Location Tracking in Mobile Ad Hoc Networks using Particle Filter

Rui Huang and Gergely V. Záruba
Computer Science and Engineering Department
The University of Texas at Arlington
416 Yates, 300NH, Arlington, TX 76019
email: rxh1725@omega.uta.edu, zaruba@uta.edu

January 10, 2006

Abstract

Mobile ad hoc networks (MANET) are dynamic networks formed on-the-fly as mobile nodes move in and out of each others' transmission ranges. In general, the mobile ad hoc networking model makes no assumption that nodes know their own locations. However, recent research shows that location-awareness can be beneficial to fundamental tasks such as routing and energy-conservation. On the other hand, the cost and limited energy resources associated with common, low-cost mobile nodes prohibits them from carrying relatively expensive and power-hungry location-sensing devices such as GPS. This paper proposes a mechanism that allows non-GPS-equipped nodes in the network to derive their approximated locations from a limited number of GPS-equipped nodes. In our method, all nodes periodically broadcast their estimated location, in term of a compressed particle filter distribution. Non-GPS nodes estimate the distance to their neighbors by measuring the received signal strength of incoming messages. A particle filter is then used to estimate the approximated location, along with a measure of confidence, from the sequence of distance estimates. Simulation studies show that our solution is capable of producing good estimates equal or better than the existing localization methods such as APS-Euclidean for the more difficult scenario when the network connectivity is low.

1 Introduction

Mobile ad hoc networks (MANET) are constructed on the fly as the network nodes move in and out of the transmission range of each other. A major challenge in protocol design for this type of networks is to provide mechanisms that deal with the dynamical topology change. Constant topology change makes it more difficult for fundamental tasks such as routing since the routing algorithm cannot simply rely on its previous knowledge of the network topology. Furthermore, even after a route has been successfully established, it can still be disrupted at any time due to the movement of the intermediate nodes. For this reason, most protocols originally designed for static networks cannot be adopted to ad hoc networks without significant change. Many protocols have to be redesigned for ad hoc networks in order to cope with the topology change.

Studies have shown that innovative algorithms can aid mobile ad hoc network (MANET) protocols if the nodes in the network are capable of obtaining their own as well as other nodes' location information. For instance, algorithms such as LAR [8], GRID [11], and GOAFR+ [10] rely on the location information to provide more stable routes during unicast route discovery. The location information is also applied to geocast (multicast based on geographic information) [7] for algorithms such as LBM [9], GeoGRID [12] and PBM [14]. To minimize the power consumption, The GAF algorithm [23] uses the location information to effectively modify the network density by turning off certain nodes at particular instances.

The algorithms listed earlier all rely on the availability of reasonably accurate location information. This assumption is valid for networks in which some location sensing devices, such as GPS receivers, are available at all nodes. However, in reality this is rarely the case; although GPS receivers are increasingly cheaper to produce and becoming more widely available, they are still relatively expensive and power-hungry. GPS receivers also require line-of-sight to satellite, which precludes indoor usage. Therefore, it is too general to assume that they will be applicable to every node in the ad hoc

networks. For this reason different algorithms have been proposed to derive approximated locations of *all* nodes based on the relaxed assumption that direct location sensing devices (such as GPS) are available to only a *subset* of the nodes.

This paper presents a solution to the location tracking problem based on particle filters. Given an ad hoc network with limited number of location-aware nodes, our solution estimates the locations of all other nodes by measuring sensory data, in this particular case the received signal strength indication (RSSI), from neighbors. For each node, the estimated location is viewed as a probabilistic distribution maintained by a particle filter. Unlike other location tracking methods, our solution has low overhead because it is purely based on local broadcasting and does not require flooding of the location information over the entire network. Simulation studies show that even without flooding, our solution can still generate good estimates comparable to other existing methods, given that the percentage of GPS nodes is not extremely low. In addition when connectivity is low, our algorithm is still able to derive location information which is not the case with most of the other approaches. While most algorithms either attempt to increase the accuracy of the estimate or to increase the coverage, our algorithm recognizes the tradeoff between the two and provides a quantative measure for both. From the implementation point of view, our algorithm can be easily implemented in distributed manner for both stationary and mobile networks. Most importantly, our algorithm provides a probabilistic framework in which other sensory data (such as angle of arrival) can be naturally incorporated in the future.

2 Related Works

Given a network graph $G = (V, E)$ in which the number of location-aware nodes (also called *anchor* nodes) $|V_{gps}| \leq |V|$, the objective of the location tracking algorithm is to find the locations of *non-anchor* nodes $\{V\} - \{V_{gps}\}$. In this section we survey the previous work on the location tracking problem in ad hoc networks.

Generally speaking, there are two categories of distributed localization methods depending on whether sensory data are used. The methods that do not use sensory data are simpler but tend to perform poorly especially when anchor ratio is low or the network is sparse. The methods that do use sensory data generally perform better but tend to be significantly more complex. The performance in the latter case is also largely affected by the noise introduced to the sensory data which tends to aggregate rapidly as sensory data is propagated through the network.

The Centroid method [2] provides the most straight-forward solution that does not use sensory data. Assuming that a non-anchor node is capable of receiving the location information from multiple anchor nodes, the Centroid method derives the location of a non-anchor node as the average of its neighboring anchor nodes' locations. The method is simple and efficient, but it requires the anchor nodes to redundantly cover large areas for an acceptable performance. The APIT method [5] estimates the node location by isolating the area using various triangles formed by anchor nodes. The location of the node is narrowed down by analyzing overlapping triangles to determine whether the node is contained within the triangles. Both the Centroid method and the APIT method require the transmission range of anchors to be much greater than non-anchors (by an order of magnitude [5]) in order for nodes to obtain reasonable location estimates.

The DV-Hop method [18] allows the location information from anchor nodes to propagate through multiple hops. The locations of anchors are periodically flooded throughout the network much like the routing packets in a distance vector routing protocol. The locations of non-anchor nodes are derived geometrically by performing trilateration of the distance estimates from at least three anchor nodes. Here the distance estimates are obtained by multiplying the number of hops to the anchor node to a predefined average-distance-per-hop value. The DV-Hop method does not require a greater transmission range of anchors, and it works well even when the ratio between anchor and non-anchor nodes is low. However, the message complexity is rather high due to the flooding of the location information. Furthermore, because the average-distance-per-hop is an estimated value over the entire network, the accuracy of the location estimation suffers when the nodes are not uniformly placed over the network.

Other, significant location tracking methods make use of additional sensors. In [13], the location, velocity and acceleration of mobile nodes are estimated by measuring the received signal strength indicator (RSSI) from multiple base stations in a cellular network. The measured power levels are fed into a Kalman filter to smooth out (filter) the erratic readings and thus be able to derive the distance. Since base station locations are assumed to be well-known in a cellular network, mobile nodes can use them as reference points for location estimation. In [17], the authors assume that non-anchor nodes are equipped with devices that measure the incoming signal directions. The directional information allows the receivers to obtain the angle of arrival (AoA) of the signal thus allowing more accurate location estimates than the pure DV-Hop

method. The DV-Distance method [18] is similar to the DV-Hop method but uses the estimated distance instead of the hop count during trilateration. In [20] after obtaining the initial location estimates from the DV method, the nodes obtain the estimated locations from the neighbors via local broadcast. The RSSI readings also provide the distance estimates from the neighbors. Using the distance estimates along with the estimated locations from the neighbors, the nodes can refine their initial location estimates via trilateration.

Hardware-wise, sensors that measure RSSI are widely available to mobile devices. Indeed, most off-the-shelf technologies implicitly provide such information (e.g., most Wi/Fi cards provide with RSSI). Based on RSSI and an underlying signal propagation model, the distance to the sender can be estimated. Because of the noise caused by multipath fading and far field scattering during the signal transmission, the distance estimates derived from RSSI suffer accordingly, especially when a large number of obstacles present. However, a number of mechanisms have been proposed to improve the accuracy of such estimates, such as the ones that use a more robust acoustic ranging system [3], device calibration on the RSSI sensors [22], and Kalman filters to smooth out the odd readings from the sensors [6]. Experiments have shown that the distance estimation error can be drastically reduced by using those methods. Thus, the RSSI-based methods are becoming more practical solutions to the location tracking problem in ad hoc networks.

3 Particle Filter Solution

"Geometrically speaking," in order to find the location of a node in a 2-dimensional space, the distances and locations of at least three anchors need to be known (as each of these anchors define a circle where the target node could be). In a network where the percentage of anchors is low, the major challenge is to obtain the distances and locations of anchors when the node is several hops away from the anchors. Previous works resolve this problem by either 1) assuming a greater transmission range of anchors [2, 5] (thus, anchors are always 1-hop away), or 2) broadcasting the anchor locations hop-by-hop over the entire network [18, 17, 20]. The assumption made in the first solution requires the network to be heterogeneous in the node types (in which anchors' radios are considered different than those of non-anchors) and requires homogeneity (uniformity) for anchor nodes' location over the area. The flooding of the location packets in the second solution requires extra overhead. This overhead can be especially heavy when nodes are mobile, where location packets need to be re-broadcasted repeatedly by nodes. Furthermore, most methods in [18, 17, 21] requires multiple phases of operations such as a phase of initial location discovery followed by a phase of refinement. However, in a more general network model in which nodes are can come online and go offline at different time, it becomes more difficult to define the start and the end of a phase. Lastly, due to the geometric and algorithmic limitations, most existing methods produce the location estimates for a limited percentage of nodes. But, their estimates lack a measure that qualifies the estimates. In other words, one cannot tell how good those estimates are.

Recognizing various shortcomings of previous approaches, we propose a different location tracking method that is based on Bayesian filters using Monte Carlo sampling (also known as particle filters) introduced in [4]. Our method can be considered as 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 the distance estimates from RSSI reading. Essentially, the nodes estimate their own locations by interchanging the location distributions with their neighbors.

Our method has the following advantages over most existing localization methods:

1. *Provide a measure of estimation quality*. DV based algorithms can generate location estimates to a subset of nodes. The coverage of the estimates depends on the nature of the algorithm. There is always tradeoff between the coverage and the quality of the estimates. Some algorithms (such as DV-Hop) give better coverage, while other (such as Euclidean) gives better estimates. Our method, however, generates location estimates for all nodes in the network. Each estimate is qualified by a variance, which serves as the 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 while other might desire better coverage. Previously, different localization methods need to be applied separately to accomplish the two objectives. Our method, however, produces the result satisfies both scenarios all in 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. The multilateration methods [21] contains three phrases of 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 GPS nodes followed by phase of further refinement. The initial location estimate suffers because infomation from non-GPS nodes are not used. The refinement phase is needed so that information from non-GPS nodes can be incorporated into the estimates. Our method does not need separate phases, as the information from non-GPS nodes is automatically applied as soon as it becomes available. In particular, as non-GPS nodes becomes more aware of their locations, their variances decrease, which allows 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 the periodical Hello messages used by mobile nodes in ad hoc networks to declear their existence. No new type of control messages is needed. Furthermore, our simulation shows that comparing to existing method such as APS, our method generally converges with less message overhead.

4. *Mobile ready*. Because of our algorithm eliminates multiple phases and uses a simple communication model, it can be applied directly to mobile networks without any modification. While previous works do not generally provide simulation result for mobile scenarios, we demonstrate via simulation that our method can be effectively used in mobile ad hoc networks.

5. *Extensibility*. Peering away the dependency to the RSSI signal readings, the core of our algorithm is a probabilistic framework based on particle filtering that is extremely versatile. The framework can be easily extended to different signal and network models. For instance, unlike DV-Hop, our method does not assume that all nodes have the same transmission range. Unlike Centeriod or APIT, our method does not require a greater range for GPS nodes, which allows it to work in homogeneous networks. Furthermore, the framework is not tied to a particular signal propagation model or a particular sensory data. Although we have not implemented it, we expect other sensory data such as angle of arrival (AoA) can be used in place of RSSI as the input to our algorithm. More interestingly, the same probabilistic framework will allow multiple sensory data working together to localize the network. In other words, a subset of nodes is capable of AoA readings while another subset is capable of RSSI readings. The framework provided by our algorithm can be adapted to solve such problem.

A similar Bayesian based approach has been proposed in [24] for the in-door location tracking problem. In [24], because of the different obstacles (walls, windows and doors) presented in the in-door floor-plan, a signal strength (RSSI) map needs to be obtained via measurement ahead of time. The location tracking problem then becomes a decision-making problem. The problem can be solved using a measurement model that compares RSSI with the signal strength map to find the location in the map that contains the largest probability of matching the current RSSI characteristics. While similar, our solution is designed for our-door environment in which obstacles are assumed to be minimum, and fairly reliable distance estimates can be obtained from RSSI readings and the signal propagation model. Based on those assumptions, our solution does not require the RSSI map. The probability distributions of location estimates are updated solely from the distance and location estimates from neighbors.

Fig. 1 demonstrates how our method solve the localization problem in a simple scenario. Here, node 2, 3 and 4 are GPS nodes, and node 0 and 1 are non-GPS nodes. Of the non-GPS nodes, node 0 can receive signal from 1 and 4 only, and node 1 can receive signal from node 1, 2 and 3 only. The probability distribution of the estimated location is represented by the particles (dots) in the graph. In (a), node 0 can only receive signal from node 4. Thus, as the particle distribution indicate, the probability distribution where node 0 locates at is a circle around node 4. In (b), node 1 can receive signal from node 2 and 3. Thus, the probability where node 1 locates centers around two areas where circles around node 2 and 3 intersect. Intuitively, in order to localize itself a node-GPS node needs to receive location information from a minimum three GPS-nodes either directly or indirectly. In both case (a) and case (b), the exact location of the non-GPS nodes 0 and 1

cannot be deduced because they do not receive location information from all three GPS nodes. In (c) and (d), node 0 and 1 are able to communicate to each other and exchange their probability distributions. Thus, their exact locations are identified even though neither node receives location information from the all three GPS nodes directly.

3.1 Classic Monte Carlo Sampling-Based Bayesian Filtering

This section describes the theoretical background behind Bayesian filtering and how it can be applied to location estimation using RSSI. Let us envision a grid system superimposed over the entire tracking area, and let the state s_t be the location of the node to be tracked in the grid system at the time t . Our goal is to estimate the posterior probability distribution, $p(s_t|d_1,\ldots,d_t)$, of potential states - s_t, using the RSSI measurements, $d_1,\ldots,d_t$. The calculation of the distribution is performed recursively using a Bayes filter:

$$p(s_t|d_1,\ldots,d_t) = \frac{p(d_t|s_t)\cdot p(s_t|d_1,\ldots,d_{t-1})}{p(d_t|d_1,\ldots,d_{t-1})}$$

Assuming that the Markov assumption holds, i.e., $p(s_t|s_{t-1},\ldots,s_0,d_{t-1},\ldots,d_1) = p(s_t|s_{t-1})$, the above equation can be transformed into the recursive form:

$$p(s_t|d_1,\ldots,d_t) = \frac{p(d_t|s_t)\cdot \int p(s_t|s_{t-1})\cdot p(s_{t-1}|d_1,\ldots,d_{t-1})ds_{t-1}}{p(d_t|d_1,\ldots,d_{t-1})},$$

where $p(d_t|d_1,\ldots,d_{t-1})$ is a normalization constant. In the case of the localization of a mobile node from RSSI measurements, the Markov assumption requires that the state contains all available information that could assist in predicting the next state and thus, an estimate of the non-random motion parameters of the nodes is required as part of the state description. Starting with an initial, prior probability distribution, $p(s_0)$, a system model, $p(s_t|s_{t-1})$, representing the motion of the mobile node (the mobility model), and the measurement model, $p(d|s)$, it is then possible to drive new estimates of the probability distribution over time, integrating one new measurement at a time. Each recursive update of the filter can be broken into two stages:

Prediction: Use the system model to predict the state distribution based on previous readings

$$p(s_t|d_1,\ldots,d_{t-1}) = \int p(s_t|s_{t-1})\cdot p(s_{t-1}|d_1,\ldots,d_{t-1})ds_{t-1}$$

Update: Use the measurement model to update the estimate

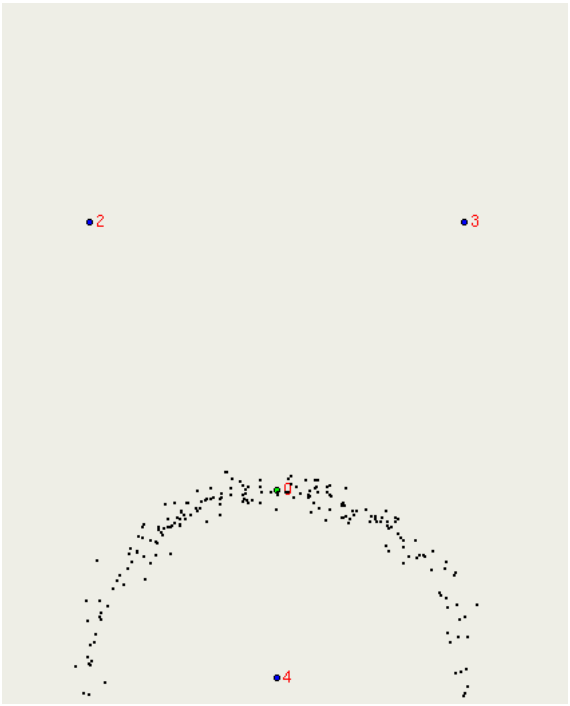
$$p(s_t|d_1,\ldots,d_t) = \frac{p(d_t|s_t)}{p(d_t|d_1,\ldots,d_{t-1})}p(s_t|d_1,\ldots,d_{t-1})$$

To address the complexity of the integration step and the problem of representing and updating a probability function defined on a continuous state space (which therefore has an infinite number of states), the approach presented here uses a sequential Monte Carlo filter to perform Bayesian filtering on a sample representation. The distribution is represented by a set of weighted random samples and all filtering steps are performed using Monte Carlo sampling operations. Since we have no prior knowledge of the state we are in, the initial sample distribution, $p_N(s_0)$, is represented by a set of uniformly distributed samples with equal weights, $\{(s_0^{(i)}, w_0^{(i)})|i\in[1,N], w_0^{(i)} = 1/N\}$ and the filtering steps are performed as follows:

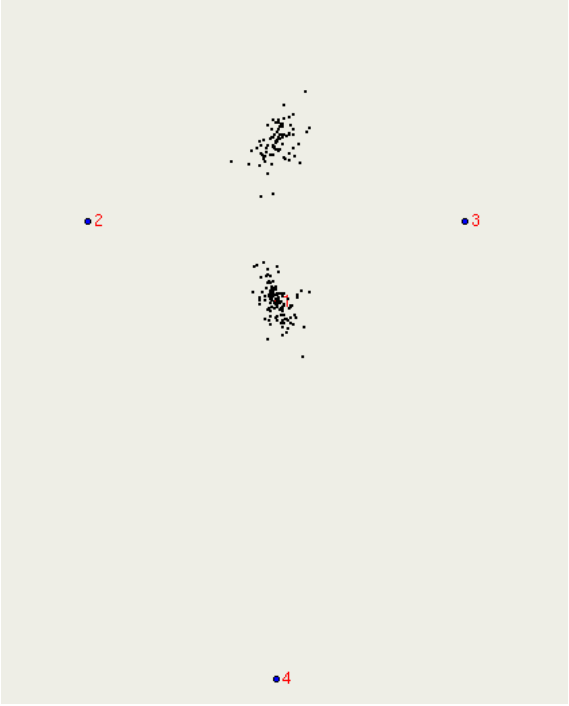
Prediction: For each sample, $(s_{t-1}^{(i)}, w_{t-1}^{(i)})$, in the sample set, randomly generate a replacement sample according to the system (mobility) model $p(s_t|s_{t-1})$. This results in a new set of samples corresponding to $p(s_t|d_1,\ldots,d_t)$:

$$\{(\tilde{s}_t^{(i)}, w_t^{(i)})|i\in[1,N], w_t^{(i)} = 1/N\}$$

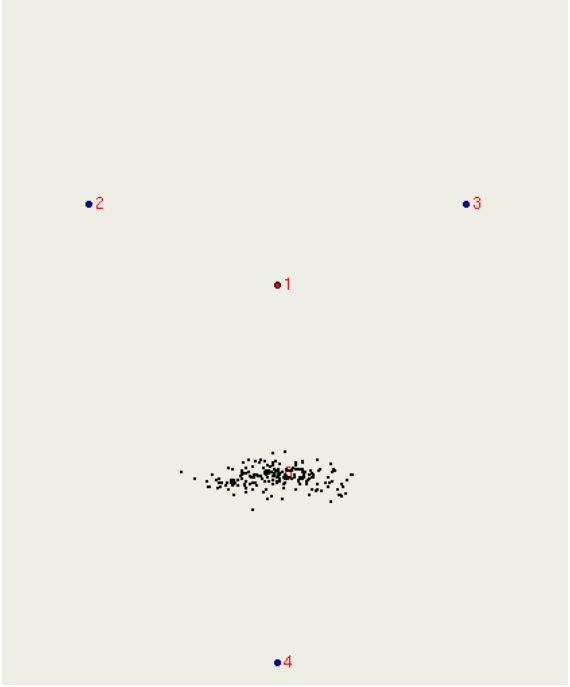
Update: For each sample, $(\tilde{s}_t^{(i)}, w_t^{(i)})$, set the importance weight to the measurement probability of the actual measurement, $\tilde{w}_t^{(i)} = p(d_t|\tilde{s}_t^{(i)})$. Normalize the weights such that $\sum_i \eta\cdot\tilde{w}_t^{(i)} = 1.0$, and draw N random samples for the sample set $\{(\tilde{s}_t^{(i)}, \eta\cdot w_t^{(i)})|i\in[1,N]\}$ according to the normalized weight distribution. Set the weights of the new samples to $1/N$, resulting in a new set of samples $\{(s_t^{(i)}, w_t^{(i)})|i\in[1,N], w_t^{(i)} = 1/N\}$ corresponding to the posterior distribution $p(s_t|d_1,\ldots,d_t)$.



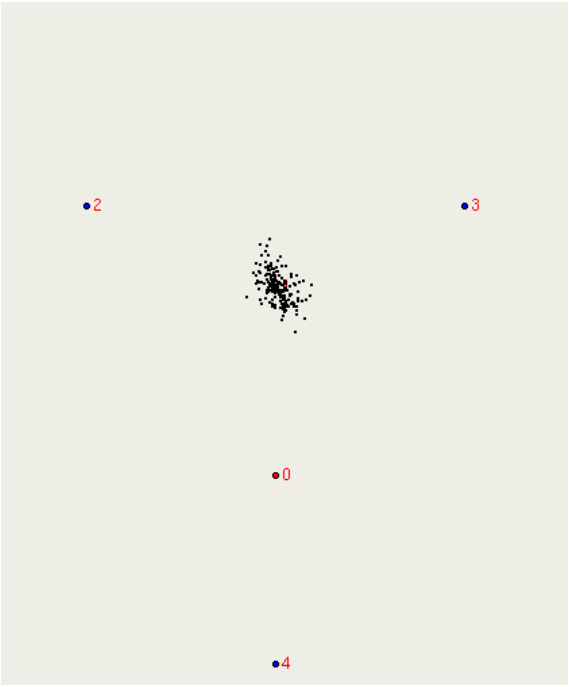
(a) Particle distribution of node 0 when node 1 is *not* presented.



(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.

Figure 1: Location Distribution in Simple Scenarios.

3.2 Modified Particle Filtering for Location Estimations

The classical Monte Carlo method is often implemented using particle filters. To apply the filter to the location tracking problem a system model and a measurement model must be provided. We use a simple random placement model as our system model (please note that this is the mobility model used in the filter which is different from the mobility model used in the simulations to enable node movement). The model assumes that at any point in time the node moves with a random velocity drawn from a Normal distribution with a mean of $0m/s$ and a fixed standard deviation σ. 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 while such system model should work well in stationary networks, it's not best suited for mobile networks. In reality, mobile nodes follow a certain kind of movement profile instead of random motion. The system model should closely resemble the current movement profile of the node. However, since it's difficult to obtain a reliable movement profile when the location is unknown, the assumption of random movement is probably the best we can do at this stage.

The measurement data are obtained by observing the periodical location data broadcast from neighbors. To minimize the impact of the measurement error, we apply a simple Kalman filter to the RSSI sensor readings [6] before feeding the measurement data to the particle filter. When a node u receives broadcast location data from node v, the broadcast data consist of the unique identifier of v, and the probability distribution, X_v, of the location estimate of v at time t. The X_v distribution is a compressed version of the actual particle distribution at v. The detail method of compressing and decompressing the particle distribution is the topic of the next section. For now, let us assume that X_v contains a set of sample particles that represents v's location. Along with the RSSI reading of the broadcast, $RSSI_v$, the complete measurement metrics d_t is therefore $(id, X_v, RSSI_v)$.

After the measurement from the neighbor v is collected, the particle filter at node u is updated. In the classic particle filtering, particles are re-sampled based on weights, which are in turn assigned based on the measurement. More weights are assigned to the particle values that are more consistent with the measurement reading. After re-sampling, the particle distribution becomes more consistent with the current measurement. In our situation we have a unique scenario where the measurement itself consists of a particle distribution, X_v. Furthermore, both X_u and X_v are *imprecise*. Our task during the update step is to modify the particle distribution X_u so that it becomes more consistent with $RSSI_v$ while taking into account the inherent impreciseness of X_u and X_v. First, we obtain a distance estimate from the inverse of the signal propagation model P:

$$D^{(RSSI)} = P'(RSSI_v)$$

Note that P can be arbitrary as long as it depends on the distance from the sender to the receiver. Noise can be added to the model, but we disregard it when calculating the inverse and let it be filtered out by the particle filtering (note, that in the simulations noise is indeed added to the RSSI measurements).

For each particle x_u in X_u, we randomly select a particle x_v in 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 re-sampling 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 point 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, i.e., $D^{(RSSI)} = |A'B|$. 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, the new location for x_u should stay at A. Therefore, we select the new location based on the perceived accuracy, i.e., the variances, of the distribution X_u and X_v. Let the variance of a distribution X be $var(X)$. We select the new location of x_u, x'_u, along the line $|AA'|$ such that

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

A new particle is then randomly re-sampled by a Normal distribution centered at x'_u with the variance being the average of the variances of X_u and X_v. We consider the variances of both X_u and X_v during re-sampling because the spread of both distributions affects the spread of the updated distribution X'_u.

Comparing to the re-sampling method of classic particle filters, our method is different in that we do not use a weight based re-sampling method. Instead, we re-sample by comparing the two distributions together against the measurement reading. But, the concept is the same as we are updating the distribution to fit the measurement readings. Our re-sampling

method has a number of advantages over the traditional method. First, our method does not re-sample directly from the original particle location using a weight based Gaussian distribution. Instead, it re-samples from a more accurate location influenced by neighbor's distribution. Thus, our method requires less amount of random probing and converges more quickly. Secondly, since our method requires less amount of random probing, a significantly smaller number of particles are required. With less particles, the particle filter update procedure computes more efficiently.

3.3 Compressing and Decompressing Particle Filter Distribution

The previous section makes the assumption that the complete location distribution is received from the neighbor. Since the complete distribution consists of a large number of particles with their location data, doing so is obviously not very practical due to the limited bandwidth of ad hoc networks. Therefore, we propose a simple yet effective compressing mechanism that allows the particle distribution to be transmitted in a compact form.

Given a particle distribution X, we locate the expected value, $\hat{x}$, as the particle in the distribution that has the minimum overall distance between itself and other particles, i.e., $\hat{x} = \arg\min_{x \in X} (\sum_{y \in Y} |x - y|)$. 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 the predefined range r. We then calculate the variance, σ^2 within those n particles. Thus, we obtain a quadruple $(\hat{x}, r, n, \sigma^2)$. 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 sequences of quadruples that approximates the original particle distribution. When the quadruples are received by the receivers, a decompressing algorithm runs to reproduce the distribution by randomly generating particles based on the expected value, range, particle number and variance for each quadruple.

For each broadcast, a fixed number of aforementioned quadruples are transmitted. The following algorithm is used to progressively increase the range r for each quadruple.

Q := number of quadruples desired

R := max range that covers the entire area

$minQuota := |X|/Q$

$rIncrement := X^{2/3}/R$

$xCount := 0$

$r := 0$

$curRange := 0$

$q' := 1$

FOR q = 1 to Q

    $maxRange := q \cdot rIncrement^{3/2}$

    WHILE $curRange < maxRange$ AND

    number of particles in $curRange + xCount < minQuota \cdot q$ DO

        $curRange := q' \cdot rIncrement^{3/2}$

        $q' := q' + 1$

    $r_q := curRange$

    $xCount$:= $xCount$ + number of particles in $curRange$

The algorithm starts with an initial range of $X/R^{3/2}$ 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 the total number of particles covered thus far. At

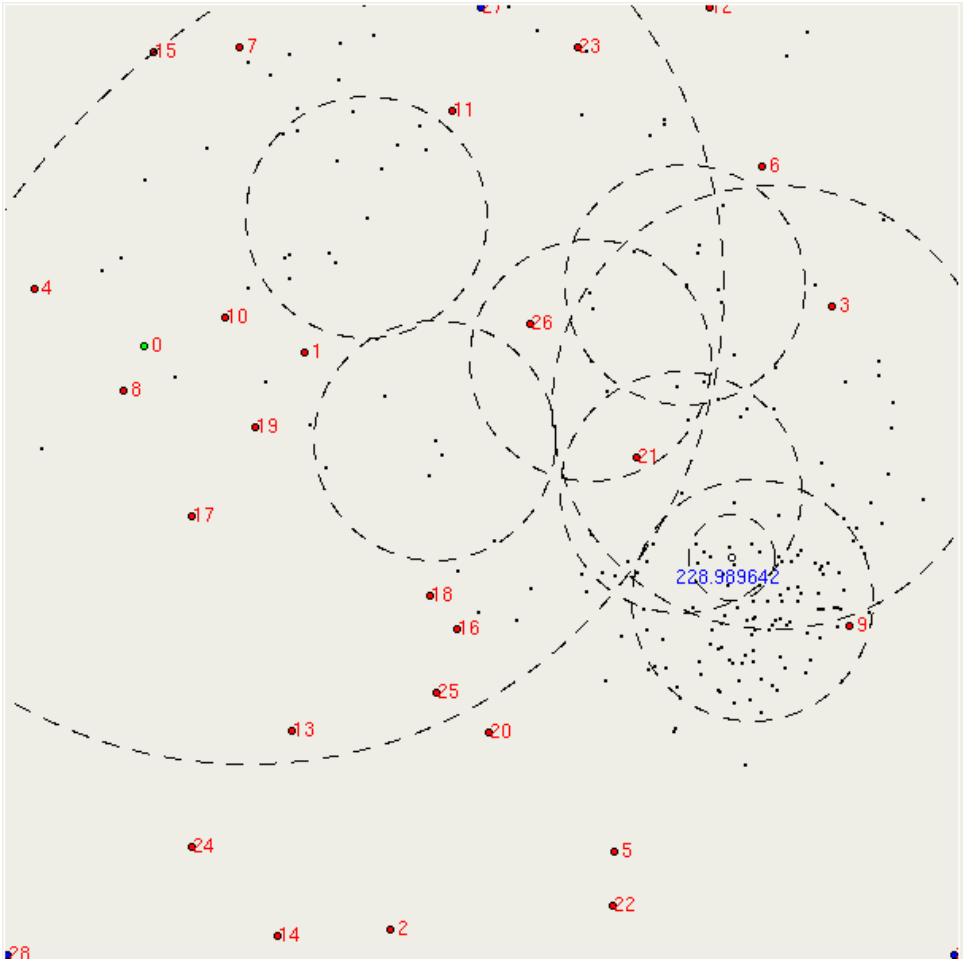


Figure 2: Compressing the particle filter distribution.

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 quadruple, and the overall trends of the original distribution are maintained. Meanwhile, by using a quota limit with the exponential range increment, more heavily populated areas are preserved with finer detail. Our experiment has shown that the compression method reduces the amount of data exchange by nearly 90 percent without a significant increase to the location estimation error

Fig. 2 shows the compressed distribution, where the circles shows the ranges.

4 Simulation Results

We have conducted a number of experiments to validate the effectiveness of our particle filter based solution. Our experiments attempt to duplicate real world scenarios as closely as possible. In our simulations we assume a network in which all nodes have an identical transmission power, with a certain percentage of nodes (simulation parameter) being GPS nodes. For a network of fixed size, the connectivity of the network depends (almost solely) on the transmission range. When a node is located within the transmission range of another node, we assume that it is capable of receiving signal from the sender when noise is not present. The received signal strength depends on the distance to the sender as well as a signal propagation

model and a noise model.

The signal propagation model is given by $P = c \cdot d^{-2}$, in which the power of the received signal P is inversely proportional to the second power of the distance d. Here, c is an arbitrary constant. When the received signal power P is below a threshold P_{min}, it is considered too weak to be captured by the receiver thus the link breaks. For our simulations, we select $c = 10^6$ and $P_{min} = 1$. Note the c and P_{min} selection does not affect the overall simulation results, as long as the same values are used in the observation model of the filters. In fact, the same can be said about all other signal propagation models - all we require is a model that represents the receiving power as a function of distance, and we let the filter to filter out the noise. For the particle filter itself, we use a total number of 200 particles at each node.

We use two types of networks, isotropic and anisotropic, of 100 randomly placed nodes. With isotropic networks, nodes are randomly placed into a square with an average degree of 7.6. With anisotropic networks, nodes are placed into a C shape area with an average degree of 7. Noise is added to the signal strength calculated via the signal propagation model as a percentage of the calculated signal strength. For instance, a 10 percent noise means that the received signal strength may vary within a plus-minus 10 percent range of the calculated signal strength (uniformly distributed). Note that our network configuration and noise model is identical to that of the isotropic topology in [16], so that we can effectively compare our method with APS.

We start by running the simulation on stationary networks, which resembles sensor network in the real world.

4.1 Filter Convergence

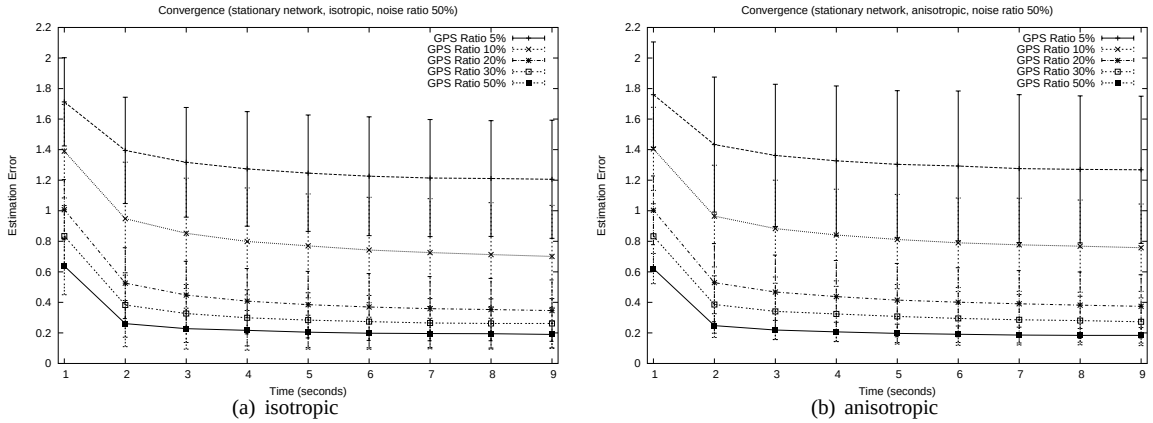
Figure 3 shows how the estimation error converges as more measurement readings are processed in a static network. We are interested in how long and how many messages it takes for the error to reach an acceptable level from which it only reduces marginally. We added a noise level of 50 percent to the measurement readings. The estimation error is calculated as the difference between the most likely value given by the particle distribution and the actual location. The difference is then measured in term of the ratio against the maximum transmission range. Thus, an estimation error of 1.0 means that difference between the expected value and actual location equals to the maximum transmission range. The data is collected of enough simulation runs to claim a 95 percent confidence, which shows as the vertical scale at each data point; the error ratio is the average of all non-GPS nodes (i.e., the perfect "estimates" of GPS nodes are not biasing the results).

Two obvious facts can be observed from Figure 3: i) networks with higher GPS ratio produce better estimations and ii) estimation error reduces quicker with higher GPS ratio. Both of those observations can be explained by the fact that GPS ratio determines how fast and how accurate location information can be propagated through the network. With a higher GPS ratio, non-GPS nodes will be able to obtain the necessary location information faster because non-GPS nodes are closer (i.e., less number of hops) to GPS nodes. Also, since measurement error is aggregated at each hop, the location information will be more accurate with higher GPS ratio. GPS ratio also affects the confidence interval. This is because when GPS ratio is low, the estimation error depends greatly on the position of the GPS nodes. When their position does not spread out evenly through the network (for instance, GPS nodes is concentrated around one edge of the network), it becomes more difficult for the the nodes further away to obtain good estimates. As the GPS ratio increase, the chances of bad positions reduces, and thus the variance of the estimation error reduces.

Figure 3 also shows that the estimation error converges to the minimum between 2 to 5 seconds depending on the GPS ratio. Considering that the location broadcast occurs every 0.5 seconds, it takes about 4 to 10 rounds of broadcasts for the error to reach the minimum. Since the average degree of the network is 7.5 with a total of 100 nodes, each round of broadcast is equivalent to 750 messages. Therefore, it takes about 3000 to 7500 messages to minimize the error depending on the GPS ratio. Note that when even in the worst case where the GPS ratio is low, error converges very quickly and is close the minimum after 2 seconds. The results are at least as good as those of APS, where the "DV-distance" method uses 6500 messages (when GPS ratio is 0.1) to 9000 messages (when GPS ratio is 0.9), and the "Euclidean" method takes from 3000 to 8500 messages (results taken from Figures 7 and 11 of [16]). Note that our method converges quicker and takes less number of messages when the GPS ratio is high.

4.2 Minimum Estimation Error

Fig. 4 compares the estimation error of our particle filter method with other methods. Again, the simulation scenario is duplicated from that of the isotropic topology in APS [16], with a rather dense network of degree averages at 7.6. One



(a) isotropic (b) anisotropic

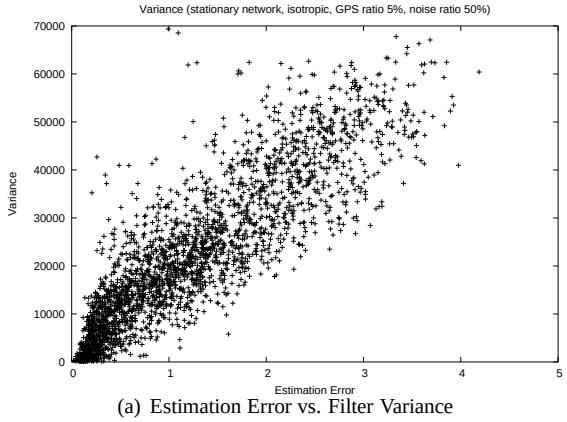
Figure 3: Filter Convergence.

advantage of our method is that our method 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. Fig. 4(a) plots the actual filter variances against the estimates for the scenario where GPS ratio is to 5 percent. Here, a trend can be observed that the variances increase linearly with the estimates. Fig. 4(b) shows relationship between the coverage and estimation error when the GPS ratio varies. Other methods such as DV-Hop, DV-Distance and Euclidean generate estimates with fixed coverage, and thus they are plotted as single points in the graph. Note that when GPS Ratio is 10 percent, our method generates similar estimation error as Euclidean when adjusted to the same coverage, but DV-Hop and DV-Distance give better estimation with the same coverage.

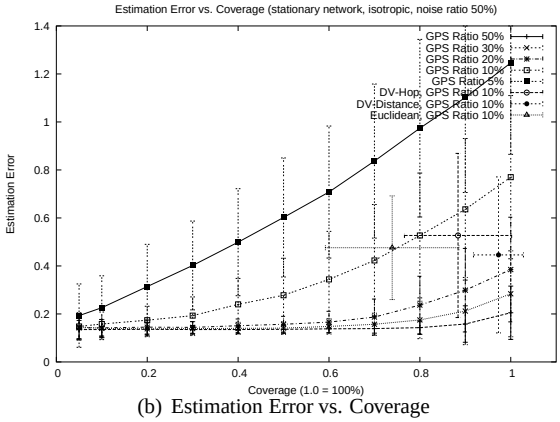
Fig. 4(c) through Fig. 4(e) shows the result of a more detailed comparison against DV-Hop, DV-Distance and Euclidean. We compare the estimation error of our method with other methods by obtaining the average estimation error when its corresponding coverage matches the other method. The figures show that DV-Hop and DV-Distance give lower estimation error when the GPS ratio is less than 20 percent. With a higher GPS ratio, our method gives better result. Similar result can be observed when comparing to Euclidean, but the cut-off point here is around 10 percent GPS ratio. The higher error at low GPS ratio can be explained by the fact that the particle filter method prefers the scenarios where GPS nodes are located around all edges of the network forming a near convex hull, in which case the location information from various GPS nodes can utilized more effectively. When GPS ratio is low, such ideal scenarios are less likely to occur, and there will be nodes outside the convex hull that are more difficult to localize. DV based methods, however, are affected less by those scenarios, since they uses globally collected data such as distance-per-hop to perform the triangulation.

4.3 Connectivity

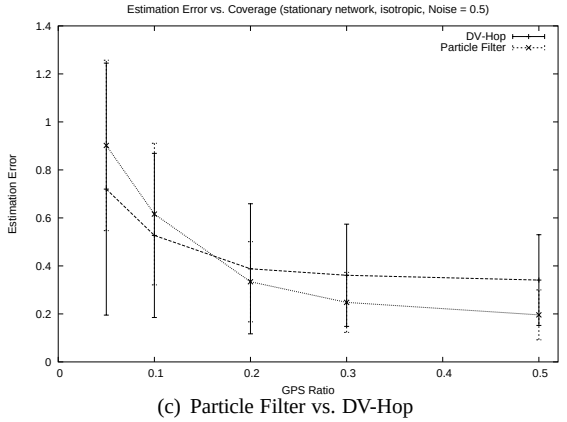
Simulation results in previous work are based on a rather dense network with an average degree of 7.67. Similar networks were used in [16], and thus allow a more sensible comparison. Fig. 5 shows the estimation error of our particle filter based localization method in more sparse networks. Here, we vary the network connectivity by changing the transmission range while maintaining the network size (100 nodes). Fig. 5(a) shows that as expected more error is introduced in sparser networks. Roughly speaking, the error halves when the network connectivity doubles. Fig. 5(b) through Fig. 5(d) shows the result of direct comparison the estimation error between our method and others under the same coverage. When the network is sparser, our method clearly out-performs all other three methods. In theory a node needs to receive signal readings from a minimum of three neighbors in order to pinpoint its location. Thus, a network with degree of at least three will be needed to localize all its nodes. With our localization method, respectable estimations are obtained even with very sparse networks of degrees less than three (no other approaches are able to derive estimates for such situations).



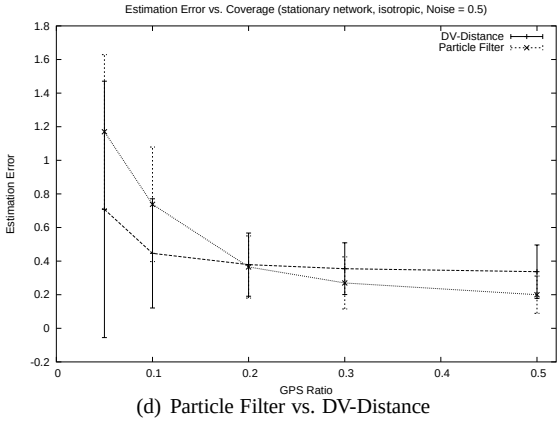
(a) Estimation Error vs. Filter Variance



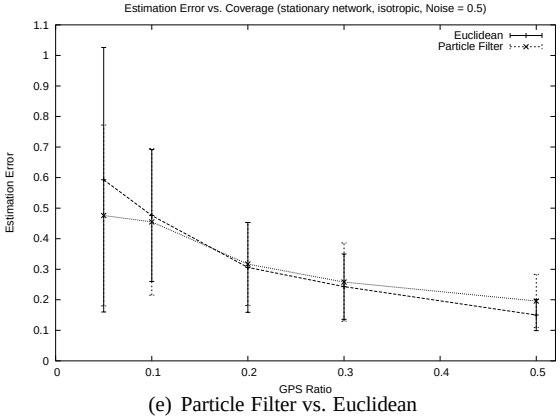
(b) Estimation Error vs. Coverage



(c) Particle Filter vs. DV-Hop

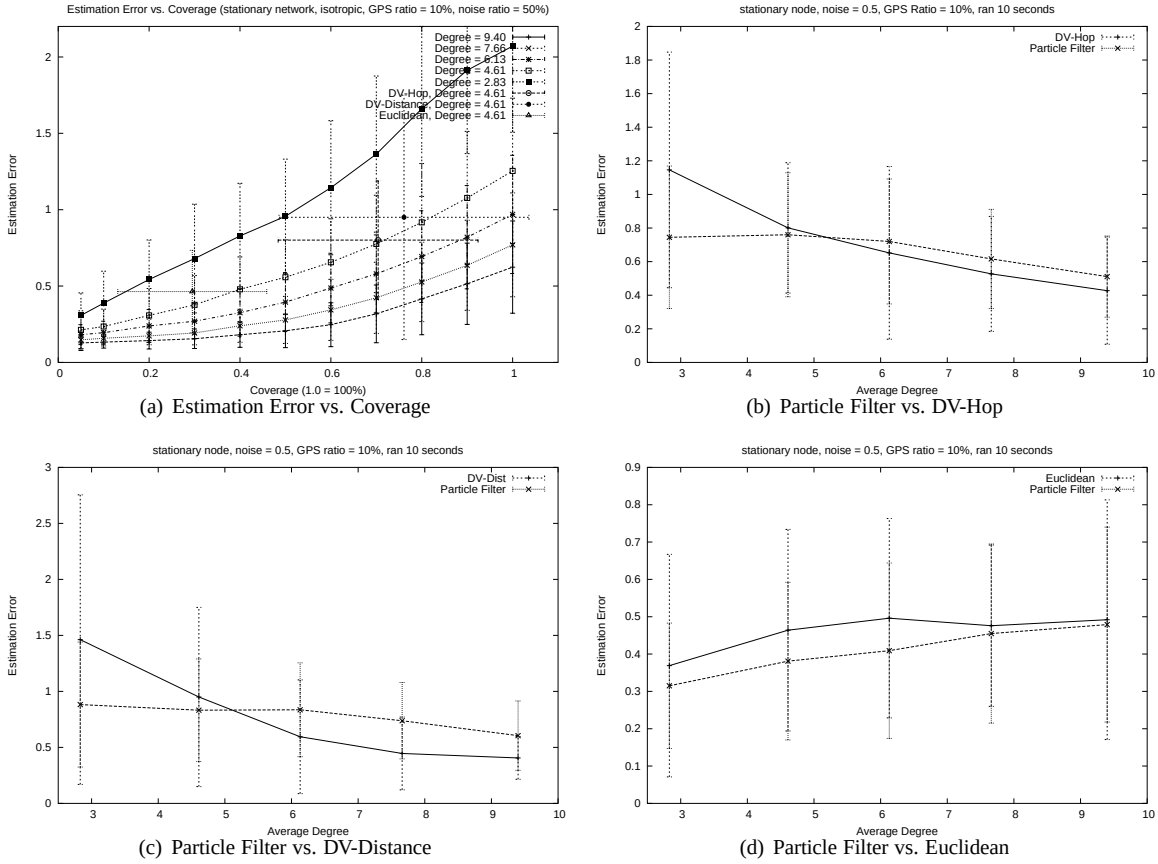


(d) Particle Filter vs. DV-Distance



(e) Particle Filter vs. Euclidean

Figure 4: Effect of GPS Ratio on Estimation Error.



(a) Estimation Error vs. Coverage

(b) Particle Filter vs. DV-Hop

(c) Particle Filter vs. DV-Distance

(d) Particle Filter vs. Euclidean

Figure 5: Effect of Connectivity on Estimation Error.

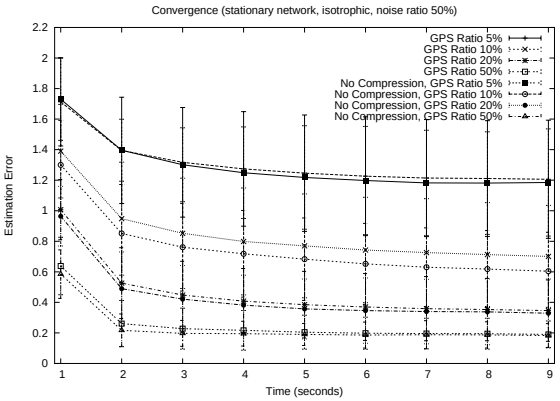


Figure 6: Effect of Compression on Filter Convergence.

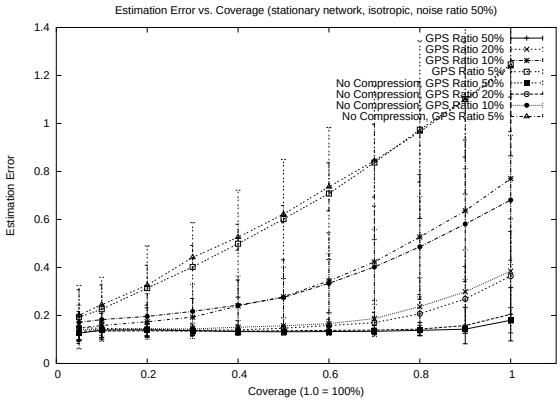


Figure 7: Effect of Compression on Estimation Error.

4.4 Compression vs. No Compression

Fig. 6 and Fig. 7 demonstrate the effectiveness of our compression algorithm on transmitting the particle distribution. The same scenario is repeated when a complete particle distribution is transmitted instead of the compressed version. The results are compared side by side. While the localization algorithm works better when the complete distribution is sent, the differences are rather minimal. While the original particle distribution consists of 200 particles, each of which contains two decimal numbers to designate the location, the compressed version consists of 10 quadruples, each of which contains five decimal numbers. The compression method archives a total bandwidth saving of 87.5% at each location exchange. Given that the network bandwidth can be expensive, one can easily justifies the minimal tradeoff of performance using our compression scheme.

4.5 Results on Mobile Networks

Previous work on MANET localization generally do not contain extensive simulation and analysis when the network is indeed mobile (as the definition of MANETs imply). As discussed earlier, many 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. In the worst case, the entire localization scheme has to be rerun. Our method, however, are specifically designed to work in mobile networks.

This section discusses simulation results on running our the particle filter localization method on mobile networks. Again, we use a network with a population of 100 nodes and average degree of 7.5. We use the epoch-based mobility model of [15] to simulate node movement, which is widely accepted as a good mobility model for ad hoc networks - more realistic than, e.g., simple Brownian motion models. The entire movement path of the node is defined by a sequence of "epochs," i.e., $(e_1, e_2, \cdots, e_n)$. The duration of each epoch is I.I.D. exponentially distributed with a mean of $1/\lambda$. Within 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 2π. The absolute value of the velocity is I.I.D. normal with a mean μ of and a variance of σ^2. Our simulation uses a fixed mean and variance such that $\mu = \sigma$. The result is obtained by varying μ and σ from 0m/s to 40m/s. The expected amount of time a node maintains its current velocity is set to 5 seconds, i.e., $\lambda = 5$.

Figure 8 shows the filter convergence on mobile networks with measurement noise level set to 50%. Here, all nodes in the network moves at 10m/s in average, i.e., $\mu = \sigma = 10$. Comparing to the results of stationary networks in Figure 3, the random movement of the nodes causes the estimation error to swing. However, the error variances are not very high once the nodes determine their initial locations after the first couple of seconds. This indicates that the filter is able to adapt to the node movement well enough to maintain its overall estimation accuracy. Figure 9 shows the average estimation error of mobile networks. The error does increases gracefully as the speed increases. Considering that neighbors exchange

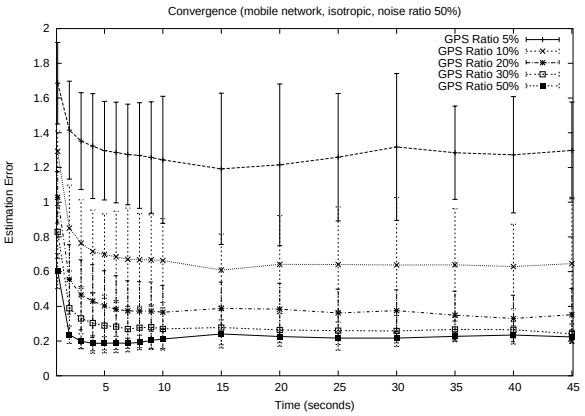


Figure 8: Filter Convergence with Mobile Networks.

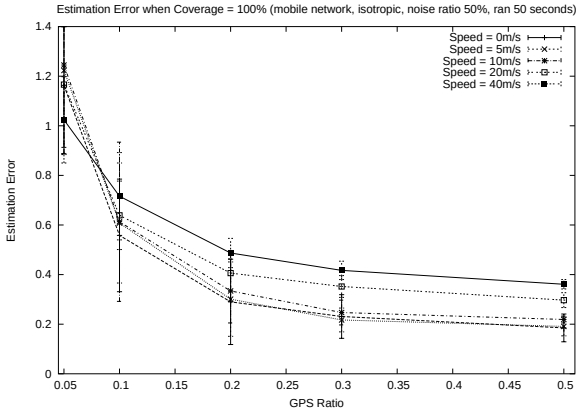


Figure 9: Estimation Error of Varying GPS Ratio and Network Mobility.

location information every 0.5s, in a network with an average nodal speed of 40m/s nodes move an average of 20 meters per observation; yet our method is capable of producing usable location estimates. From the above result, one can foresee that it is possible to reduce the rate of localization exchanges between the neighbors after the initial localization completes, while still maintaining reasonable good estimates. A challenge of future works is to find out the ideal rate of exchanges for a given amount of network mobility.

5 Conclusions

This paper described a novel solution to the location tracking problem for mobile ad hoc networks that uses a Monte Carlo sampling-based Bayesian filtering (i.e., particle filtering) method. The estimated location for nodes is regarded as a probability distribution represented by a collection of sample points. The location information from the GPS nodes is propagated through the network via local broadcasting of the location estimates. When a node receives the location estimates from neighbors, it updates its location distribution using the particle filtering method. Simulation study has shown that the particle filter solution is capable of producing good estimates equal or better than the existing localization methods such as APS-Euclidean. Our solution also performs quite well when the network connectivity is low. Study has also shown that the solution is resilient to network topology change, making it suitable for ad hoc networks with significant mobility.

Our particle filter based localization method currently uses RSSI as the sole measurement. However, because our method is based on a rather generic algorithm of probabilistic filters, it can be easily extended to incorporate other measurement types such as angle of arrival (AoA). To do so, only the filter update step needs to be changed in order to meaningfully update the filter according to the properties of the new measurement, but the basic algorithm remains the same. In fact, it is easy to implement our method with multiple types of measurements coexisting in the network. The same particle filter method can be used in a network where an arbitrary portion of nodes are capable of measuring RSSI, another part of the nodes are capable of measuring AoA, and some are capable of measuring both. This makes our method truly versatile and ideal for such heterogeneous networks.

References

[1] P. Bahland and V.N. Pamanabhan, RADAR: An in-Building RF-Based user Location and Tracking System, In Proceedings of the IEEE INFOCOM'00, March 2000.

[2] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," IEEE Personal Communications Magazine, vol. 7, no. 5, pp. 28–24, October 2000.

[3] L. Girod and D. Estrin, Robust Range Estimation using Acoustic and Multimodal Sensing, In Proceedings of IROS 01, Maui, Hawaii, October 2001.

[4] N. Gordon, Bayesian Methods for Tracking, PhD thesis, University of London, 1993.

[5] T. He, C. Huang, B. M. Blum,J. A. Stankovic,and T. F. Abdelzaher. "Range-Free Localization Schemes in Large Scale Sensor Networks", CS-TR2003 -06. Submit to MobiCom 2003.

[6] R. Huang, G.V. Záruba, and M. Huber, Link Longevity Kalman-Estimator for Ad Hoc Networks, To appear in the Proceedings of the VTC2003, IEEE 54th Vehicular Technology Conference, 2003.

[7] X. Jiang and T. Camp. Review of geocasting protocols for a mobile ad hoc network. In Proceedings of the Grace Hopper Celebration (GHC), 2002.

[8] Y. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) Mobile Ad Hoc Networks," MOBICOM '98, Dallas, TX, 1998.

[9] Y. Ko and N. Vaidya. Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms. In IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 1999.

[10] F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger: Geometric ad-hoc routing: of theory and practice. PODC 2003: 63-72.

[11] W.-H. Liao, Y.-C. Tseng, and J.-P. Sheu. GRID: A fully location-aware routing protocol for mobile ad hoc networks. Telecommunication Systems, 18(1):37–60, 2001.

[12] 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. Journal of Internet Technology, 1(2):23–32, 2000.

[13] T. Liu, P. Bahl, and I. Chlamtac, "A hierarchical position-prediction algorithm for efficient management of resources in cellular networks," *Proceedings of the IEEE GLOBECOM'97*, Phoenix, Arizona, November 1997.

[14] M. Mauve, H. Fuler, J. Widmer, and T. Lang. Position-Based Multicast Routing for Mobile Ad-Hoc Networks. Technical Report TR-03-004, Department of Computer Science, University of Mannheim, 2003.

[15] A. B. McDonald and T. Znati, "A mobility-based framework for adaptive clustering in wireless ad-hoc networks," *IEEE Journal on Selected Areas in Communication* Special Issue on Wireless Ad-Hoc Networks, vol. 17, no. 8, August 1999.

[16] D. Niculescu and B. Nath, "Ad hoc positioning system (APS),"*Proceedings of the IEEE GLOBECOM'01*, San Antonio, 2001.

[17] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AoA," *Proceedings of the IEEE INFOCOM*, San Francisco, 2003.

[18] D. Niculescu and B. nath, DV Based Positioning in Adhoc Networks, to appear in journal of Telecommunication Systems, 2003.

[19] P.J. Nordlund, F. Gunnarsson, and F. Gustafsson, "Particle filters for positioning in wireless networks," *Proceedings of the XI. European Signal Processing Conference (EUSIPCO)*, 2001.

[20] C. Savarese, J. Rabay and K. Langendoen, Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks, USENIX Technical Annual Conference, Monterey, CA, June 2002.

[21] A. Savvides, H. Park and M. B. Srivastava, The n-Hop Multilateration Primitive for Node Localization Problems. Mobile Networks and Applications 8(4): 443-451, August 2003.

[22] K. Whitehouse and D. Culler, Calibration as Parameter Estimation in Sensor Networks, In First ACM International Workshop on Wireless Sensor Networks and Application, Atlanta GA, September 2002.

[23] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for adhoc routing. In Proc. seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pages 70–84, 2001.

[24] G. V. Záruba, M. Huber, and F. A. Karmangar, Monte Carlo Sampling Based In-Home Location Tracking with Minimal RS Infrastructure Requirements, to be published, 2003.