HexNet: Hexagon-Based Localization Technique For Wireless Sensor Networks

Hady S. AbdelSalam
Computer Science Dep.
Old Dominion University,
Norfolk, VA 23508
asalam@cs.odu.edu

Stephan Olariu
Computer Science Dep.
Old Dominion University,
Norfolk, VA 23508
olariu@cs.odu.edu

***Abstract*—Developing efficient localization techniques to allow sensors to estimate their positions has been a hot research topic for long time. Sensory readings are almost meaningless if they are not associated with the locations in which these readings were taken. Getting location information through recording positions manually or through an expensive GPS chip are not valid options for sensor networks. We propose and evaluate, a new localization protocol for sensor networks. Although our technique is presented using only a single anchor, it can be extended easily to benefit from the existence of several anchors. The proposed technique starts by hypothetically tiling the deployment area around the anchor node using identical hexagons. After that, the closest sensor node to the center of each of these hypothetical hexagons is determined, we refer to these nodes as backbone sensors. The positions of hexagon centers hence the positions of backbone sensors are estimated geometrically. After that, backbone sensors are treated as beacons and the positions of all non-backbone sensors are estimated using the centroid approach. Simulation results under noise-free and noisy conditions show that the proposed protocol achieves a localization accuracy that makes it useful for most WSN applications.**

I. Introduction

Sensory readings are meaningless unless associated with the locations in which they were taken. To make such association possible, a sensor must have access to information about its exact or at least approximate position. Manually recording and entering the position of each sensor is impractical for large sensor networks with massive deployment. Also adding an expensive GPS chip to each sensor is not an optimal option for such tiny low-cost devices.

The problem of determining the geographic positions of sensors is referred to as "sensor localization" problem. This problem has attracted the attention of many researchers in the literature. Several localization algorithms have been proposed, each has its advantages and its disadvantages. As there is no specific protocol on top of others, we just outline the technical foundations of each technique in the next section.

II. Related Works

Localization techniques can be broadly classified into two categories: *Range-based* and *Range-free*. Range-based techniques depend on range estimation between nodes that know their positions and nodes that do not. Ranges are usually estimated based on measurements of received signal strength (RSS), time of arrival (TOA), time difference of arrival (TDOA) and angle of arrival (AOA). Although Range-based techniques are more accurate than Range-free techniques, they share a common drawback, they require additional hardware to be available with each node to be able to take the required measures. Range-free techniques have been proposed to overcome the stringent hardware requirements of range-based techniques. The main idea behind these techniques is to exploit radio connectivity information among neighboring nodes to infer their location without taking any range measurements. This way range-free techniques eliminate the need of the specialized hardware needed on each node.

In the Centroid method [1], a sensor node estimates its location as the centroid of the polygon whose vertices coincide in location with all the beacons it could hear. If beacons are positioned uniformly, localization error can be reduced however this can not be guaranteed in ad hoc or non static deployments. Another similar approach for complex shapes was proposed in [4]. The APIT method [2] divides the deployment area into triangular sections using beacon nodes. Each sensor applies an approximate PIT test to decide whether it is inside each possible triangular or not. After that it uses a grid scan algorithm to estimate the maximum likelihood area within which it resides. The Ad-hoc Positioning System (APS) [6] was proposed to allow non-GPS enabled nodes to estimate their locations in a hop by hop fashion. Three different propagation methods were investigated, and the DV-Hop algorithm was the best to perform for most purposes. In the DV-hop algorithm, each beacon sends a message that contains its location information to all the sensors nodes around it. Sensors that receive these messages forward them to their neighbors and so the messages are flooded through the whole network. Within each message, there is a field that indicates number of hops this message has been forwarded. Basically, this field is initialized to 1 at the beacon node, and incremented at each hop. This way, sensors can determine how far they are (in terms of number of hops) from different beacons. The average distance per hop is calculated and each

[0]This work was supported in part by NSF grant CNS-0721563 i.e. Collaborative Research-NeTS-NOSS: AutoNomouS netWorked sEnsoR systems (ANSWER).

978-1-4244-3304-9/09/$25.00 ©2009 IEEE

sensor can estimate its distance to different beacons. After that, a sensor node can use the estimated distances between itself and three or more different beacons to figure out its position. The Amorphous[5] localization protocol depends on a similar idea. The location coordinates of each beacon are flooded throughout the network with the number of hops to the source beacon tracked in each message. This enables each node to maintain a list of hop-count to each beacon along with the location of that beacon. Each node that does not know its location can use this list to estimate its location. Cricket [8], an indoor location support system proposed by MIT, allows nodes to learn their physical location by using listeners that hear and analyze information from anchors (beacons). Anchors concurrently use radio and ultrasonic signals to send their location information. The listener inference modules on each node use TDOA to localize sensors and overcome multipath and interference to improve localization accuracy. AHLoS [10] is similar to Cricket and uses RF and ultrasound for indoor localization. TDOA techniques like these techniques rely on extensive hardware that might be expensive and energy consuming, making it less suitable for sensor networks. Another drawback of TDOA techniques that uses ultrasound, they require dense deployment as ultrasound signals propagate for a limited distance only. The lighthouse system [9] uses a rotating anchor that produces a parallel light beam of fixed width. A sensor node detects the light beam for a period of time that depends on the distance between the anchor and the sensor. If the rotation speed and the width of the beam are known to the sensors, then each sensor can measure the time it detects the light beam and estimates the distance and the angle to the anchor. Acoustic-based ranging techniques like BeepBeep [7] was proposed to localize sensors based on the two-way time of flight of the beeps between two communicating devices. APL was suggested in [3] to address localization in road networks where most range based localization techniques fail due to the sparse nature of deployment. APL uses binary vehicle-detection timestamps to obtain distance estimates between any pair of sensors on roadways. Although the literature has many other interesting localization techniques, we stop our survey at this point for the limited space.

Despite this long list of techniques, our proposed technique still has its advantages over other techniques. We summarize these advantages in the following: (1) the proposed technique is very flexible in terms of number of available anchors. The protocol is presented with only a single anchor, however it can be easily extended to benefit from the existence of several anchors for better localization accuracy. (2) our technique implicitly provides a clustering mechanism that divides the network into clusters (hexagons) where the backbone sensor at the center of each hexagon is the cluster head. (3) The proposed protocol constructs a virtual ternary coordinate system that simplifies geographic routing in different network parts. (4) The proposed technique does not require sensors to be equipped with sophisticated hardware to measure AOA or TDOA.

The remainder of the paper is organized as follows: in section III, we describe our assumptions for the underlying network model. The details of the proposed localization protocol is presented in section IV. Performance evaluation is summarized in section V. Section VI concludes the paper.

III. Network Model

In our network model, we assume the following:

- A sensor node refers to a tiny electronic device with limited sensing, computational and communicational capabilities.
- Sensors are powered by non-renewable on-board energy source. When this energy supply is exhausted, a sensor becomes totally in-operational; hence sensors sleep and wake up alternatively to save their energy. Sleep and wake up cycles for different sensors occur asynchronously.
- Sensors are deployed massively, once deployed, they must work *unattended*. Due to the massive deployment, it is impractical or infeasible to devote attention to individual sensors hence sensors should be treated as if they were *anonymous* – with no fabrication-time identities.
- The network should have one or more anchor nodes that are distributed uniformly across the deployment area. Anchor nodes are assumed to be aware of their own positions either through manual insertion of these positions or through using GPS devices. Anchors have steady energy supply, hence they have no energy constraints and remain awake all the time.
- Each anchor node is equipped with two powerful transceivers, one can transmit directionally while the other can transmit omnidirectionally.
- Each sensor has a maximum transmission range, denoted by t_x, assumed to be much smaller than R (the maximum transmission range of anchor nodes).[1] This implies that messages sent by a sensor can only reach sensors in its proximity.
- The reception circuitry in sensors should be able to determine the received signal strength RSS. [2]

IV. Localization Protocol

We summarize the main idea behind our localization protocol in the following steps. (1) Sensors are geographically clustered into identical disjoint hexagons that perfectly tile the whole deployment area around anchor nodes. (2) Within each hexagon, the closest sensor to the hexagon center is determined and referred to as hexagon backbone sensor. (3) The position of the center of any hexagon is calculated geometrically relative to its nearest anchor node. (4) Backbone sensor positions are approximated by the positions of the hexagon centers they occupy. (5) Finally, backbone sensors are treated as beacons and the positions of all non-backbone sensors are estimated using the centroid approach.

[1] Of course, t_x depends on the particular type of sensors deployed. Under technology available at the time of writing this paper, t_x is about 50m for micro-sensors.

[2] Sensors with these capabilities do exist. Of course, the accuracy of the measured RSS depends on the particular type of sensors used.

Since sensor communication is omnidirectional, using circular regions may look as a more appropriate choice. However, since circles can not perfectly tile the deployment area, we decided to use hexagons as shown in figure 1. The size of the tiling hexagons is determined such that transmissions of any backbone sensor are overheard by all its immediate neighboring backbone sensors. In particular, if we set the distance between any two neighboring backbone sensors to t_x, the maximum transmission range of a sensor, then hexagon size as determined by the radius of the circle that passes through its vertices should be $\frac{t_x}{\sqrt{3}}$.

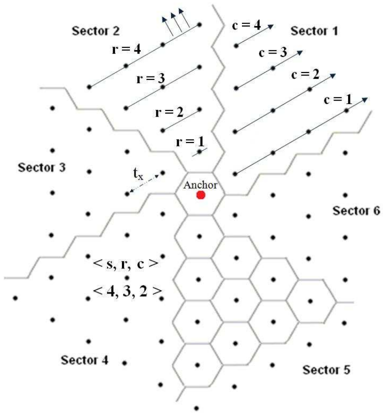


Fig. 1. **Dividing Deployment Area into Sectors**

Tiling hexagons starts from anchor nodes outward. The first hexagon is placed so its center coincides with the anchor node. Other hexagons are placed side by side in six different directions (i.e. $\frac{\pi}{6}$, $\frac{3\pi}{6}$, $\frac{5\pi}{6}$, $\frac{7\pi}{6}$, $\frac{9\pi}{6}$, and $\frac{11\pi}{6}$). The geometry of the gaps between the hexagons in any two consecutive directions allows perfect tiling using sequentially increasing number of hexagons.

We use a ternary coordinate system $\langle$sector,row,column$\rangle$ to uniquely identify hexagons and backbone sensors. The deployment area is divided into six sectors. In each sector, hexagons are stacked in rows. In the first row, there is only one hexagon(column), in the second row, there are two, in the third there are three ... and so on. The hexagon in column c in row r in sector s is uniquely identified using the ternary tuple $\langle s, r, c\rangle$. Sectors and columns are numbered as shown in figure 1.

To allow fast localization, backbone sensors should be selected in all directions at the same time. This makes the selection process very tricky especially when we try to avoid redundant selections and reduce transmission collisions to save sensor energy. We propose a selection strategy that starts when an anchor node selects the backbone sensor in the first row in each sector. After that, selection continues recursively where backbone sensors in row i select backbone sensors in row $i+1$ and so on. This continues for a maximum number of rows or until we reach the boundaries of the deployment area where no more backbone sensors can be selected.

Given backbone sensor S, we list the rules that govern

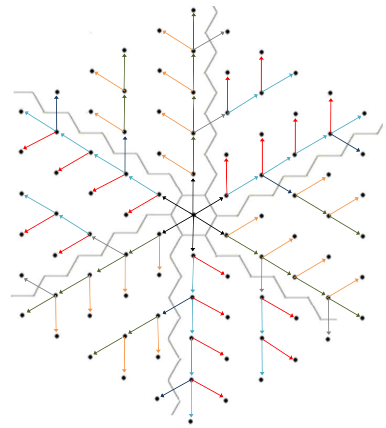


Fig. 2. **Directions of Searching for Backbone Sensors**

which backbone sensors S should select:

1) Only backbone sensors with odd column coordinate are allowed to select.
2) Sensor $S_{\langle s,r,2c-1\rangle}$ selects sensors $S_{\langle s,r+1,2c-1\rangle}$ and $S_{\langle s,r+1,2c\rangle}$. One exception is for backbone sensors in the first column in even rows. Sensor $S_{\langle s,2r,1\rangle}$ selects sensors $S_{\langle s,2r+1,1\rangle}$, $S_{\langle s,2r+1,2\rangle}$ and $S_{\langle s-1,2r+1,2r+1\rangle}$.
3) Selection of a backbone sensor takes one time epoch.
4) Selection of backbone sensors with the same row and column coordinates in different odd sectors (i.e. $s = 1, 3, 5$) occurs in the same time epoch. A similar rule applies for even sectors (i.e. $s = 2, 4, 6$).

From the above rules we emphasize on the following. backbone sensors with even column coordinate do not select. Search in odd rows in a single sector requires 2 time epochs, however search in even rows requires 3 time epochs. The total number of time epochs needed to search in all the six sectors is 4 time epochs for odd rows and 6 time epochs for even rows. Figure 2 shows search directions for the first 4 rows in different sectors. Up to this point, we only specified the directions of selection of backbone sensors and distributed the responsibilities of the selection over network sensors. The reader may wonder how these sensors are selected and this will be described in details in the upcoming sections.

As we mentioned earlier, backbone sensors are chosen to be the closest sensors to the centers of their hexagons hence their positions can be approximated by the positions of these centers. From the geometry of the design, it can be easily shown that the position (x, y) of the hexagon center identified by the tuple $\langle s, r, c\rangle$ is

$$x = x_a + r.t_x.\cos\left(\frac{\pi}{3}(s-\frac{1}{2})\right) + (c-1).t_x.\cos\left(\frac{\pi}{3}(s+\frac{3}{2})\right) \quad (1)$$

$$y = y_a + r.t_x.\sin\left(\frac{\pi}{3}(s-\frac{1}{2})\right) + (c-1).t_x.\sin\left(\frac{\pi}{3}(s+\frac{3}{2})\right) \quad (2)$$

Where (x_a, y_a) is the position of the nearest anchor node, t_x sensor maximum transmission range and at the same time the distance between the centers of two neighboring backbone sensors.

In the next subsection, we show how backbone sensors are selected. The proposed approach requires sensors to be aware of the angle between the positive x axis and the line connecting the sensor to the anchor node. Hence, we find it more appropriate to describe how sensors can measure this angle. The reader should recall that in our model, we assumed that anchors are capable of transmitting directionally and omnidirectionally. The directional antennas at anchor nodes have a very small beam-width and can be rotated in any direction. Initially, each anchor node uses its omnidirectional antenna to send a sequence of messages to wake up sleeping sensors so they can estimate their angles to the anchor node. The number of messages should be large enough to guarantee that all the sensors around the anchor have received at least one copy of the message. After the last message, the anchor node uses its directional antenna to send a sequence of angle determination messages while slowly rotating its antenna. Within each such message, the anchor sends its position (x_s, y_s) and the transmission angle used to transmit this message. Sensors that receive these messages estimate their angle as the average of the transmission angles within these messages. The whole process can be repeated several times for more accurate results with an obvious trade off between the achieved accuracy the number of cycles needed.

A. Selection of backbone Sensors

In this section, we show how backbone sensors are selected. Initially, the searching entity which might be an anchor node or another backbone sensor computes ϕ, the angle between the nearest anchor node and the center of the hexagon that contains the backbone sensor to be selected. Assuming this hexagon is identified by the tuple $\langle s, r, c\rangle$, we can determine ϕ geometrically using,

$$\phi = \frac{\pi}{3}\left(s - \frac{1}{2} + \frac{c-1}{r}\right) \tag{3}$$

After that the searching entity broadcasts a message to all sensors in its neighborhood calling for sensors with angle ϕ with the anchor node to declare themselves. Sensors that hear this call for the first time estimates e_i, the distance between itself and the center of the target hexagon and initialize an internal countdown timer to this value. When the timer of any of these sensors expires, the sensor realizes it is the closest sensor to the center of the target hexagon, hence it broadcasts a message to all his neighbors announcing itself as the backbone sensor of the hexagon. Sensors that hear this message stop their timers and use this message among other messages they receive from other backbone sensors to determine their location. When collisions happen, ties can be broken by allowing colliding sensors to compete in another countdown round starting from a randomly selected value.

Now, we show how a sensor can estimate e_i, the distance between itself and the center of the target hexagon. Here, we distinguish between two different cases.

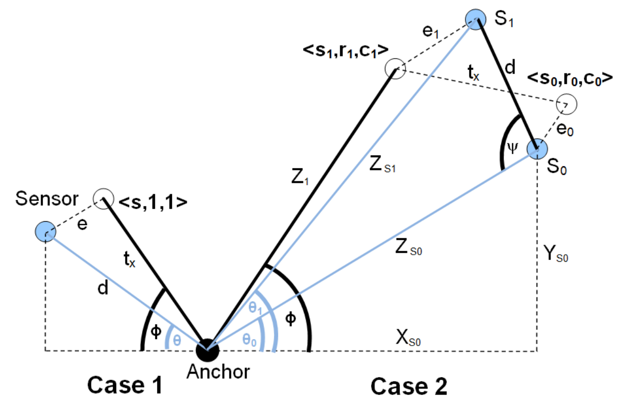


Fig. 3. **Estimation of Selection Error in Case 1 and Case 2**

<u>Case 1:</u> in which we estimate e_i for the six backbone sensors around the anchor node. As shown in figure 3, the selection error of a sensor can be evaluated using,

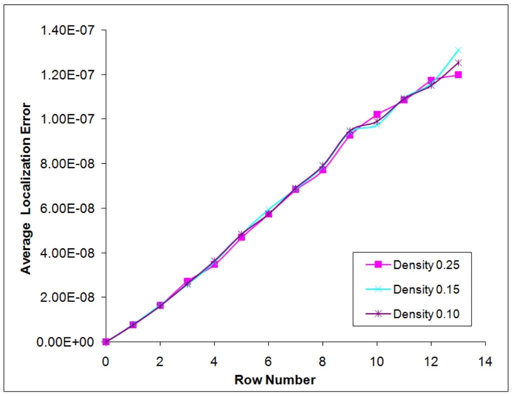
$$e_i = \sqrt{{t_x}^2 + d^2 - 2.t_x.d.\cos(\theta_2 - \theta_1)} \tag{4}$$

Where d is the distance between the anchor and the sensor and obviously it can be estimated using received signal strength.

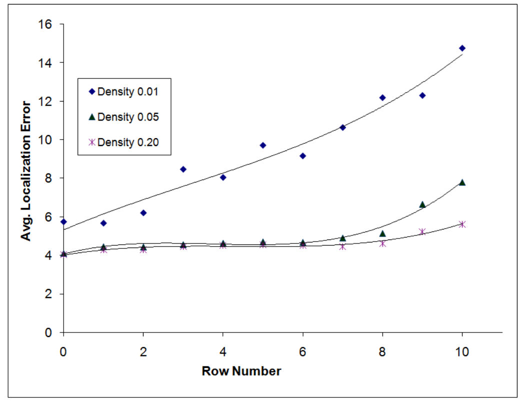
<u>Case 2:</u> in this case, we estimate e_i for backbone sensors other than those handled by Case 1. As shown in figure 3, we assume the existence of another backbone sensor s_0 that was previously selected by the protocol to represent the hexagon $S_{\langle s_0, r_0, c_0\rangle}$. The selection error of sensor s_0 is e_0 and represents the distance between s_0 and the center of the hexagon $S_{\langle s_0, r_0, c_0\rangle}$. Based on selection rules described earlier, it is s_0 turn to select sensor s_1 to represent the hexagon $S_{\langle s_1, r_1, c_1\rangle}$. s_1 is selected such that the selection error e_1 is minimum where (e_1 is the distance between s_1 and the center of the hexagon $S_{\langle s_1, r_1, c_1\rangle}$). Now, our goal is to provide an expression for e_1 that can be evaluated by each sensor independently.

As we mentioned earlier, the searching sensor s_0 sends a message to sensors in its neighborhood calling for the closest sensor to the required hexagon center to announce itself to others. To avoid repeating the same calculations at each sensor, we propose that sensor s_0 evaluate all common expressions and send them within the message it sends to its neighboring sensors. These expressions include the following:

1) $Z_{s0} = \sqrt{X_{s0}^2 + Y_{s0}^2}$, the estimation of the Euclidean distance between the anchor node and sensor s_0.
2) $\theta_0 = \tan^{-1}\frac{Y_{s0}}{X_{s0}}$, the measured angle between the line connecting the anchor node to s_0 and the positive X axis.
3) In addition to Z_{s0} and θ_0, s_0 also evaluates and sends Z_1 and θ_2. Z_1 is the distance between the center of the target hexagon $\langle s_1, r_1, c_1\rangle$ and the anchor node. ϕ is the angle surrounded by the line connecting the anchor



(a)



(b)

Fig. 4. **Average Localization Error of Backbone/Non-Backbone Sensors for Different Network Densities.**

node to the center of the target hexagon $S_{\langle s_1, r_1, c_1 \rangle}$ and the positive x axis.

Mathematically, Z_1 and ϕ can be evaluated using,

$$\begin{aligned} Z_1 &= \sqrt{X_1^2 + Y_1^2} \\ \phi &= \frac{\pi}{3}\left(s_1 - \frac{1}{2} + \frac{c_1 - 1}{r_1}\right) \end{aligned}$$

Where (X_1, Y_1) is the position of the center of the hexagon $S_{\langle s_1, r_1, c_1 \rangle}$ and can be evaluated using equations 1 and 2.

After receiving Z_{s0}, θ_0, Z_1 and ϕ, each sensor continues the estimation of e_1 on its own. Let, θ_1 be the angle surrounded between the line connecting the sensor to the anchor node and the positive x axis. Sensors can measure this angle as we described earlier. Using RSSI, each sensor estimates d, the distance between the sensor itself and sensor s_0. Using the trigonometric law of sines, each sensor calculates Z_{s1}, the distance between itself and the anchor node as follows,

$$\begin{aligned} \frac{d}{\sin(\theta_1 - \theta_0)} &= \frac{Z_{s0}}{\sin(\pi - \theta_1 + \theta_0 - \psi)} \\ \psi &= \pi - \theta_1 + \theta_0 - \sin^{-1}\left(\frac{Z_{s0}.\sin(\theta_1 - \theta_0)}{d}\right) \\ Z_{s1} &= \sqrt{d^2 + Z_{s0}^2 - 2.d.Z_{s0}.\cos(\psi)} \qquad (5) \end{aligned}$$

Finally, each sensor can apply the trigonometric law of cosines to evaluate e_1 as,

$$e_1 = \sqrt{Z_1^2 + Z_{s1}^2 - 2.Z_1.Z_{s1}.\cos(\phi - \theta_1)} \qquad (6)$$

It is also worthwhile to mention that sensors which evaluate e_1 to be larger than some threshold value e_{max} should not initialize their internal timers. This implicitly provides the stopping criterion upon which the boundaries of the deployment area is determined. After selection of backbone sensors, each sensor estimates its position as

$$(x_a + Z_{s1} \cdot cos(\theta_1), y_a + Z_{s1} \cdot sin(\theta_1))$$

After being localized, backbone sensors can be treated as beacons to localize non-backbone sensors in their neighborhood. The geometric structure of the hexagons typically allows a non-backbone sensor to hear messages from three or four backbone sensors, hence they are able to apply the centroid approach to localize themselves.

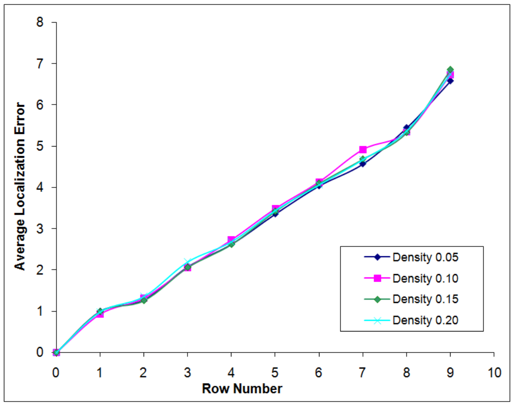
V. Performance Evaluation

To verify the performance of the proposed protocol, we built a simulator of a WSN that implements our protocol to localize deployed sensors. In our simulation, we assumed a square deployment area with side length 200m. We used a single anchor node that was placed at the center of the deployment area. We used a standard uniform pseudo-random generator to distribute sensors in the deployment area. Sensor maximum transmission ranget_x was set to $30m$. We run several simulation experiments using different sensor densities that ranges from 0.01 to 0.25 sensors$/m^2$. We estimated localization error as the average of the Euclidean distance between the actual and the estimated positions of sensors. Mathematically, this was expressed as,

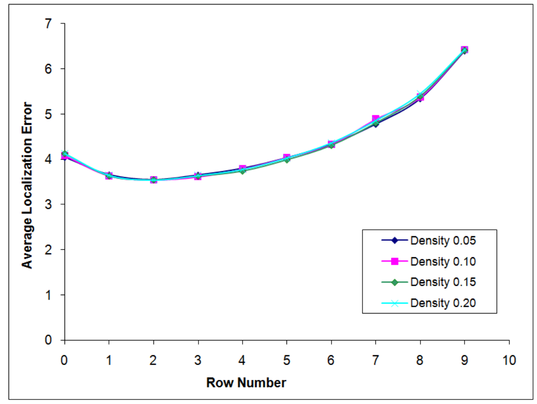
$$\text{Avg. Error} = \frac{\sum_{i=1}^{n} \sqrt{(x_{ai} - x_{ei})^2 + (y_{ai} - y_{ei})^2}}{n}$$

where (x_{ai}, y_{ai}) is the actual position of sensor i and (x_{ei}, y_{ei}) is its estimated position.

Since localization of non-backbone sensors depends on backbone sensors, we estimate localization errors of both types separately. Initially, we run the protocol assuming exact distance measurements to verify the correctness of our simulation implementation. Basically, in the absence of distance measurement errors, sensors should be localized accurately except for truncation errors, if any. Figures 4(a) and 4(b) respectively, show the average localization error in backbone and non-backbone sensors in the absence of measurement errors. The almost exact estimation of backbone sensor positions verifies the correctness of the implementation. On the other hand, errors in localization of non-backbone sensors are inevitable when using the centroid approach even in the absence of measurements errors. Figure 4(a) shows that errors in backbone sensor localization are independent of the underlying



(a)



(b)

Fig. 5. **Average Localization Error for Backbone/Non-Backbone Sensors in the Existence of Errors in Distance and Angle Estimations**

network density. In fact this was expected because in sparse networks although the distances between the hexagon centers and backbone sensors might be large, this should have no effect on localization accuracy because the proposed protocol considers selection errors while localizing backbone sensors. On the other hand figure 5(b) shows that the localization errors for non-backbone sensors increase significantly for sparse networks. This can be explained by the fact that in sparse networks, although backbone sensors are localized accurately, they might be distributed sparsely, hence their positions do not allow them to be appropriate beacons to localize non-backbone sensors.

After verifying our simulation implementation, we conducted several experiments to test the performance of the proposed protocol in the existence of errors in distance and angle measurements. We represented measured distances and angles as Gaussian distributed random variables. Figures 5(a) and 5(b) respectively show the average localization errors for backbone and non-backbone sensors under different network densities. The figures show that the localization error is almost independent from the underlying network density as long as the density is not too small with a slight increase in localization errors for backbone sensors as the row number increases. Localization errors for non-backbone sensors are very close to their values in the absence of distance error (around $3.5m$).

VI. Conclusions

We proposed a new localization protocol for WSN. In terms of number of anchors, our protocol is very flexible as it can work even with a single anchor as long as it can transmit directionally and omnidirectionally. Initially, the deployment area around the anchor is hypothetically tiled using identical hexagons. A ternary coordinate system is defined to distinguish between different hexagons. Each hexagon is associated with a single sensor (backbone sensor) which is chosen to be the closest sensor to the center of the hexagon. When backbone sensors are localized, non-backbone sensors can use them as beacons to localize themselves using the centroid approach. We simulated the proposed technique under different network densities in noisy and noise-free environments. Simulation results showed that the localization accuracy achieved by the protocol makes it appropriate for most sensor network applications. The proposed protocol has advantages over other localization protocols: it can work even with a single anchor, it provides an implicit clustering mechanism with backbone sensors as cluster heads, furthermore, it simplifies routing through its virtual ternary coordinate system.

References

[1] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 7(5):28–34, Oct 2000.

[2] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization schemes in large scale sensor networks, 2003.

[3] J. Jeong, S. Guo, T. He, and D. Du. Apl: Autonomous passive localization for wireless sensors deployed in road networks. *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 583–591, April 2008.

[4] S. Lederer, Y. Wang, and J. Gao. Connectivity-based localization of large scale sensor networks with complex shape. *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 789–797, April 2008.

[5] R. Nagpal, H. E. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *IPSN*, pages 333–348, 2003.

[6] D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 3:1734–1743 vol.3, 30 March-3 April 2003.

[7] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 1–14, New York, NY, USA, 2007. ACM.

[8] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, New York, NY, USA, 2000. ACM.

[9] K. Römer. The lighthouse location system for smart dust. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 15–30, New York, NY, USA, 2003. ACM.

[10] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM.