Range-Free Localization Using Expected Hop Progress in Wireless Sensor Networks

Yun Wang, *Member*, *IEEE*, Xiaodong Wang,
Demin Wang, *Member*, *IEEE*, and Dharma P. Agrawal, *Fellow*, *IEEE*

Abstract—Localization algorithm continues to be an important and challenging topic in today's wireless sensor networks (WSNs). In this paper, a novel range-free *localization algorithm using expected hop progress (LAEP)* to predict the location of any sensor in a WSN is proposed. This algorithm is based on an accurate analysis of hop progress in a WSN with randomly deployed sensors and arbitrary node density. By deriving the expected hop progress from a network model for WSNs in terms of network parameters, the distance between any pair of sensors can be accurately computed. Since the distance estimation is a key issue in localization systems for WSNs, the proposed range-free LAEP achieves better performance and less communication overhead as compared to some existent schemes like DV-Hop and RAW. In addition, we study the effect of anchor placement on the algorithm performance by deriving the corresponding mean position error range. Extensive simulations are performed and the results are observed to be in good agreement with the theoretical analysis.

Index Terms—Anchor placement, hop progress, range-free localization, sensor position, wireless sensor network.

1 Introduction

Wireless sensor network (WSN) is emerging as a key tool for various applications including search and rescue, disaster relief, target tracking, and smart environments due to its reliability, accuracy, cost effectiveness, and ease of deployment [1]. A WSN is a collection of small, cheap, and low-powered sensors which can dynamically form a network without any underlying infrastructure support [2]. The responsibilities of sensors are to sense data from the environment (e.g., temperature, light, etc), process the data, and finally transmit the data to the base station (or sink) directly or in multihop fashion.

The inherent characteristics of a WSN make sensors' locations an important part of their state and the network operations. In other words, most of the sensing data generated by the sensors in WSN applications has a common characteristic, i.e., *they are useful only when considered in the context of the location from where the data has been measured* [3]. Besides the typical application of correlating sensor readings with physical locations, approximate geographical localization is also needed for many WSN applications such as location-aided routing and data aggregation [4]. In fact, some routing protocols [5] exploit the location information of sensors to reduce the routing overhead.

- *Y. Wang is with the Department of Computer Science, Southern Illinois University, Edwardsville, IL 62026-1656. E-mail: yuwang@siue.edu.*
- *X. Wang is with the Office of the Chief Scientist of Qualcomm Inc., San Diego, CA 92121. E-mail: wangx@qualcomm.com.*
- *D. Wang is with Microsoft, One Microsoft Way, Redmond, WA 98052. E-mail: demin.wang@ieee.org.*
- *D.P. Agrawal is with the Center of Distributed and Mobile Computing, Department of Computer Science, University of Cincinnati, 833 Rhodes Hall, PO Box 210030, Cincinnati, OH 45221-0030. E-mail: dpa@cs.uc.edu.*

Manuscript received 12 July 2008; revised 23 Sept. 2008; accepted 15 Oct. 2008; published online 24 Oct. 2008.
Recommended for acceptance by A. Smailagic.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2007-07-0235.
Digital Object Identifier no. 10.1109/TPDS.2008.239.

Unfortunately, not every sensor in a WSN can be equipped with a range-based localization system like GPS to compute its location. Range-based localization system depends on absolute point-to-point distance estimates (range) or angle estimates to calculate the location [6]. It also requires expensive and energy-consuming electronics to satisfy the strict requirements on synchronization, i.e., GPS needs to precisely synchronize with a satellite's clock to perform positioning. These are against the basic characteristics of a sensor which should be small, light, and cheap.

In WSNs, positioning errors can often be masked by features associated with WSNs, including fault tolerance, node redundancy, data aggregation, and other means [7], [8]. This makes coarse accuracy of sensor localization sufficient for most WSN applications. Range-free localization is therefore pursued as a cost-effective alternative for expensive range-based scheme, by having no dependence on the availability or validity of hardware to provide range information [9]. The key idea is to place a small fraction of anchors with known coordinates across the network. Positioning of sensors is obtained from the estimated distance to multiple anchors and their coordinates, according to trilateration algorithm.

Many range-free localization algorithms have been proposed in the literature [9], [10], [11], [12], [13]. Most solutions are based on the methodology proposed in ad hoc positioning system (APS) [9], and APS is considered as a benchmark for range-free localization algorithms for WSNs. In APS, a distributed, hop-by-hop positioning algorithm is presented to provide approximate location information for all sensors in a WSN. This algorithm can be interpreted as an extension of both distance vector (DV) routing and GPS positioning. Although the APS is simple and useful, it can only provide acceptable location estimation if the node distribution in the network is dense and uniform. The performance and accuracy of APS deteriorates greatly if the node distribution is sparse or nonuniform.

The inefficiency of APS mainly comes from two perspectives. One is the separation of anchors' location information broadcasting and the following estimated average hop distance broadcasting. The other is the nonlocalized derivation of average hop distance, which is obtained from the global information of a WSN. The derivation can also generate larger-than-necessary errors if the network is not uniformly and densely deployed. In view of this, we propose a novel range-free localization algorithm to estimate the positions for all sensors in a WSN. The algorithm is based on an accurate analysis of hop progress in a WSN with randomly deployed sensors and arbitrary node density. The proposed algorithm is totally localized, where sensors make decisions solely based on the information available from their neighbors. In addition, the proposed localization algorithm can fulfill the task of broadcasting anchors' location information as well as the relative distance estimation to each sensor simultaneously.

In this paper, we introduce a *hop progress analytical model*, to estimate the optimal path distance between any pair of sensors in a WSN. More specifically, we derive the distance estimation between any two sensors in terms of expected hop progress and hop counts from a network model with arbitrary node density and sensor transmission range. Based on the analytical results, a *range-free localization algorithm using expected hop progress* (LAEP) is proposed and validated. The algorithm leads to a good position determination and reduced communication overhead for WSNs, as compared to some existent positioning schemes. The main contributions of this paper are as follows:

- We characterize the distribution of hop progress for a WSN with arbitrary network parameters, and prove that the expected hop progress is a function of node density, node connectivity, and transmission range. We also show that the distance estimation based on the expected hop progress provides satisfactory accuracy.
- We propose a novel range-free localization algorithm, called LAEP, with enhanced performance, reduced communication overhead, and decreased computing complexity as compared to existing range-free localization schemes, such as RAW and DV-Hop [9] in realistic network settings.
- We study the impact of anchor placement on the system performance of the proposed LAEP algorithm, and we show that a triangle-pattern-based anchor placement leads to better performance than square pattern placement, with fewer numbers of required anchors.

The remainder of this paper is organized as follows: Section 2 builds the network model. Section 3 analyzes the distribution of hop progress and derives the expected hop progress. Section 4 presents a range-free *LAEP* in a WSN. Section 5 illustrates the theoretical and simulation results. Section 6 presents some related works, and finally, this paper is concluded in Section 7.

2 Network Model

When a WSN is randomly deployed, we cannot assume any regularity in spacing or pattern of the *sensors.* This is due to the fact that most WSN deployments are performed through low flying airplanes or unmanned ground vehicles. However, *anchors* (i.e., special sensors with known positions) are generally arranged in the form of a regular tile across the network so as to help in estimating sensors' positions [3]. The formation of regular tiles can be triangle, square, and regular hexagon since only these forms can be repeated over a continuous field. In this paper, we take triangle and square anchor placement into account. It is because hexagon anchor placement can be seen a special case of triangle anchor placement, where six neighboring triangles constitute a hexagon.

Consider a WSN in a 2D plane with N **sensors**, donated by a set $\mathbf{N} = (n_1, n_2, \ldots, n_N)$, where n_i is the $i\text{th}$ sensor. All sensors are uniformly and independently deployed in a square area $A = L \times L$. Such a random deployment results in a 2D Poisson distribution of sensors with node density $\lambda = \frac{N}{L \times L}$ [14]. All sensors are assumed to be homogeneous, omnidirectional, and stationary. Therefore, the network can be seen as static or regarded as a snapshot of mobile ad hoc sensor networks.

Definition 1. *Let $Area(n_i, r_0)$ define the transmission range or the radio coverage area of sensor n_i, where sensor n_i is the center and r_0 is the transmission radius.*

The transmission coverage area $Area(n_i, r_0)$ of any sensor is assumed to be circular and symmetrical. Thus, $Area(n_i, r_0) = \pi r_0^2$. Any other sensor(s) located in this area can communicate directly with the sensor n_i, and is defined as its neighbor(s). Obviously, if a sensor n_i is the neighbor of sensor n_j, sensor n_j is also the neighbor of sensor n_i. It is because all the sensors are assumed to have the same transmission capability in the network model.

Definition 2. *Let $E(C)$ be the average number of sensors located in the transmission coverage area. The average connectivity, denoted by E_C, is defined as the average number of neighbors located in a sensor's transmission coverage area.*

Following Definition 1, the transmission coverage area is $Area(n_i, r_0) = \pi r_0^2$ for each sensor. The node density of the network is assumed as λ. Therefore, it has $E(C) = \lambda \pi r_0^2$ and $E_C = \lambda \pi r_0^2 - 1$.

Definition 3. *Anchor A_i is defined as a special sensor that is aware of its own location (X_i, Y_i), either through GPS or manual programming during deployment.*

In the network model, only a few anchors should be placed in a square or triangle pattern across the network area and act as location references. Figs. 1 and 2 show the corresponding local layout of network topology for a square and a triangle anchor-placement WSN, respectively. In both figures, the link between any two sensors represents that they are neighbors and can communicate directly.

This uniformly WSN model can also be extended to a WSN model with nonuniformly deployed sensors as illustrated in Figs. 3 and 4. In both figures, the node density for different region of the WSN is different while the node deployment in each small tile still conforms to Poisson distribution.

3 Hop Progress Analysis

3.1 Preliminary

In a densely distributed WSN, a shortest multihop path is likely to exist between any pair of the sensors. Fig. 5 depicts such a situation. By advancing one hop away from the source node, the accumulative distance is likely to be

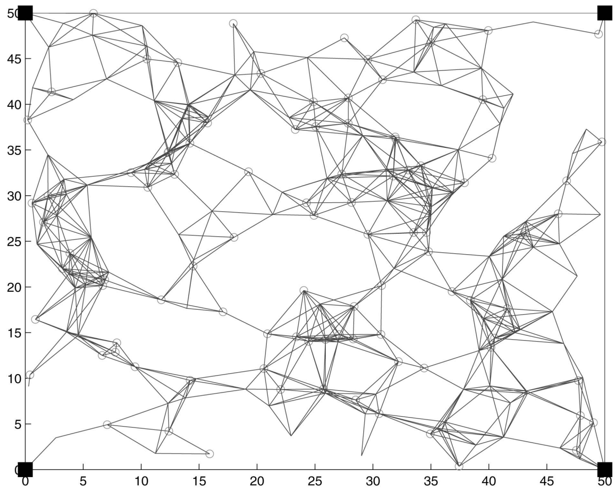

Fig. 1. Uniformly deployed WSN with anchors placed in square pattern.

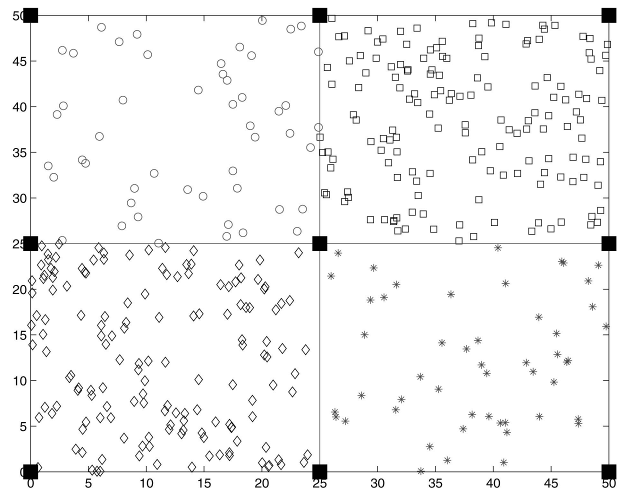

Fig. 3. Nonuniformly deployed WSN with anchors placed in square pattern.

increased by one transmission range. Therefore, if all the sensors have the same transmission range, the distance between any pair of the sensors (e.g., S/D) can be approximately estimated by the product of transmission range and the corresponding hop counts between them [12]. The following equation expresses such a relationship:

$$d = h * r_0, \tag{1}$$

where h is the hop counts between S and D. This distance estimation scheme is referred as RAW in the rest of this paper.

However, if the sensors are sparsely deployed as shown in Fig. 6, RAW is prone to introduce substantial inaccuracy in distance prediction. This is due to the fact that the node density in a sparsely deployed WSN is not adequate to construct a straight and shortest multihop path between sensors. In this case, for any intermediate sensor along the path, the probability that the next forwarding sensor is located close to the boundary of its transmission range is extremely low. The corresponding discrepancy between the actual hop progress and the transmission range would be appreciable. It makes the distance estimation by using RAW far from accurate.

In order to circumvent this problem, we propose to estimate the distance between any S/D pair in a WSN by using expected hop progress denoted by the following equation:

$$d = h * E(R), \tag{2}$$

where h is the hop counts between S and D, and $E(R)$ is the expected hop progress between neighboring sensors in a WSN. To derive the expected hop progress in an arbitrary WSN, we introduce a *hop progress analytical model* with arbitrary node density and uniform transmission range in the following discussion.

3.2 Hop Progress Formulation

It is of paramount importance to quantify the relationship between the path distance and the network parameters such as the sensor's transmission range and node density to

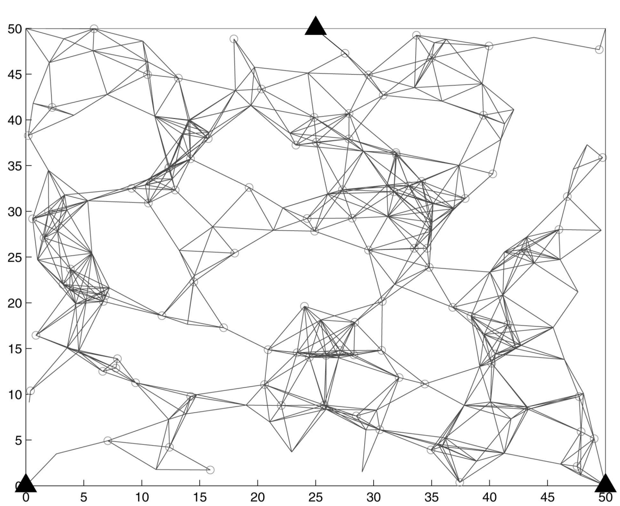

Fig. 2. Uniformly deployed WSN with anchors placed in triangle pattern.

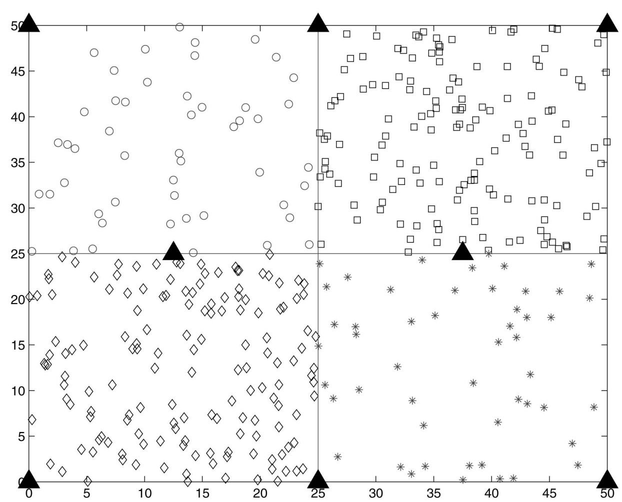

Fig. 4. Nonuniformly deployed WSN with anchors placed in triangle pattern.

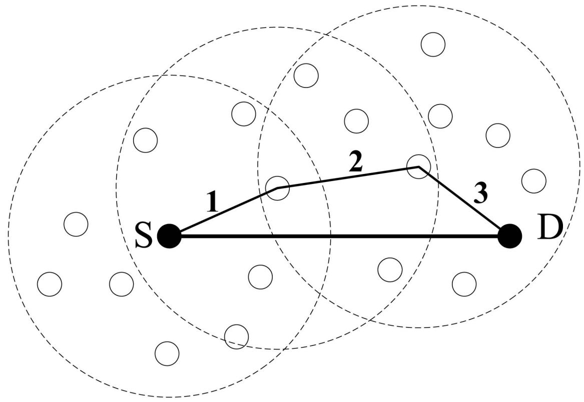

Fig. 5. Sparse WSN deployment.

characterize hop progress in a WSN. The path distance is defined as the length of the path from a source node to a destination node that has the shortest (minimum) length among all such paths [15]. Given a source node S, the hop progress toward the destination D at each hop is denoted as R, and R is a random variable. In 1D case, a node's next hop is always the farthest node on the line toward the destination node [17]. However, in the 2D case, the next hop might not be on the line to the destination. Fig. 7 shows such a situation.

For the convenience of the description and the analysis, let us define the following parameters:

- r_i: Hop distance from S to its next potential forwarding sensor n_i (e.g., r_1 and r_2 in Fig. 7).
- R_i: Hop progress from S to its next potential forwarding sensor n_i. It is the projection of r_i on the line connecting S and D (e.g., R_1 and R_2 in Fig. 7).
- θ_i: The angle between the line connecting S to its next forwarding sensor n_i and the line connecting S and D (e.g., θ_1 and θ_2 in Fig. 7).

From Fig. 7, it is clear that in S's transmission range, sensor n_1 is closer to the boundary of S's transmission range as compared with sensor n_2. However, the corresponding hop progress R_1 (from S to sensor n_1) is less than R_2 (from

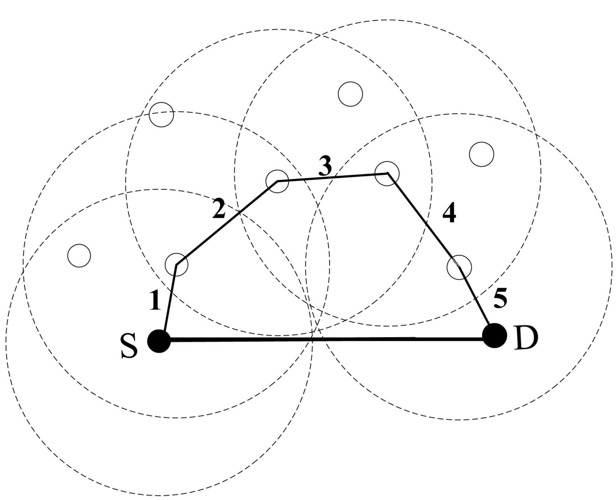

Fig. 6. Sparse WSN deployment.

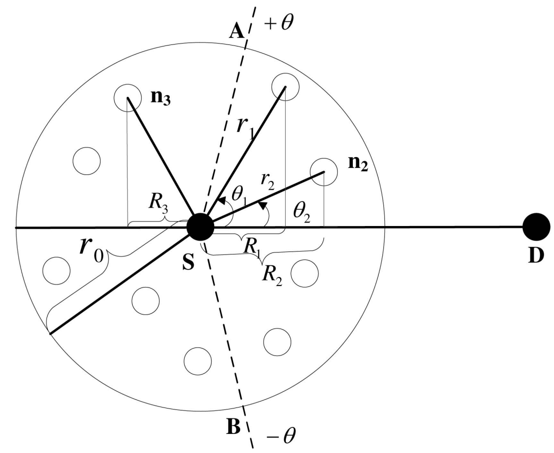

Fig. 7. Hop progress analysis.

S to sensor n_2), i.e., $R_2 > R_1$ while $r_2 < r_1$. Then, for S to communicate with D, sensor n_2 is preferred over sensor n_1 to be the next forwarding node. The relationship between hop distance and hop progress can be represented as $R_i = r_i * \cos\theta_i$.

It should be noted that for any intermediate sensor, only neighbors closer to the destination D than the current sensor are considered for forwarding. For instance, sensor n_3 is also located in the transmission range of S, but it is farther away from destination D than S, so it cannot be chosen by S as the next forwarding node to arrive D. This observation makes it necessary to determine the potential forwarding zone for any sensor along the path from S toward destination D.

3.3 Potential Forwarding Zone Determination

For simplicity of analysis, the potential forwarding zone from source S to destination D is assumed to be a cone area between dotted line $\overline{SA}$ and $\overline{SB}$, determined by $(-\theta, \theta)$ as illustrated in Fig. 7. In other word, θ determines the potential forwarding zone from S toward D.

In order to obtain a suitable size of θ, we draw two circles as shown in Fig. 8. One is the circle $C(S)$ with S as its center and transmission range r_0 as its radius. The other is the circle $C(D)$ centered at D with the euclidean distance d between S and D as its radius. These two circles intersect with each other at points A_1 and B_1. Obviously, the shadowed area in S's transmission radio range between $\overline{SA_1}$ and $\overline{SB_1}$ can be regarded as a good estimate for the potential forwarding zone from S to D. This is because any sensor located in this shaded region can be a potential forwarder for S, with less distance or closer to D. Let θ_s be the angle between line $\overline{SA_1}$ and line $\overline{SD}$, it conforms to the following equation:

$$\theta_s = \arccos\frac{r_0}{2d}. \tag{3}$$

Note that, with respect to sensor S, its next hop progress toward D can achieve the maximum value r_0, if there is a sensor located in P as shown in Fig. 8. In view of this, we draw a third circle $C(P)$ centered at P with radius r_0. $C(P)$ intersects with circle $C(S)$ at A_2 and B_2 and is illustrated in

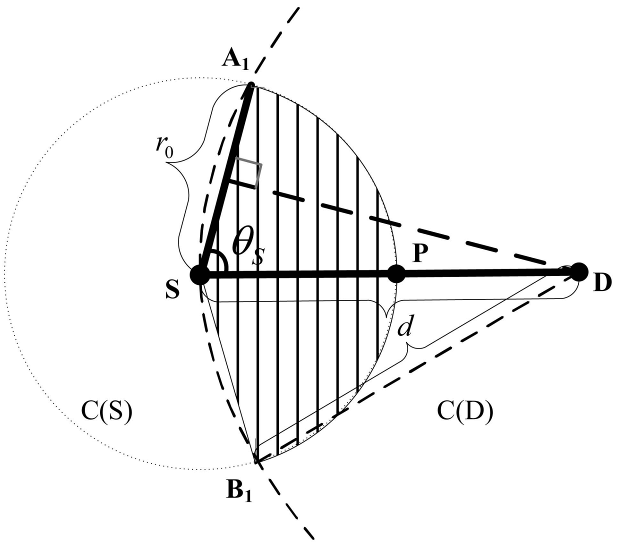

Fig. 8. Potential forwarding zone.

Fig. 9. The overlapped area of $C(S)$ and $C(P)$ can also be a good estimate of the potential forwarding zone for source S. The size of θ_p (i.e., $\angle A_2SP$) is determined by

$$\theta_p = \lim_{d \to r_0} \arccos\left(\frac{r_0}{2d}\right) = \arccos\left(\frac{1}{2}\right) = \frac{\pi}{3}. \tag{4}$$

Here, we choose θ_p over θ_s to determine the potential forwarding zone for S. The underlying rationality is illustrated in Fig. 10. In the figure, $\overline{SA_1}$, $\overline{SB_1}$, $\overline{SA_2}$, and $\overline{SB_2}$ divide S's transmission range into three zones in the forward direction to D. That is, z_1, z_2, and z_3, i.e., section determined by angle $\angle A_1SA_2$, $\angle A_2SB_2$, and $\angle B_2SB_1$, respectively. Clearly, angle θ_s determines $z_1 + z_2 + z_3$ and angle θ_p decides z_2. The three zones can be easily ranked according to their possible hop progress toward the destination D. For instance, the maximum hop progress in zone z_1 and z_3 is $\frac{r_0}{2}$ if there is a sensor located in positions A_2 and B_2, respectively. On the other hand, if there is any sensor located in the shadowed area of z_2, the resulting hop progress is larger

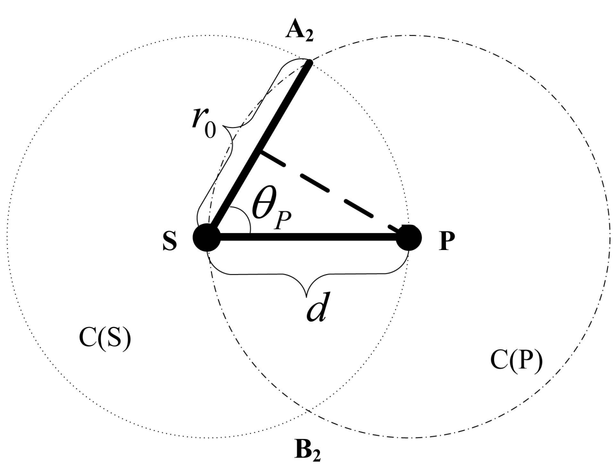

Fig. 9. Potential forwarding zone (continued).

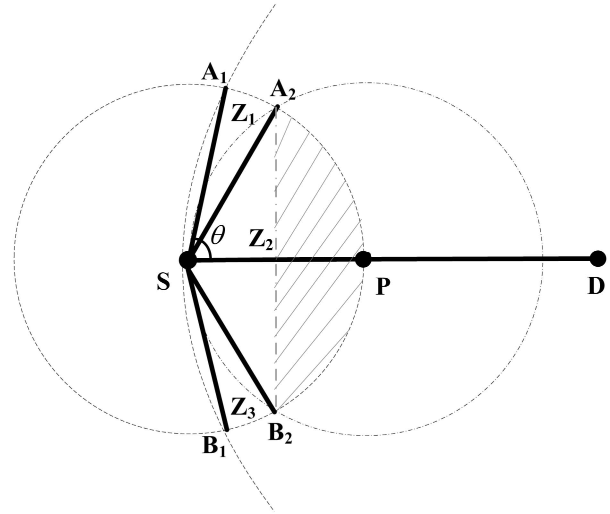

Fig. 10. Determination of potential forwarding zone.

than $\frac{r_0}{2}$. This observation implies that any sensor in the shadowed area of z_2 leads to a better hop progress than the maximum hop progress that can be achieved from either z_1 or z_3. Coupled with the high probability of at least one sensor located in the shadowed area of z_2 in a connected WSN, it is reasonable for us to assume that potential forwarding zone for S is z_2, giving the range of $\theta = (-\frac{\pi}{3}, \frac{\pi}{3})$. Note that S is an arbitrary sensor in a uniformly deployed WSN, the potential forwarding zone determination for S holds for any other sensor of the network.

3.4 Expected Hop Progress Derivation

In the network model described in Section 2, all sensors are deployed in a square area, conforming to the 2D Poisson distribution with node density $\lambda = \frac{A}{L \times L}$. Then, the probability of m sensors located within a sensor's transmission range $Area(n_i, r_0) = \pi r_0^2$ can be expressed as [16]

$$\mathrm{P}(m, r_0) = \frac{(\lambda \pi r_0^2)^m}{m!} e^{-\lambda \pi r_0^2}. \tag{5}$$

Let $P(m, \theta, r_0)$ be the probability of m sensors located in a section between $(-\theta, \theta)$ of a sensor's transmission range, and $E(m, \theta, r_0)$ be the corresponding average number of sensors located in the θ-section. Then, for any sensor, the probability that there are m sensors located in its potential forwarding zone can be given by $P(m, \theta, r_0) = \frac{(\lambda \theta r_0^2)^m}{m!} e^{-\lambda \theta r_0^2}$. The average number of sensors located in a sensor's potential forwarding zone can be expressed as $E(m, \theta, r_0) = \lambda \theta r_0^2$. In addition, the probability that m nodes located in a sensor's transmission range can be further expressed as $P(m, \pi, r_0) = \frac{(\lambda \pi r_0^2)^m}{m!} e^{-\lambda \pi r_0^2}$. Note that the average connectivity E_C for any sensor in the network model is defined as the average number of its neighbors. Then,

$$E_C = E(m, \pi, r_0) - 1 = \lambda \pi r_0^2 - 1. \tag{6}$$

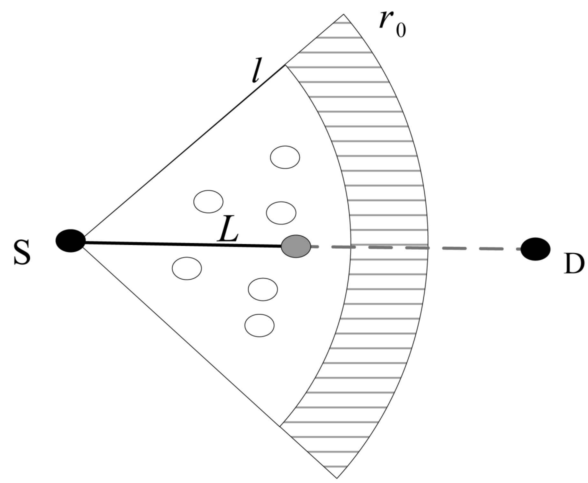

Fig. 11. Determination of potential forwarding zone (continued).

Fig. 11 depicts a snapshot of sensor deployment in a sensor's (i.e., S's) potential forwarding zone to D. Let L be the hop distance between S and its next forwarding sensor. L is a random variable [17], and the probability of L being less than ℓ can be given as

$$\begin{aligned} P[L \leq \ell] &= \sum_{m=0}^{N} P(m, \theta, r_0) P(m, \theta, \ell) \\ &= P[\texttt{no nodes in the shadowed } \theta\texttt{-sector} \\ &\qquad \texttt{area between } r_0 \texttt{ and } \ell] \\ &= e^{-\lambda\theta(r_0^2 - \ell^2)}. \end{aligned} \tag{7}$$

This is due to the fact that if there is no sensor in the shadowed θ-sector between r_0 and ℓ, the hop distance L cannot exceed ℓ, as illustrated in Fig. 11.

As (7) gives the cumulative density function (CDF) of L, the corresponding probability density function (PDF) [16] can be derived by

$$f_L(\ell) = \frac{d[P[L \leq \ell]]}{d\ell} = 2\lambda\theta\ell e^{-\lambda\theta(r_0^2 - \ell^2)}. \tag{8}$$

According to the definition of hop progress R, the projection of hop distance on the line connecting the source and the destination nodes, we get

$$R = L * \cos\omega, \tag{9}$$

where ω is the angle between the line connecting source and its next forwarding sensor toward destination, and the line connecting source and destination.

Then, the expected hop progress $E(R)$ [17] can be derived as follows:

$$\begin{aligned} E(R) &= \frac{\int\limits_0^\theta \int\limits_0^{r_0} f_L(\ell) \ell cos\omega d\omega d\ell}{\theta} \\ &= 2\lambda \sin(\theta) \int\limits_0^{r_0} \ell^2 e^{\left(-\lambda\theta r_0^2 + \lambda\theta\ell^2\right)} d\ell. \end{aligned} \tag{10}$$

By combining (10) with (6), we get

$$\begin{aligned} E(R) &= 2k\sin(\theta) \int\limits_0^{r_0} \ell^2 e^{\left(-k\theta(r_0^2 - \ell^2)\right)} d\ell \\ &= 2k\sin(\theta) \int\limits_0^{r_0} \ell^2 e^{\left(-\frac{E_C+1}{\pi}\theta(1 - \frac{\ell^2}{r_0{}^2})\right)} d\ell, \end{aligned} \tag{11}$$

where $k = \frac{E_C+1}{\pi r_0^2}$.

From (10), the expected hop progress can be derived if the node density and the transmission range are given. In addition, from (11), the expected hop progress can be computed if the expected node connectivity is available.

3.5 Integration of Power Management Protocol

Sensors are usually powered with embedded battery(s), which are limited and nonrechargeable in energy. This makes power management schemes greatly desirable and widely adopted in WSN applications. In other words, the sensor power should be put on/off periodically to save energy and enhance network lifetime in most of the WSN applications [14], [18], [19], [20]. In this paper, we also take a power management scheme into consideration, in terms of node availability rate.

As we have introduced in [14], the most basic prescheduled power management scheme can be implemented by a Random Independent Sleeping (RIS), in which the time is equally divided into cycles based on a time synchronization method. Each sensor needs to independently decide whether to awake or sleep at the beginning of each cycle. Namely, each sensor can choose to be active with probability p or go to sleep with probability $1 - p$ in each cycle. Thus, the network lifetime is increased by a factor up to $1/p$ [21]. Here, we incorporate this RIS scheme in our derivation of expected hop progress by introducing node availability rate in the network model. To be specific, we assume all sensors have the same availability probability p_a. It means that in every communication cycle, each sensor has the probability of p_a to transmit or forward, and the probability of $1 - p_a$ to be off or sleep. In this way, the energy consumption for each sensor can be reduced, and the network lifetime can be improved.

For calculating the average hop progress after introducing the node availability rate in the network model to provide network lifetime enhancement, we should note that if a Poisson stream with mean rate τ is split into k substreams such that the probability of a job going to be the $i\text{th}$ substream is p_i, each substream is also Poisson with a mean rate of $p_i\tau$ [14], [22]. In our network model, all sensors are randomly deployed and conform to a Poisson distribution. Based on the characteristics of Poisson distribution, our analysis above can be extended to incorporate power management scheme RIS with a node availability rate p_a, by replacing the previous node density λ with $p_a\lambda$ in the above derivation. That is,

$$\begin{aligned} E(R) &= \frac{\int\limits_0^\theta \int\limits_0^{r_0} f_L(\ell) \ell cos\omega d\omega d\ell}{\theta} \\ &= 2p_a\lambda \sin(\theta) \int\limits_0^{r_0} \ell^2 e^{\left(-p_a\lambda\theta r_0^2 + p_a\lambda\theta\ell^2\right)} d\ell. \end{aligned} \tag{12}$$

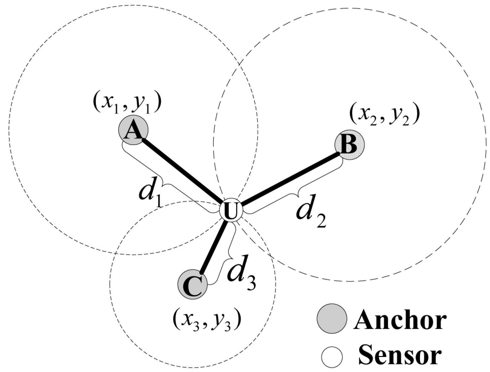

Fig. 12. Positioning by measuring distance to beacons.

It is clear that (12) can be reduced to (10) for $p_a = 1$.

Verification of the analysis in Section 3 will be presented later in Section 5.

4 Localization Algorithm Using Expected Hop Progress in WSNs

In this section, we first review the basic infrastructure of localization algorithm for positioning in WSNs. Then, we describe the range-free LAEP based on the analysis in Section 3.

4.1 Basic Trilateration Algorithm

Positioning of sensors in a WSN can be done by observing the properties of a triangle. The coordinates of any interior point in a triangle can be computed by knowing the coordinates of the three vertices and the corresponding distances to them [23]. More specifically, the location of any interior point in a triangle can be computed by calculating the intersection of three circles. These circles are determined by the vertices's coordinates that act as the centers, and their relative distances to the interior point that act as the radii. Fig. 12 shows such an situation, an interior node U in triangle $\triangle ABC$ is aware of the coordinates of A, B, and C, and the corresponding distances, denoted by (x_1, y_1, d_1), (x_2, y_2, d_2), and (x_3, y_3, d_3), respectively. Then, the location of U can be calculated by solving an array of nonlinear equations [6]: $(x - x_i)^2 + (y - y_i)^2 = d_i^2$ for $i = 1, 2, 3$. Further these equations can be reduced to linear ones by subtracting. It yields $2x(x_i - x_1) + 2y(y_i - y_1) = d_1^2 - d_i^2 + x_i^2 - x_1^2 + y_i^2 - y_1^2$. This linear system can be solved using standard methods such as the pseudoinverse [23].

Based on the above trilateration techniques, many GPS-free localization algorithms for positioning in WSNs have been proposed [9], [12], [3]. These mechanisms assume some *anchors* with known coordinates (e.g., nodes A, B, and C in Fig. 12) in the network as landmarks. Other *sensors* such as node U in Fig. 12 are designed to estimate their own locations by using trilateration algorithm, provided the required information such as coordinates and corresponding distance to at least three anchors. In other word, any sensor calculates its own location by employing the trilateration algorithm after acquiring the coordinates and corresponding distance to at least three anchors.

TABLE 1
Notations

Notation	Meaning
P_o	expected hop progress
n_i	ID of sensor n_i
A_i	ID of anchor A_i
N_a	number of anchors in the WSN
Γ_i	set of sensor n_i's neighbors
τ_i	number of sensor n_i's neighbors
$\sum_{P_c}^{i}$	current cumulative hop progress to A_i
$\sum_{H_c}^{i}$	current cumulative hop counts to A_i
$\sum_{P_r}^{i}$	received cumulative hop progress to A_i
$\sum_{H_r}^{i}$	received cumulative hop counts to A_i

4.2 LAEP Algorithm Description

We describe the range-free localization algorithm (LAEP), which is using the trilateration techniques and the expected hop progress analytical results in Section 3. In the stage of WSN deployment, each sensor is assumed to store a small database in its limited memory space. The database maps the relationship between node connectivity to the corresponding expected hop progress by using (11) or (12), which depends on whether or not a power management scheme such as RIS is adopted. The notations used in this section for describing LAEP are listed in Table 1.

In the initial stage of LAEP, each sensor in the WSN exchanges *hello* packets with its neighbors to obtain the local connectivity, i.e., the number of neighbors in its transmission coverage area. The exchanging of *hello* packets between neighboring sensors is limited in one hop communication, and no relaying of *hello* packets is allowed in the WSN. To be specific, each sensor maintains data structure Γ_i and τ_i, i.e., the set and number of neighbors, respectively, in its transmission range. This information can be obtained from the *hello* packets from its neighbors. For example, each *hello* packets contains the identification (ID) of the sender (e.g., n_j). Then, any sensor (e.g., n_i) in n_j's transmission range can get the *hello* packet. After that, n_i checks if it is a duplicate packet. If so, it just

Algorithm 4.1: INITIALIZER($Packet * p$)

comment: sensor n_i initializes itself.

for $k = 0$ **to** $k = N_a$
$$\begin{cases} \sum_{P_c}^{k} = 0; \sum_{H_c}^{k} = 0; \\ \Gamma_i = \emptyset; \tau_i = 0; \end{cases}$$
comment: updates by receiving p from n_j.

if $n_j \notin \Gamma_i$

then $\begin{cases} \Gamma_i = \Gamma_i \bigcup n_j; \\ \tau_i + +; \end{cases}$

else $Ignore(p)$;

comment: maps τ_i to expected hop progress P_o.

$P_o = f(\tau_i)$;

Fig. 13. Pseudocode for network initialization.

ignores the packet. Otherwise, it updates its local database by the process illustrated in Fig. 13.

After the network initialization, the process of *LAEP* starts from the broadcasting of anchors, which is the main paradigm of communication in WSNs [24]. In other word, each anchor (e.g., A_i) launches the algorithm by initiating a broadcast containing the information $Info = [(A_i, X_i, Y_i), \sum_{H_c}^i, \sum_{P_c}^i]$, where (X_i, Y_i) is the *anchor* A_i's known coordinate in a 2D space, $\sum_{H_c}^i$ and $\sum_{P_c}^i$ are the cumulative hop counts and cumulative hop progress to anchor A_i. Clearly, $\sum_{H_c}^i = 0$ and $\sum_{P_c}^i = 0$ in the beginning. After receiving the packet broadcasted from the anchor A_i, any sensor that is one-hop away from A_i records the anchor's information, updates its database, and then performs another broadcast to its neighbors. The new broadcast packet contains the anchor's constant information such as (A_i, X_i, Y_i) and the updated $info$ like $\sum_{P_c}^i = \sum_{P_r}^i + P_o$ and $\sum_{H_c}^i = \sum_{H_r}^i + 1$. This process of anchor's $info$ packet broadcasting continues until the packet arrives at any other anchor(s). In addition, each sensor keeps track of the number of anchors, whose $info$ packet has already arrived at least once, represented by a variable *Count*. Initially, *Count* is set to 0. The event of *Count* beyond 3 triggers the trilateration algorithm to compute a sensor's own location. The procedure for each sensor responding to the event of $info$ packet arriving is summarized as following: If a $info$ packet comes, the sensor first checks the packet's origin. If it is from a new anchor, it increases its local variable *Count* by 1, records the coordinate (A_i, X_i, Y_i), and updates $[\sum_{H_c}^i, \sum_{P_c}^i]$ by $\sum_{P_c}^i = \sum_{P_r}^i + P_o$ and $\sum_{H_c}^i = \sum_{H_r}^i + 1$, respectively. It then rebroadcasts the new $Info = [(A_i, X_i, Y_i), \sum_{H_c}^i, \sum_{P_c}^i]$ to its neighbors; Else if the sensor has already received the $info$ from the specific anchor, it then checks whether $\sum_{P_c}^i > \sum_{P_r}^i + P_o$ or not, if holds, it updates $[(A_i, X_i, Y_i), \sum_{H_c}^i, \sum_{P_c}^i]$ and rebroadcasts the new $Info$ as if the packet comes from a new anchor. Otherwise, it just ignores the packet. The pseudocode is shown in Fig. 14.

One thing worth mentioning is that, unlike the DV-Hop [9], which broadcasts the coordinates of anchors and the average hop distance between neighboring sensors in two different phases, and introduces more than necessary volume of traffic and overhead and increases the communication delay, our proposed LAEP broadcasts the anchor coordinates and the corresponding estimated distance to each sensor at the same time, i.e., in a single phase, therefore dramatically reducing the network traffic and the communication delay.

Algorithm 4.2: PACKETHANDLER($Packet * p$)

comment: p = $[(A_i, X_i, Y_i), \sum_{H_c}^i, \sum_{P_c}^i]$

$$\begin{array}{l}
\sum_{H_r}^i \leftarrow \sum_{H_c}^i; \\
\sum_{P_r}^i \leftarrow \sum_{P_c}^i; \\
\textbf{if } new(A_i) \\
\quad \textbf{then } \begin{cases} Record(A_i, X_i, Y_i); \\ Count++; \\ Update \begin{cases} \sum_{H_c}^i = \sum_{H_r}^i + 1; \\ \sum_{P_c}^i = \sum_{P_r}^i + P_o; \\ P\prime = [(A_i, X_i, Y_i), \sum_{H_c}^i, \sum_{P_c}^i]; \end{cases} \\ Rebroadcast(p\prime); \end{cases} \\
\quad \textbf{else if } (\sum_{P_c}^i > \sum_{P_r}^i + P_o) \\
\quad \textbf{then } \begin{cases} Update \begin{cases} \sum_{H_c}^i = \sum_{H_r}^i + 1; \\ \sum_{P_c}^i = \sum_{P_r}^i + P_o; \\ P\prime = [(A_i, X_i, Y_i), \sum_{H_c}^i, \sum_{P_c}^i]; \end{cases} \\ Rebroadcast(p\prime); \end{cases} \\
\quad \textbf{else } Ignore(p); \\
\textbf{if } (Count >= 3) \\
Trilateration();
\end{array}$$

Fig. 14. Pseudocode for sensor nodes.

5 Theoretical and Simulation Results

In this section, we present the outcome of theoretical and simulation experiments with two aims:

- To study the properties of hop progress in WSNs and verify the theoretical results by simulation outcomes. In particular, we show the interplay between expected hop progress and network parameters. We also illustrate the correctness of distance estimation for random S/D pairs by using expected hop progress.
- To evaluate the effectiveness of the proposed *LAEP* in WSN positioning system. We measure the performance in three metrics, i.e., the distance estimation error, position estimation error, and the mean error range. These results are compared with the same measures collected using algorithms RAW and DV-Hop.

5.1 Hop Progress Verification

Theoretical and simulation results on hop progress analysis are discussed in this section.

5.1.1 Theoretical Results

We use Matlab 6.5 to do the theoretical analysis. Network deployment area is set to be a square with length $L = 50$ m. The area of the deployment field is therefore $A = L \times L = 2{,}500\ \mathrm{m}^2$. A number of 50 to 1,000 sensors are to be deployed in the square area, with node density λ varying from $0.02/\mathrm{m}^2$ to $0.4/\mathrm{m}^2$. The transmission range r_0 of each sensor is changing from 8 m to 12 m. All sensors are uniformly deployed, and the resulting sensor deployment follows 2D Poisson distribution. Under such network settings, the results for the hop progress analysis in Section 3 are illustrated in Figs. 15 and 16, where the expected hop progresses are normalized to the transmission range r_0, i.e., by dividing the transmission range.

Fig. 15 compares the theoretical results of expected hop progress for different node densities. It shows that the expected hop progress improves when the node density increases, and it converges to about $0.8r_0$ when the node density goes to $0.4/\mathrm{m}^2$. It can also be observed from the figure that the expected hop progress is larger for a higher transmission range. This is because when the node density is fixed, the higher the transmission range, more sensors can

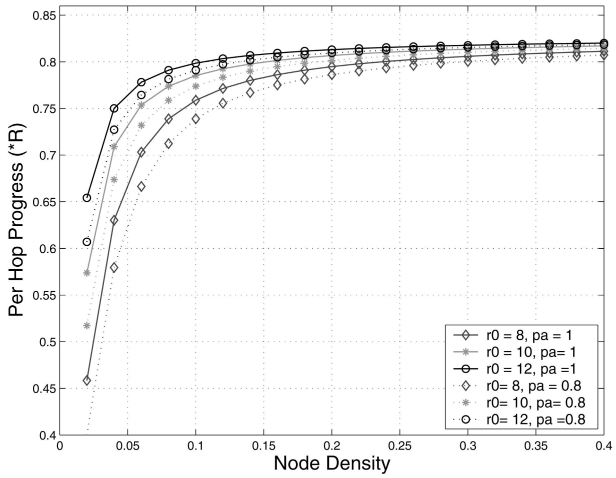

Fig. 15. Results on expected hop progress and node density.

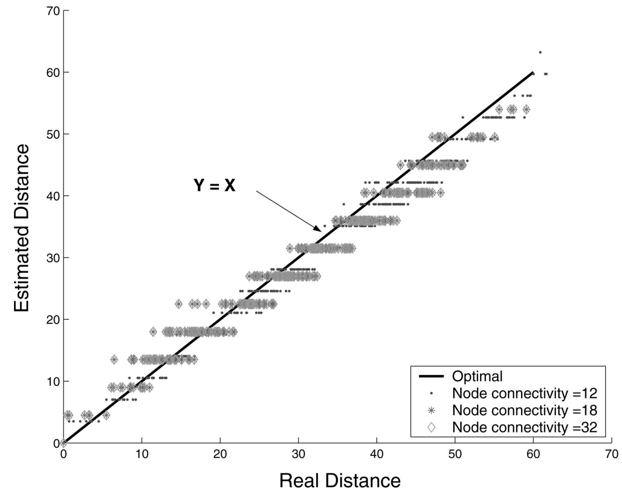

Fig. 17. Simulation results on distance estimation.

be covered in a sensor's transmission coverage area, resulting in the larger hop progress.

Due to the broadcast nature of wireless channel, each sensor can easily find out the number of neighbors within its transmission range, i.e., it can directly obtain its local connectivity. In view of this, Fig. 16 displays the results of the expected hop progress in terms of node connectivity from the same experiment depicted in Fig. 15. From Fig. 16, it is clear that the resulting curves for different network settings almost overlapped, i.e., the same connectivity leads to almost the same expected hop progress for all the cases. This substantiates our intuition that the expected hop progress mainly depends on the node connectivity. By taking advantage of this property, the distance between any pair of sensors in the network can be easily estimated if the hop counts between them are obtained, according to (2) and (11). In addition, Fig. 16 shows that the larger transmission range corresponds to longer curves. This is because, if the network deployment is fixed, the connectivity increases with the enlarging of transmission range.

In this experiment, we also show the effect of incorporating energy-saving scheme such as RIS on the expected hop progress by setting the node availability rate p_a. In the energy-saving state, each sensor keeps awake for 80 percent of a cycle (i.e., $p_a = 0.8$). While in the normal state, all sensors keep awake for 100 percent of a cycle (i.e., $p_a = 1.0$). The results in Figs. 15 and 16 show a trend that the expected hop progress drops if the waking time of the sensor is reduced (e.g., $0.8 < 1$). This is because the effective node density of the network drops with the introduction of RIS scheme. The results also illustrate that the effective node connectivity (i.e., number of waking sensors in a sensor's transmission range) leads to the same expected hop progress.

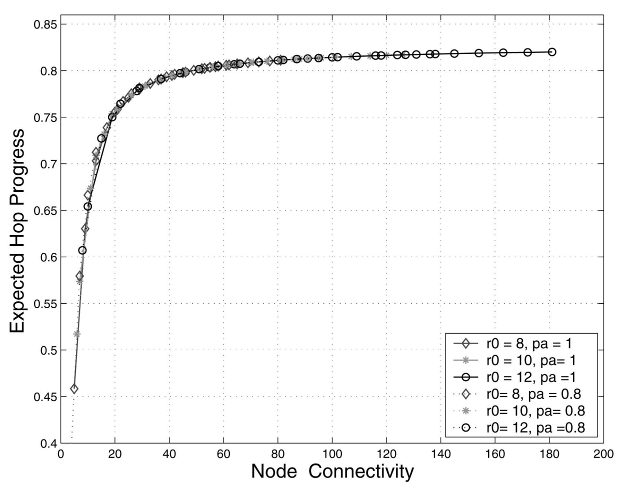

Fig. 16. Results on expected hop progress and node connectivity.

5.1.2 Simulation Results

A simulator using Microsoft Visual C++ 6.0 is built to verify the accuracy of the theoretical results. In order to facilitate the simulation, we randomly distribute $N = 400$ sensors in an area of $A = 50 \times 50 \text{ m}^2$, admitting the 2D Poisson distribution. We vary the average node connectivity from 12 to 32 by increasing the transmission range r_0 accordingly. An arbitrary sensor in the network is chosen as the source S, and all other sensors act as the destinations D. In other words, we are going to evaluate the distance estimation among 399 S/D pairs under the given network settings, by using the proposed expected hop progress analysis in Section 3.

Fig. 17 compares the estimated distance with real distance among all the S/D pairs. Real distance is defined as the euclidean distance between any S/D pair, while the estimated distance is obtained from the analysis of hop progress, i.e., $d = h * E(R)$ by combining (2) and (11). All the results shown in the figure are averaged over 50 trials. It is obvious that most of the estimated results are very close to the real distance. For brevity of comparison, we show the optimal line defined by $Y = X$ as reference, where Y denotes the estimated distance between each S/D pair and X denotes the corresponding real distance between them. Fig. 17 validates our hop progress analysis. It also illustrates the effectiveness of distance estimation using expected hop progress.

5.2 LAEP Algorithm Evaluation

We evaluate the performance of the proposed LAEP by building a simulator in Microsoft Visual C++ 6.0. We compare the performance of the proposed LAEP, baseline

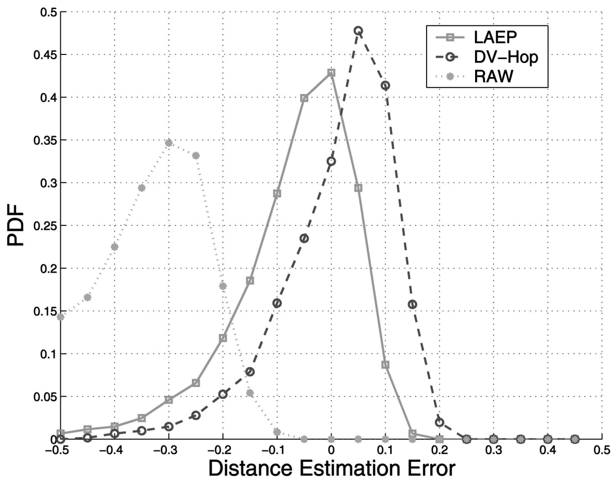

Fig. 18. Distribution of distance estimation error.

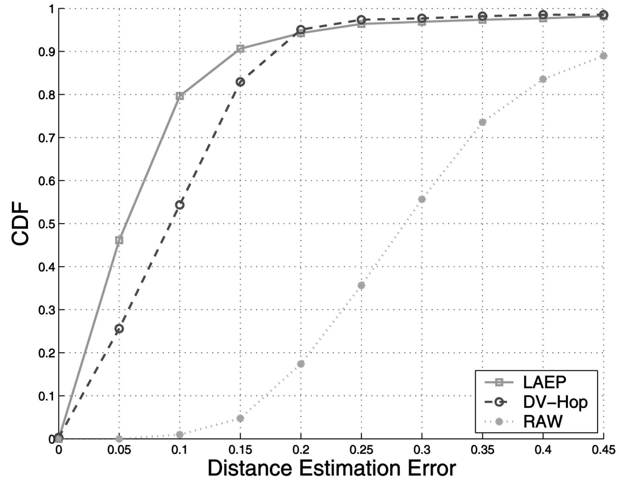

Fig. 19. Distribution of distance estimation error (continued).

DV-Hop, and the corresponding RAW algorithm under the same network settings in terms of three well-defined metrics. In addition, we examine the impact of anchor placement on the proposed LAEP scheme.

5.2.1 Evaluation Metrics Definitions

The definition of the three metrics are listed as follows:

- *Distance estimation error.* It represents the distance estimation error with respect to the real distance and is expressed as

$$E_D^{ij} = \frac{\left\| D_{est}^{ij} - D_{real}^{ij} \right\|}{D_{real}^{ij}}, \tag{13}$$

  where D_{est}^{ij} is the estimated distance between sensor n_i and anchor A_j, and D_{real}^{ij} is the real distance between them.
- *Position estimation error.* It represents the position estimation deviation with respect to transmission range and can be given by

$$E_R^i = \frac{\sqrt{\left(x_{est}^i - x_{real}^i\right)^2 + \left(y_{est}^i - y_{real}^i\right)^2}}{r_0}, \tag{14}$$

  where (x_{est}^i, y_{est}^i) and (x_{real}^i, y_{real}^i) are sensor n_i's estimated coordinates and real coordinates, respectively. The errors of position estimates are normalized to r_0 (i.e., 50 percent position errors means half of the transmission range).
- *Mean error range.* It represents the average of the accumulative sensor position estimation deviation relative to transmission range and can be given as

$$E(E_R) = \frac{\sum_{i=1}^{N} E_R^i}{N}, \tag{15}$$

  where N is the total number of sensors in the WSN.

5.2.2 Simulation Results

In the simulation, a number of 200 sensors are deployed in a square area conforming to a 2D Poisson distribution. The sensor's transmission range is set as 10 m and the network deployment area is set as $A = 50 * 50\ \mathrm{m}^2$. Whenever necessary, we vary the number of deployed sensors to get different node density and connectivity. The simulation results shown here are averaging over 100 trials.

Figs. 18 and 19 illustrate the PDF and CDF of distance estimation error and compares the accuracy of distance estimation among LAEP, DV-Hop, and RAW, respectively. Fig. 18 shows that the probabilities of exact distance estimation, i.e., estimation error equal to 0, by using algorithm LAEP, DV-Hop, and RAW are 0.425, 0.325, and 0, respectively. Fig. 19 shows that the probabilities of distance estimation deviation in the interval of $\mp 0.05 D_{real}$ are 0.48, 0.25, and 0.01 by using algorithm LAEP, DV-Hop, and RAW, respectively. The probabilities of distance estimation deviation in the interval of $\mp 0.1 D_{real}$ are 0.8, 0.6, and 0.02. It is clear that both LAEP and DV-Hop algorithms show better performance than the RAW scheme, and LAEP improves the accuracy of distance estimation further as compared to the baseline DV-Hop.

All the three localization algorithms take advantage of the distance estimation to anchors for estimating sensors' positions in a WSN. It is no doubt that more accurate distance measurement leads to better position estimation. Fig. 20 illustrates the simulation results on the position

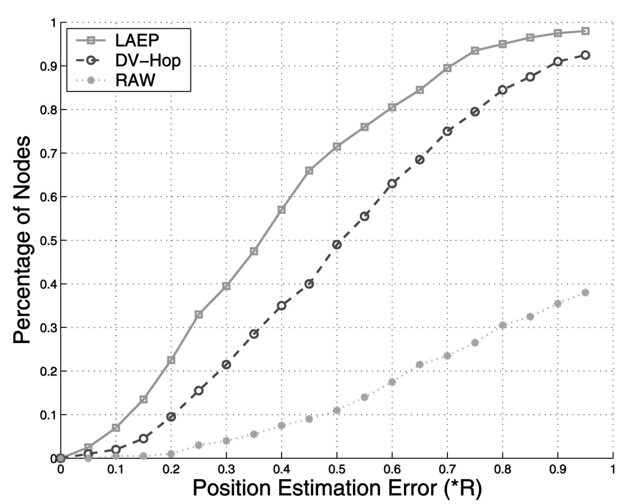

Fig. 20. Distribution of position error.

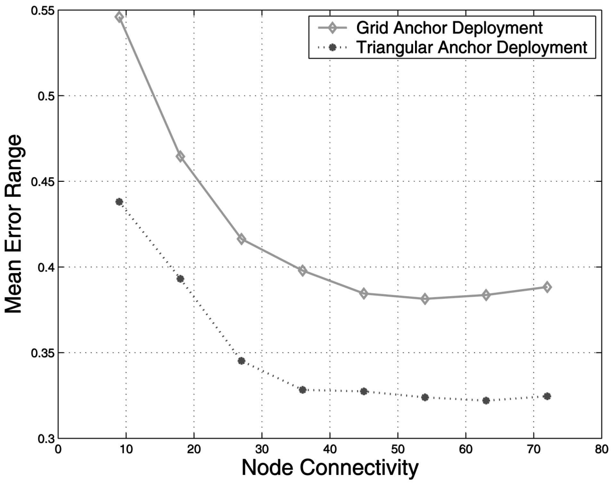

Fig. 21. Error range as a function of node connectivity and anchor placement.

estimation accuracy in terms of position estimation error. It shows that by using LAEP, over 90 percent of the sensors could estimate their positions with deviation in one transmission range from the actual locations, while only 38 percent of the sensors could be located in the same accuracy by using RAW. $(90 \text{ percent} >> 38 \text{ percent})$ illustrates that LAEP performs significantly better than RAW. In addition, using LAEP, over 72 percent sensors can manage to estimate their locations within half the transmission range deviation from their real locations. Comparatively, using DV-Hop, only around 49 percent of the sensors can achieve the same performance. $(72 \text{ percent} > 49 \text{ percent})$ demonstrates that the performance of the proposed LAEP is much better than DV-Hop.

Next, we study the impact of *triangle* and *square* anchor placements on the performance of LAEP. Here, we compare the two anchor placement strategies under the same network parameters in terms of mean error range (i.e., mean position estimation error). We vary the number of sensors enclosed in the region of interest from 50 to 400, then the network parameters such as node connectivity changes accordingly. Fig. 21 shows the mean error range under different node connectivity for *triangle* and *square* anchor placement strategies. It is clear that the mean error range decreases as node connectivity increases in both anchor placement cases. This is due to the fact that the higher the node connectivity, the more the sensors could be able to construct a straight and shortest multihop path to the anchors, the more accurate the position estimation can be. In addition, the results demonstrate that the mean error range of the triangle anchor placement is less than that of the square anchor placement under the same network settings even with fewer anchors. Then, we can conclude that the placement of anchors is essential to the localization system in WSNs, and triangle anchor placement is better than the square anchor placement in a randomly deployed WSN with regularly placed anchors.

6 Related Works

To provide location information for sensors in WSNs, a number of localization algorithms have been proposed in the literature [9], [10], [11], [12], [13], [25]. With regard to the mechanisms used in location determination, the localization schemes can be divided into two categories: range-free schemes and range-based schemes [26].

Range-based localization schemes use point-to-point distance or angle information to calculate locations [27]. Time of Arrival (TOA) [10], Time Difference of Arrival (TDOA) [11], Angle of Arrival (AOA) [28], and Received Signal Strength (RSS) [29] are the common techniques used in this area. The most basic range-based localization system is GPS, which uses a constellation of 24 satellites as reference nodes in conjunction with ground stations to provide positioning services for the ground nodes [2]. However, attaching a GPS receiver to each sensor in WSNs is prohibitively expensive because of the large volume of sensors [9]. In addition, the positioning error can be of the order of $10 \sim 20$ m for GPS system, which can be much larger than the hop size of some WSNs and therefore render the GPS location information useless. Finally, the size constraints of a sensor node such as Rene board [30] makes it difficult to integrate with GPS. All of these constraints of sensors create an essential demand for efficient and cost-effective range-free localization algorithms in WSNs.

Range-free localization algorithms are cost-effective due to the fact that no distance/angle measurement between communicating nodes is required. In a WSN, errors can be masked by its underlining fault-tolerant operation, redundancy computation, and data aggregation [7]. As far as the tradeoff between location precision and deployment cost is concerned, range-free localization schemes are more suitable for WSNs than expensive range-based ones.

In APS [9], a distributed, hop-by-hop positioning algorithm (i.e., DV-Hop) has been proposed. This can be interpreted as an extension of both DV routing and GPS positioning so as to provide approximate location information for all nodes in ad hoc networks [9]. DV-Hop introduces a lot of traffic load and communication delay in a WSN, due to the fact that it works in two steps. First, anchors broadcast their location information throughout the network and each sensor keeps track of the hop counts to each of the anchors. Second, any anchor estimates the average hop distance between neighboring sensors after it gets the distances to other anchors. Then, it broadcasts the corresponding average hop distance throughout network. After the two-step broadcasting, each sensor can estimate its distance to each anchor (i.e., the product of average hop distance and corresponding hop counts), and perform the trilateration algorithm to get its estimated location. Although the DV-Hop is simple and useful, it can only provide good position estimation if the node distribution in the network is dense and uniform. The performance and accuracy of DV-Hop deteriorates if the node distribution is sparse or nonuniform. In [12], a density-aware hop-count localization algorithm has been proposed that gives a better performance than DV-Hop under the case of nonuniform node distribution. They introduced a range ratio to adjust the accumulated hop counts and reduce the distance overestimation. However, they only relied on simulations to determine the range ratios and the classification is too coarse. In [31], a GPS-free positioning algorithm has been proposed to define and compute relative positions of nodes in an ad hoc network, by building a relative coordinate system in which the node positions are computed in two dimensions. In [32], a simple centroid model is built to estimate sensor's position by using proximity information. This work introduces low overhead but high inaccuracy as compared to others. On the contrary, a key advantage of our approach over previous methods is the utilization of connectivity-based expected hop progress analysis in determining the locations of all the sensors in a WSN. This leads the proposed LAEP to achieve better performance with reduced traffic load and communication delay.

7 CONCLUSION

For sensed data to be meaningful in a WSN, an estimation of the physical locations for sensors is critical. In this paper, a range-free LAEP has been proposed to compute the locations of sensors. The proposed LAEP algorithm soundly outperforms existing localization schemes such as DV-Hop and RAW in terms of accuracy and efficiency. Furthermore, the effect of square and triangle anchor placement on the algorithm performance is examined, and triangle anchor placement is shown to have better performance with fewer anchor requirements.

ACKNOWLEDGMENTS

This work has been supported by the Ohio Board of Regents Doctoral Enhancements Funds and the US National Science Foundation under Grant BES-0529063.

REFERENCES

[1] S. Tilak, N.B. Abu-Ghazaleh, and W. Heinzelman, "A Taxonomy of Wireless Micro-Sensor Network Models," *ACM Mobile Computing and Comm. Rev.*, vol. 6, no. 2, Apr. 2002.

[2] D.P. Agrawal and Q.-A. Zeng, *Introduction to Wireless and Mobile Systems.* Brooks/Cole Publishing, Aug. 2003.

[3] X. Ji and H. Zha, "Sensor Positioning in Wireless Ad-Hoc Sensor Networks Using Multidimensional Scaling," *Proc. IEEE INFOCOM,* 2004.

[4] H. Chan, M. Luk, and A. Perrig, "Using Clustering Information for Sensor Network Localization," *Proc. First IEEE Int'l Conf. Distributed Computing in Sensor Systems (DCOSS),* 2005.

[5] Y.B. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Proc. ACM MobiCom '98,* pp. 66-75, 1998.

[6] B.H. Wellenhoff, H. Lichtenegger, and J. Collins, *Global Positions System: Theory and Practice.* Springer Verlag, 1997.

[7] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proc. ACM MobiCom,* 2003.

[8] Y. Wang, X. Wang, D. Wang, and D.P. Agrawal, "Localization Algorithm Using Expected Hop Progress in Wireless Sensor Networks," *Proc Third IEEE Int'l Conf. Mobile Ad Hoc and Sensor Systems (MASS '06),* Oct. 2006.

[9] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '01),* Nov. 2001.

[10] N. Patwari, A.O. Hero, M. Perkins, N.S. Correal, and R.J. ODea, "Relative Location Estimation in Wireless Sensor Networks," *IEEE Trans. Signal Processing,* vol. 51, no. 8, pp. 2137-2148, Aug. 2003.

[11] Vossiek, M. Wiebking, L. Gulden, P. Wieghardt, J. Hoffmann, and C. Heide, "Wireless Local Positioning," *IEEE Microwave Magazine,* vol. 4, no. 4, pp. 77-86, 2003.

[12] S.Y. Wong, J.G. Lim, S. Rao, and W.K. Seah, "Density-Aware Hop-Count Localization in Wireless Sensor Networks with Variable Density," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '05),* Mar. 2005.

[13] A. Nasipuri and K. Li, "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks," *Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '02),* pp. 105-111, Apr. 2002.

[14] Y. Wang, X. Wang, B. Xie, D. Wang, and D.P. Agrawal, "Intrusion Detection in Homogenous and Heterogeneous Wireless Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 7, no. 6, 2008.

[15] K.A. Berman and J. Paul, *Fundamentals of Sequential and Parallel Algorithms.* PWS, 1996.

[16] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering,* second ed. Addison-Wesley, July 1993.

[17] X. Wang, "QoS Issues and QoS Constrained Design of Wireless Sensor Network," PhD dissertation, Univ. of Cincinnati, 2006.

[18] C. Gui and P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks," *Proc. ACM MobiCom,* 2004.

[19] A. Ephremides, "Energy Concerns in Wireless Networks," *IEEE Wireless Comm.,* vol. 9, no. 4, pp. 48-59, Aug. 2002.

[20] L. Wang and Y. Xiao, "A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks," *Mobile Network Applications,* vol. 11, no. 5, pp. 723-740, 2006.

[21] S. Kumar, T.H. Lai, and J. Balogh, "On k-Coverage in a Mostly Sleeping Sensor Network," *Proc. ACM MobiCom,* 2004.

[22] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling.* Wiley-Interscience, 1991.

[23] D. Niculescu and B. Nath, "Ad hoc Positioning System (APS) Using AOA," *Proc. IEEE INFOCOM,* 2003.

[24] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Wireless Sensor Networks," *IEEE Comm. Magazine,* vol. 40, no. 8, pp. 102-114, Aug. 2002.

[25] L. Doherty, L. Ghaoui, and K. Pister, "Convex Position Estimation in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* 2001.

[26] M. Rudafshani and S. Datta, "Localization in Wireless Sensor Networks," *Proc. Sixth IEEE/ACM Int'l Conf. Information Processing in Sensor Networks (IPSN),* 2007.

[27] L. Fang, W. Du, and P. Ning, "An Anchor-Less Location Discovery Scheme for Wireless Sensor Networks," *Proc. IEEE INFOCOM,* 2005.

[28] J. Yli-Hietanen, K.K. Vi, and J. Astola, "Low-Complexity Angle of Arrival Estimation of Wideband Signals Using Small Arrays," *Proc. Eighth IEEE Signal Processing Workshop Statistical Signal and Array Processing (SSAP '96),* p. 109, 1996.

[29] P. Bahl and V. Padmanabhan, "Radar: An In-Building rf-Based User Location and Tracking System," *Proc. IEEE INFOCOM,* 2000.

[30] B. Parkinson and J. Spilker, *Global Positioning System: Theory and Application.* Am. Inst. Aeronautics and Astronautics, 1996.

[31] S. Capkun, M. Hamdi, and J.-P. Hubeaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," *Cluster Computing,* vol. 5, no. 2, pp. 157-167, Apr. 2002.

[32] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm. Magazine,* special issue on Smart Spaces and Environments, 2000.

Yun Wang received the BS degree in computer science and engineering from Wuhan University, Hubei, China, in 2001 and the PhD degree in computer science and engineering from the University of Cincinnati, Ohio, in 2008. She is currently an assistant professor in the Department of Computer Science, Southern Illinois University, Edwardsville, Illinois. Her research interests include network modeling and analysis, performance evaluation and optimization, wireless ad hoc and sensor networks, wireless LAN, and wireless mesh networks. She is a member of the IEEE.

Xiaodong Wang received the BS degree in communication engineering in 1995, the MS degree in electric engineering in 1998, and the PhD degree in computer engineering from the University of Cincinnati in 2006. He has been working on system design, prototyping, standardization on various wireless systems in the industry, including China Telecom, Lucent Technologies, Motorola, and Qualcomm. His research interests include multimedia over wireless, performance analysis of wireless MAC protocols, QoS issues in wireless sensor networks, and cross-layer design of wireless ad hoc networks. He joined the office of chief scientist of Qualcomm in 2008.

Demin Wang received the BS degree in computer science and the MS degree in safety technology and engineering from the University of Science and Technology of China, Hefei, China, in 2000 and 2003, respectively. He received the PhD degree in computer science and engineering from the University of Cincinnati, Cincinnati, Ohio, in 2008. He joined Microsoft in 2009. His research interests include coverage and energy problems in wireless sensor networks and implementation of wireless sensor networks and wireless mesh networks. He is a member of the IEEE and the IEEE Computer Society.

Dharma P. Agrawal is the Ohio Board of Regents distinguished professor of computer science and the founding director for the Center for Distributed and Mobile Computing, Department of Computer Science, University of Cincinnati, Cincinnati, Ohio. He was a visiting professor of electrical and computer engineering at the Carnegie Mellon University, on sabbatical leave during the Autumn 2006 and Winter 2007 Quarters. He was a faculty member at the North Carolina State University, Raleigh (1982-1998) and the Wayne State University, Detroit (1977-1982). His recent research interests include resource allocation and security in mesh networks, efficient query processing and security in sensor networks, and heterogeneous wireless networks. His coauthored introductory textbook on wireless and mobile computing has been widely accepted throughout the world and a second edition was published in 2006. The book has been reprinted both in China and India and translated in to Korean and Chinese languages. His second coauthored book on ad hoc and sensor networks published in Spring 2006 has been named as a bestseller by the publisher. He has given tutorials and extensive training courses in various conferences in the US, and numerous institutions in Taiwan, Korea, Jordan, Malaysia, and India in the areas of ad hoc and sensor networks and mesh networks. He is an editor for the *Journal of Parallel and Distributed Systems*, the *International Journal on Distributed Sensor Networks*, the *International Journal of Ad Hoc and Ubiquitous Computing* (IJAHUC), and the *International Journal of Ad Hoc and Sensor Wireless Networks.* He has served as an editor of *IEEE Computer Magazine*, the *IEEE Transactions on Computers*, and the *International Journal of High Speed Computing.* He has been a program chair and a general chair for many international conferences and meetings. He has received numerous certificates and awards from the IEEE Computer Society. He was awarded a "Third Millennium Medal" by the IEEE for his outstanding contributions. He has also delivered keynote speeches for five international conferences. He also has five patents in the wireless networking area. He has also been named as an ISI Highly Cited Researcher in Computer Science. He is a fellow of the IEEE, the ACM, the AAAS, and the WIF.