

# Incorporating Data from Multiple Sensors for Localizing Nodes in Mobile Ad Hoc Networks

Rui Huang, *Member, IEEE*, and Gergely V. Záruba, *Member, IEEE*

**Abstract**—The ad hoc network localization problem deals with estimating the geographical location of all nodes in an ad hoc network, focusing on those nodes that do not have a direct way (for example, GPS) to determine their own location. Proposed solutions to the ad hoc localization problem (AHLP) assume that nodes are capable of measuring received signal strength indication (RSSI) and/or are able to do coarse (sectoring) or fine signal angle-of-arrival (AoA) measurements. Existing algorithms exploit different aspects of such sensory data to provide either better localization accuracy or higher localization coverage. However, there is a need for a framework that could benefit from the interactions of nodes with mixed types of sensors. In this paper, we study the behavior of RSSI and AoA sensory data in the context of AHLP by using both geometric analysis and computer simulations. We show which type of sensor is better suited for which type of network scenario. We study how nodes using either, both, or none of these sensors could coexist in the same localization framework. We then provide a general particle-filtering framework, the first of its kind, that allows heterogeneity in the types of sensory data to solve the localization problem. We show that, when compared to localization scenarios where only one type of sensor is used, our framework provides significantly better localization results. Furthermore, our framework provides not only a location estimate for each nonanchor, but also an implicit confidence measure as to how accurate this estimate is. This confidence measure enables nodes to further improve on their location estimates using a local, iterative one-hop simple message exchange without having to rely on synchronized multiphase operations like in traditional multilateration methods.

**Index Terms**—Mobile computing, position measurement, radio position measurement, probabilistic algorithms, ad hoc networks.

## 1 INTRODUCTION

MOBILE ad hoc networks (manets) are infrastructureless networks that form on the fly as network nodes move in and out of each other's transmission range. Since a manet serves as an abstract model and concept that can be seen as a superset of diverse subareas such as sensor networks and mesh networks, or as an enabler for pervasive computing, it has attracted heavy research interest in the past several years. A major advantage of manets over regular wired or wireless networks is in their infrastructureless nature, as they can potentially be deployed more rapidly and less expensively than infrastructure-based networks. However, the lack of an underlying explicit infrastructure also becomes a major disadvantage in adapting manets to a wider array of applications, since existing network algorithms and protocols are not "plug-in" solutions for such dynamic networks. New algorithms need to be and are being designed for such fundamental network tasks as addressing, topology discovery, and routing.

Location discovery is emerging as one of the more important tasks, as it has been observed and shown that (semi)accurate location information can greatly improve the performance of other manet tasks such as routing, conserving energy, or maintaining network security. For instance, algorithms such as Location-Aided Routing (LAR) [13],

Grid [16], Greedy Other Adaptive Face Routing (GOAFR+) [15], and Terminode Local Routing/Terminode Remote Routing (TLR/TRR) [1] rely on location information to provide more stable routes during unicast route discovery. The availability of location information is also required for geocast (multicast based on geographic information [12]) algorithms such as the Location-Based Multicast (LBM) algorithm [14], GeoGrid [17], and Position-Based Multicast (PBM) routing [19]. To minimize power consumption, the Geographical Adaptive Fidelity (GAF) algorithm [29] uses location information to effectively modify the network density by turning off certain nodes at particular instances. In [9], it has been shown that wormhole attacks can be effectively prevented when location information is available. Localization techniques are also used in event detection within sensor networks [28].

A direct way of obtaining location information is to install GPS receivers on each node. However, this is currently impractical, as GPS receivers are still relatively expensive, power-hungry, and require a clear line of sight (that is, making indoor use impossible) to several Earth-bound satellites. In ad hoc networks, devices may be small while operating on a very restricted power source; thus, it may not be feasible to install GPS receivers onto all nodes. In this paper, we present a localization algorithm that allows non-GPS nodes to be localized based on heterogeneous measurements including ranging, angle, and connectivity.

The rest of the paper is organized as follows: Section 2 defines the localization problem and the motivation of our research. Related work is summarized in Section 3. Section 4 introduces our particle filter framework, the performance of which is evaluated via simulations in Section 5. A

• The authors are with the Computer Science and Engineering Department, The University of Texas at Arlington, 416 Yates, 300NH, Arlington, TX 76019. E-mail: rxh1725@omega.uta.edu, zaruba@uta.edu.

Manuscript received 15 Apr. 2006; revised 2 Oct. 2006; accepted 13 Nov. 2006; published online 7 Feb. 2007.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0105-0406. Digital Object Identifier no. 10.1109/TMC.2007.1015.

discussion on the advantages of the particle filter method over existing localization methods is presented in Section 6. Section 7 concludes the paper.

## 2 PROBLEM DEFINITION AND MOTIVATION

We define the ad hoc localization problem (AHLP) as the task of finding estimates for the physical location of all nodes given that only a subset of the nodes (*anchors*) know their exact location. More formally, given a network graph  $G = (V, E)$ , where a subset of the nodes  $\{V\}$  are location-aware anchor nodes  $\{V_{gps}\} \subset \{V\}$ , the objective of an ad hoc localization algorithm (AHLA) is to find the locations of *nonanchor* nodes  $\{V\} - \{V_{gps}\}$ . The AHLP is nontrivial for a number of reasons:

1. *Geometric limitations.* To pinpoint its location, a node needs to know the locations of at least three anchor nodes together with its distance from each of these anchor nodes. Alternatively, nodes could calculate their location based on a distance and an (absolute) angle measurement from one anchor. Even if obtaining such measurements was possible and the measurements were exact, guaranteeing that (several) anchor nodes surround each regular node is impossible, as manets may be randomly deployed and, in general, only a small percentage of nodes are anchors. Thus, a good AHLA needs to take advantage of multihop information, that is, estimating node locations based on other nodes' location estimates.
2. *Availability of measurements.* For AHLAs that require distance or angle measurements, certain sensory devices will need to be available to provide such readings. However, it is likely that not all nodes have the same sensory capacity. In other words, there is a need for the AHLA to work in a heterogeneous environment with different location sensory capacities.
3. *Measurement error and error propagation.* Even when received signal strength indication (RSSI) and/or angle-of-arrival (AoA) sensors are available, those measurements are prone to errors. For instance, a distance measurement based on an RSSI reading is prone to multipath fading and far field scattering. The error can be especially high when the quantity and quality of obstacles in between sender and receiver is significant. Since most AHLAs require measurements from nodes several hops away, the measurement error is likely to aggregate along the path and eventually completely throw off the location estimate.

Previous work on the AHLP tries to address the above issues from various angles. However, we are not aware of any previous work specifically and explicitly addressing the second issue. Thus, in this paper, we propose a probabilistic AHLA based on particle filtering that is specifically designed to work in heterogeneous networks where different nodes may have different sensory capacities. To the best of our knowledge, our framework encompasses the first AHLA dealing with ad hoc networks where location

sensor capabilities may be heterogeneous. Using the same probabilistic framework, our algorithm enables nodes with different location sensory capacities (for example, RSSI, AoA, or no sensory reading at all) to cooperate in solving the AHLP. Our AHLA is also highly distributed and loosely coupled in the sense that the localization process does not need centralized control. Each node performs its own localization aided by a simple message exchange with its neighbors; there is no requirement for maintaining multihop connectivity data. We have performed extensive simulations of our AHLA by varying different parameters such as sensor capacities, sensory noises, anchor ratio, and network density. Our analysis and simulation study indicate a clear advantage of combining RSSI and AoA sensors in the same network instead of relying on the same type of sensor for the entire network. The simulations also demonstrate the effectiveness of our particle-filtering framework in adapting to the networks of different sensory capacities, making it particularly suitable for such a heterogeneous environment.

## 3 RELATED WORK

Currently, a majority of the AHLAs rely on distance measurements using RSSI or time of arrival (ToA). In RSSI, the receiver measures the received signal strength and compares it with the transmitted signal strength. The difference (in decibels) is then applied to the inverse of the signal propagation model to provide a distance measurement. Sensors that measure RSSI are widely available to mobile devices. Indeed, most off-the-shelf technologies implicitly provide such information (for example, most Wi-Fi, Bluetooth, and IEEE 802.15.4 chipsets do). The drawback of RSSI-based measurements is that they can be very inaccurate because an exact model of the propagation environment is unavailable. Although ToA is used for radio signals in GPS, it is mostly used in the context of acoustic signals in inexpensive ToA tracking (as propagation speeds are five orders of magnitude less). Thus, ToA measures the time-acoustic signals travel from the sender to the receiver. The distance is obtained by multiplying this time with the signal propagation speed. In general, distance measures based on ToA are more accurate than RSSI-based measures. However, special acoustic transceivers have to be employed on each node, and strict time synchronization among nodes needs to be in place. As mentioned earlier, ToA may also be used together with radio signals, but current technology is not mature enough to provide a satisfactory precision over smaller distances inexpensively.

A number of AHLAs that operate on distance measurements have been proposed. Methods such as DV-Hop, DV-Distance, and euclidean [21], [22] estimate the absolute node locations; the GPS-Free method [3] calculates relative node locations from the distance measurements. Different methods generally exploit the trade-off between the estimation accuracy and the estimation coverage. For instance, given the same network scenario, the euclidean method is capable of generating more accurate location estimates of a smaller subset of nodes, whereas the DV-Hop method has better coverage but worse accuracy.

Regardless of the trade-off, a common characteristic shared by distance-based AHLAs is that they require a relatively high network density in order to archive better results. Based on the extensive simulation of DV-Distance, euclidean, and multilateration methods performed in [4], it can be concluded that those distance-based AHLAs “require an average degree of 11–12 nodes within the ranging neighborhood in order to achieve 90 percent localization coverage with 5 percent accuracy.”

Simulation studies in [4] also show that, when AoA is used in conjunction with distance measurements, the localization accuracy and coverage can be drastically improved. This should not come as a surprising conclusion, as nodes need to communicate with only one neighbor to perform localization if they can obtain both AoA and distance measurements. The work in [4] also presents three variations of a weighted mean square error algorithm that localizes the nodes, each of which is designed to work with one of the three measurement types: 1) distance-only measure, 2) distance plus a more accurate AoA measure (up to 8 degrees of precision), and 3) distance plus a less accurate AoA measure (up to 60 degrees of precision). The less accurate AoA measurement method is sometimes referred to as *sectoring*. Simulations in [4] show that the localization accuracy and coverage can be greatly improved with coarse sectoring as well.

Currently, there is no off-the-shelf device that offers AoA sensing capability. However, a number of prototype devices are available. For instance, Cricket Compass [24] is a small form device that uses ultrasonic measurements and fixed anchors to obtain acoustic signal orientations. In [23], a rotating directional antenna is attached to an 801.11b base station. By measuring the maximum received signal strength, a median error of 22 degrees can be obtained from the sensor. The challenge here is to design the AoA sensing device that has small form factor and low energy consumption. In [4], Chintalapudi et al. outline a solution with a ring of charge-coupled devices (CCDs) to measure AoA with relatively low energy consumption.

Other than range and angle-based methods, there is a different class of localization methods that rely solely on connectivity information. Assuming that all nodes have the same transmission range, the localization problem becomes a problem of fixing a set of unit disks into a graph given the connectivity constraints. The Centroid method [2] estimates the location of an unknown node as the average of its neighbor’s locations. The Approximate Point-in-Triangulation (APIT) method [7] estimates the node location by isolating the area using various triangles formed by anchors.

The localization algorithm proposed in this paper is a probabilistic method employing important sampling techniques (particle filters). Here, each unknown node’s location is viewed as a probability distribution over the deployment area. The goal of the localization algorithm is to shape the distribution based on a sequence of measurements until the distribution becomes focused and collapses onto a small area. The probabilistic method and particle filters have been used in visual target tracking [11] and computer vision location systems [5], [27] in the context of

robotics. The particle filter method is also used in [18] to triangulate the mobile node location based on received signal strengths from several known-location base stations in wireless cellular networks. The probability Grid system in [26] is a centralized probabilistic localization algorithm that updates the distribution based on a Grid system.

Perhaps the closest to our work are the Monte Carlo localization (MCL) method [8] and the indoor location tracking algorithm proposed in [30]. In [8], a similar probabilistic and particle filter approach was taken to localize mobile nodes in ad hoc networks. As nodes move in and out of range of each other, the probability distribution is updated via particle filtering based on connectivity information. However, our work differs from theirs in that we consider sensory-based (RSSI or AoA) localization, whereas the localization in [8] is range-free. Given the different measurement models, the filtering process is also completely different. Furthermore, our method works for both stationary and mobile networks, whereas the model in [8] is designed for mobile networks only. Finally, localization in [8] only occurs when a regular node hears from an anchor directly, whereas our algorithm allows collaborative localization among regular nodes.

In [30], we have dealt with a known environment; thus, different obstacles can be represented in a floorplan and a signal strength (RSSI) map can be obtained via measurements and calculations ahead of time. The location tracking problem then becomes a decision-making problem, where a solution may use a measurement model that compares the current RSSI with the signal strength map to find the location with the highest probability of matching the current RSSI reading. Although roots are similar, the solution described in this paper is designed for outdoor environments and infrastructureless networks, where major continuous obstacles (such as walls) are assumed to be minimum and fairly reliable distance estimates can be obtained from RSSI readings and the signal propagation model. The probability distributions of location estimates are updated solely from the distance and location estimates from neighbors.

## 4 PARTICLE FILTER FRAMEWORK FOR LOCATION ESTIMATION

In this paper, we propose a localization method based on Bayesian filters by using Monte Carlo sampling (also known as particle filters) introduced in [6]. Our method can be considered a probabilistic approach, in which the estimated location of each node is regarded as a probability distribution captured by samples, thus, the term particles. The distribution of particles (the probability distribution of a node’s location over the area) is continuously updated, as the node receives location estimates from its neighbors along with certain types of sensory readings such as RSSI and AoA. (More precisely, particles represent samples drawn from a continuous probability density function; thus, using the term “probability distribution” is justified.) Essentially, nodes estimate their own locations by exchanging the location distributions directly with their neighbors.

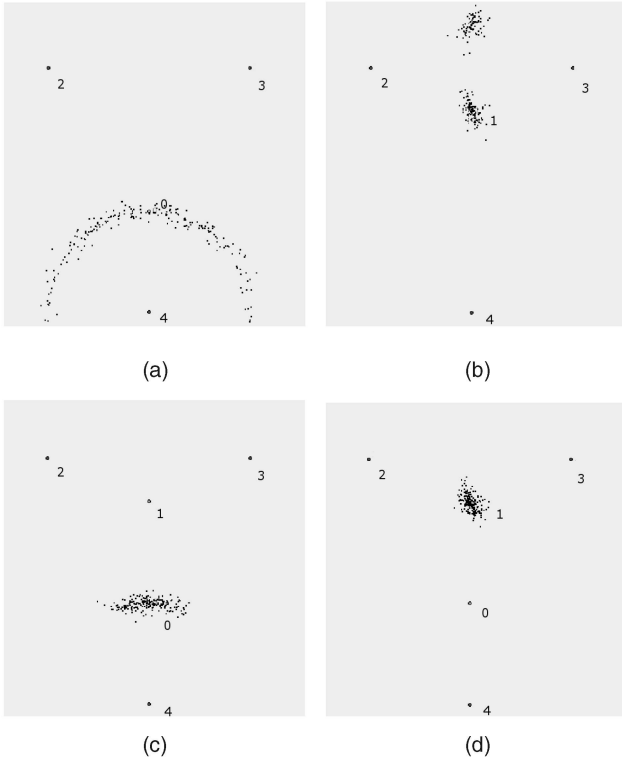


Fig. 1. Location distribution in simple scenarios with distance measures. (a) Particle distribution of node 0 when node 1 is *not* presented. (b) Particle distribution of node 1 when node 0 is *not* presented. (c) Particle distribution of node 0 when node 1 *is* presented. (d) Particle distribution of node 1 when node 0 *is* presented.

Fig. 1 demonstrates how our method solves the AHLP in a simple scenario (where RSSI-based distance measurements are used). Here, nodes 2, 3, and 4 are anchors, whereas nodes 0 and 1 are nonanchors. Of the nonanchors, node 0 can receive signals from nodes 1 and 4, and node 1 can receive signals only from nodes 0, 2, and 3. From the signal strength readings, nonanchors estimate their distances to their neighbors. The probability distribution of the estimated location is represented by the particles (dots) in the graph. In Fig. 1a, where node 1 is removed, node 0 can only receive signals from node 4; thus, as the particle distribution indicates, the probability distribution where node 0 is most likely located concentrates on a circle around node 4. In Fig. 1b, where node 0 is removed, node 1 can receive signals from nodes 2 and 3; thus, the most likely locations for node 1 center around two areas where “transmission circles” around nodes 2 and 3 intersect. Intuitively, in order to localize itself, a nonanchor needs to receive location information from a minimum of three anchors either directly or indirectly. In both cases (Figs. 1a and 1b), the exact location of nonanchor nodes 0 and 1 cannot be deduced because they do not receive location information from all three anchors. In Figs. 1c and 1d, where all nodes are available, nodes 0 and 1 are able to communicate with each other and exchange their particle distributions. Thus, their probability densities will represent their actual locations much closer even though neither node receives location information from all three anchors directly.

The theory and mathematics behind our particle filter solution are explained in detail in [10]. In a nutshell, the particle filter updates its probability distribution in two steps: estimation and correction. In the first step, the filter updates the distribution,  $X$ , based on a system model where the new location of the node is **estimated** based on its previous location. Here, we select a simplistic model by assuming that, at any point in time, the node moves with a random velocity drawn from a normal distribution with a mean of 0 m/s and a fixed standard deviation  $\sigma$ . No information about the environment is included in this model and, as a consequence, the filter permits the estimates to move along arbitrary paths. Thus, our system model is simply  $p(s_t|s_{t-1}) = N(0, \sigma)$ , where  $N$  is a normal distribution. Note that, although such a system model should work well in stationary networks and networks where user mobility is extremely uncertain, there may be better models for mobile networks. In reality, mobile nodes follow a certain kind of movement profile instead of random motion. The closer the system model resembles the actual movement profile of the node, the better the filter will perform. However, since it is difficult to obtain a reliable movement profile when the location is unknown, the assumption of random movement is probably the best we can do. (Note that a more accurate mobility model would greatly improve the estimate of the filter.)

When a reading  $m$  is obtained from the RSSI or AoA sensor, the particle filter undergoes a *correction step*, in which the measurement is used to correct the output of the system model. In particular, the correction step modifies the particle distribution  $X$  so that it becomes more consistent with the current measurement. In our case, the correction is calculated based on both the measurement reading and the location distribution of the neighbor. In other words, when node  $u$  receives a measurement reading  $m$  from node  $v$ , the correction step updates the location distribution of  $X_u$  based on  $m$  as well as  $X_v$ . The correction step is challenging due to the fact that both  $m$  and  $X_v$  are imprecise. The measurement reading  $m$  could be noisy due to environmental and sensory characteristics. The location distribution  $X_v$  could also be imprecise, unless  $v$  is an anchor. Thus, the correction step needs to modify the particle distribution  $X_u$  so that it becomes more consistent with  $m$  while taking into account the inherent impreciseness of  $X_u$  and  $X_v$ .

After each correction step, the estimated location of the node is obtained by finding the mode of the particle distribution. The mode is calculated by comparing the distances between samples. The particle that is the closest to all other particles (that is, the mode of the density) is selected to be the most likely estimated location at the current time. For stationary networks, we can monitor the expected location over multiple updates and set the stop condition when the change of the location becomes sufficiently small for the application using the location data. Alternatively, we can monitor the filter uncertainty (in terms of variance in particles’ locations) and set the stop condition when it falls below a certain threshold. For mobile networks, the particle filter can run continuously to keep track of locations as nodes move.

**Algorithm 1.** Particle Filter Update

```

 $X_u \leftarrow$  Uniform particle distribution over the deployment
area
repeat
  for all neighbor  $v$  in the neighbor set do
    Receive ( $id_v$ ,  $X_v$ , and  $m$ )
    Decompress  $X_v$ 
    for all  $x_u \in X_u$  do
      Randomly select a  $x_v \in X_v$ 
      Find location  $x'_u$  based on  $x_v$ ,  $m$ ,  $stdev(X_u)$ , and
 $stdev(X_v)$ 
      Update  $x_u$  with  $N(x'_u, ((stdev(X_u) + stdev(X_v))/2)^2)$ 
    end for
  end for
until  $var(X_u)$  is below a threshold

```

Algorithm 1 shows the pseudocode for the particle filter update algorithm. Note that the method to find the new location  $x'_u$  depends on the sensory capacity of the receivers. We can now consider three different types of behavior for the correction step depending on what sensors are available: RSSI, AoA, or none ("blind node"). We then analyze both RSSI and AoA methods in the context of the AHLF with and without measurement noises. Finally, we explain the *Decompress* step, which is necessary to reduce the communication overhead.

**4.1 RSSI Sensor Availability**

When an RSSI sensor is available, we obtain a distance estimate from the inverse of the signal propagation model  $A^{RSSI} = P(f, D, A^0)$ , where  $D$  is the distance,  $f$  is the frequency used,  $A^0$  is the transmission power, and  $A^{RSSI}$  is the received signal strength. Noise can be added to the model, but we disregard it when calculating the inverse and let it be reduced by the particle filter. (This does not mean that we are ignoring noise in our evaluation, as noise will indeed be added to the RSSI measurements in the simulation model).

As shown in Fig. 2a, we let node  $v$  be the sender and node  $u$  be the receiver. For each particle  $x_u$  in the current location distribution  $X_u$ , we randomly select a particle  $x_v$  in the sender's location distribution  $X_v$  and calculate their distance  $D(x_u, x_v)$ . We then measure the difference between  $D(x_u, x_v)$  and  $D^{(RSSI)}$  and select a new location for resampling based on the difference, as well as the variances, of the particle distribution  $X_v$  and  $X_u$ . For instance, before the update step,  $x_u$  and  $x_v$  are located at points  $A$  and  $B$ , respectively. Thus,  $D(x_u, x_v) = |AB|$ . Let  $A'$  be the location of  $x_u$  based on the RSSI reading on the same line, that is,  $D^{(RSSI)} = |A'B|$ . Intuitively, if the location estimate given by the distribution  $X_v$  is accurate and the actual location for node  $v$  is indeed at  $x_v$ , then the new location for particle  $x_u$  should be at point  $A'$ . Conversely, if the location estimate of the distribution  $X_u$  is accurate, then the new location for  $x_u$  should stay at  $A$ . Therefore, we select the new location based on the perceived accuracy, that is, the variances, of the distributions of  $X_u$  and  $X_v$ . Let the standard deviation of a distribution  $X$  be  $stdev(X)$ . We select the new location of  $x_u$ ,  $x'_u$ , along the line  $|AA'|$  such that

$$\frac{|Ax'_u|}{|x'_uA'|} = \frac{stdev(X_u)}{stdev(X_v)}.$$

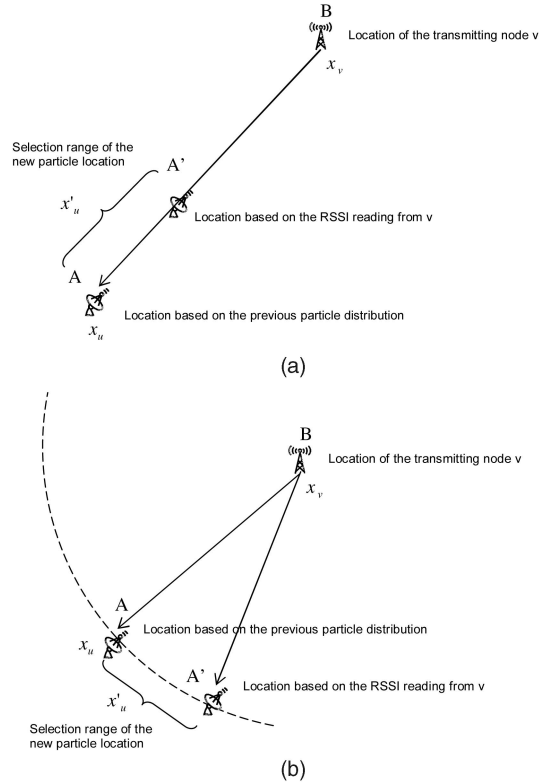


Fig. 2. Particle filter update. (a) RSSI. (b) AoA.

A new particle is then randomly resampled using a normal distribution centered at  $x'_u$ , with the standard deviation being the average of those of  $X_u$  and  $X_v$ . We consider the standard deviations of both  $X_u$  and  $X_v$  during resampling because the spread of both distributions affects the spread of the updated distribution  $X'_u$ .

**4.2 AoA Sensor Availability**

When an AoA sensor is available, we compare the standard deviations of the sender's and receiver's distribution as before, but in this case, we modify the receiver's particles based on AoA. Again, let  $v$  be the sender and  $u$  be the receiver. As shown in Fig. 2b, for each particle  $x_u$  in the current location distribution  $X_u$  and particle  $x_v$  in the sender's location distribution  $X_v$ , the AoA between  $x_u$  and  $x_v$  can be calculated along with the distance  $D(x_u, x_v) = |AB|$ . We draw a line through  $B$  according to the current AoA reading and select a point  $A'$  by maintaining the distance so that  $|A'B| = |AB|$ . The new particle location  $x'_u$  is then located on the arc between points  $A$  and  $A'$  with the radius being  $|AB|$ . If the location estimate from  $v$  is more accurate, then  $x'_u$  should be closer to point  $A'$  along the arc. Conversely, if the location estimate from  $u$  is more accurate, then  $x_u$  should be closer to  $A$ . Thus, the new location of  $x_u$ ,  $x'_u$ , is the following:

$$\frac{\angle ABx'_u}{\angle x'_uBA'} = \frac{stdev(X_u)}{stdev(X_v)}.$$

Similarly to the way done at the RSSI sensors, a new particle is resampled from the selected location based on a normal distribution with the standard deviation equaling the

average of  $stdev(X_u)$  and  $stdev(X_v)$ . When both RSSI and AoA sensors are available, the aforementioned update steps can be effectively combined. In such a case, the particles are first updated based on the RSSI reading followed by an update based on the AoA readings. Thus, both sensor readings are applied to the location estimation.

### 4.3 Blind Nodes

We also consider a third type of nodes, the blind nodes, where neither RSSI nor AoA sensors are available. Blind nodes have to rely on pure connectivity information to estimate their locations. One approach is to use a variation of the Centroid method [2], where nodes estimate their locations by simply averaging all their neighbors' locations. Intuitively, such a method should work reasonably well when the network is well connected (so that there is more location data from the neighbors to work with) and the anchor ratio is high (so that the location data from the neighbors are more accurate). We adopt the same idea here but adapt it to the context of particle filtering.

We consider the same scenario as before, where node  $u$  receives a location update message from a neighbor  $v$ . Let  $x_u$  be a random particle within the location distribution  $X_u$  before the update. Let  $x_v$  be a random particle at neighbor's distribution  $X_v$ . Since there is no sensor reading on ranging or angle, we cannot tell exactly the distance or the direction between  $u$  and  $v$ , but we know they are sufficiently close since the nodes are in each other's transmission range. In this case, we update particle  $x_u$ 's location (to between  $x_u$  and  $x_v$ ) as

$$\frac{|x_u x_u'|}{|x_u' x_v|} = \frac{stdev(X_u)}{stdev(X_v)} \cdot c.$$

Again, when  $X_u$  is perceived as more accurate (that is, when  $stdev(X_u)$  is smaller),  $x_u'$  becomes closer to the previous location  $x_u$ ; otherwise,  $x_u'$  moves closer to  $x_v$ . We multiply the weighting factor with a constant  $0 < c < 1$  so that the new location is closer the previous location. Thus, we are able to retain the location information through a sequence of location updates from different neighbors.

### 4.4 Analysis

In this section, we attempt to infer, by simple geometric analysis, whether RSSI or AoA sensors would be preferred for localization. We also consider the presence of noise. In particular, our goal is to derive a relationship between the noise parameters of each sensor type so that their impact can be compared during simulation.

Let us first consider the perfect scenario where no measurement noise interferes with the sensor reading. Recall that, to precisely locate a node, at least three RSSI readings from different anchors are required, whereas only two AoA readings are needed. When both measurement types are available, only one RSSI reading and one AoA reading from the same anchor are required to locate the node. In such case, AoA readings should provide better coverage (that is, locating more nodes) than RSSI readings.

In networks where connectivity is low and/or the anchor ratio is low, it is likely that nodes have to estimate their location based on other nodes' estimates. Let us consider

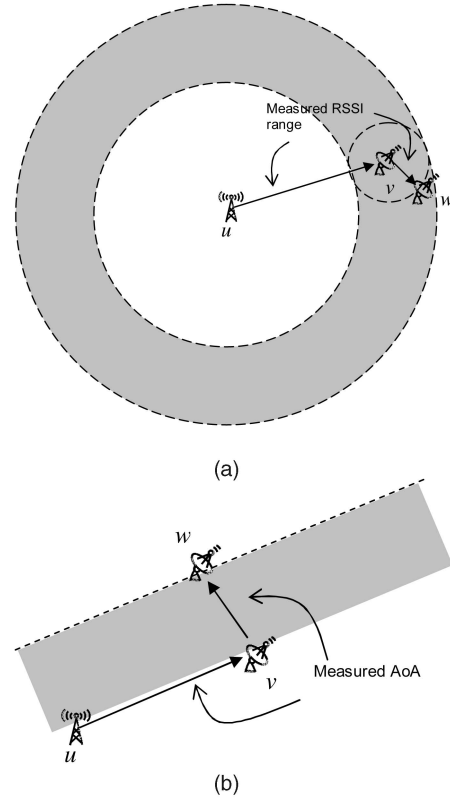


Fig. 3. Location range for two hops. The shaded area indicates the range of all possible locations. (a) RSSI. (b) AoA.

how well location data is propagated in all sensor type cases. Let node  $u$  be an anchor and  $v$  and  $w$  be nonanchors. Let  $u$  connect to  $v$  and  $v$  connect to  $w$ , but let there be no connection between  $u$  and  $w$ . Therefore,  $w$  has to infer its possible location based on the indirect information from  $v$ . In the case of RSSI sensors, the possible range of  $w$  is a hollow disk with inner radius  $r - s$  and outer radius  $r + s$ , where  $r$  is the distance between  $u$  and  $v$  and  $s$  is the distance between  $v$  and  $w$  (Fig. 3a). In the case of AoA sensors, the possible location of the intermediate node  $v$  is a beam with the origin at  $u$ 's location. To derive the possible locations of  $w$ , we have to draw a beam with the origin at every possible point of the previous beam. Thus, the possible range of  $w$  is an area bounded only by the beams and the network boundaries (Fig. 3b).

From the above discussion, we can infer that RSSI sensors are better suited for localization when location data need to propagate through multiple hops. AoA sensors, however, are better when anchors are only a single hop away. Thus, when networks are sparse and/or the anchor ratio is low, RSSI sensors are better suited. When networks are dense and/or the anchor ratio is high, AoA sensors would be a better choice.

When noise is added to the measurement readings, the precise location of nodes cannot be obtained even when the minimal geometric requirements are met. Instead, the location needs to be estimated, and the accuracy of this estimate is affected by the noise (and, thus, the noise model in simulation and mathematical evaluations). Furthermore, unlike the ideal scenario, where the minimal geometric

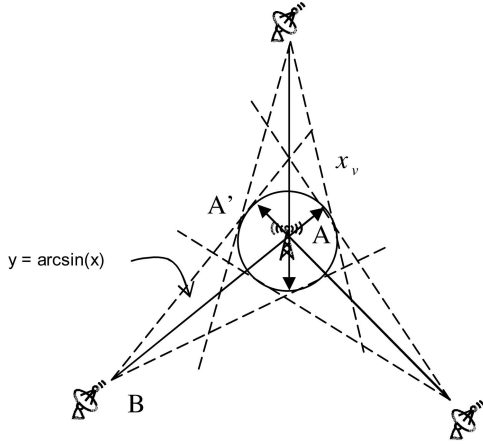


Fig. 4. Finding similar error range between RSSI and AoA.

requirement is sufficient to localize the node, the accuracy of the estimated location improves with readings from additional neighbors. As in other related works, we qualify the noise in terms of a *noise ratio*, which presents noise as a percentage of the unbiased measure. In the case of RSSI, a noise ratio of  $x$  over the unbiased distance  $D$  would depend on the actual reading:  $D' = D \cdot \text{Uniform}(1 - x, 1 + x)$ . In other words, the distance measured at the receiver contains an error in the range  $(-xD, xD)$ , which is uniformly distributed. The error in the distance measurement accounts for the power loss due to factors such as multipath fading and far field scattering. Modeling noise, in this way, with a uniform distribution, is drastically simplified; however, we have chosen this simplified model so that our result can be effectively compared to results of other AHLAs (as this method has been excessively used in the AHLP literature). The AoA measurement noise is modeled in a similar way using a uniform distribution; with a noise ratio of  $y$ , the measured AoA at the receiver is then  $A' = A \cdot \text{Uniform}(1 - y, 1 + y)$ .

To compare the estimates based on RSSI and AoA under a noisy environment, we need to establish a relationship between noise ratios  $x$  and  $y$ . In other words, with a given  $x$  for RSSI, a corresponding  $y$  for AoA needs to be selected to generate a more or less similar noisy environment. Fig. 4 shows how such a relationship could be established. Here, we consider a scenario in which a node is localized from three anchors of equal distance away. Let the actual location of the node be  $A$  and the location of one of the anchors be  $B$ ; thus, the actual distance between them is  $D = |AB|$ . Let us first consider the RSSI sensor, where  $x$  is the noise ratio. The range of all possible estimated locations should be within a circle of radius  $D \cdot x$  centered at the actual location  $A$ .

Now, consider the case of the AoA sensor. To replicate a similar range from the same three neighbors by AoA sensor, the closest range that can be possibly formed is to project AoA error  $y$  along line  $AB$  so that the entire circle is covered (by selecting the beam angle appropriately). Doing so in all three anchors, the actual location error range forms an outer hexagon that encloses the circle. Here,  $\sin(y) = |AA'|/|AB|$ , where  $|AA'|/|AB| = x$ . Thus, a relationship can be established between the RSSI noise parameter  $x$  and the AoA noise parameter  $y$  in that  $y = \arcsin(x)$ . Note that this

relationship does not generate identical error range between the two sensor types; in fact, the error range from AoA will always be greater. However, since it is not feasible to find the noise parameter that generates an identical error range that would work for all cases, our simplification can be justified. After all, our intention is to provide a relatively similar error range, so we can reasonably compare the results from the two sensor types.

#### 4.5 Compressing/Decompressing Particle Filter Distribution

In Sections 4.1, 4.2, and 4.3, it is assumed that a complete set of particles is received from each of the neighbors. Since the complete distribution consists of a large number of particles with their location data, doing so is obviously not very practical. Therefore, we propose a simple yet effective compressing mechanism that allows a representative for the particle distribution to be transmitted in a compact form.

Given a particle distribution  $X$ , we locate the most likely value  $\hat{x}$  as the particle in the distribution that has the minimum overall distance to other particles, that is,  $\hat{x} = \arg \min_{x \in X} (\sum_{y \in X} |x - y|)$ . In other words,  $\hat{x}$  is the most representative particle of the entire distribution. From  $\hat{x}$ , we count the number of particles  $n$  within a predefined range  $r$ . We then calculate the variance  $\sigma^2$  within those  $n$  particles. Thus, we obtain a quadruple  $(\hat{x}, r, n, \text{ and } \sigma^2)$ . From there, we remove the  $n$  particles in the previous quadruple from the distribution and repeat the process of finding the expected value, a larger range (explained later), and the variance. By continuing the same process until all particles have been covered, we obtain a sequence of quadruples that approximates the original particle distribution. When the quadruples are received by the receivers, a decompressing step is executed to reproduce the distribution by randomly generating particles based on the expected value, range, particle number, and variance for each quadruple. Using this method, the particle distribution can keep all of its "modes," even when the distribution shows several likely location areas for the node.

##### Algorithm 2. Range Increase for Particle Compression

```

 $Q \leftarrow$  number of quadruples desired
 $R \leftarrow$  max range that covers the entire area
 $\text{minQuota} \leftarrow |X|/Q$ 
 $r\text{Increment} \leftarrow R^{1/3}/Q$ 
 $x\text{Count} \leftarrow 0$ 
 $r \leftarrow 0$ 
 $\text{curRange} \leftarrow 0$ 
 $q' \leftarrow 1$ 
for  $q = 1$  to  $Q$  do
   $\text{maxRange} \leftarrow q \cdot r\text{Increment}^{3/2}$ 
  while  $\text{curRange} < \text{maxRange}$  AND the number of
  particles in  $\text{curRange} + x\text{Count} < \text{minQuota} \cdot q$  do
     $\text{curRange} \leftarrow q' \cdot r\text{Increment}^{3/2}$ 
     $q' \leftarrow q' + 1$ 
  end while
   $r_q \leftarrow \text{curRange}$ 
   $x\text{Count} \leftarrow x\text{Count} + \text{number of particles in curRange}$ 
end for

```

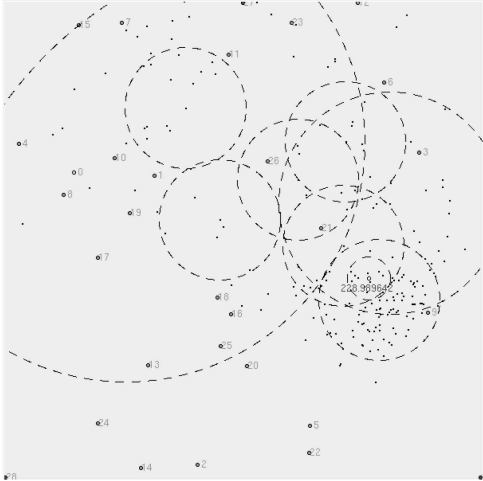


Fig. 5. Compressing the particle filter distribution.

For each broadcast, a fixed number of aforementioned quadruples are transmitted. Fig. 2 shows the algorithm used to progressively increase the range  $r$  for each quadruple. The algorithm starts with an initial range of  $R^{1/2}/Q$  and a minimum quota of particle size  $|X|/Q$  for each quadruple. As each quadruple is defined, a running sum  $xCount$  keeps track of the total number of particles covered thus far. At each step, the range is incremented exponentially at each quadruple by  $r := r^{3/2}$ , unless the running sum already exceeds the minimum quota. The algorithm guarantees that all particles are covered by the predefined number of quadruples and the overall trends of the original distribution are maintained. Meanwhile, by using a quota limit with the exponential range increment, denser particle areas are preserved in more detail. As reported in [10], our experiments showed that the compression method reduces the amount of data exchange by nearly 90 percent without a significant increase in the location estimates' error. Fig. 5 shows a typical compressed distribution where the circles represent the ranges.

## 5 SIMULATION RESULTS

We have conducted a number of simulation experiments to validate the effectiveness of our solution. In the following discussions, we will concentrate on the performance of the filter based on the AoA reading (pure AoA as well as mixed sensors of both AoA and RSSI coexisting in the same network); detailed results on the performance of the pure RSSI-based filter can be found in [10]. Although our particle filter framework has no such restriction, we assume a network in which all nodes have an identical transmission power. Thus, we can effectively control the network connectivity by varying the transmission range. A certain percentage of nodes (simulation factor) are designated as anchors that know their coordinates. When a node is located within the transmission range of another node, we assume that it is capable of receiving a signal from the sender. The RSSI depends on the distance to the sender (based on the employed signal propagation model) and a noise model. The measured AoA reading is affected by a noise model and its parameters as well.

The signal propagation model that we use is the general free-space propagation model of  $P = c \cdot d^{-2}$ , where the power of the received signal  $P$  is inversely proportional to the second power of the distance  $d$  and  $c$  is a constant that includes transmission power and frequency among others. When the received signal strength  $P$  is below a threshold  $P_{min}$ , it is considered too weak to be captured by the receiver; thus, the link breaks. Note that the selection of  $c$  and  $P_{min}$  does not affect the overall simulation results, as long as the same values are used in the observation model of the filters. For the particle filter itself, we use a total number of 200 particles at each node. We randomly place 100 nodes into an isotropic (square) network. Noise is added to both RSSI and AoA readings; for noise, we are using the models outlined in Section 4.4.

Regardless of the sensor types available, nodes localize themselves by running our particle filter framework using the location exchanges among neighbors. The location information is exchanged between the neighbors at 0.5-second time intervals. In other words, on the average, a node is able to obtain the location information from all of its neighbors every 0.5 seconds. The 0.5-second time interval is further randomized by a truncated normal distribution  $N(0.5, 0.5)$  to simulate the unpredictability of message arrival time in real networks. We simulate each type of scenario 50 times; the results are averaged and a 95 percent confidence interval is calculated and displayed (using vertical bars in our figures, each of which will be discussed in detail). Estimation errors in our graphs are given as a ratio to the transmission range; that is, an average error of 1 means that, on the average, the location estimate's error is the same as the nominal transmission range. We start by showing simulation results on stationary multihop networks and then move to manets.

### 5.1 Connectivity

The primary performance metric for any location algorithm is the estimation error, which indicates how close the estimated location is to the actual location. We compare the estimation error of our particle filter algorithm, including several combinations of RSSI, AoA, and blind nodes, against results of existing methods. Here, we use a noise parameter of  $x = 20$  percent and an anchor ratio of 10 percent. One advantage of our method is that it produces the location estimate along with a variance indicating its quality. Thus, by varying the variance threshold, we are able to control the effective estimation coverage; the lower the coverage, the better the estimation accuracy. The estimation error in our figures is therefore plotted against the desired coverage.

Fig. 6 shows the estimation error against coverage using four different variations of the particle filter algorithm while varying network connectivity. For each variation, we modify the sensor types within the network as follows:

1. 100 percent of the nodes have RSSI-only capacity,
2. 100 percent of the nodes have AoA-only capacity,
3. 50 percent of the nodes have RSSI and another 50 percent of the nodes have AoA capacity, and
4. 100 percent of the nodes have both RSSI and AoA capacities.



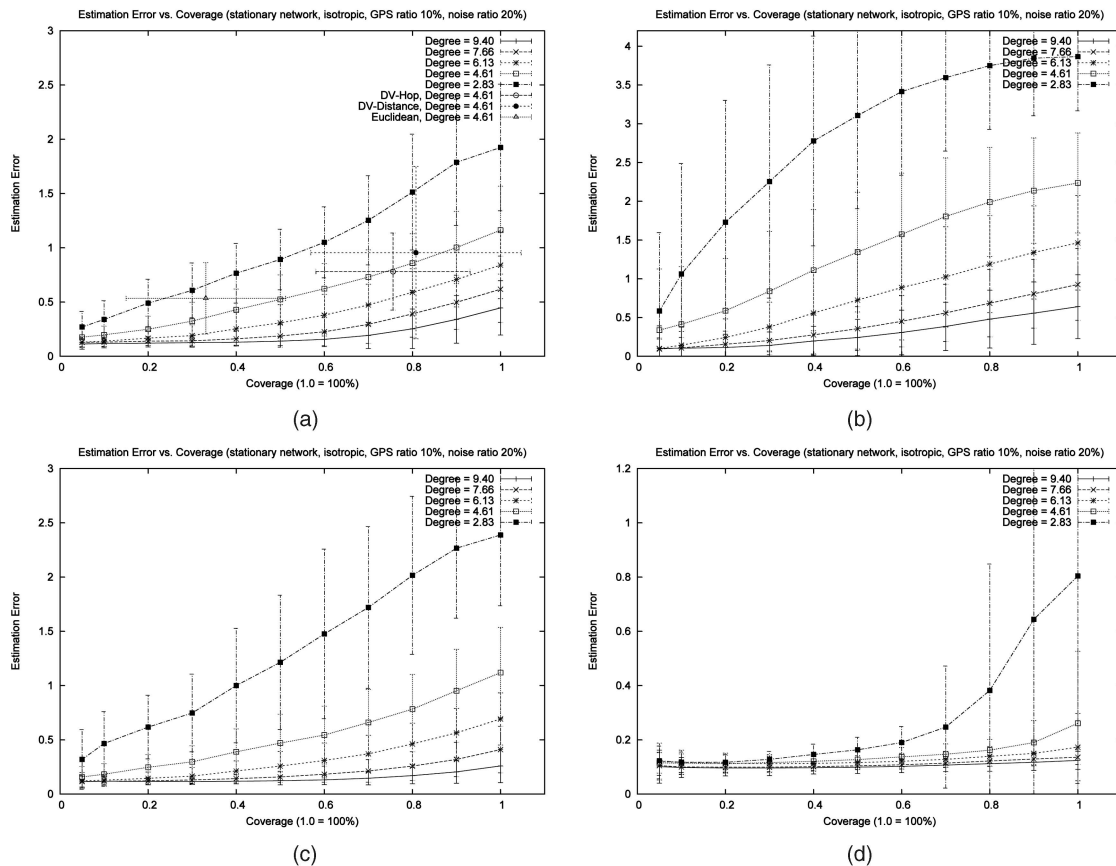


Fig. 6. Effect of network connectivity on the estimation error. (a) RSSI: 100 percent; AoA: 0 percent. (b) RSSI: 0 percent; AoA: 100 percent. (c) RSSI: 50 percent; AoA: 50 percent. (d) RSSI: 100 percent; AoA: 100 percent.

As shown in Fig. 6, network connectivity plays an important role determining the localization accuracy. In particular, nodes in sparser networks tend to be more difficult to localize (that is, resulting in higher estimation error) primarily due to two reasons. First, it is less likely for the nodes in sparser networks to be one hop away from the anchors. Thus, more nodes would have to rely on location data from anchors several hops away. Furthermore, as the network becomes sparser, it is likely to become disconnected. The location data from the anchors might not be able to propagate to all nodes. In extreme cases, which are much more common in sparser networks, there might be nodes that do not meet the minimum geometric requirement of localization; therefore, those nodes cannot be localized.

In Fig. 6a, where the localization relies solely on the RSSI sensor data, the estimation error from the particle filter method is plotted along with the results from DV-Hop, DV-Distance, and euclidean methods that produce location estimates with a fixed coverage. Thus, their estimation errors are shown as single data points in the figure. Comparing the single data points of the above methods, the comparable plot (when degree = 4.61) from the particle filter method closely follows the single data points from the DV and euclidean methods. In general, algorithms such as euclidean would trade off coverage for accuracy, whereas algorithms such as DV-Hop would trade off accuracy for coverage. Instead of relying on different AHLAs for

different trade-off objectives, our particle filter algorithm is capable of exploiting such a trade-off by selecting the coverage based on the filter variance. Such characteristics make the particle filter solution versatile in adapting to the different localization requirements. For instance, for the application that prefers accuracy to coverage, only those estimates with smaller variances can be considered valid estimates. Conversely, for applications that prefer better coverage, estimates with larger variances can also be considered as valid.

Fig. 6b shows results where only AoA sensors are available. It can be observed that the estimation error becomes high when the network is sparse. As explained in the aforementioned analysis, this can be attributed to the fact that AoA sensors are less capable of propagating location information through multihops. However, when both RSSI and AoA sensors are implemented (as shown in Figs. 6c and 6d), the estimation error can be drastically reduced. In particular, when the average degree is greater than 4.61, the 50 percent RSSI and 50 percent AoA sensor combination outperforms the cases of 100 percent RSSI or AoA sensors (Fig. 6c). This indicates that mixing different sensor data can be very beneficial in ad hoc localization. When all nodes have both RSSI and AoA capacities, the performance is even better (Fig. 6d). This result confirms the results in [4], where the authors claim that localization results can be drastically improved when both RSSI and AoA capacities are available at all nodes.

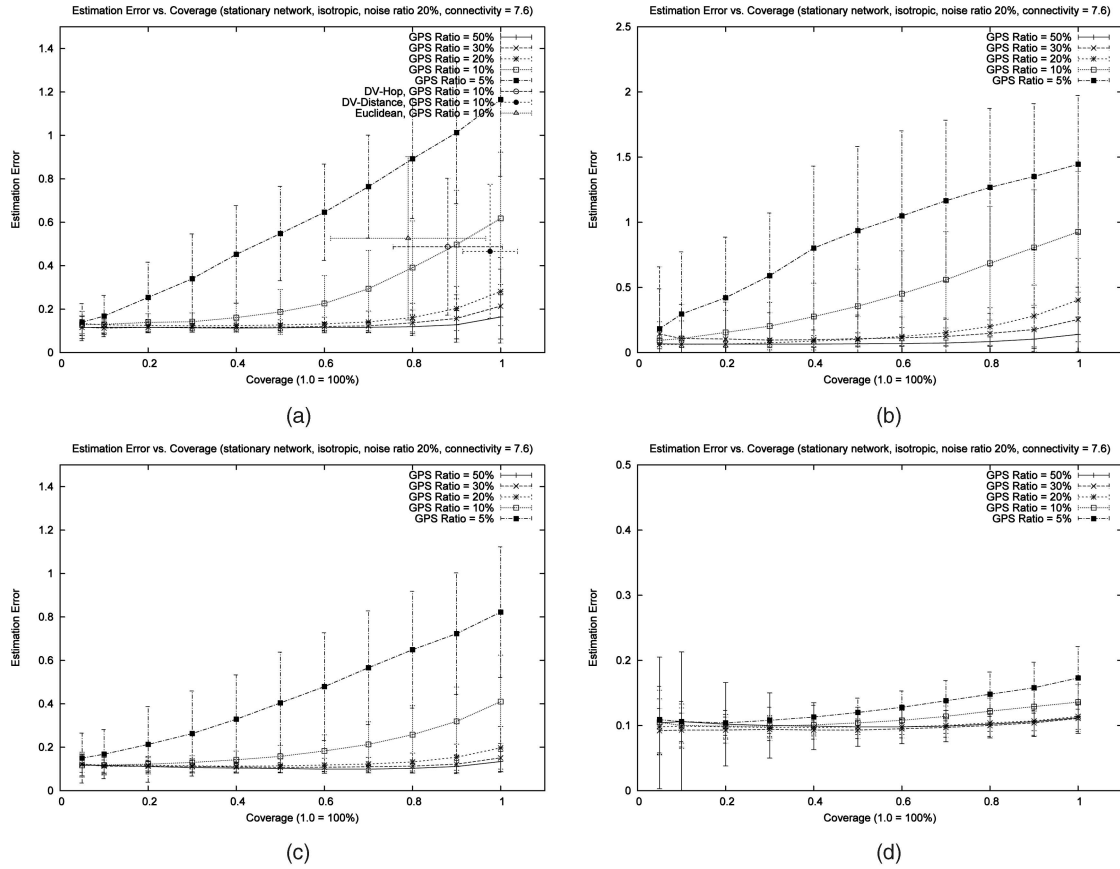


Fig. 7. Effect of anchor ratio on the estimation error. (a) RSSI: 100 percent; AoA: 0 percent. (b) RSSI: 0 percent; AoA: 100 percent. (c) RSSI: 50 percent; AoA: 50 percent. (d) RSSI: 100 percent; AoA: 100 percent.

## 5.2 Anchor Ratio

The effect of the anchor ratio on the estimation error is shown in Fig. 7. Again, we consider four different types of sensor configurations. As expected, higher anchor ratio lowers the overall estimation error in all cases. Furthermore, when half of the nodes have RSSI sensor capacity and the other half have AoA capacity (Fig. 7c), the result is better than using just one sensor type (Figs. 7a and 7b). This further proves the advantage of using mixed types of sensors in ad hoc localization.

## 5.3 Noise

The effect of the noise ratio  $x$  on the location estimates is shown in Fig. 8. In general and as expected, a lower noise ratio leads to lower estimation error. However, there is an exception in cases when only RSSI sensors are used and the coverage is high (Figs. 8a and 8c). In those cases, higher noise ratio, such as 0.2 versus 0.1 and 0.01, would actually result in a lower estimation error when the coverage exceeds 30 percent. This is indeed a side effect of the noise model. (Recall that the noise model is based on a simple uniform distribution.) A higher noise ratio means a node could hear from more neighbors because the uniform noise sometimes increases the actual transmission range. Therefore, when the noise ratio is high, even though the distance estimates become less accurate, more neighbors can be heard. In other words, more nodes are likely to obtain their

general locations, but those location estimates are not very accurate.

Another observation to be made is that the estimation error difference between various noise ratios decreases as the actual estimation error increases. For instance, the difference between the four noise ratios is quite small when the coverage is 100 percent, and only AoA sensors are used (Fig. 8b). We can also observe that, when only AoA sensors are used, fewer nodes can localize themselves simply because of the geometric limitation of the AoA method. Thus, the average localization error for all nodes increases, and such an increase is attributed more to the geometric limitation than to the signal noise. Thus, the effect of the noise ratio becomes less apparent.

## 5.4 Mixed Sensor Types

To further evaluate the localization result of mixed sensor types, we added two additional types of sensor configurations that include blind nodes. In particular, we consider networks with one-third of the nodes having RSSI, one-third having AoA sensors, and one-third blind. We will also consider networks in which all nodes are blind. For these experiments, we use a moderately connected (the average degree is 7.66) network, an anchor ratio of 10 percent, and a noise ratio of 20 percent. Fig. 9 shows the localization performance of the network with the above configuration versus the four other types of sensor configurations that we saw earlier. In terms of overall estimation error versus

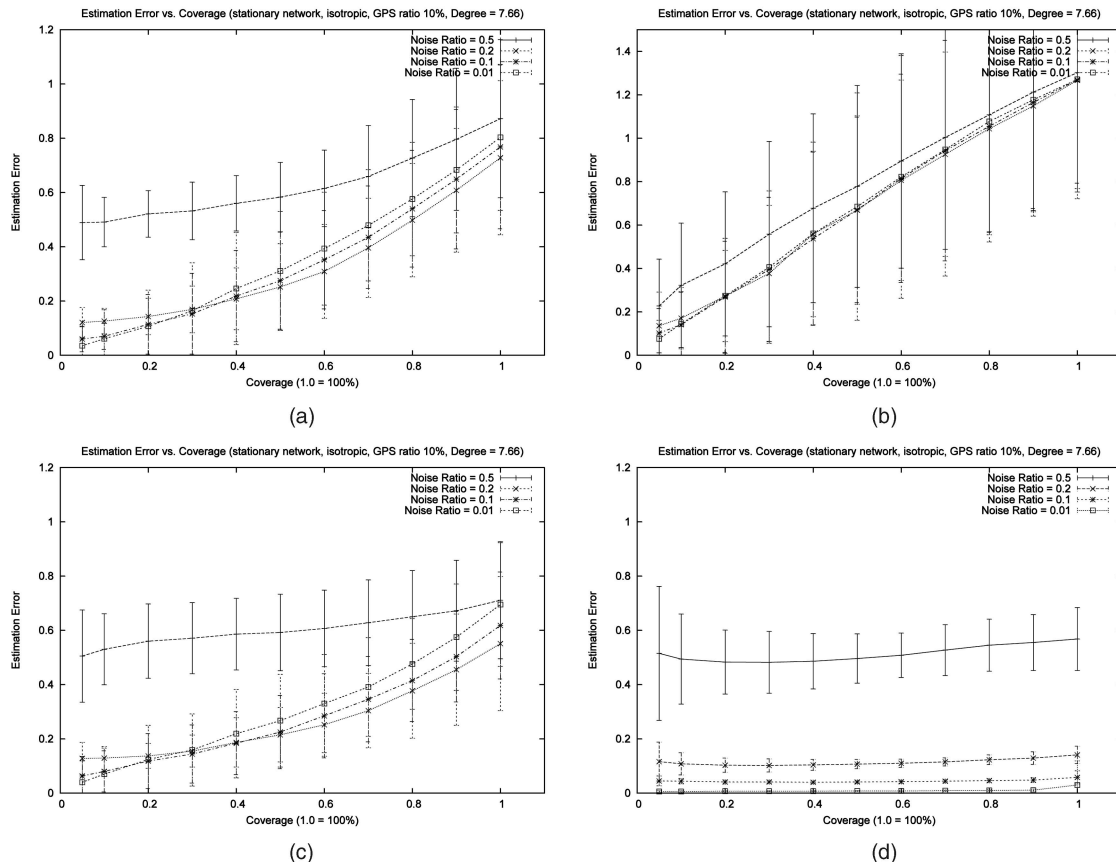


Fig. 8. Effect of noise ratio on the estimation error. (a) RSSI 100 percent; AoA 0 percent. (b) RSSI 0 percent; AoA 100 percent. (c) RSSI 50 percent; AoA 50 percent. (d) RSSI 100 percent; AoA 100 percent.

coverage (Fig. 9a), networks with one-third blind nodes perform reasonably well. Compared to networks with 100 percent RSSI capacity, the networks with one-third blind nodes have a higher localization error when the coverage is low, but they catch up when the coverage increases. This means that, although the estimated locations of individual nodes might not be as accurate as in the networks with 100 percent RSSI sensors, more nodes are capable of identifying their rough locations. When all nodes are blind nodes, our particle filter framework still gives reasonable estimates. About half of the nodes are localized within 60 percent of the transmission range, and all the nodes are localized within 100 percent of the transmission range.

Fig. 9b shows how the estimation error behaves for different types of sensor configurations in a function of time. For each of the simulation runs, we capture estimated locations at every second and compare them to the actual locations. As expected, the estimated locations become more accurate as more information is exchanged among neighbors. As such, location information from anchors eventually propagates throughout the network and allows nonanchors to localize themselves. Note that the convergence happens smoothly in all sensor configurations. This can be attributed to our filter update methods, which rely on the variances of the particle distributions. The differences between the variances allow the nodes with more accurate location information to impact the nodes with less accurate location information, and not vice versa.

## 5.5 Results on Mobile Networks

Previous works on the AHLF generally do not contain extensive simulation and analysis when the network is indeed mobile (as the definition of manets implies). As discussed earlier, most previous methods are specifically designed to work in stationary sensor networks, in which it is sufficient to complete one round of localization and there is no requirement for further adjustment when topology changes. Thus, adapting them to work in mobile networks can be quite challenging (see Section 6 for a more detailed discussion on this). In the worst case, the entire localization scheme has to be redone every time the network changes. Our method, however, is specifically designed to work in mobile networks. In our case, since the entire AHLA relies on a simple location exchange between the neighbors and there is no complicated multiphase operation, we can let the same particle filter framework run continuously as nodes move about.

For our mobile node localization simulations, we set the population to 100 nodes with an original average degree of 7.6. We use the epoch-based mobility model of [20] to simulate node movement, which is widely accepted as a good mobility model for ad hoc networks (in general, it is deemed more realistic than a Brownian motion model). The entire movement path of the node is defined by a sequence of “epochs,” that is,  $(e_1, e_2, \dots, e_n)$ . The duration of each epoch is independently and identically distributed (I.I.D.) and exponentially distributed with a mean of  $1/\lambda$ ; within

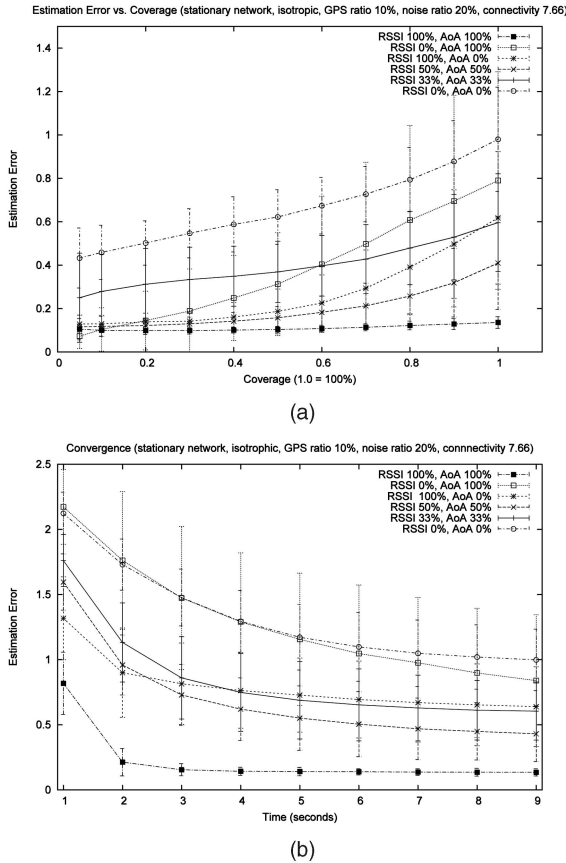


Fig. 9. Result of mixed sensor types. (a) Estimation error. (b) Convergence.

each epoch, nodes move with a constant velocity vector. At the end of each epoch, nodes randomly select a new velocity vector. The direction of the movement is I.I.D. uniform between 0 and  $2\pi$ . The absolute value of the velocity is I.I.D. normal with a mean of  $\mu$  and a variance of  $\sigma^2$ . Our simulation uses a fixed mean and variance such that  $\mu = \sigma$ . The result is obtained by using two different means and standard deviations of 5 and 20 m/s. The expected amount of time that a node maintains its current velocity is set at 5 seconds ( $\lambda = 5$ ).

Fig. 10 shows the filter convergence on mobile networks for the first 30 seconds. In both low (5 m/s) and high (20 m/s) mobile networks, the estimation error drops quickly in the first several seconds. This represents the phase when the nodes localize themselves initially. In slowly moving networks (Fig. 10a), the estimation error stays around the same level after the initial localization, which indicates that the localization process is happening quickly enough to adapt to the location change. In networks with rapid node movements (Fig. 10b), however, rapid location change increases the overall estimation error. It is interesting to note that network mobility tends to have less impact in the case when AoA sensors and/or blind nodes are used. In fact, although the stationary networks with RSSI-only nodes outperform the stationary networks with AoA-only nodes, the opposite is true for mobile networks. This indicates that, for our particle filter framework, nodes with AoA sensors tend to adapt to mobility faster than nodes with RSSI sensors.

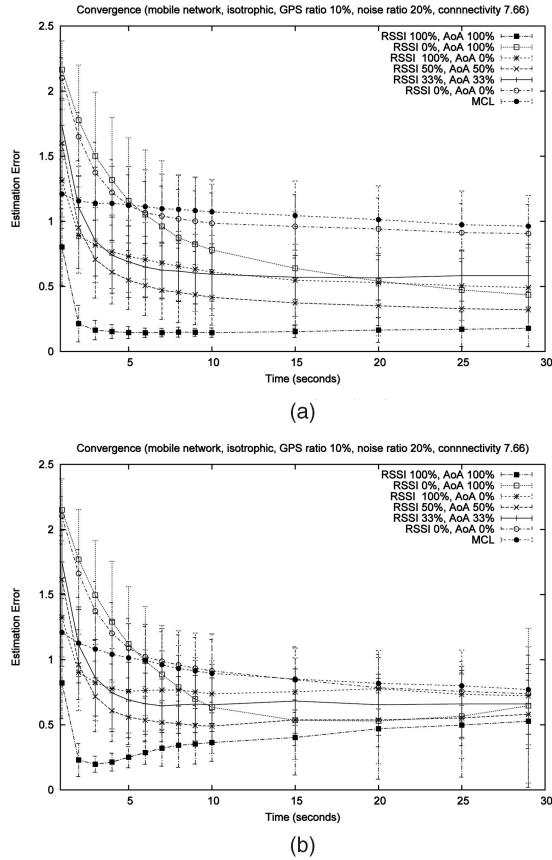


Fig. 10. Filter convergence with mobile networks. (a) average speed = 5 m/s. (b) average speed = 20 m/s.

For comparison, we also implemented the Monte Carlo localization (MCL) method proposed in [8], which is another probabilistic localization method based on particle filtering. MCL is designed for mobile networks and relies only on the connectivity information; thus, it behaves much like the blind node scenario in our algorithm. The main difference is that MCL uses the anchor locations (up to two hops away) as the measurement, whereas our algorithm uses both anchor *and* nonanchor locations (one hop away). As shown in Fig. 10, MCL has an initial lower localization error because the location information from the anchors two hops away can be used directly. Our algorithm uses the location information from the anchors more than one hop away indirectly via the exchange of the location distributions among the neighbors and, thus, it would take more iterations to converge. However, our algorithm has the advantage that the location information from the anchors is implicitly contained in the location distribution of each node. Via the exchange of the location distributions, the anchor information essentially propagates freely, and there is no limiting factor of two hops like MCL. Thus, we observe in Fig. 10 that, once particles collapse, our algorithm has a lower localization error than MCL.

In mobile networks, the frequency of filter updates can have an impact on the localization accuracy. Since the nodes are constantly moving, the current location estimation can become obsolete very quickly. Thus, the particle filters need to be updated at a sufficient frequency

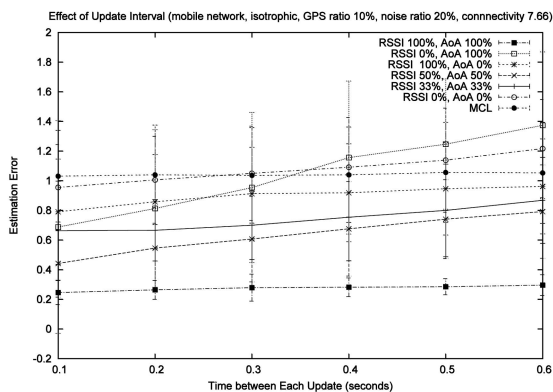


Fig. 11. Effect of update interval size in mobile networks (average speed = 20 m/s).

to keep up with the node movement. The effect of the filter update frequency on the localization error is shown in Fig. 11. Here, we use an average node movement speed of 20 m/s and vary the filter update frequency from 0.1 to 0.5 second (previous simulations use 0.5 second). As expected, the localization error generally increases with larger interval size. However, the increase is minimal in the case of 100 percent RSSI and 100 percent AoA, as well as in MCL, since they converge faster than others do. (We argue that, by increasing the update rate appropriately, an arbitrarily rapid mobility could be tracked. Indeed, the update rate should be determined based on the current local mobility rate of nodes.)

## 6 DISCUSSION

Most previous AHLAs attempt to pinpoint each node at a single location. Although this is the ultimate goal of any AHLA, it is fundamentally impossible for most topological scenarios. Thus, the challenge in the AHLA really lies in *estimating* the location of nodes, especially those that are more than one hop away from the anchors. In such case, a node might not know its *exact* location, but it may estimate a number of *possible* locations (thus obtaining partial information). A probabilistic approach provides a natural way of representing such partial information. In particular, the node location may not be presented as a fixed value, but as a probability distribution. Initially, a node unknown of its own location has its location distribution uniformly spread over the entire deployment area. As the localization process proceeds, the location distribution is updated, and it is expected to converge to a more concentrated location estimate (ideally, to a single high probability location). In our previous work [10], we applied such a probabilistic model by using RSSI as our measurements and particle filtering as the method of updating the location distributions. We showed that the probabilistic method achieves a much better balance of the trade-off between the estimation error and coverage when considering the limitation of DV-Hop/DV-Distance (high estimation error) and euclidean (low coverage). Our method has the following advantages over most existing AHLAs:

1. *Measure of estimation quality.* DV-based algorithms can generate location estimates covering only a subset of nodes. The coverage of the estimates depends on the nature of the algorithm. There is always a trade-off between the coverage and the quality of the estimates. Some algorithms (such as DV-Hop) give better coverage, whereas others (such as euclidean) give better estimates. Our method, however, generates location estimates for all nodes in the network. Each estimate has an implicit variance associated with it, serving as a quality measure. Thus, the coverage of our estimates is not a fixed value, but a function of the variances. In practice, certain applications might desire better estimation quality, whereas others might desire better coverage. For instance, LAR protocols [13] might only need a very rough location estimation of the destination node when the request is still far away, but as the request moves closer to the destination node, more accurate location estimation is needed. Previously, different AHLAs had to be applied separately to accomplish the two objectives. Our method produces results satisfying both scenarios and does it in the same probabilistic framework.
2. *Single-phase operation.* Many algorithms employ multiple phases during the localization process. For instance, DV-Hop requires a first phase to calculate per-hop distance and a second phase to propagate the result. Multilateration methods [25] require three phases: initial estimation, grouping, and refinement. Our method, however, has the advantage of a single-phase operation. From the implementation point of view, our algorithm can be easily implemented in distributed fashion because nodes do not have to collectively maintain the state information of “which phase are we in?” From the functional point of view, the probabilistic nature of our method simplifies the algorithm by eliminating the need for multiple phases. In multilateral methods, an initial estimate is obtained based on a certain measure (distance or hops) to anchors followed by the phase of further refinement. The initial location estimate suffers because information from nonanchors is not used. The refinement phase is needed so that information from nonanchors can be incorporated into the estimates. Our method does not need separate phases, as the information from nonanchors is automatically applied as soon as it becomes available. In particular, nonanchors become more and more certain of their locations (their location variances decrease), allowing their estimates to be used by neighboring nodes.
3. *Simple communication model and fast convergence.* Our method employs a simple computation and communication model which relies solely on local broadcast (broadcast to neighbors only). This allows our method to be naturally integrated into periodical Hello messages (as generally used by mobile nodes in ad hoc networks to declare their existence); we do not require a new type of control message. Furthermore, our simulations show that, compared to

existing methods such as APS, our method generally converges with less message overhead.

4. *Mobile readiness.* Because our algorithm eliminates multiple phases and uses a simple communication model, it can be applied directly to mobile networks. Whereas previous works do not generally provide a simulation result for mobile scenarios, we demonstrate via simulation that our method can be effectively used in manets.

To the best of our knowledge, our work is the first that incorporates multiple sensory data for localization using the same algorithmic framework and enables localization for nodes in both static and mobile multihop networks. The need for such algorithms is established in [4], but the authors only consider the limited case when both RSSI and AoA are available at nodes. Our work is more general in that we consider the cases when the nodes can either have RSSI, AoA, both, or no sensory data at all. The same particle filter framework allows the nodes with different sensory capacities to collaborate in the localization.

## 7 CONCLUSION AND FUTURE WORK

In this paper, we proposed a particle filter framework that solves the localization problem in both stationary networks and manets. Compared to previous localization algorithms, our framework is general enough to accommodate different sensory capacities regardless of the availability of ranging (such as RSSI) or sectoring (such as AoA). More importantly, our framework allows the networks consisting of nodes with different sensor types to collaborate in the localization process. By leveraging the filter variances, nonanchor nodes localize themselves by simple location data exchanges with their neighbors without going through an initial localization phase and a refinement phase; the filter variances also give a measure of the estimation accuracy. The differentiation in estimation accuracy could be very useful, as different applications might have different accuracy requirements.

Our analyses and simulation studies showed that, in stationary networks, AoA sensors, by themselves, do not work well when the network connectivity or the anchor ratio is low. Whereas using only AoA sensors would result in low localization coverage, good results can be achieved by combining AoA-only sensors and RSSI-only sensors in the same network. In fact, networks where 50 percent of the nodes have only AoA sensors and another 50 percent of the nodes have only RSSI sensors can achieve a better localization result than networks dominated by one type of sensors. Furthermore, our analysis shows that employing AoA sensors is highly beneficial in mobile networks, especially when the network mobility is high. Simulation studies also validate the effectiveness of incorporating different sensor capacities (RSSI, AoA, or blind) using the same particle filter framework. With our framework, reasonable results can be obtained even if one-third of the nodes do not have any location sensors.

Although this work addresses network mobility, we believe that there is room for further evaluation. In particular, our current framework assumes that particle filters are continuously updated with the same rate. The update rate, however, could be tied to the mobility in such a

way that the location updates are executed on demand. This way, when the node movement is minimal, we can eliminate the unnecessary updates and thus reduce the network traffic and save the power consumption by switching off the sensors. Furthermore, we are currently using a rather simplistic movement model in the particle filter (Gaussian displacement). This model could be improved by learning movement patterns (models) and their parameters (such as velocity and acceleration); we are targeting our research at those enhancements for mobile networks in the near future.

## ACKNOWLEDGMENTS

The authors would like to thank the editor and reviewers for their assistance in revising this work.

## REFERENCES

- [1] L. Blazevic, J.-Y. Le Boudec, and S. Giordano, "A Location-Based Routing Method for Mobile Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol. 4, no. 2, Mar.-Apr. 2005.
- [2] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm. Magazine*, vol. 7, no. 5, pp. 28-24, Oct. 2000.
- [3] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-Free Positioning in Mobile Ad Hoc Networks," *Proc. 34th Hawaii Int'l Conf. System Sciences*, vol. 9, p. 9008, 2001.
- [4] K. Chintalapudi, R. Govindan, S. Sukhatme, and A. Dhariwal, "Ad Hoc Localization Using Ranging and Sectoring," *Proc. INFOCOM*, pp. 2662-2672, Apr. 2004.
- [5] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," *Proc. Nat'l Conf. Artificial Intelligence*, July 1999.
- [6] N. Gordon, "Bayesian Methods for Tracking," PhD thesis, Univ. of London, 1993.
- [7] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T.F. Abdelzaher, "Range-Free Localization Schemes in Large-Scale Sensor Networks," *Proc. MobiCom*, pp. 81-95, 2003.
- [8] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," *Proc. MobiCom*, pp. 45-57, 2004.
- [9] Y.-C. Hu, A. Perrig, and D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks," *Proc. INFOCOM*, vol. 3, pp. 1976-1986, Apr. 2003.
- [10] R. Huang and G.V. Záruha, "Location Tracking in Mobile Ad Hoc Networks Using Particle Filters," *Proc. Int'l Conf. Ad-Hoc Networks and Wireless (ADHOC-NOW '05)*, pp. 85-98, 2005.
- [11] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *Proc. Fourth European Conf. Computer Vision*, vol. 1, pp. 343-356, 1996.
- [12] X. Jiang and T. Camp, "Review of Geocasting Protocols for a Mobile Ad Hoc Network," *Proc. Grace Hopper Celebration (GHC '02)*, 2002.
- [13] Y. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Wireless Networks*, vol. 6, no. 4, pp. 307-321, July 2000.
- [14] Y. Ko and N.H. Vaidya, "Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms," *Proc. Second IEEE Workshop Mobile Computer Systems and Applications (WMCSA '99)*, p. 101, 1999.
- [15] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric Ad Hoc Routing: Of Theory and Practice," *Proc. 22nd ACM Symp. Principles of Distributed Computing (PODC '03)*, pp. 63-72, 2003.
- [16] W.-H. Liao, Y.-C. Tseng, and J.-P. Sheu, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," *Telecomm. Systems*, vol. 18, nos. 1-3, pp. 37-60, 2001.
- [17] W.-H. Liao, Y.-C. Tseng, K.-L. Lo, and J.-P. Sheu, "Geogrid: A Geocasting Protocol for Mobile Ad Hoc Networks Based on Grid," *J. Internet Technology*, vols. 1-2, pp. 23-32, 2000.
- [18] T. Liu, P. Bahl, and I. Chlamtac, "A Hierarchical Position-Prediction Algorithm for Efficient Management of Resources in Cellular Networks," *Proc. IEEE Global Telecomm. Conf. (GLOBE-COM '97)*, vol. 2, pp. 982-986, Nov. 1997.

- [19] M. Mauve, H. Fuler, J. Widmer, and T. Lang, "Position-Based Multicast Routing for Mobile Ad Hoc Networks," Technical Report TR-03-004, Dept. Computer Science, Univ. of Mannheim, 2003.
- [20] A.B. McDonald and T. Znati, "A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks," *IEEE J. Selected Areas in Comm.*, special issue on wireless ad hoc networks, vol. 17, no. 8, Aug. 1999.
- [21] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '01)*, vol. 5, pp. 2926-2931, 2001.
- [22] D. Niculescu and B. Nath, "DV-Based Positioning in Ad Hoc Networks," *Telecomm. Systems*, vol. 22, nos. 1-4, pp. 267-280, Jan.-Apr. 2003.
- [23] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AoA," *Proc. INFOCOM*, vol. 3, pp. 1734-1743, 2003.
- [24] N. Priyantha, A. Miu, H. Balakrishnan, and S. Teller, "The Cricket Compass for Context-Aware Mobile Applications," *Proc. MobiCom*, pp. 1-14, 2001.
- [25] A. Savvides, H. Park, and M.B. Srivastava, "The n-Hop Multi-lateration Primitive for Node Localization Problems," *Mobile Networks and Applications*, vol. 8, no. 4, pp. 443-451, Aug. 2003.
- [26] R. Stoleru and J.A. Stankovic, "Probability Grid: A Location Estimation Scheme for Wireless Sensor Networks," *Proc. First IEEE Conf. Sensor and Ad Hoc Comm. and Networks*, pp. 430-438, Oct. 2004.
- [27] S. Thrun, "Particle Filters in Robotics," *Proc. 17th Ann. Conf. Uncertainty in AI (UAI '02)*, Aug. 2002.
- [28] M. Waelchli, M. Scheidegger, and T. Braun, "Intensity-Based Object Localization and Tracking with Wireless Sensors," *Proc. ACM Workshop Real-World Wireless Sensor Networks (REALWSN '06)*, June 2006.
- [29] Y. Xu, J. Heidemann, and D. Estrin, "Geography-Informed Energy Conservation for Ad Hoc Routing," *Proc. MobiCom*, pp. 70-84, 2001.
- [30] G.V. Záruba, M. Huber, and F.A. Karmangar, "Monte Carlo Sampling Based In-Home Location Tracking with Minimal RF Infrastructure Requirements," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '04)*, Dec. 2004.



**Rui Huang** received the BS degree in computer science from Texas Christian University in 1998 and the PhD degree in computer science from the University of Texas at Arlington in 2006. He has held a number of positions in the software industry. His research interests include wireless networks, sensor networks, network protocols, and algorithms. He is a member of the IEEE and the IEEE Communications Society.



**Gergely V. Záruba** received the MS degree in computer engineering from the Department of Telecommunications and Telematics, Technical University of Budapest, in 1997 and the PhD degree in computer science from the University of Texas at Dallas in 2001. He is an assistant professor of computer science and engineering at the University of Texas at Arlington (CSE@UTA). He has served on many organizing and technical program committees for leading conferences and has been a guest editor of journals. His research interests include wireless networks, algorithms and protocols, performance evaluation, and current wireless and assistive technologies. He is a member of the IEEE and the IEEE Communications Society.

► **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**