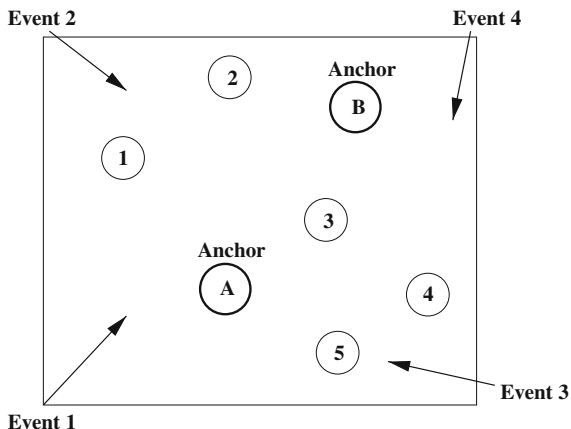
$$E_x(t) = \{p_k \mid k \in [0, l], p_k = (t * s, k)\} \text{ for } t \in [0, l/s] \text{ and}$$

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

The localization function can then be derived from the intersection of the vertical and horizontal scans, i.e.,

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

**Fig. 13** The area cover implementation



- In the **Area Cover** function, Spotlight devices that can cover an entire area are being used. With this approach, a sensor field  $A$  is partitioned into multiple sections, each assigned a unique binary identifier, called *code* (Fig. 13). Each section  $S_k$  within area  $A$  has then a unique code  $k$  and events are generated according to the encoded bit in the bitstring. The area cover function is then:

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

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

The localization function can then be described as

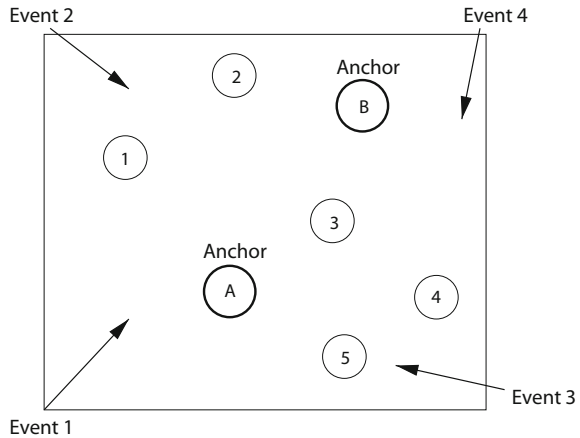
$$L(T_i) = \{p \mid p = COG(S_k), BIT(k, t) = true \text{ if } t \in T_i, BIT(k, t) = false \text{ if } t \in T - T_i\}.$$

### Multi-Sequence Positioning

Finally, the Multi-Sequence Positioning (MSP) approach is another example of event-based localization [39] and works by extracting relative location information from multiple simple one-dimensional orderings of sensor nodes.

For illustration of the concept behind MSP, consider Fig. 13, which shows a small sensor network with five nodes with unknown locations and two reference nodes with known locations. There are multiple event generators placed around the network and each generator produces an event at different points in time. For example, such events can be ultrasound signals or laser scans with different angles. The nodes in the sensor network detect these events at different times, depending on their distances to the event generators. For each event, we can then establish a node sequence, i.e., a node ordering (which includes both the sensor and the

**Fig. 14** Basic concept of MSP



**Table 1** Node sequence for the four events generated in Fig. 14

Event 1	A - 1 - 5 - 3 - 4 - 2 - B
Event 2	1 - 2 - B - A - 3 - 5 - 4
Event 3	4 - 5 - B - 3 - A - 2 - 1
Event 4	B - 2 - 3 - 1 - 4 - A - 5

anchor nodes) based on the sequential detection of the event. An example of such ordering for the situation shown in Fig. 13 is presented in Table 1. Then, a multi-sequence processing algorithm narrows the potential locations for each node to a small area and finally, a distribution-based estimation method estimates the exact locations.

The main concept behind the MSP algorithm is to split a sensor network area into small pieces by processing node sequences. For example, in the example in Fig. 14, performing a straight-line scan from top to bottom results in a node sequence 2-B-1-3-A-4-5. The basic MSP algorithm uses two straight lines to scan an area from different directions, treating each scan as an event. A left-to-right scan leads to a node sequence 1-A-2-3-5-B-4.

Since the reference nodes have well-known locations, the two reference nodes in Fig. 14 split the area into nine parts. This process can be extended to cut the area into smaller pieces by increasing the number of reference nodes and scans (from different angles). The basic MSP algorithm processes each node sequence to determine the boundaries of a node (by searching for the predecessor and successor reference nodes for the node) and shrinks the location area of this node according to the newly obtained boundary information. Finally, a centroid estimation algorithm sets the center of gravity of the resulting polygon as the estimated location of the target node.



## 4 Discussion and Comparison

While the primary goal of range-free techniques is to produce localization solutions that are inexpensive (and ideally also with low computational and communication overheads), while typically providing less accurate results compared to range-based techniques. This section qualitatively compares the different localization schemes described in this chapter, discusses their underlying techniques, including the advantages and disadvantages of these techniques.

### Centralized Versus Distributed Localization

In centralized localization schemes, node information is collected by a centralized server, which then executes the localization algorithm. Then, the server can use the computed locations for a variety of sensor network management tasks (e.g., to establish routes) or it can send the locations to the individual sensor nodes. Since the server can learn the entire network topology, the outcome can be optimized. On the other hand, the computational requirements of the server are typically very high. Additional downsides of a centralized approach are the large communication overheads (i.e., topology information must be collected centralized and location information may have to be disseminated to all nodes) and that location changes may take a long time to be recognized (i.e., these techniques work best in stationary networks).

In a distributed localization scheme, each sensor node independently determines its location, typically by exchanging information with neighboring sensor nodes, communicating with neighboring reference nodes, or by observing events (such as light beams). In contrast to centralized solutions, the overheads (communication and computation) are typically lower (making distributed solutions also more scalable), but since every node only knows localized information, the localization error tends to be larger.

Localization protocols such as MDS, convex position estimation, and Spotlight are centralized solutions due to their reliance on network-wide information for positioning. Other protocols such as APIT, APS, REP, and SeRLoc are distributed by nature. Finally, some protocols, e.g., hop-based positioning schemes, can be implemented both in distributed or centralized fashion.

Table 2 summarizes some of the key advantages and disadvantages of typical state-of-the-art centralized and distributed localization schemes. Generally, the increased communication overhead in centralized localization schemes limits the scalability of these schemes, but can lead to significantly more accurate positions. On the other hand, with distributed schemes, localization is performed locally only, leading to typically scalable solutions that produce results with larger errors.

### Communication and Computational Complexity

The localization schemes discussed in this chapter vary widely in their communication and computational overheads (and consequently also their energy requirements).

**Table 2** Comparison of centralized and distributed localization schemes

	Centralized schemes	Distributed schemes
Sensor cost	Low	Low-Medium
Number of anchors	Small	High
Accuracy	Medium	Low
Scalability	Medium	High
Computation overhead	Sensor: Low Server: High	Low
Communication overhead	High	Low
Robustness to anisotropic topologies	Low	Medium

For example, centralized solutions typically have large communication overheads, where network-wide information is collected centrally and positioning results may have to be reported back to each individual sensor node in the network. Algorithms based on graph theory (e.g., convex position estimation) or data analysis techniques (e.g., MDS) also have high computational requirements on the centralized server. For example, MDS has a computational cost of [24, 30], although variants of MDS with improved scalability have also been proposed [32], including a distributed version of MDS [9]. Due to the high communication overheads, centralized solutions typically scale poorly and react very slowly to topology changes. However, compared to the requirements for the server, the computational requirements for sensor nodes are usually very limited, which can limit the energy costs.

In contrast, in decentralized solutions, each sensor nodes communicates with neighboring nodes to determine its position. The overall communication overheads in the network can therefore be lower, but each node must execute its own version of the localization scheme and the overheads introduced depend on the complexity of the localization algorithm. For example, APIT is a rather computationally intensive distributed algorithm that performs best when the number of reference nodes within the vicinity of a sensor node is large (i.e., greater than 20). However, with 20 reference nodes, the intersections of 1140 areas need to be analyzed. On the positive side, since localization is performed locally, distributed schemes can detect changes in topology rather quickly (e.g., depending on the frequency of information exchange between reference nodes and sensor nodes) and therefore the latencies in positioning can be kept low.

### Localization Accuracy

The accuracy of localization schemes and the error in the computed positions depend on a variety of parameters, including the number of reference nodes. For example, as mentioned previously, APIT performs best when the number of reference nodes “visible” to a sensor node is rather large, which requires a highly dense network of reference nodes. Further, schemes such as APIT perform best when the reference nodes are randomly distributed throughout the network, which is not always feasible in a real deployment. Compared to distributed solutions, centralized schemes have

the advantage that they collect network-wide information, thereby allowing them to optimize their positioning algorithms. As a consequence, centralized localization algorithms typically provide positions with higher accuracy than distributed solutions that determine sensor location using only limited (local) information.

The achievable accuracy also depends significantly on the network topology, e.g., REP applies a path rendering technique to remove the impact of holes in the network. Simulations have shown that the accuracy of REP is significantly better than both APS and APIT in isotropic networks [6, 23]. However, REP assumes (i) that the network has high node density to allow it to achieve accurate boundary recognition, (ii) that the shortest path between two nodes is close to a straight line (if there are no holes), and (iii) that the length of an arc can be estimated from node connectivity along the circle boundary.

Schemes based on concepts in graph theory, such as MDS, can provide accurate results if the network is not too sparse. For example, simulation results [30] have shown that connectivity-based MDS methods achieve an average localization error of  $0.31R$  (where  $R$  is the radio range in a randomly deployed network with mean 1-hop connectivity of 12.1).

In localization schemes that rely on localization events, a typical goal is to utilize an asymmetric system design to keep the overheads at the sensors low, i.e., sensor nodes can use simple techniques to detect the events. In the Lighthouse approach, an accuracy of 10 cm in a square space of  $5\text{ m} \times 5\text{ m}$  has been demonstrated when the event emitter is carefully calibrated. The event distribution needs to be very accurately timed and typically line-of-sight is required for event detection. However, uneven terrain and the reliance on mechanical parts (e.g., rotating emitter) make careful calibration difficult. In contrast, MSP has the advantage that events are emitted by multiple event sources in different locations of the network (however, a sensor node must be able to detect events from at least two different emitter sources) and that precise control of event generation is not required, i.e., events can be generated anytime (and only event observation is necessary for localization).

Tables 3 and 4 summarize key advantages and disadvantages of the localization protocols discussed in this chapter. The goal is to emphasize the key functional and performance differences between the different techniques, to facilitate the selection of an appropriate localization solution given a specific network topology or application and to help identify potential research directions.

## 5 Research Directions

The last decade has seen the emergence of a variety of localization techniques and protocols, most of which belong to the class of area- or hop-based schemes. Event-based systems belong to a newer category that has received a significant amount of attention recently. In the area of event-based systems, more works is needed to address challenges such as accurate calibration, positioning accuracy, and how to maintain line-of-sight. When line-of-sight is not given, a secondary localization technique can

**Table 3** Summary of localization techniques (part 1)

Localization protocol	Type	Centralized or distributed	Reference nodes	Security features
Convex Point Estimation	Area	Centralized	Medium-high number required	No
APS	Hop	Distributed	Medium-high number required	No
APIT	Area	Distributed	Many required	No
MDS	Hop	Centralized	None	No
REP	Hop	Distributed	Few required	No
Ring Overlapping	Area	Distributed	Medium number required	No
SeRLoc	Area	Distributed	Sectored antennas	Yes
Lighthouse	Event	Distributed	Light emitter (1+)	No
Spotlight	Event	Centralized	Event generator	No
MSP	Event	Centralized	Event generator (2+)	No

**Table 4** Summary of localization techniques (part 2)

Localization protocol	Overhead (comput.)	Overhead (commun.)	Accuracy	Scalability
Convex point estimation	High (server) Low (sensor)	Medium	Good	Medium
APS	Low	Low	Medium-Good	Good
APIT	High	Low	Medium	Good
MDS	High	Medium	Medium	Medium
REP	High	Medium	High (anisotropic) Medium (regular)	Good
Ring Overlapping	Medium	Low	Medium-Good	Good
SeRLoc	Medium	Low	Medium-Good	Good
Lighthouse	Low	Low	Good	Medium
Spotlight	Medium (server) Low (sensor)	Low	High	Medium
MSP	Medium (server) Low (sensor)	Medium	Good-High	Good

be employed, possibly with reduced accuracy. The concept of *multimodal localization* (i.e., use of multiple localization techniques simultaneously) is not only attractive because of its robustness, but also its ability to further reduce localization error and to provide verifiability. Such integration of multiple techniques could also be used to combine range-based with range-free technologies in order to take advantage of the strengths of each approach. For example, range-based techniques can provide high accuracy, but often require additional ranging hardware, extensive environmental profiling, and careful system tuning and calibration. On the other hand, range-free techniques can be more economic at the resource-constrained sensors. More research is required to investigate new protocols that combine cost-effectiveness and accuracy.

Among the protocols discussed in this chapter, only SeRLoc addresses the need for security in wireless sensor networks. Localization is critical to most sensor networks and attacks can render them ineffective. However, very little work has been done to provide robust and secure localization schemes. SeRLoc has the ability to protect against wormholes, Sybil attacks, and attacks intended to compromise sensor nodes. While it provides some security, more work will be needed, especially for sensor network application in emergency response or military settings. In [11], the authors attempt to address some of the shortcomings of SeRLoc, specifically focusing on providing robust localization and verification of the location claim of a sensor node. The work in [11] does not require centralized management and is also not vulnerable to jamming.

More work is also needed to limit the impact of localization techniques on the communication and computation overheads, hardware costs, the need for centralized computation, and the need for dedicated event generators. Most sensor networks are composed of very low-cost technology, where the accuracy of localization can be affected by the hardware limitations of the sensors or by environmental noise affecting signal or event detection. By applying techniques such as uncertainty-based averaging, error propagation control, or data fusion, these impacts could be mitigated. Finally, there remain many challenges to be addressed in network topology and density, e.g., REP is an effort to address holes and irregular shapes in a network that could be problematic for connectivity-based localization techniques. These are complex challenges that remain open problems, especially for large networks and networks with varying densities.

Finally, most localization techniques focus on position estimation in two-dimensional space and very little prior work has focused on three-dimensional space. Techniques based on graph theory (e.g., convex position estimation) can be extended to three-dimensional space when the positions are extended to include a third component ( $x, y, z$ ), however leading to significantly increased computational complexity. Hop-based techniques can also be extended rather easily if all three coordinates of the reference nodes are known. Area-based techniques will typically require more significant changes to compute the intersections of spheres and other three-dimensional objects instead of simple circles and triangles. Finally, while event-based techniques such as MSP may rather easily be adapted to three-dimensional scenarios, other techniques that are based on light events (e.g., Lighthouse) may be difficult to implement or require significant amounts of additional hardware (e.g., multiple simultaneous light beams in different directions in the Lighthouse scheme).

## 6 Summary

Localization in sensor networks is essential for many sensing applications and network management activities. This chapter provided a survey of range-free localization techniques in wireless sensor networks, including techniques based on hop count, area information, neighborhood information, and event detection. For many of

these techniques, it is required that there are sufficient reference nodes and that those nodes are evenly distributed throughout the network. While the accuracy obtained by range-free localization techniques is typically lower than the accuracy of range-based techniques, a main advantage of range-free localization is that typically no additional hardware is needed and localization can therefore be performed at a lower cost. This chapter compared a variety of centralized and distributed localization techniques, including techniques belonging to the event-based positioning category. These techniques vary significantly in their overheads, performance, scalability, and the number of reference nodes. This chapter concluded with a comparison of these techniques, including a discussion of open research questions.

## References

1. I. Borg, P. Groenen, in *Modern Multidimensional Scaling: Theory and Applications* (Springer, New York, 1997)
2. S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, Linear matrix inequalities in system and control theory. in *Society for Industrial and Applied Mathematics (SIAM)* (Philadelphia, PA, 1994)
3. Y.-T. Chan, W.-Y. Tsui, H.-C. So, P.-C. Ching, Time-of-arrival based localization under NLOS conditions. *IEEE Trans. Veh. Technol.* **55**(1), 17–24 (2006)
4. C.R. Comsa, A.M. Haimovich, S.C. Schwartz, Y.H. Dobyns, J.A. Dabin, Time difference of arrival based source localization within a sparse representation framework. in *Proceedings of the 45th Annual Conference on Information Sciences and Systems (CISS)* (Baltimore, MD, 2011)
5. L. Doherty, K.S.J. Pister, L. El Ghaoui, Convex position estimation in wireless sensor networks. in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies* (Anchorage, AK, 2001)
6. T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks. in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)* (San Diego, CA, 2003)
7. J. Hightower, G. Borriello, Location system for ubiquitous computing. *Comput. J.* **34**(8), 57–66 (2001)
8. B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, in *Global Positioning System: Theory and Practice*, 5th edn., (Springer, 2008)
9. X. Ji, H. Zha, Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)* (Hong Kong, China, 2004)
10. I. Khan, H. Mokhtar, M. Merabti, A new self-detection scheme for sensor network boundary recognition. in *Proceedings of the 34th Conference on Local Computer Networks* (Zurich, Switzerland, 2009)
11. L. Lazos, S. Capkun, R. Poovendran, Robust position estimation in wireless sensor networks. in *Proceedings of the International Workshop on Information Processing in Sensor Networks (IPSN)* (2005)
12. L. Lazos, S. Capkun, R. Poovendran, SeRLoc: Secure range-independent localization for wireless sensor networks. in *Proceedings of the ACM Workshop on Wireless Security (WiSe)* (2004)
13. H. Lee, A. Klappenecker, K. Lee, L. Lin, Energy efficient data management for wireless sensor networks with data sink failure. in *Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)* (Washington, DC, 2005)

14. M. Li, Y. Liu, Rendered path: range-free localization in Anisotropic sensor networks with holes. in *Proceedings of the Thirteenth Annual International Conference on Mobile Computing and Networking (MobiCom)* (Montreal, Canada, 2007)
15. C. Liu, K. Wu, T. He, Sensor localization with ring overlapping based on comparison of received signal strength indicator. in *Proceedings of the IEEE International Mobile Ad-Hoc and Sensor Systems (MASS) Conference* (Fort Lauderdale, FL, 2004)
16. Y. Liu, Z. Yang, X. Wang, L. Jian, Location, localization, and localizability. *J. Comput. Sci. Technol. (IJST)* **25**(2), 274–296 (2010)
17. K. Lorincz, D. Malan, T.R.F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, S. Moulton, M. Welsh, Sensor networks for emergency response: challenges and opportunities. *IEEE Pervasive Comput.* **3**(4), 16–23 (2004)
18. Y. Lu, W. Wang, Y. Zhong, B. Bhargava, Study of distance vector routing protocols for mobile ad hoc networks. in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom)* (Dallas - Fort Worth, TX, 2003)
19. V. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, N. Shroff, Design of surveillance sensor grids with a lifetime constraint. in *Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN)*, (2004), pp. 263–275
20. L.S. Monteiro, T. Moore, C. Hill, What is the accuracy of DGPS? *J. Navig.* **58**(2), 207–225 (2005)
21. Y. Nesterov, Interior point polynomial algorithms in convex programming. in *Society for Industrial and Applied Mathematics (SIAM)* (Philadelphia, PA, 1994)
22. D. Niculescu, B. Nath, Ad hoc positioning system (APS). in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)* (San Antonio, TX, 2001)
23. D. Niculescu, B. Nath, DV based positioning in ad hoc networks. *Telecommun. Syst.* **22**(1–4), 267–280 (2003)
24. N. Patwari, A.O. Hero, J.A. Costa, Learning sensor location from signal strength. in *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks, Advances in Information Security Series*, vol. 30 (Springer, 2006)
25. R. Peng, M.L. Sichitiu, Angle of arrival localization for wireless sensor networks. in *Proceedings of the 3rd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks* (2006)
26. R. Rivest, The MD5 message-digest algorithm. RFC 1321, April 1992
27. K. Roemer, The lighthouse location system for smart dust. in *Proceedings of the 1st International Conference On Mobile Systems, Applications And Services* (2003), pp. 15–30
28. A. Savvides, C.-C. Han, M.B. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors. in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking* (Rome, Italy, 2001)
29. Y. Shang, W. Ruml, Improved MDS-based localization. in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)* (2004)
30. Y. Shang, W. Ruml, Y. Zhang, M.P. Fromherz, Localization from mere connectivity. in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (Annapolis, Maryland, 2003)
31. Y. Shang, W. Ruml, Y. Zhang, M. Fromherz, Localization from connectivity in sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **15**(11), 961–974 (2004)
32. Y. Shang, W. Ruml, M.P. Fromherz, Positioning using local maps. *Ad Hoc Netw.* **4**(2), 240–253 (2006)
33. I.G. Siqueira, L.B. Ruiz, A.A.F. Loureiro, J.M. Nogueira, Coverage area management for wireless sensor networks. *Int. J. Netw. Manag.* **17**(1), 17–31 (2007)
34. I. Stojmenovic, Position based routing in ad hoc networks. *IEEE Commun. Mag.* **40**(7), 128–134 (2002)
35. R. Stoleru, T. He, J.A. Stankovic, Range-free localization, Chapter in secure localization and time synchronization for wireless sensor and Ad Hoc networks. in *Advances in Information Security Series*, ed. by R. Poovendran, C. Wang, S. Roy, vol. 30 (Springer, 2007)

36. R. Stoleru, T. He, J.A. Stankovic, D. Luebke, A high-accuracy low-cost localization system for wireless sensor networks. in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)* (San Diego, CA, 2005)
37. J. Undercoffer, S. Avancha, A. Joshi, J. Pinkston, Security for Sensor Networks. in *Proceedings of the CADIP Research Symposium* (2002)
38. O. Younis, S. Fahmy, Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach. in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom)* (Hong Kong, 2004)
39. Z. Zhong, T. He, MSP: multi-sequence positioning of wireless sensor nodes. in *Proceedings of the 5th International Conference On Embedded Networked Sensor Systems* (2007), pp. 15–28