A New Approach towards Solving the Location Discovery Problem in Wireless Sensor Networks

(Technical Report CS-TR-4551 and UMIACS-TR-2003-119)

Guang Han, Shaoxiong Hua and Gang Qu

Department of Electrical and Computer Engineering
University of Maryland, College Park, MD, 20742
{hanguang, shua, gangqu}@glue.umd.edu

December 12, 2003

ABSTRACT

Location discovery in wireless sensor network (WSN) is the process that sensor nodes collaborate to determine the position for unknown sensor nodes. Anchors, sensors that know their locations, are expensive but are required to be deployed into the WSN to solve this problem. Thus it is desirable to minimize the number of anchors for this purpose. In this paper, we propose an anchor deployment scheme and a novel bilateration locationing algorithm to achieve this goal. The basic idea of anchor deployment method is to have three anchors deployed as a group, and locate sensors around them expansively. The novelty of our bilateration algorithm is that it in general requires only two neighbor sensors to determine a node's location. Comparing with the state-of-the-art location discovery approaches, our algorithm gives location estimation with high accuracy, low communication cost and very small anchor percentage. We conduct theoretical analysis about location estimation error and extensive simulation shows that our algorithm can derive sensor location within 4% location error and much less communication cost compared with other algorithms.

Categories and Subject Descriptors

C.2.3 [**Computer Systems Organization**]: Network Operations

General Terms

Algorithms, Performance

Keywords

Weighted multilateration, Bilateration primitive, Location estimation, Wireless Sensor Networks

1. INTRODUCTION

Wireless sensor networks (WSNs) have found a lot of military and civil applications recently. Some of the reasons that WSN gains its popularity include 1) the low cost of deployment mainly because of MEMS technology, 2) relatively long network lifetime due to low power and battery technology advances, 3) the more and more computation power the sensor nodes have. Basically, each sensor will monitor its local environment and they collaborate as a whole to provide information about the sensor field. Knowing its own physical location with great accuracy and precision is vital for the sensing unit in a WSN to provide its environmental information. It is possible to acquire a sensor node's location from the rather reliable Global Positioning System (GPS). Such sensor nodes that know their locations will be referred to as **anchor nodes** or **beacons**. However, GPS requires expensive infrastructure and costs about $100/receiver. Therefore, many location discovery systems have been proposed recently to compute the sensors' positions from limited number of anchors [1, 7, 12, 17, 19], or even without anchors [21]. A good locationing algorithm should provide high accuracy on location estimates and high scalability with low energy and communication cost. In this paper, we propose a locationing algorithm that achieves these goals based on the bilateration primitive and a deliberate anchor deployment scheme.

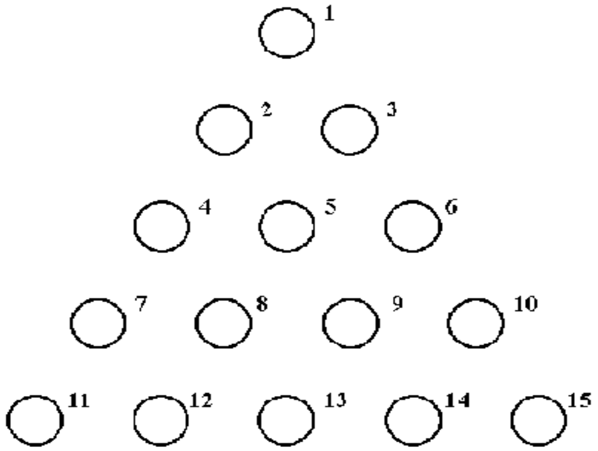

Fig. 1. Node distribution

A Motivational Example

We illustrate the basic idea of our approach by the following example. Consider the 15 sensor nodes as shown in Figure 1, the distance between any two neighboring nodes is equal to the communication range of the sensor node. Thus each node can only communicate with its neighbors; for example, node 1 can talk to only nodes 2 and 3, while nodes 2,3,4,6,8,9 are all in the communication range of node 5. Suppose that we have three anchors, the problem is where should we deploy these three anchors such that we can locate the other sensor nodes accurately and efficiently.

Current approaches such as those in [8] and [14] place the three anchors along the perimeter of the area in order to reduce location estimation error. In this scenario, nodes 1, 11 and 15 will be chosen as anchors. Both DV-hop and DV-distance, two state-of-the-art locationing algorithms [11], will give the same location estimation fro the rest of the sensor nodes. If the standard deviation of the range error is 5% (normalized to the communication range). Based on these two approaches, the average location error of these sensor nodes will be about 35% (normalized to the communication range).

However, if we place the three anchors at nodes 5, 8 and 9, node 6 will be able to locate itself. This is because that although node 6 has only two anchors 5 and 9 within its communication range, its mirror image (position where node 8 locates) can be eliminated by using anchor 8 as a reference: node 6 cannot talk to node 8, but its mirror image can. Similarly we can locate nodes 4 and 13. Once these three nodes are discovered, they behave as anchors and we can iteratively determine the location of other nodes by such *bilateration calculation*. In this approach, the average location error is about 5% for nodes 4,6 and 13; 10% for nodes 2,3,7,10,12 and 14; and 15% for nodes 1, 11, and 15. The overall location error will be about 10%, much more accurate than the 35% by the traditional approaches.

This example illustrates our approach: deploying three anchors close to each other and locate other sensor nodes around them expansively. We expect small location errors because distance measurement is performed only between direct (one-hop) neighbors. The proposed algorithm will also have a low communication and energy cost because there is no need to flood anchor's position to the entire network or to refine location. Furthermore, less anchors will be used to discover a given sensor field because three anchors are put together as a group and the sensors are located expansively. These expectations are validated by theoretical location estimation error analysis and extensive simulations.

Paper Organization

The rest of the paper is organized as follows. In Section 2, we survey the existing location discovery algorithms. We then formulate the problem in Section 3. We elaborate the anchor deployment scheme and the bilateration primitive in Section 4. In Section 5, we analyze the performance of our proposed approach in terms of communication cost, computation cost, coverage, and location accuracy. The simulation results are presented in Section 6 before we conclude in Section 7.

2. RELATED WORK

Many algorithms are proposed to solve the sensor localization problem in the past several years. We can group these algorithms according to different criteria: infrastructure-based or infrastructure free (ad-hoc); centralized or distributed; range-based or range-free; iterative or non-iterative; etc. The algorithms in the same group have similar problems, so we can actually analyze the common characteristic of these groups in order to have a global picture of these locationing algorithms.

Infrastructure-based algorithms [2, 12] rely on an external infrastructure to locate sensors. This kind of approach may produce good results, but it is not a favorable method for ad-hoc wireless sensor networks.

Centralized algorithms [4,18] can produce high accuracy results, but it requires significant computation and communication. Since sensor nodes are often limited in power and computational capacities, this is not a good choice either.

The recent research work [6] by He et al. divides the locationing algorithms into two categories: range-free and range-based. They propose APIT (All Point-In-Triangulation Test), which calculates the center of gravity of the intersection of all of the triangles in which a sensor resides to determine its location. However, their algorithm requires large anchor range that is 10 times the normal sensor range. This assumption may increase the installation cost of sensor networks. Another range-free algorithm [18] by Shang et al. only needs three anchors to locate all the sensor nodes in the network. But this approach is intrinsically a centralized approach with significant computation ($O(n^3)$) and communication cost. In general, although range-free approach [2, 6, 18] does not rely on range estimation, it can only produce coarse-grained results that may not be used in applications with stringent accuracy requirements.

Langendoen and Reijers present a quantitative comparison between three range-based locationing schemes [8]. This detailed comparison gives us some basic idea on the common characteristic of the current locationing algorithms. As stated in the paper, three phases are included in the algorithm: anchor distance estimation; initial location estimation and location refinement. However, a general drawback of these algorithms is big anchor distance estimation error introduced by the first phase of these algorithms. When distance information is propagated over several hops, the final distance estimation error becomes significant, especially for sparse and irregular networks. This large distance estimation error leads to large location error of initial location estimation, which in turn results in a heavy burden for refinement phase. As mentioned in [8], most communication cost is consumed in the refinement phase and system coverage is also decreased since some location estimations cannot be improved to be acceptable.

Iterative multilateration algorithm [16] by Savvides et al., localized location discovery algorithm [9] by Meguerdichian et al., and position dissemination algorithm by Albowicz [20] et al. describe the use of iterative algorithm to locate sensor nodes in wireless sensor networks. The appealing feature of iterative algorithm is free of anchor distance estimation and location refinement. However, it requires high connectivity or high percentage of anchors that significantly increase the installation cost. In addition, error accumulation should also be considered due to the use of unknown nodes as anchors.

The expansive locationing algorithm proposed in this paper is intrinsically a range-based iterative algorithm. Compared with other iterative algorithms, we can achieve similar location accuracy without high connectivity or high anchor percentage. The detailed description of the algorithm will be given in Section 4.

3. PROBLEM FORMULATION

Consider a sensor field ***S*** with unknown number of sensors and no sensor knows its location. However, sensors within the communication range ***R*** can talk to each other and therefore measure the distance between them. We assume that such measured distance carries an error that is modeled as an independent Gaussian random variable with zero mean and variance σ^2. That is, if node i and node j are located at positions (x_i,y_i) and (x_j,y_j) respectively, then their measured distance satisfies the following

$$\hat{d}_{ij} = d_{ij} + e_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + e_{ij} \quad (1)$$

where d_{ij} is the real distance between nodes i and j in the two-dimensional field ***S*** and e_{ij} is the estimation error.

We now deploy ***M*** anchor nodes into the same field ***S*** at locations (X_i,Y_i) for $i=1,2,\ldots,M$. Because the anchor nodes know their physical locations, their neighbors (the nodes within the the communication range ***R*** from anchor nodes) can obtain location information from them and then locally estimate their own locations. We consider the following problem: *for a given sensor field **S**, how to use the minimal number of anchors to determine the locations of the sensors in field **S** precisely and efficiently?*

The solution to this problem contains two parts. First, it has to specify the anchor deployment scheme; second, once all the anchors are deployed, we have to develop an estimation algorithm. The accuracy is measured by the location error. The efficiency is measured by the energy and communication cost.

4. ALOGORITHM DESCRIPTIONS

The algorithm flow diagram is shown in Fig.2 below. The input of the algorithm is the locations of the anchors. Then a sensor has three or more "neighboring anchors" will be located by using weighted multilateration method, if both "angle test" (used to avoid large location error) and "reference test" (used to avoid wrong location solution) are passed. And the located sensor will become an anchor. Otherwise, a sensor within the communication range of only two anchors will be located through bilateration primitive, if both "angle test" and "reference test" are passed. And the located sensor will become an anchor. Then we will locate the next sensor.

The main contribution of this paper is to put three anchors as a group and to locate unknown sensors expansively. Quite different from other algorithms that put anchors separately in the field, we suggest putting three anchors together and locating unknown sensors around them iteratively. When the unknown sensors around these three initial anchors are located, they become anchors and they help to locate the sensors around themselves, which could be two hops away from the initial anchors. Such a process continues, and more and more sensors will be located. The deployment of the initial three anchors will be discussed in section 4.1.

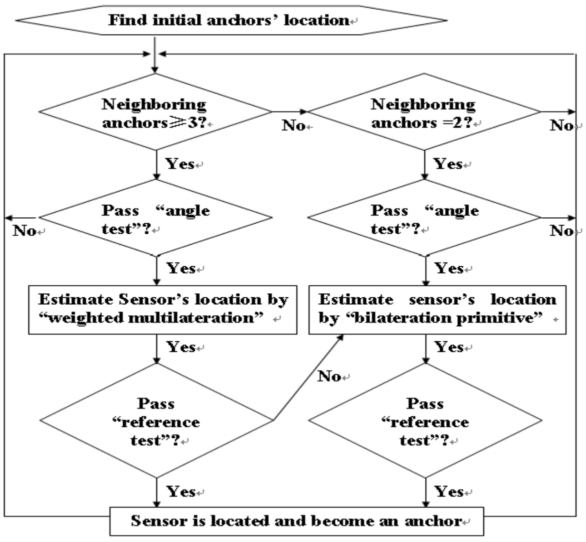
Find initial anchors' location
Neighboring anchors≥3?
No
Neighboring anchors =2?
No
Yes
Yes
Pass "angle test"?
No
Pass "angle test"?
No
Yes
Yes
Estimate Sensor's location by "weighted multilateration"
Estimate sensor's location by "bilateration primitive"
Yes
Yes
No
Pass "reference test"?
Pass "reference test"?
Yes
Yes
Sensor is located and become an anchor

Fig. 2. Algorithm flow diagram

In random deployed wireless sensor network, it is possible that sensor deployments in some directions of the network are sparse and irregular. Then in those directions we may not have three anchors to locate the unknown sensor, so we propose bilateration algorithm that can locate a sensor with just two anchors. This bilateration primitive needs at least an additional anchor two hops away from the unknown sensor to make a decision between two possible solutions produced by two one-hop anchors. Different from other approaches that an additional anchor may not be available, the sensors are located expansively from the initial anchors in our algorithm, thus we always have additional two-hop anchors on the opposite direction of system expansion. The details of locating sensors in the next round, both weighted multilateration and bilateration primitive, will be described in section 4.2. In addition, "reference test", which is used to choose one solution from bilateration primitive or solve "collinear anchors problem"; and "angle test", which is used to limit error propagation speed are presented in section 4.2.

4.1 Deployment of Initial Anchors

In this location discovery algorithm, we suggest putting three anchors as a group and several sensors randomly near these three anchors. These sensors will be located at the beginning and they will act as anchors to assist locating more sensors. There are two problems that should be considered when putting initial three anchors. First, how far away are these anchors from each other? If no sensor is deliberately placed near these anchors, the distances between anchors should be at least equal to the communication range or it is possible that no sensor can be located at the beginning. Meanwhile, they still should be smaller than two times the communication range even if some sensors are deliberately deployed. However, the distances cannot be too small or fewer sensors will be benefited by the initial anchors and location estimation error will be larger. In our algorithm, distances between anchors are set to be 1.5 times the communication range. Second, how many sensors do we need to be deliberately deployed? It is good to deploy more sensor nodes since more sensors will act as anchors at the beginning. But too many additional sensors will increase the installation cost with little influence on the location accuracy. In our algorithm, the number of additional sensors is set to be 7.

Putting three anchors together can be easily realized when anchors are deployed manually or by robots. However, when anchors are dropped from an airplane, some techniques are needed to limit the distances between the anchors. A possible approach is suggested here: Three anchors, several sensors and a timer are bound together by several springs. The anchors and sensors will be spring out when they are close to ground. (This can be controlled by the timer) Three anchors will be away from each other within appropriate distances and the sensors will be scattered around the anchors.

4.2 Locating Sensor in the Next Round

4.2.1 Weighted Multilateration Algorithm

It has been discussed extensively that a sensor can be located when it is within the communication range of three or more than three anchors [9, 11, 14, 15, 16, 20]. A weighted multilateration algorithm is used to minimize the residual of location estimation. Actually, we are trying to minimize

$$f(x,y) = \sum_{i \in one\text{-}hop\ anchors} \omega_i \times \left(\sqrt{(x_i - x)^2 + (y_i - y)^2} - \hat{d}_i\right)^2 \quad \mathbf{(2)}$$

Where (x, y) refers to unknown sensor location, (x_i, y_i) $i = 1,2,3...n$ refers to the location of the i th one-hop anchor, $\hat{d}_i$ refers to the measured distance between unknown sensor and the i th one-hop anchor and ω_i refers to the weight assigned to the i th one-hop anchor. The higher certainty of the location of the one-hop anchor, the larger the weight. The detailed weight assignment scheme will be discussed in section 5.4. Rather than linearizing this nonlinear least square problem [15, 16], we prefer to solve this nonlinear least square problem directly in order to increase location accuracy. In addition, location estimation with big residual will not be accepted. However, nonlinear least square computation requires more computation cost. We will talk about this computation vs. accuracy tradeoff in section 5.1.

4.2.2 Bilateration Primitive

When the unknown sensor can only talk to two anchors, we will try to find its location with only two anchors. As seen from Fig.3, given the location of anchor M(x_M, y_M) and N(x_N, y_N), and the distance to the unknown sensor, r_1 and r_2, the location of the unknown sensor can be computed based on geometry relation between sensor and two anchors. Thus the key point to locate a sensor with two anchors is to choose between two solutions (A and B, as shown in Fig.3). In order to solve this problem, "reference test" is presented in section 4.2.3. In addition, "reference test" can be used to solve "collinear anchors problem" [17, 20], which refers to the situation that all the anchors (>2) are on the same line.

4.2.3 Reference Test

In our algorithm, the one-hop anchors will send a specific number of additional anchors' location as well as range measurement information to the unknown sensor. (We limit

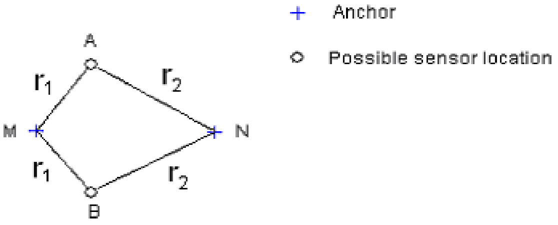

Fig.3. Locate a sensor with two anchors

the number of additional reference anchors because we want to limit the communication cost) So the wrong solution will be recognized if it is within the communication range of these additional reference anchors. "Reference test" can be used in two situations: choose the right solution from bilateration primitive and find the location when anchors (>2) are on the same line. The details of "reference test" are summarized as pseudo codes in Fig.4 and Fig.5 for both situations:

I. When the unknown sensor talks to two neighboring anchors:

```
Inputs:  A(Solution A),
B(solution B), communication
range R, other reference anchors
Outputs: Correct solution
Correct_solution
(A, B, R, reference anchors)
$N_A = 0$
$N_B = 0$
for each anchor C in the
reference anchors
    $d_{AC} = \sqrt{(x_A - x_c)^2 + (y_A - y_c)^2}$
    $d_{BC} = \sqrt{(x_B - x_c)^2 + (y_B - y_c)^2}$
    if  $d_{AC} < R$  $N_A = N_A + 1$ end
    if  $d_{BC} < R$  $N_B = N_B + 1$ end
end
if $N_A >= 1$ & $N_B >= 1$
    return No solution
else if $N_A >= 1$ return B
else if $N_B >= 1$ return A
else return No solution
end
```

Fig.4. "Reference test" for solution selection

In the pseudo code from Fig.4, N_A and N_B refer to number of additional anchors that can be heard for two solutions A and B. If the number of additional anchors can be heard is greater than one, this solution will be discarded. If the number of additional anchors can be heard is just one, we will check the number of heard additional anchors of another solution to decide which solution is correct. This additional check is due to the possibility that an additional anchor is "moved" closer to the unknown sensor under the influence of location estimation error. When "No solution" is returned, two-anchor locationing will not be used.

II. When the unknown sensor talks to more than two neighboring anchors:

When the unknown sensor talks to more than two anchors, we will still check the result using additional anchors to avoid "collinear anchors problem". The result will not be accepted if it is within the communication range of any additional anchor. Then we can pick two one-hop anchors with the angle of the unknown sensor closest to 90^0 to perform bilateration calculation in order to get the result.

```
Inputs:  A(Solution A),
         communication range R,
         other reference anchors
Outputs: Correct solution
Correct_solution
(A, R, reference anchors)
$N_A = 0$
for each anchor C in the
reference anchors
    $d_{AC} = \sqrt{(x_A - x_c)^2 + (y_A - y_c)^2}$
    if  $d_{AC} < R$  $N_A = N_A + 1$  end
end
if $N_A >= 1$ return No solution
else return A
end
```

Fig.5. "Reference test" for "collinear anchors problem"

4.2.4 Angle Test

Geographic dilution of precision (GDOP) can be used to characterize the influence on sensor location estimation from geometry of anchors. A clear representation of GDOP is given in [21]:

$$GDOP = \sqrt{\frac{N}{\Sigma_i \Sigma_{j,j>i} |\sin(\theta_{ij})|^2}} \qquad \textbf{(3)}$$

Where N is the number of reference anchors and θ_{ij} is the angle between anchor i and anchor j. Equation (3) shows that the location error will be very large if all the angles between pairs of anchors are very small or all are close to π. Thus we compute the angles between pairs of anchors based on the ranges and anchors' location, and to see if any of them is within the angle limitation range $[\alpha, \pi - \alpha]$. Location estimation will not be performed when none of the angles is in this range, which means the anchors are clustered to each other $(< \alpha)$ or almost on the same line $(> \alpha)$. The reason why we use "angle test" instead of using GDOP is that "angle test" has similar function as GDOP and we just want to screen out very bad topology rather than to know the exact influence from the anchor geometry. Actually, whereas GDOP only considers range error, we can estimate the sensor location error that considers both range error and anchor location error. The details of location estimation are given in section 5.4.

The angle limitation range will increase when connectivity increases or range error becomes smaller. In our simulation, α is taken as 25^0, when connectivity is 9 and range error is 1%.

5. PERFORMANCE ANALYSES

5.1 Communication Cost

Similar to other localization approaches[8, 14, 16], we suppose the communications between the neighboring nodes are in the form of broadcasts. Then the communication cost will depend on the number of broadcast messages a node transmits and receives. In our algorithm, it is clear that each node will just broadcast one packet when its location is available. And the average number of received broadcast packets by one node will be two, three or four. (Since we set the communication limit to be four, only under very poor geometry of anchors do we need more than four anchors) This makes in total at most five messages per node. Because no position refinement is needed in this algorithm, communication cost will be significantly reduced.

5.2 Computation Cost

The computation performed at each node involves the computation of nonlinear least-squares estimation. Actually the computation cost of nonlinear least square computation can be translated to n times linear least square computation, here n refers to the number of iterations involved in the nonlinear least square computation. (Average of n is about 10 from simulation results). Thus the computation cost of our algorithm will be higher than other algorithms when only one location computation on one node is compared. However, our overall computation cost will still be less since other algorithms need to perform linear least square computation a lot of times [8] in the location refinement phase. Generally, a common processor can handle such computation and the energy consumption will be mainly resulted from communication cost.

5.3 Coverage

Suppose that an error threshold is set to be the maximum acceptable error for a located sensor. We can see that as long as a sensor is not far away from initial anchors, it will find its location in our algorithm. While on the other hand, the traditional approaches may not locate a sensor because computation performed at the sensor cannot converge. In addition, with smaller range error or increased network density, more sensors will be located with three initial anchors. From simulation results in section 6, sensors that can be located will be extended far away from the initial anchors when range error is 1%. And such coverage will be improved further if range error is smaller. We can later learn from section 6 that the coverage of our algorithm is much higher than other algorithms, because other algorithms may not converge at some sensor nodes due to the large distance estimation error.

If the whole field with many "three anchor" groups is considered, the coverage area will depend on how "three anchor" group are deployed. If they are deployed in a grid-like manner, the coverage will be the largest. If they are deployed randomly, there will be overlaps between the fields that can be covered by several "anchor groups". Then the coverage will be smaller or we may need more "anchor groups" to maintain the coverage. If "anchor groups" are uniformly distributed in a random manner, two times "anchor groups" are needed to maintain the coverage. According to the simulation results in section 6, even if two times "anchor groups" are needed, the anchor percentage will still be very low.

5.4 Accuracy

In this section, we describe how to estimate location error of each sensor and how to assign weight to each anchor in order to minimize the location estimation error. The basic steps can be listed as follows:

1. Estimate approximate location of the sensor
2. Weight assigned to each neighboring anchor
3. Estimate final location of the sensor
4. Estimate location error of the sensor

In addition, we present a geometry explanation of location error to show how anchor location error affects sensor location error.

5.4.1 Location Error Estimation

In this expansive locationing algorithm, error propagation can be a serious problem. The farther the unknown point is from the initial three anchors, the larger the location error. Without careful control of the error propagation, the location estimation several hops away from the initial anchors will be unacceptable. An anchor with large location error will adversely affect the location error of its neighboring sensors. In order to prevent such anchors from propagating their errors to the whole network, we propose a novel weight assignment scheme to decrease the influence from large error anchors. Quite different from other weight assignment scheme [14] before, our weight assignment does not only depend on anchors' location error, but also range error and the relative location between the unknown sensor and anchors.

Whenever a sensor is located and becomes an anchor, we will estimate its location error. And when this anchor is used to locate other sensors, we will assign weight to this anchor based on its location error estimation, range error and its relative location to the unknown sensor. We need one linear least square computation to get the approximate location of the unknown sensor (used to assign weight to anchors) and one weighted nonlinear least square computation (the approximate location is used as the initial value) to get the final location estimation.

Suppose the unknown sensor (x_u, y_u) is within the communication range of several reference anchors (x_i, y_i) $i = 1,2,3,4,...n$.

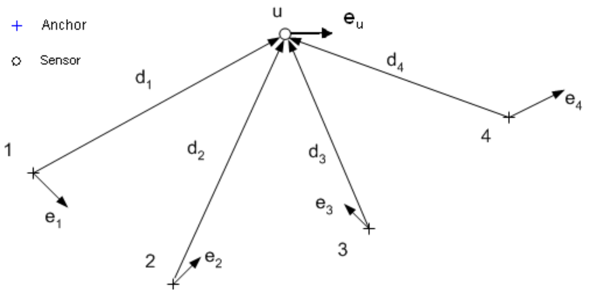

Fig. 6. Sensor location estimation with anchors

As shown in Fig.6, we suppose the estimated unknown sensor to be $(\overline{x_u}, \overline{y_u})$ and estimated reference anchors to be $(\overline{x_i}, \overline{y_i})$ $i = 1,2,3,4,...n$. $\overrightarrow{d_i} = i\frac{\overline{x_u} - \overline{x_i}}{\overline{d_i}} + j\frac{\overline{y_u} - \overline{y_i}}{\overline{d_i}}$ is the anchor distance vector, $\overrightarrow{e_i} = i\Delta x_i + j\Delta y_i$ is the anchor location error vector and $\overrightarrow{e_u} = i\Delta x_u + j\Delta y_u$ is the unknown sensor location error vector. (Note that "i" and "j" do not mean the index of the anchor. It is the unit vector on "x" and "y" direction. In addition, error vectors are bi-directional due to their random nature)

Distance between unknown sensor and reference anchor:

$$d_i = \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2} \qquad (4)$$

Distance between estimated unknown sensor and estimated reference anchor:

$$\overline{d_i} = \sqrt{(\overline{x_u} - \overline{x_i})^2 + (\overline{y_u} - \overline{y_i})^2} \qquad (5)$$

By using Taylor's series around the estimated anchor location and estimated sensor location, we can obtain:

$$\Delta d_i = \frac{\overline{x_u} - \overline{x_i}}{\overline{d_i}}\Delta x_u - \frac{\overline{x_u} - \overline{x_i}}{\overline{d_i}}\Delta x_i + \frac{\overline{y_u} - \overline{y_i}}{\overline{d_i}}\Delta y_u - \frac{\overline{y_u} - \overline{y_i}}{\overline{d_i}}\Delta y_i \qquad (6)$$

$$i = 1,2,3,4,...n$$

where $\Delta d_i = d_i - \overline{d_i}$

Δd_i consists of two error components: range error and least squares estimation error. We ignore the latter component since it is relatively small compared with the range error. (Big residual result will not be accepted in least squares estimation)

From equation [2], we may notice that weight is assigned to anchor nodes to ensure each anchor node has an appropriate level of influence on location estimation. A "high quality" anchor influences the result more than a "low quality" anchor. Optimal results, which minimize the uncertainty in the location estimation, are obtained when the weights, ω_i, assigned to each anchor are inversely proportional to the variances at each combination of predictive variable values of each anchor.[22] Thus in equation (2), ω_i should be inversely proportional to $\mathrm{var}(\sqrt{(x_i - x)^2 + (y_i - y)^2} - d_i)$, which is approximately equal to $\mathrm{var}(\Delta d_i + \frac{\overline{x_u} - \overline{x_i}}{\overline{d_i}}\Delta x_i + \frac{\overline{y_u} - \overline{y_i}}{\overline{d_i}}\Delta y_i)$, note that unknown sensor's location (x, y) should be considered accurate when calculate the variance. By linearizing equation (2), we get the approximate sensor location estimation through linear least square computation [14, 16]. And we assign weight based on this approximate sensor location result.

After weight assignment part, equation (2) is used to solve the final sensor location.(through nonlinear least square computation) In order to estimate the error of this final solution, we write (6) in a compact matrix formulation as:

$$W\begin{bmatrix} \Delta d_1 + \frac{\overline{x_u} - \overline{x_1}}{\overline{d_1}}\Delta x_1 + \frac{\overline{y_u} - \overline{y_1}}{\overline{d_1}}\Delta y_1 \\ \Delta d_2 + \frac{\overline{x_u} - \overline{x_2}}{\overline{d_2}}\Delta x_2 + \frac{\overline{y_u} - \overline{y_2}}{\overline{d_2}}\Delta y_2 \\ \dots \\ \Delta d_n + \frac{\overline{x_u} - \overline{x_n}}{\overline{d_n}}\Delta x_n + \frac{\overline{y_u} - \overline{y_n}}{\overline{d_n}}\Delta y_n \end{bmatrix} = W\begin{bmatrix} \frac{\overline{x_u} - \overline{x_1}}{\overline{d_1}} & \frac{\overline{y_u} - \overline{y_1}}{\overline{d_1}} \\ \frac{\overline{x_u} - \overline{x_2}}{\overline{d_2}} & \frac{\overline{y_u} - \overline{y_2}}{\overline{d_2}} \\ \dots & \dots \\ \frac{\overline{x_u} - \overline{x_n}}{\overline{d_n}} & \frac{\overline{y_u} - \overline{y_n}}{\overline{d_n}} \end{bmatrix}\begin{bmatrix} \Delta x_u \\ \Delta y_u \end{bmatrix} \qquad (7)$$

where W is the weight matrix, a diagonal matrix in which w_{ii} is the weight of the i th anchor.

Suppose,

$$\Delta d_i' = \Delta d_i + \frac{\overline{x_u} - \overline{x_i}}{\overline{d_i}} \Delta x_i + \frac{\overline{y_u} - \overline{y_i}}{\overline{d_i}} \Delta y_i \ , \quad i = 1,2,3...n$$

then we get simplified form of (7):

$$W\Delta d' = WA\Delta X \qquad \textbf{(8)}$$

Solve position error through weighted least squares method and take covariance of both sides, it is not hard to get

$$\mathrm{cov}(\Delta X) = (WA)^{-} W \,\mathrm{cov}(\Delta d') W^T ((WA)^{-})^T \qquad \textbf{(9)}$$

where $(WA)^{-} = ((WA)^T WA)^{-1} (WA)^T$ (pseudo inverse matrix of WA). We assume that the location errors of different sensors are uncorrelated. (It is not exactly true since one sensor may derive its location based on another sensor's location information. But the covariance is quite small compared with the variance of $\Delta d_i'$) Then $\mathrm{cov}(\Delta d')$ can be approximated as a diagonal matrix and the diagonal element is

$$\mathrm{var}(\Delta d_i') = \mathrm{var}(\Delta d_i) + \mathrm{var}(\frac{\overline{x_u} - \overline{x_i}}{\overline{d_i}} \Delta x_i + \frac{\overline{y_u} - \overline{y_i}}{\overline{d_i}} \Delta y_i) \qquad \textbf{(10)}$$

where the first component is the variance of the range error, and the second component can be derived from the location error of the i th anchor, which is computed in the last round.

Thus we can estimate the location error of each sensor node by equation [9]. And location accuracy will be improved since our weight assignment scheme not only considers location error, but also range error and relative location between unknown sensor and anchors.

5.4.2 Geometry Interpretation of Location Error

Given the definition of $\overrightarrow{d_i}, \overrightarrow{e_i}$, equation (7) can be simplified as: (Here $\langle \ \rangle$ refers to inner product)

$$\begin{bmatrix} \Delta d_1 + \langle \overrightarrow{d_1}, \overrightarrow{e_1} \rangle \\ \Delta d_2 + \langle \overrightarrow{d_2}, \overrightarrow{e_2} \rangle \\ \cdots \\ \Delta d_n + \langle \overrightarrow{d_n}, \overrightarrow{e_n} \rangle \end{bmatrix} = \begin{bmatrix} \langle \overrightarrow{d_1}, \overrightarrow{e_u} \rangle \\ \langle \overrightarrow{d_2}, \overrightarrow{e_u} \rangle \\ \cdots \\ \langle \overrightarrow{d_n}, \overrightarrow{e_u} \rangle \end{bmatrix} \qquad \textbf{(11)}$$

The left side of equation (11) consists of two parts Δd_i and $\langle \overrightarrow{d_i}, \overrightarrow{e_i} \rangle$ (shadow vector of $\overrightarrow{e_i}$ on $\overrightarrow{d_i}$). Actually, $\Delta d_1 + \langle \overrightarrow{d_1}, \overrightarrow{e_1} \rangle$ can be viewed as "equivalent range error" when anchor location error is considered. Thus the influence on the location estimation of the unknown sensor from both anchor location error and range error becomes clear. We can see from equation (11) that when anchor location error (e_i) is smaller than range error (Δd_i), sensor location error is decided mainly by range error; whereas when anchor location error is larger, sensor location error will be mainly decided by anchor location error.

Furthermore, we notice from (11) that $\langle \overrightarrow{d_i}, \overrightarrow{e_u} \rangle$ is confined by $\Delta d_i + \langle \overrightarrow{d_i}, \overrightarrow{e_i} \rangle$, which means that the length of the shadow vector of $\overrightarrow{e_u}$ on $\overrightarrow{d_i}$ will be approximately equal to the "equivalent range error". This tells us that sensor location error will be confined in the directions of $\overrightarrow{d_i}$ $i = 1, 2, 3, 4, ...n$. However, location error can be very large in the directions that are away from $\overrightarrow{d_i}$. Thus more anchors will not help to get accurate estimation if they are on the almost same direction of the unknown sensor. Furthermore, since these vectors are bi-directional, we can get accurate location estimation even if anchors are on the one side of the unknown sensor, which is the situation for our algorithm.

Although the above argument is not a mathematically rigorous proof, it does contain the key idea that anchor location error affects sensor location error through its shadow on the distance vector; and the sensor location error will be confined in each direction of the distance vector. We should not assign weights just based on the uncertainty of the anchors; range error and the relative location between an anchor and the unknown sensor should also be considered.

6. SIMULATION RESULTS

The goal of the simulation performed in this section is to investigate the characteristics of the following parameters:

1. Location estimation error vs.
    1) Communication limit
    2) Range error
    3) Communication range (connectivity)
    4) Sensor deployment pattern
    5) Initial anchor deployment pattern

2. Coverage (located sensor percentage)
3. Communication cost

In our experiments, we ran this expansive locationing algorithm on various topologies of networks in Matlab. The result is the average of 500 topologies. Three anchors are placed as an equilateral triangle. And 7 (This is a heuristic number as we mentioned in section 4.1) sensors are randomly placed inside the circle circumscribing the triangle. Two sensor placement models are considered: 1) uniform placement, in which sensors are placed uniformly within a circle area 2) grid placement, in which sensors are placed in a grid-like position within a circle area. The grid length is normalized to 1. The sensors are placed on grid points with placement error modeled as Gaussian noises $N(0,0.25^2)$. The center of the circle area is set to be (0,0) and the radius is set to be 5 in order to maintain a similar anchor percentage with other algorithms. Node density in this circle area is equal to one. We change the communication range from 1.6 to 2.6. The range error is modeled as $N(0,e_r^2)$, in which e_r changes from 0.001 to 0.05 (normalized to communication range R). Anchor percentage is about 3.8% based on this setup.

An unknown sensor can receive a message from each of its one-hop neighbor anchors. However, receiving more messages will increase the communication cost. So we set a communication limit to limit the number of messages an unknown sensor can receive. To get a tradeoff between accuracy and efficiency, we investigate how the average location error will change with communication limit that limits the number of messages this unknown sensor can receive. As shown in Fig. 7, the simulation results suggest that the location error is very large when communication limit is set to 3. However, there is a sharp decrease between 3 and 4. And when communication range is larger than 4, the location error gradually decreases but the difference is rather small. Thus, in our experiment, the communication limit is set at 4 to get a better tradeoff between location accuracy and communication cost. Just as what we discussed in section 5.4, the reason why location error decreases when communication limit increases is we have more "directions". Considering the random choice of neighboring anchors, the "saturation" phenomenon is because we already have enough "directions" to limit the location error of the sensor.

Fig 8 and Fig 9 show the average location error under uniform deployment pattern and grid deployment pattern. Based on the figures, location error becomes smaller when communication range (connectivity) increases. This is because that the unknown sensor can talk to more anchors, thus location error is confined in more "directions". In addition, grid deployment produces better results than uniform deployment because grid deployment does not have bad sensor topologies whereas uniform deployment may have bad sensor topologies that will result in large location error. When range error is smaller than 1%, the average location error will be always smaller than 4% (both normalized to communication range).

Fig.10 and Fig.11 show the changes of the system coverage under different sensor deployment patterns. When connectivity is 7, range error is 1%, the coverage will be more than 78% under uniform deployment and 98% under grid deployment. However, other approaches [8] cannot achieve high coverage and small location error at the same time. As mentioned in [8], coverage of these traditional approaches will drop by 50% when location refinement phase is performed But when refinement phase is not performed, the location error of these approaches will be large (more than 25%) even if the range error is very small (equal to zero) and connectivity is very high (equal to 15). (Fig.11 in [8]) Thus our algorithm can achieve high accuracy as well as high coverage.

When anchors are dropped from the airplane, the relative location between initial three anchors cannot be precisely controlled. Since initial anchor deployment pattern may influence the average location error, we investigate the following five initial anchor deployment patterns:

Pattern1: {(0, 1.7321), (-1.5, -0.866), (1.5, -0.866)}
Pattern2: {(0, 1.5011), (-1.3, -0.7506), (1.3, -0.7506)}
Pattern3: {(0, 1.1547), (-1.0, -0.5774), (1.0, -0.5774)}
Pattern4: {(0, 1.1547), (-1.6, -1.0970), (1.6, -0.5196)}
Pattern5: {(0, 1.3856), (-1.4, -0.9238), (1.4, -0.6928)}.
Each pattern refers to the locations of three anchors, as shown in Fig. 12. The results indicate that average location errors don't change much with deployment patterns and thus this algorithm is robust to the anchor deployment pattern. In addition, the larger the distance between the initial anchors, the smaller the average location error. This can be clearly recognized from the curves of pattern 1, 2 and 3. Actually, when anchors are farther from each other, more sensors will be benefited by the initial anchors, thus location estimation accuracy is improved.

A comparison between MDS-MAP approach [18], Sum-dist approach [8], DV-hop approach [8, 11, 15] and our Expansion approach is given in Table.1. The range error is 5% for all these approaches. Under similar anchor percentage, connectivity and range error, our expansive approach can achieve better accuracy, higher coverage and significantly less communication cost. Detailed communication cost is not given in MDS-MAP paper[18], but this approach is essentially a centralized method which requires large communication cost.

Our algorithm produces very good results when range error becomes smaller, as shown in Table.2. Table.2 describes the localization upper bound (L_u) as a function of location error upper bound(E_u) and range error (e_r). Here E_u refers to the largest location error that can be accepted by location-based applications, L_u refers to the expected largest extending distance from initial anchors given error upper bound and range error. The simulation is performed with connectivity equal to 11. When range error is 1% and location error upper bound is 40%, localization upper bound becomes 36.46. This means under similar connectivity level with other approaches, anchor percentage can be less than 0.1%! This result is much smaller than other approaches that require anchor percentage to be at least 5%. Moreover, when range error is 0.1%, small location error can be achieved as well as low anchor percentage. When localization upper bound is fixed to be 50, range error at 0.1% and connectivity equal to 11.6, less than 5% average location error can be achieved with only 0.04% anchor percentage!

The objective for localization in wireless sensor networks is not only to minimize location estimation error, but also to minimize cost, which includes installation cost, computation cost and communication cost. In conclusion, the advantage of our approach is to locate sensor with less localization error, less installation cost (much less anchors needed) and less communication cost. Encouraging results are obtained when range error is small. (<1%)

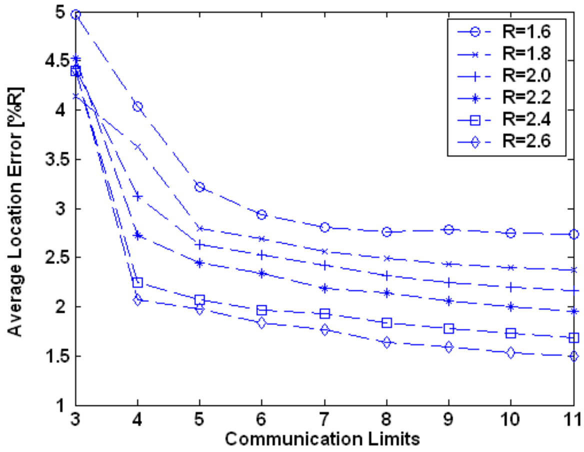

Fig. 7. Average location error Vs communication limits (Range error 0.01, Uniform deployment)

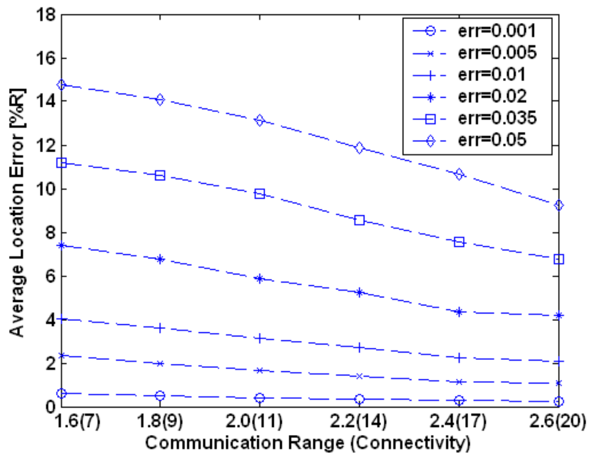

Fig. 8. Average location error vs. communication range under uniform deployment

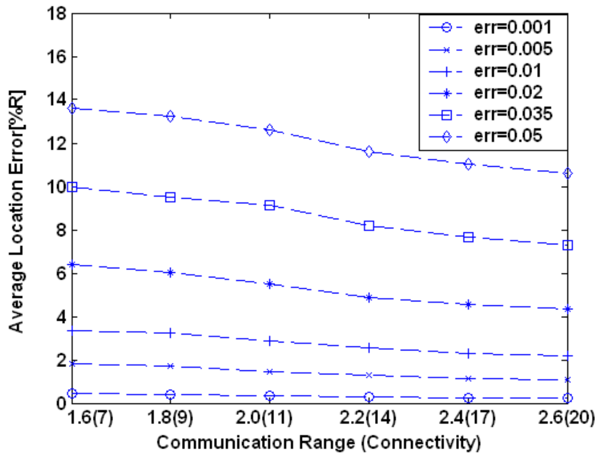

Fig. 9. Average location error vs. communication range under grid deployment

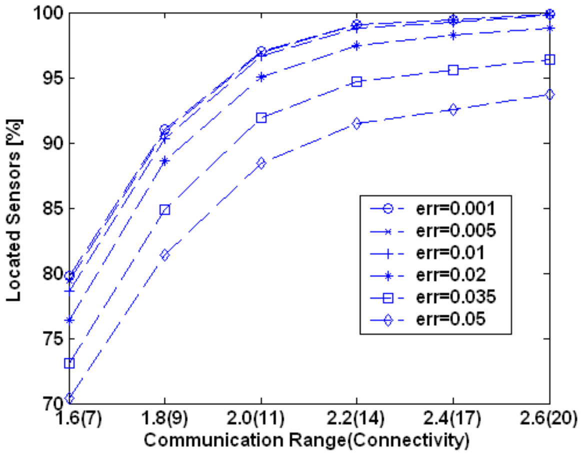

Fig. 10. Average coverage vs. communication range under uniform deployment

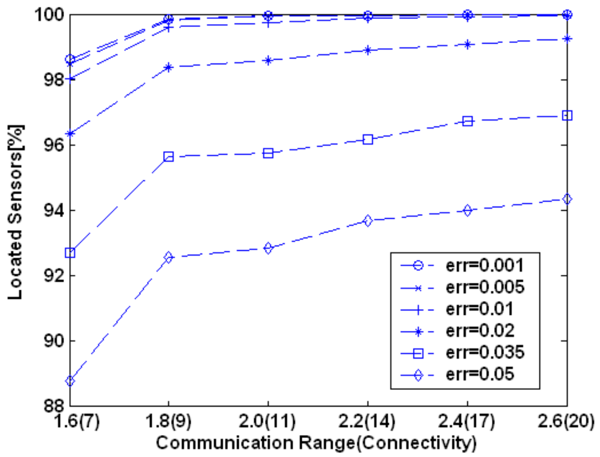

Fig. 11. Average coverage Vs communication range under grid deployment

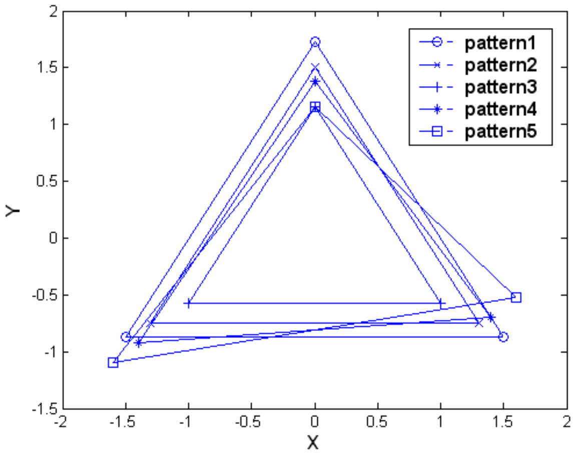

Fig. 12. Different initial anchor deployments

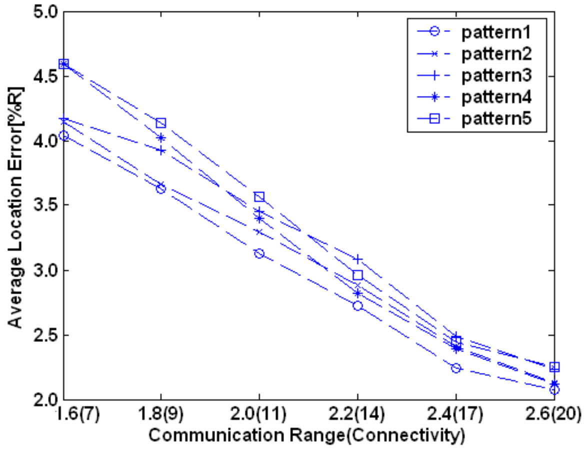

Fig. 13. Average location error for different patterns of initial anchor deployment

Table.1. Performance comparison

	MDS-MAP	Sum-dist	DV-hop	Expansion
Anchor percentage	4%	5%	5%	3.8%
Connectivity	12.242	12	12	11.6
Location error	21%	17%	17%	13%
Coverage	100%	<60%	<60%	89%
Communication messages	large	36.3	33.8	<5

Table.2. Localization upper bound L_u

E_u e_r	0.05	0.10	0.20	0.30	0.40
0.01	6.36	12.78	22.75	30.71	36.46
0.02	4.11	7.56	12.43	19.46	24.30
0.05	——	4.36	5.75	8.92	10.87

7. CONCLUSION

In this paper, we propose an expansive locationing approach. The novelty of this paper is to put three anchors together, and locate sensors expansively around them. To the best of our knowledge, we are the first to propose such localization scheme. Extensive theoretical error analyses are presented in order to control error propagation. In addition, bilateration primitive is used to improve coverage and "collinear anchors problem" is solved. Our simulation results demonstrate that our algorithm can derive sensor location within 4% location error and much less communication cost compared with other locationing algorithms. It provides a useful tool for location discovery in WSNs.

8. REFERENCES

[1] Bahl, P., Padmanabhan. V.N., RADAR: An In-Building RF-Based User Location and Tracking System. *IEEE INFOCOM.* Vol. 2., pp. 775–784, 2000.

[2] Bulusu, N., Heidemann, J., Estrin, D., GPS-Less Low Cost Outdoor Localization for Very Small Devices, *IEEE Personal Communications, Special Issue on Smart Spaces and Environments*, Vol. 7(5), pp. 28-34, 2000.

[3] Capkun, S., Hamdi, M., and Hubaux, J.-P., GPS-free positioning in mobile Ad-Hoc networks. *Hawaii International Conference on System Sciences*, pp. 3481-3490, 2001.

[4] Doherty, L., Pister, K., and Ghaoui, L. El., Convex Position Estimation in Wireless Sensor Networks, *IEEE Infocom* pp.1655–1663, 2001.

[5] Girod, L., Estrin, D., Robust Range Estimation Using Acoustic and Multimodal Sensing. *IROS*, 2001

[6] He, T., Huang C., Blum, B. M., Stankovic, J. A., and Abdelzaher, T., Range-Free Localization Schemes for Large Scale Sensor Networks, *Mobicom*, pp. 81-95, 2003.

[7] Hightower, J., Want, R., Borriello, G., SpotON: An indoor 3d Location Sensing Technology Based on RF Signal Strength, *UW CSE University of Washington, Seattle, 2000.*

[8] Langendoen, K., and Reijers, N., Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison. Computer Networks (Elsevier), *special issue on Wireless Sensor Networks, 2003.*

[9] Meguerdichian, S., Slijepcevic, S., Karayan, V., and Potkonjak, M., Localized Algorithms In Wireless Ad-Hoc Networks: Location Discovery And Sensor Exposure. *MobiHOC,* 2001.

[10] Nagpal R., Shrobe, H., and Bachrach, J., Organizing a Global Coordinate System from Local Information on an AD Hoc Sensor Network, *2nd International Workshop on Information Processing in Sensor Networks*, 2003.

[11] Niculescu, D., Nath. B., Ad-Hoc Positioning System, *IEEE GlobeCom,* 2001.

[12] Priyantha, N., Chakraborthy, A., and Balakrishnan, H., The Cricket Location-Support System, *Proceedings of International Conference on Mobile Computing and Networking,* pp. 32-43, 2000.

[13] Priyantha, N., Miu, A., Balakrishnan, H., Teller, S., The Cricket Compass for Context-Aware Mobile Applications. *ACM Sigmobile (Mobicom,* pp. 1–14, 2001.

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

[15] Savarese, C., Rabaey, J., and Beutel, J., Locationing in Distributed Ad-hoc Wireless Sensor Networks, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2037-2040, 2001.

[16] Savvides, A., Han, C. C., and Srivastava, M. B., Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors, *Proceedings of the fifth annual international conference on Mobile computing and networking, Mobicom*, pp. 166-179, 2001.

[17] Savvides, A., Park, H., Srivastava, M. B., The bits and flops of the n-hop multilateration primitive for node localization problems, *Mobicom Workshop on Wireless Sensor Networks and Applications (WSNA 2002),*, pp.112-121, 2002.

[18] Shang, Y., Ruml, W., Zhang, Y., Fromherz, M. P. J., Localization from Mere Connectivity, *MobiHoc* , pp. 201-212, 2003.

[19] Ward, A., Jones, A., Hopper, A., A New Location Technique for the Active Office, *IEEE Personal Communications*, Vol. 4(5), 1997.

[20] Albowicz, J., Chen, A., Zhang, L., Recursive Position Estimation in Sensor Networks, *ICNP*, 2001.

[21] Srdan Capkun, Maher Hamdi, and Jean-Pierre Hubaux, "GPS-free positioning in mobile ad-hoc networks," *34th IEEE Hawaii Int. Conf. on System Sciences (HICSS-34),* Jan. 2001.

[22] http://www.itl.nist.gov/div898/handbook/pmd