Novel Self-Configurable Positioning Technique for Multihop Wireless Networks

Hongyi Wu, *Member, IEEE*, Chong Wang, *Student Member, IEEE*, and Nian-Feng Tzeng, *Senior Member, IEEE*

***Abstract*—Geographic location information can effectively improve the performance (e.g., in routing or intelligent coordination) of large wireless networks. In this paper, we propose a novel self-configurable positioning technique for multihop wireless networks, based on a Euclidean distance estimation model and a coordinates establishment scheme. A number of nodes serve as the landmarks to establish a coordinates system. Specifically, any pair of landmarks estimate their Euclidean distance according to the shortest path length between them and establish the coordinates system by minimizing an error objective function. Other nodes in the network can accordingly contact the landmarks and determine their own coordinates. The proposed technique is independent of the Global Navigation Satellite Systems (GNSSs), and the established coordinates can be easily tuned to GNSS if at least one node in the network is equipped with GNSS receiver. Our simulation results show that the proposed self-configurable positioning technique is highly fault-tolerable to measurement inaccuracy and can effectively establish the coordinates for multihop wireless networks. More landmarks yield more accurate results. With the rectification of our Euclidean distance estimation model, four to seven landmarks are usually sufficient to meet the accuracy requirement in a network with hundreds of nodes. The computing time for coordinates establishment is in the order of milliseconds for a GHz CPU, acceptable for most applications in the mobile ad hoc networks as well as the sensor networks.**

***Index Terms*—GPS-free, positioning techniques, self-configurable, wireless networks.**

I. INTRODUCTION

THE packet switching wireless networks are experiencing tremendous growth with the advances in integrated circuits and radio technologies. Due to limited bandwidth and error-prone channels compared with their wired counterparts, however, the wireless networks face several major challenges in the protocol design in various layers. Extensive studies have shown that the geographic location information of wireless nodes can effectively improve the performance of large wireless networks. For example, several location based routing protocols (such as [1], [2], [3]) have been proposed for mobile ad hoc networks [4], [5] to reduce routing overhead and to improve system throughput. Additionally, it is crucial for the nodes in large-scale sensor networks to locate themselves for intelligent coordination, data collection, and energy-efficient routing (see [6] and [7] for examples).

A number of Global Navigation Satellite Systems (GNSSs), such as the Global Positioning System (GPS), the GLObal NAvigation Satellite System (GLONASS), and the upcoming Galileo system, as well as several terrestrial-based systems such as the Signpost Navigation System and the Cellular Geolocation System, have evolved over the years to provide location information for outdoor mobile users [8]–[10]. On the other hand, several indoor positioning systems [11], [12] have been developed recently to provide location information in places where the signals from satellite (or other outdoor systems) are not available. All of these systems rely on existing infrastructure (e.g., satellites, cellular base stations, or pre-deployed anchor nodes) and normally need special user hardware. For example, the GPS system requires each user to equip with a GPS receiver, which may not be available (or affordable) by all nodes in a wireless network. Moreover, the GPS reception might be obstructed by buildings (e.g., for indoor sensor networks) or climate conditions. In fact, *global* positioning information is certainly desirable but not necessary for all nodes, in order to effectively assist the communication in a multihop self-configurable wireless network. For instance, a relative *local* coordinates system agreed by all nodes in the network will provide sufficient information to support location based routing protocols.

The objective of this paper is to establish a local positioning system *within* a wireless network with the following features:

- *self-configurability*: the local positioning system should work based on the coordination of the nodes inside the wireless network, without any assistance from other infrastructure;
- *independence*: the local positioning system should be independent of other global positioning systems. Based on the local coordinates, however, the wireless nodes can also tune to the global coordinates if there is at least one node in the network equipped with the GNSS receiver;
- *robustness*: the positioning technique should tolerate possible measurement inaccuracy, e.g., in the estimation of distance between adjacent nodes (within the transmission range of each other) that is usually needed to calculate the coordinates;
- *high accuracy*: the system should provide location information that is accurate enough to support target applications (e.g., location-based routing in sensor networks or mobile ad hoc networks).

We propose a self-configurable positioning technique that is built upon two models. First, for a given node distribution

Manuscript received November 1, 2003; revised May 21, 2004, approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Ajmone Marsan. This work is supported in part by the National Science Foundation under CAREER Award CNS-0347686, by the U.S. Department of Energy (DoE) under Award Number DE-FG02-04ER46136, and by the Board of Regents, State of Louisiana, under Contract Number DOE/LEQSF(2004-07)-ULL.

The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70504 (e-mail: wu@cacs.louisiana.edu).

Digital Object Identifier 10.1109/TNET.2005.850202

(e.g., a uniform distribution), the Euclidean distance between two nodes (usually multiple hops away) is estimated according to the length of the shortest path obtained by sending a control packet. Second, a number of stable nodes are selected to serve as landmarks. Every landmark estimates its distance to other landmarks and exchanges obtained distance information with each other. Once a landmark has accumulated a full set of distances between any two landmarks in the network, it may start establishing the coordinates system. More specifically, the landmarks calculate the coordinates by minimizing an error objective function, which signifies the errors between the actual distance and the distance in the established coordinates system. Other nodes in the network calculate their coordinates by similarly minimizing the error in the distances to the landmarks. Our simulation results have shown that the proposed self-configurable positioning technique can tolerate 20% (or higher) measurement inaccuracy, effectively providing location information for the nodes in a wireless network. While the coordinates error decreases with more landmarks, four to seven landmarks are usually sufficient to meet the accuracy requirement for a network with hundreds of nodes. The computing time for coordinates establishment is in the order of milliseconds in our simulation (where a 2.66 GHz processor is used), acceptable for most applications in mobile ad hoc networks and wireless sensor networks.

The rest of this paper is organized as follows. Section II discusses background and related work. Section III introduces the proposed self-configurable positioning technique. Simulation results are presented in Section IV. Finally, Section V concludes the paper.

II. Background and Related Work

Received Signal Power (RSS) and Time-of-Arrival (ToA) are two basic approaches for estimating the distance between adjacent nodes that are within the transmission range of each other. RSS measures the power of the signal at the receiver and calculates the distance according to the propagation loss model. ToA measures the propagation time of the received signal and determines the distance by multiplying it with the speed of light. Multiple measurements can be averaged to obtain more accurate results. In general, RSS is easier to implement, while ToA may achieve higher accuracy. In the following discussion, we assume either of them is used to provide the estimated distance (possibly with a certain error) between two neighboring nodes.

A number of positioning systems have been proposed recently for multihop wireless networks. Ref. [13] introduces a GPS-free positioning system for mobile ad hoc networks. A node measures the one-hop distance to its neighbors and exchanges distance information with each other such that every node knows its two-hop neighbors and the distances. Based on this information, each node establishes the local coordinates with itself as the origin. Then, these local coordinates are tuned and merged to become the coordinates of the entire system. This approach needs neither GPS information nor any infrastructure support other than the nodes in the ad hoc network. However, it has three major drawbacks. First, each node needs at least three neighbors to establish the local coordinates. Second, the inaccurate one-hop distance estimation may result in significant errors or even a failure in establishing the local coordinates. Third, merging the local coordinates is complex. Another GPS free positioning system has been proposed in [14]. It utilizes connectivity information and establishes the coordinates system by using a dimensional scaling algorithm. Its major drawbacks are the high computing complexity (not scalable) and low accuracy.

Refs. [13] and [14] are the only existing positioning schemes that do not require any GPS-aware nodes. There are several other proposals that need a subset of nodes equipped with GPS receivers (or that use other global positioning systems). In [7], the authors present a GPS-less low cost localization approach for very small devices. The basic idea is to deploy a number of fixed reference points with overlapping regions covering the network. The reference points are equipped with GPS receivers and send out beacons periodically. A node in the network estimates its coordinates based on the received beacons and the coordinates of the reference points. Similarly, [15] presents a localization algorithm that calculates the location of a node based on the locations of several immediate neighboring beacons nodes.

A recursive position estimation approach that requires fewer reference nodes is proposed in [16] for sensor networks. A node that is close to at least three reference points estimates its position through nonlinear regression. After the node obtains a reasonable position estimate, it may serve as a new reference point. This process can be applied recursively until all nodes in the network have obtained their coordinates. While the recursive approach saves on hardware cost, it sacrifices accuracy, especially for the node far away from the original reference points.

Ref. [17] proposes a convex position estimation system for wireless sensor networks. A number of nodes have known positions, while the positions of other nodes are unknown. The connectivity of the network is represented by a set of convex position constraints and a linear programming model is used to obtain the coordinates of those unknown nodes. The main drawback of this approach is the high computation complexity involved in solving the linear program problem. Similarly, [18] is another connectivity-based approach, where the reference nodes are used to confine the possible locations of the unknown nodes.

Ref. [19] proposes an ad hoc positioning system (APS), where a number (at least three) of landmarks with GPS receivers are assumed to be available. The ad hoc nodes estimate the distances to these landmarks (that may be multiple hops away) according to the number of hops or the route distance obtained by a distance vector algorithm. Then, the node coordinates can be calculated using the triangulation approach. In [20], a similar approach is employed to estimate the coordinates, which are then iteratively refined for improving the accuracy.

In [21], a localization scheme is proposed for wireless sensor networks based on the angle of arrival (AoA) technique. A set of reference points with known coordinates are deployed. The reference points transmit high power signals to cover the entire network area. The sensor nodes receive the signals from at least three reference points and determine their coordinates by triangulation according to the angle bearings of the incoming signals. In APS-AoA [22], the AoA technique is applied in the ad hoc network without reference points that transmit high power signals. The nodes collect the angle information from neighbors and derive coordinates by using angle-based triangulation.

Clearly, none of the above related work meets all of our design objectives discussed in Section I. Specifically, [13] is not robust; [14] is computationally complex and potentially inaccurate; while [7] and [15]–[21] are not completely self-configurable and still depend on GNSS systems.

III. Proposed Self-Configurable Positioning Technique

The basic idea of our proposed self-configurable positioning technique is to select a number of nodes serving as the landmarks, which can exchange information and establish a coordinate system by themselves without the support of GNSS. The other nodes (called regular nodes, hereafter) in the network can accordingly contact the landmarks and compute their own coordinates. In this section, we first introduce a Euclidean distance estimation model that serves as the basic element of the proposed positioning technique. Then, we present the proposed coordinates calculation scheme and the landmark selection algorithm. Finally, we discuss the application of the proposed positioning technique in heterogeneous wireless networks. For simplicity, we illustrate the proposed approach in a two-dimensional space, while it can be readily applied to three dimensions as well.

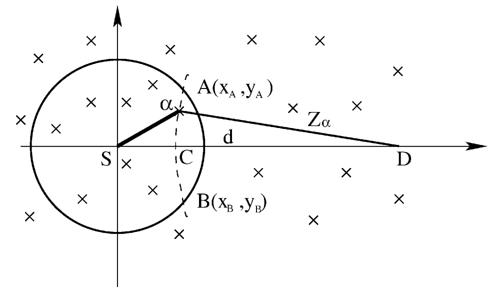
A. Euclidean Distance Estimation

In order to establish the coordinates system, it is crucial to have an accurate estimation of the distance between two landmarks or between a regular node and a landmark. If two nodes are within the transmission range of each other, RSS, ToA, or TDoA (Time Difference of Arrival, a variation of ToA) can be used to estimate their distance [15]. When the two nodes are not adjacent, however, the distance estimation becomes nontrivial.

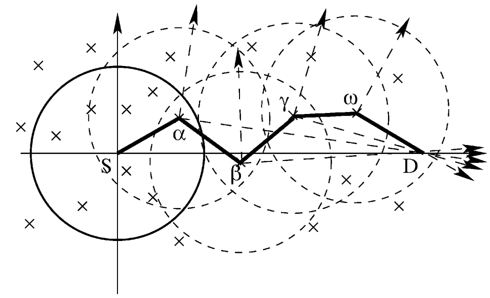
In this research, we propose a simple and effective scheme to estimate the distance between two remote nodes. The basic idea of the proposed model is to reveal the correlation of the Euclidean distance and the corresponding shortest path length between two nodes in the network. Based on such a model, a node (S) can estimate the Euclidean distance to another node (D) by sending a control packet and finding the length of the shortest path to D.

We assume that the node distribution of the wireless network is known. A uniform distribution is adopted in the following discussion, because it well approximates most applications (e.g., a sensor network consisting of randomly distributed sensors, or an ad hoc network with randomly situated mobile users). When the node distribution is not uniform, the Euclidean distance estimation model will still be effective with minor modifications (as will be discussed later).

Without loss of generality, we consider two nodes, S and D [as shown in Fig. 1(a)], in a network (or a part of a network) consisting of N nodes uniformly distributed in a 1×1 area. The Euclidean distance between S and D is d. To facilitate our discussion, we assume arbitrary coordinates, with S located at (0,



(a)



(b)

Fig. 1. Euclidean distance estimation model. The nodes (denoted by "×") are for illustration only, and they may be located anywhere in a real system.

0) and D at $(d, 0)$.[1] The transmission range of a node in the network is $r \ll 1$.

For analytic tractability, we show the correlation between the Euclidean distance and the shortest path length by finding the length of the short path (l), for the given Euclidean distance d between S and D. Since the nodes are uniformly distributed, there are in average a set Φ of $N \times \pi r^2$ nodes within S's transmission range. The distance between node D and a node i (with coordinates (X_i, Y_i)) in Φ is given by

$$Z_i = \sqrt{(X_i - d)^2 + Y_i^2} \tag{1}$$

where X_i and Y_i are random variables with a uniform distribution

$$f_{\mathbf{X_i}, \mathbf{Y_i}}(x_i, y_i) = \begin{cases} \frac{1}{\pi r^2}, & i \in \Phi \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Accordingly, we can derive the density function of Z_i

$$f_{\mathbf{Z_i}}(z_i) = \frac{2}{\pi r^2} z_i \cos^{-1} \frac{z_i^2 + d^2 - r^2}{2 z_i d}. \tag{3}$$

In order to emulate the shortest path algorithm, we assume a node α, which is within the transmission range of S and has the shortest Euclidean distance to D, is selected as the next hop along the shortest path. Since the distance between node α and node D is the shortest, we have

$$Z_\alpha = \min \{Z_i \,|\, i \in \Phi\}. \tag{4}$$

Consequently, we can derive the density function of Z_α,

$$f_{\mathbf{Z_\alpha}}(z_\alpha) = N\pi r^2 \left(1 - F_{\mathbf{Z_i}}\right)^{N\pi r^2 - 1} f_{\mathbf{Z_i}}(z_i) \tag{5}$$

[1]Note that these coordinates are used to facilitate our discussion of the Euclidean distance estimation model only. It has *no* relationship to the coordinates system to be discussed in Section III-B.

and obtain its mean value

$$E(\mathbf{Z_\alpha}) = d - r + \int_{d-r}^{d+r} (1 - F_{\mathbf{Z_i}}(z_i))^{N\pi r^2}\, dz_i \qquad (6)$$

where $F_{\mathbf{Z_i}}(z_i)$ is the cumulative probability distribution of Z_i

$$F_{\mathbf{Z_i}}(z_i) = \int_{d-r}^{z_i} f_{\mathbf{Z_i}}(z_i)\, dz_i. \qquad (7)$$

To derive the coordinates of node α, we draw an arc ACB with node D as the center and $E(\mathbf{Z_\alpha})$ as the radius [see Fig. 1(a)]. The coordinates of A and B are (x_A, y_A) and (x_B, y_B), where

$$x_A = x_B = \frac{r^2 + d^2 - E^2(\mathbf{Z_\alpha})}{2d} \qquad (8)$$

$$y_A = -y_B = r \sin\left(\cos^{-1} \frac{r^2 + d^2 - E^2(\mathbf{Z_\alpha})}{2rd}\right). \qquad (9)$$

Assuming node α is uniformly distributed along AC (or BC), we may obtain the mean length of the first hop along the shortest path from S to D

$$E(\mathbf{l_\alpha}) = \frac{2(d - E(\mathbf{Z}_\alpha))}{\theta_\alpha} \mathrm{Elliptic}E\left(\frac{1}{2}\theta_\alpha, -\frac{4dE(\mathbf{Z_\alpha})}{(d - E(\mathbf{Z_\alpha}))^2}\right) \qquad (10)$$

where *EllipticE* is the incomplete elliptic integral of the second kind, which is defined as

$$\mathrm{Elliptic}E(\phi, m) = \int_0^\phi (1 - m \sin^2\theta)^{1/2}\, d\theta \qquad (11)$$

$$\theta_\alpha = \cos^{-1} \frac{d^2 + E(\mathbf{Z_\alpha})^2 - r^2}{2dE(\mathbf{Z_\alpha})}. \qquad (12)$$

Recursively applying the above method, we can obtain the length of the remaining hops along the shortest path. More specifically, we establish new coordinates [as shown in Fig. 1(b)], where α locates at (0, 0) and D locates at $(E(Z_\alpha), 0)$. Thus, the mean length of the second hop $(E(\mathbf{l_\beta}))$ can be similarly calculated by replacing d with $E(Z_\alpha)$ in (1)–(12). This procedure is repeated until the remaining distance to D [e.g., $|\omega D|$ in Fig. 1(b)] is not longer than r. The total length of a shortest path with m hops is

$$l = \sum_{\zeta=1}^{m} l_\zeta \qquad (13)$$

where l_ζ can be obtained through (1)–(12) if $\zeta < m$, or otherwise if it is the last hop, l_ζ equals to the remaining distance to D.

The above model reveals the correlation between the Euclidean distance and the shortest path length for any given node density. For example, the results for three different N values are shown in Fig. 2. As we can see, the shortest path length becomes closer to the Euclidean distance with the increase of N. This is expected because the shortest path is almost a straight line when node density is very high. The calculation to obtain such a figure can be done either off-line by a central controller or in a distributed manner by each node. In addition, when the node distribution is not uniform, the Euclidean distance estimation model can still be effective by replacing (2) with a new density function and deriving (3)–(6) accordingly.

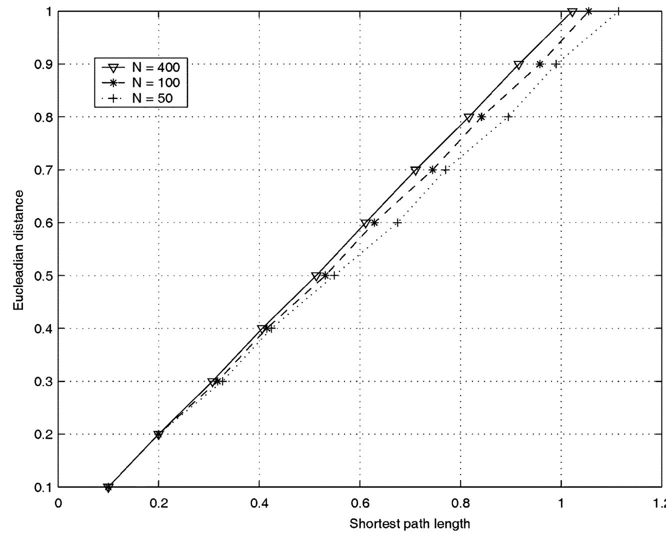


Fig. 2. Euclidean distance versus the shortest path length under $r = 0.25$.

When a node A needs the Euclidean distance to another node B, it sends to B a control packet that includes a *route length* field with an initial value of zero. When an intermediate node receives the control packet, it adds the one-hop distance (i.e., the distance between itself and the previous node, which is obtained via either RSS or ToA) to the *route length* field. We assume the control packets follow the shortest path from one node to another node.[2] Upon receiving the control packet, node B sends it back to node A immediately. After node A receives the return control packet, it reads the *route length* field and divides the value by two to get the shortest path length. Accordingly, the Euclidean distance between A and B can be determined by looking up Fig. 2.

B. Coordinates System Establishment

The coordinate system is established in two steps. First, similar to [23] that is used for Internet distance prediction in a wired network, the landmarks determine their coordinates themselves by exchanging information between each other and minimizing an error objective function. Then, other regular nodes calculate their own coordinates according to the landmarks.

1) Landmarks: A number of (at least three) nodes in a multihop wireless network are selected to be landmarks. The algorithm for landmark selection will be discussed separately in Section III-C.

After the landmarks have been identified, each of them sends a control packet to every one of all other landmarks in order

[2]In dynamic routing, the node may receive multiple replies and choose the one along the shortest path. In addition, each node may send several control packets separated by random time intervals, and choose the one that travels the shortest distance, in order to address the problem of possible packet loss along the shortest path.

to learn the Euclidean distance, as we have discussed in Section III-A. Once a landmark (e.g., landmark i) has obtained a set of distances to all other landmarks, i.e., $L_i = \{L_{ij} \,|\, K \geq j \geq 1, j \neq i\}$, it sends L_i to other landmarks in the network. After a landmark has accumulated a full set of L_i, i.e., $L = \{L_i \,|\, K \geq i \geq 1\}$, it has the distance between any two landmarks in the network. Thus any landmark may calculate the coordinates using the following method. Without loss of generality, we assume the landmark with the lowest identification (ID) number (e.g., IP address) will perform the calculation.

Our objective is to establish the coordinates that minimize the sum of the errors of the distances between any two landmarks. In other words, assuming the coordinates of a landmark i is (x_i, y_i), then the distance between two landmarks i and j in the established coordinates system is

$$\widehat{L_{ij}} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{14}$$

and the error function is defined to be

$$\epsilon = \sum_{i=1}^{K} \sum_{j=i+1}^{K} \left(\frac{L_{ij} - \widehat{L_{ij}}}{L_{ij}} \right)^2. \tag{15}$$

A Simplex method [24] (which is also adopted in [23]) is used to minimize the error function ϵ and give the coordinates for each landmark. The Simplex method is a linear fitting procedure for function minimization. It uses linear adjustment of the parameters (in our case, the coordinates variables $(x_i, y_i), 1 \leq i \leq K$) until some convergence criterion is met (i.e., ϵ becomes lower than a threshold). For example, in a network with four landmarks A, B, C, and D [as shown in Fig. 3(a)]

$$\begin{aligned}\epsilon = &\left(\frac{L_{AB} - \widehat{L_{AB}}}{L_{AB}} \right)^2 + \left(\frac{L_{AC} - \widehat{L_{AC}}}{L_{AC}} \right)^2 \\ &+ \left(\frac{L_{AD} - \widehat{L_{AD}}}{L_{AD}} \right)^2 + \left(\frac{L_{BC} - \widehat{L_{BC}}}{L_{BC}} \right)^2 \\ &+ \left(\frac{L_{BD} - \widehat{L_{BD}}}{L_{BD}} \right)^2 + \left(\frac{L_{CD} - \widehat{L_{CD}}}{L_{CD}} \right)^2\end{aligned}$$

where L_{ij} can be learned through the Euclidean distance estimation model and $\widehat{L_{ij}}$ is expressed by the coordinates variables, i.e., (x_i, y_i) where i could be A, B, C, and D, following (14). The Simplex method is then used to determine the coordinates variables such that ϵ is minimized. The Simplex has been implemented in many softwares. Our simulation (to be discussed in Section IV) adopts the Simplex module available in Matlab [25] to minimize the error objective function.

Note that, there are an infinite number of solutions, since the coordinates can be rotated or translated as long as their distances do not change. Without loss of generality, we determine the coordinates conveniently in our implementation as follows (with further discussion given in Section IV). The landmark that performs the above calculation (i.e., the one with the lowest ID) sets (0, 0) to be its own coordinates. The landmark with the second lowest ID is assumed to have coordinates $(x, 0)$ where $x > 0$ (i.e., on X-axis), while the landmark with the third lowest ID is assumed to have a negative Y value. After having been determined, the coordinates are sent to all landmarks in the network.

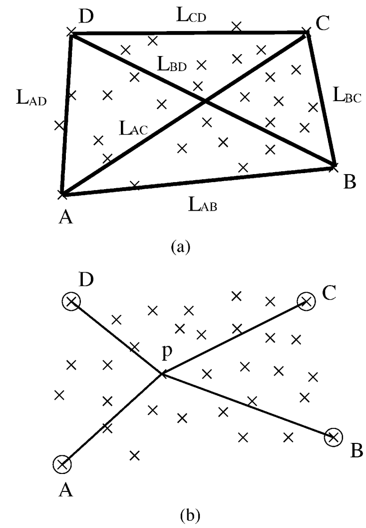


Fig. 3. Coordinates system establishment. (a) Landmarks. (b) Regular nodes.

2) Regular Nodes: Theoretically, all nodes in a network can be chosen as landmarks, with their coordinates determined in a way as discussed in Section III-B1. This approach, however, is not scalable, because the computing complexity increases exponentially with the number of landmarks. Hence, only a limited number of nodes may serve as landmarks, while other regular nodes learn the coordinates according to them.

In order to calculate its coordinates, a node needs to know the coordinates of, and its distances to, the landmarks. There are two approaches to obtain such information. In the on-demand approach, when a node [e.g., node p in Fig. 3(b)] needs its coordinates, it sends a control packet to each of the landmarks. The control packet is initially broadcasted over the network if node p does not know the IDs of the landmarks. The subsequential requests (e.g., when updated coordinates are needed due to the node's mobility) are sent via unicast or multicast in order to reduce the control overhead. Similar to that discussed in Section III-B1, the control packet includes a *route length* field to allow node p to find the shortest path lengths and accordingly the Euclidean distances to the landmarks, i.e., $L_p = \{L_i p \,|\, K \geq i \geq 1\}$. In addition, when a landmark sends the control packet back to node p, its coordinates are attached so that node p can obtain a set of coordinates of all landmarks. Alternatively, in the pro-active approach, the landmarks can periodically broadcast control packets that include their coordinates to all nodes in the networks.

After obtaining the coordinates of, and the distance to, all landmarks, node p calculates its coordinates by minimizing an error objective function similar to that introduced in Section III-B1. More specifically, assuming the coordinates of node p is (x_p, y_p), then the distance between node p and a landmark i in the established coordinates system is

$$\widehat{L_{ip}} = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2} \tag{16}$$

and the error function is defined to be

$$\epsilon_p = \sum_{i=1}^{K} \left(\frac{L_{ip} - \widehat{L_{ip}}}{L_{ip}} \right)^2. \tag{17}$$

Again, the Simplex method can be used to minimize the error function ϵ_p and determine the coordinates (x_p, y_p).

After calculating its coordinates, node p may label itself as a "semi-landmark" and respond to the requests of other regular nodes, in order to reduce communication overhead and quicken coordinates establishment. Note that, however, only very stable nodes shall be labeled as semi-landmarks to minimize the possible errors. On the other hand, other regular nodes may decide whether or not to use the information obtained from the semi-landmarks, according to their requirements on delay, accuracy, and/or computational complexity.

C. Selection of Landmarks

The selection of landmarks is a crucial design issue of the proposed self-configurable positioning technique. In this subsection, we discuss this issue in two parts: 1) How many nodes should be selected to serve as landmarks? and 2) Which nodes (in terms of, e.g., nodal location, computing power, stability, etc.) shall be selected?

1) Number of Landmarks: The more the landmarks, the higher the accuracy of the established coordinates system, because the measurement errors can be more effectively smoothed out by minimizing the error objective function. It is, however, not practical to employ a large number of landmarks since the computational complexity of establishing the coordinates system (e.g., the procedure to determine the coordinates of the landmarks) increases exponentially with the number of landmarks. In this work, we study the tradeoff between the accuracy and the complexity, and develop a distributed and flexible landmark selection scheme.

To minimize the computational complexity, the system always starts with K (a small constant) landmarks. Extensive simulations have been carried out to study the impact of K on the accuracy of the coordinates system. Our results (as to be discussed in Section IV) show that the typical values of K may vary from four to seven depending on the required accuracy. After a regular node calculates its coordinates, it may announce itself as a "semi-landmark" if it is stable and computationally powerful. As a result, there are K landmarks and M semi-landmarks in the network, which are usually sufficient (or even more than sufficient) for highly accurate coordinates calculation. Since different regular nodes may have different accuracy requirement and computing power, it is up to the individual regular nodes to decide on how many landmarks and/or semi-landmarks to use for coordinates calculation. For example, if there are only four landmarks in the network, but a regular node needs coordinates with an average error (to be discussed later in Section IV) lower than 0.006, then three additional semi-landmarks may be used for coordinates calculation (according to our simulation results shown in Fig. 29).

2) Locations of Landmarks: In order to study the impact of landmarks' locations on the accuracy of coordinates calculation, we consider four landmarks in a network with N nodes uniformly distributed in a 1×1 area. For simplicity, we assume that the four landmarks locate at the vertices of a square which is centered at (X_c, Y_c) and has an edge of G. We vary the values of (X_c, Y_c) and G to change the locations of the landmarks, and analyze the errors of coordinates calculation of the entire network, which is defined as the total errors of all regular nodes (i.e., $\sum \epsilon_p$, where ϵ_p is the error of a node p as shown in (17)). Assume the average single hop distance estimation error is

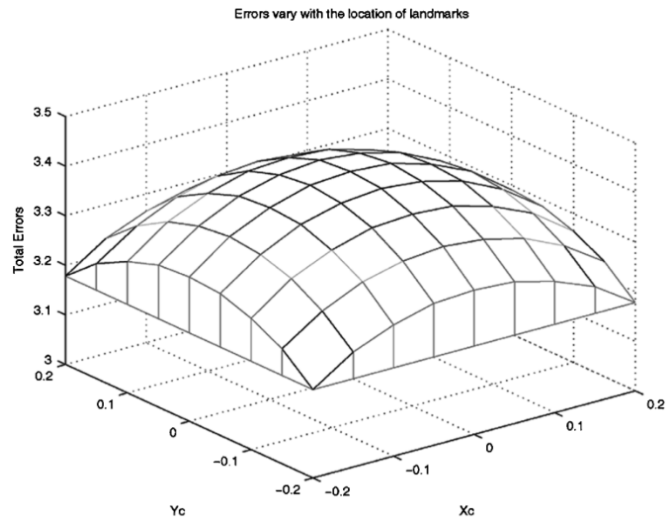


Fig. 4. $G = 0.5$.

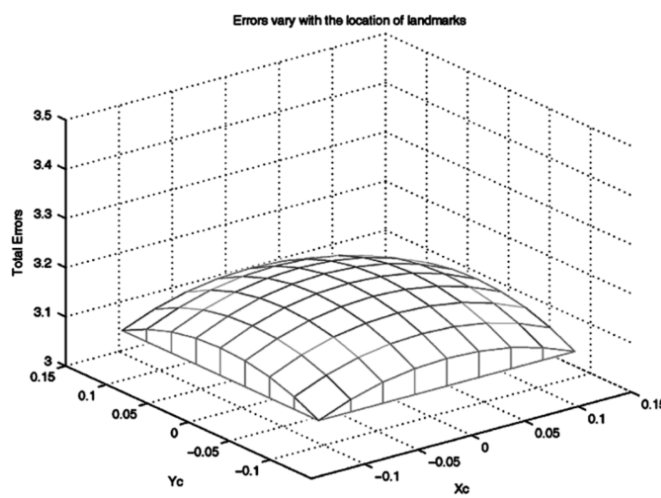


Fig. 5. $G = 0.7$.

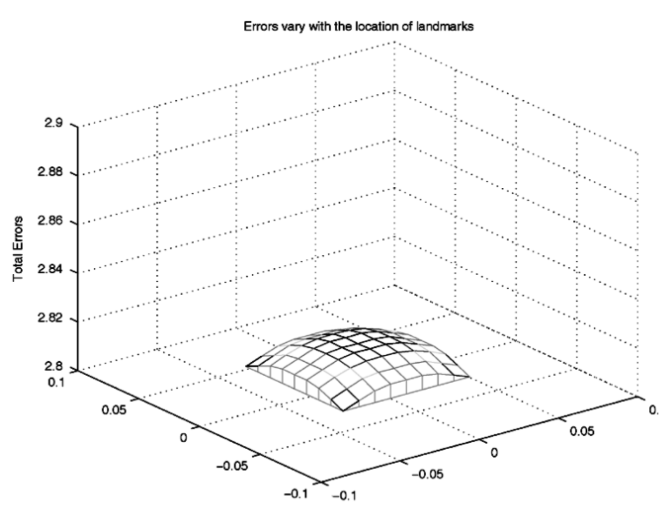


Fig. 6. $G = 0.9$.

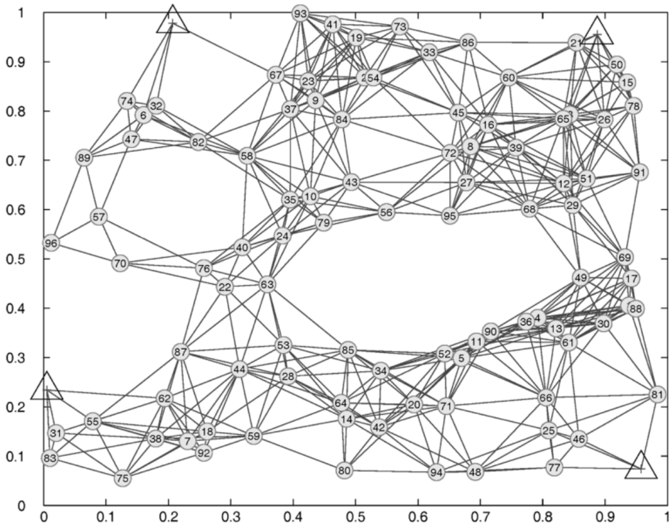


Fig. 7. $N = 100$.

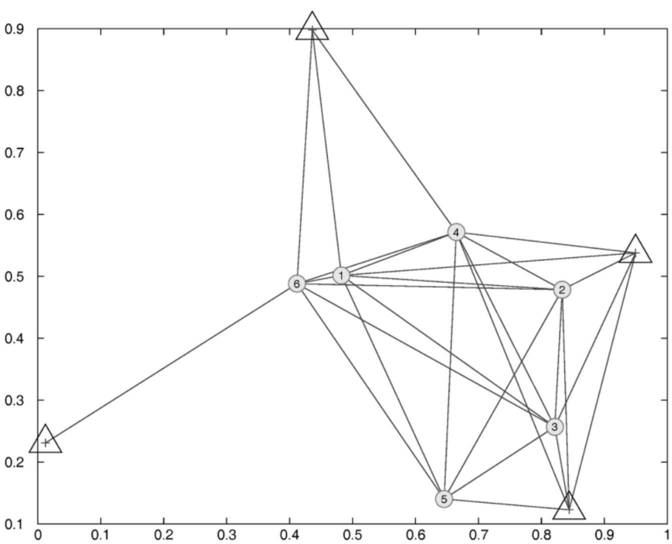


Fig. 8. $N = 10$.

ϵ_0. If the shortest path between two nodes includes n hops, the average error of the estimated distance of the entire path is

$$\epsilon_l = \frac{\epsilon_0}{\sqrt{n}}. \tag{18}$$

ϵ_l will in turn affect L_{ip} in (17) and the accuracy of coordinates calculation.

Our experimental results are shown in Figs. 4–6. For a given value of G, we observe the maximum error when $X_c = Y_c = 0$, i.e., when the square with four landmarks as vertices is at the center of the network. The error decreases as the landmarks deviate from the center, and reaches minimum as they are located at the "corners" of the network. With the increase of G, the square is stretched, and the locations of landmarks become closer to the boundary of the network. As can be seen, the errors decrease significantly with the increase of G. We have carried out similar experiments by varying the network size and the number of landmarks, and observed similar tread in all scenarios. As a result, we conclude that the landmarks should be separated as far as possible, i.e., they should locate at the corners of the network. This is reasonable because the farther the landmarks, the longer the average path length from the regular nodes to the landmarks, thus decreasing the path error [as shown in (18)] and improving the accuracy of coordinates calculation.

In order to select a number of (e.g., K) landmarks, we develop an algorithm to determine K corner nodes of the network. Initially any node is a candidate of landmark if its stability and computing power are higher than a predefined threshold. We define a set Ψ, which includes all landmark candidates. Each candidate node discovers the shortest path to all other candidate nodes in the network. We define a variable C_i, called *candidacy degree*, for node i,

$$C_i = \sum_{j=1, j\neq i}^{N} \frac{1}{S_{i,j}^2} \tag{19}$$

where $S_{i,j}$ is the length of the shortest path from i to j, if node j is in set Ψ; or otherwise, $S_{i,j}$ is taken as an infinite value.

A node i with the highest value of C_i is most probably located at the center of network, and thus should be removed from Ψ first. Then the candidacy degree of all other nodes will be updated since all $S_{i,j}$ related to the eliminated candidate node i is changed to infinity. This procedure is repeated until Ψ has the last K nodes left, which are chosen as landmarks. The proposed landmark selection scheme is summarized in Algorithm 1.

Algorithm 1 Landmark Selection

$\Psi = \emptyset$.
for $i = 1 : N$ **do**
  if node i has required stability and computing power **then**
    $\Psi = \Psi + i$.
  end if
end for
for $m = 1 : N - K$ **do**
  for all $i \in \Psi$ **do**
    $C_i = \sum_{j=1, j\neq i}^{N} (1)/(S_{i,j}^2)$.
  end for
  Search for node i, with $C_i \leq C_j$ ($j \in \Psi$ and $j \neq i$).
  $\Psi = \Psi - i$.
  $S_{i,j} = \infty$, for $1 \leq j \leq N$.
end for

Our simulation shows that the proposed landmark selection algorithm can effectively determine the corner nodes in a network. For example, Fig. 7 illustrates the result of the algorithm when it is applied to a network with 100 nodes randomly distributed in a 1×1 area. The nodes marked with small triangles are the four landmarks determined by this algorithm. As we can see, they locate largely at the corners of the network, except that node 83 seems a better choice than the one selected at the lower-left corner. This deviation is reasonable because the shortest path instead of the direct Euclidean distance is used in our algorithm in order to reduce the complexity. The proposed landmark selection algorithm also works well in a sparse network, e.g., with only 10 nodes in a 1×1 area, as shown in Fig. 8.

D. Further Discussion: Application in Heterogeneous Wireless Networks

While we have assumed a homogeneous wireless network in our discussion so far, the proposed self-configurable positioning technique can be readily applied to heterogeneous wireless networks, such as the Intel heterogeneous sensor network where an IEEE 802.11 mesh network comprised of high-end nodes is overlaid on a wireless sensor network in order to improve system throughput and scalability [26]. In such heterogeneous wireless networks, the high-end nodes (or a subset of them) with sufficient computing power, battery power, and wireless signal transmission power, can naturally serve as landmarks in the coordinates system. Since the landmarks can transmit high power signals for a long range, a regular node can usually receive signals from the landmarks directly instead of through multiple hops, thus reducing the control overhead. Moreover, the heterogeneous structure can avoid landmarks to become single point of failure and performance bottleneck (due to, e.g., traffic congestion, high computational load, and power consumption).

IV. Simulation and Discussion

We have done extensive simulations in Matlab [25] to evaluate the performance of the proposed self-configurable positioning technique. In particular, we study the impact of several parameters of interest, such as node density, one-hop distance measurement inaccuracy, the number of landmarks, and node mobility, which may dictate the accuracy, robustness, and overhead of the positioning technique. We assume a number ($N = 50$ to 400) of nodes uniformly distributed in a 1×1 unit area. Each node has a transmission range of $r = 0.25$ unit and thus has an average of about 10 to 80 neighbors, which are the typical node densities in mobile ad hoc networks and/or sensor networks [4]–[6]. The landmarks are selected according to the algorithm proposed in Section III-C.

A. Node Density

Since the Euclidean distance is estimated based on a statistical model, more nodes in a network give rise to more accurate results, consequently enhancing the accuracy of the coordinates system.

1) Euclidean Distance Estimation: Fig. 9(a)–(c) shows the Euclidean distance estimation error in the networks with different node density. In general, the proposed Euclidean distance estimation model yields results that match the simulation results very well. With an increase in node densities, the error decreases. When N reaches 100 the error becomes very small; for $N = 400$, the error is negligible. The accurate Euclidean distance estimation can effectively support the coordinates establishment and reduce the number of landmarks needed (as to be discussed later in this section).

2) Coordinates System: We simulate networks with N nodes that are depicted by the dots in Figs. 10–12. Four nodes (the dots connected by lines in the figures) are selected as landmarks. As we have discussed in Section III-B, the landmark with the lowest ID performs the calculation and sets (0, 0) as its own coordinates. The landmark with the second lowest ID is assumed on the X-axis, and the landmark with the third lowest ID is assumed to have a negative Y value. Additional 30 nodes in a triangle shape (see the figures) are considered as reference points to facilitate our studies on coordinates accuracy and to clearly show any distortion of the established coordinates system. The stars ("$*$") represent the "real" coordinates of the reference points (e.g., according to GPS), while the small circles ("o") stand for their positions in the established coordinates system.

Fig. 10(a)–(c) shows the established coordinates systems for the networks with $N = 50, 100$, and 400 nodes, respectively. As can be seen, the graph corresponding to the established coordinates system (i.e., the triangle consisting of "o") has a similar shape as the graph corresponding to the "real" coordinates system (i.e., the triangle consisting of "$*$"). When the node density is low, e.g., when N equals 50, an obvious distortion is observed in the established coordinates system, which may or may not be acceptable in a given application. With an increase of node density, the degree of distortion lessens. In particular, the two triangles are almost identical when $N = 400$.

In order to make better comparison, we rotate and translate the established coordinates system so that the center of the quadrilateral with four landmarks as vertices has the same coordinates in both the "real" coordinates system and the established coordinates system. We call this coordinates translation *center match*. The outcomes of center match are shown in Fig. 11(a)–(c), where clear deviation exists for $N = 50$ and a perfect match results when N equals 400.

To quantify the coordinates errors, we define the average coordinates error:

$$\Delta = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2} \tag{20}$$

where (x_i, y_i) are the "real" coordinates of node i and (x_i', y_i') are the established coordinates after center match translation. As shown in Fig. 13, the coordinates error decreases with an increase in node density. When $N \geq 100$, the average error is no more than 0.03 units. When $N \geq 250$, the average error converges to less than 0.01 units, which is acceptable in most ad hoc or sensor network applications [4]–[6].

In addition, if one node in the network is equipped with a GPS receiver, it can send to other nodes a packet that includes both its "real" (i.e., GPS) coordinates and the established coordinates. Upon receiving this packet, the nodes without GPS receivers can tune to the GPS coordinates accordingly. Fig. 12(a)–(c) shows the coordinates after GPS tuning, assuming one landmark has the GPS receiver. Note that the result of GPS tuning is not the same as that of center match translation, although the difference is usually insignificant. When additional GPS-aware nodes are available, their information can be used for rectification and to further improve the accuracy.

B. One-Hop Measurement Error

Neither RSS nor ToA provides absolutely accurate distance between two adjacent nodes. In fact, the primary cause of errors involved in our implementation is the inaccuracy of one-hop distance measurements. Such inaccuracy results in Euclidean distance estimation between two nodes deviated from their

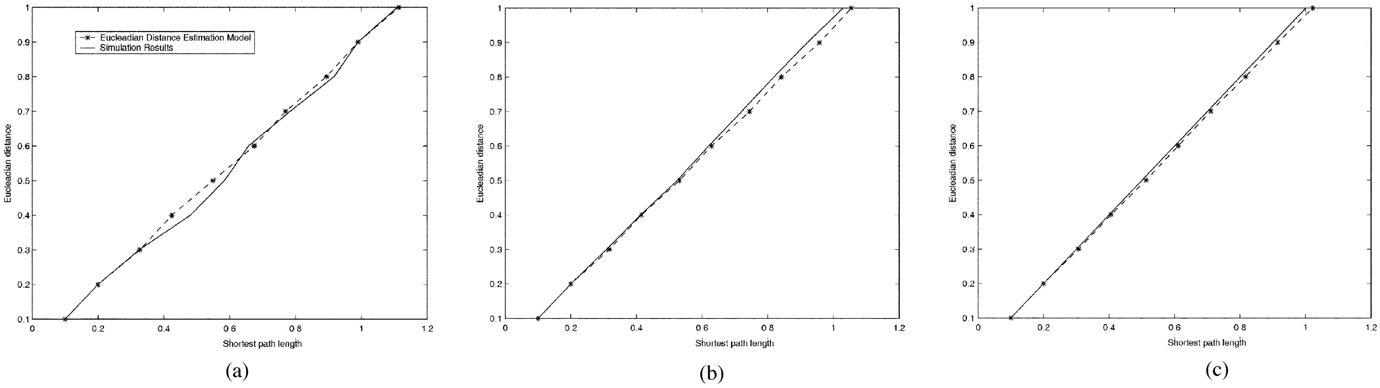


Fig. 9. Euclidean distance. (a) $N = 50$. (b) $N = 100$. (c) $N = 400$.

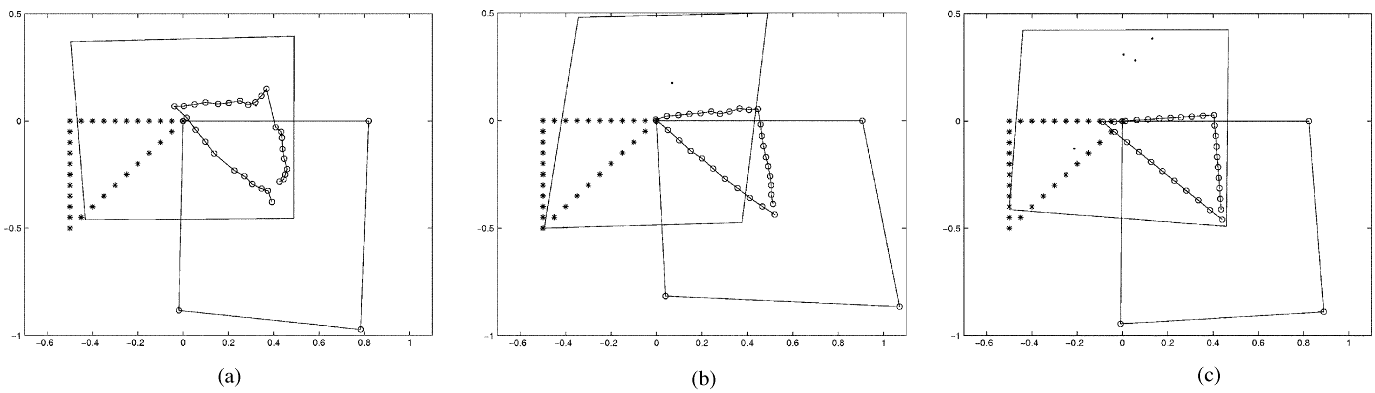


Fig. 10. No translation. (a) $N = 50$. (b) $N = 100$. (c) $N = 400$.

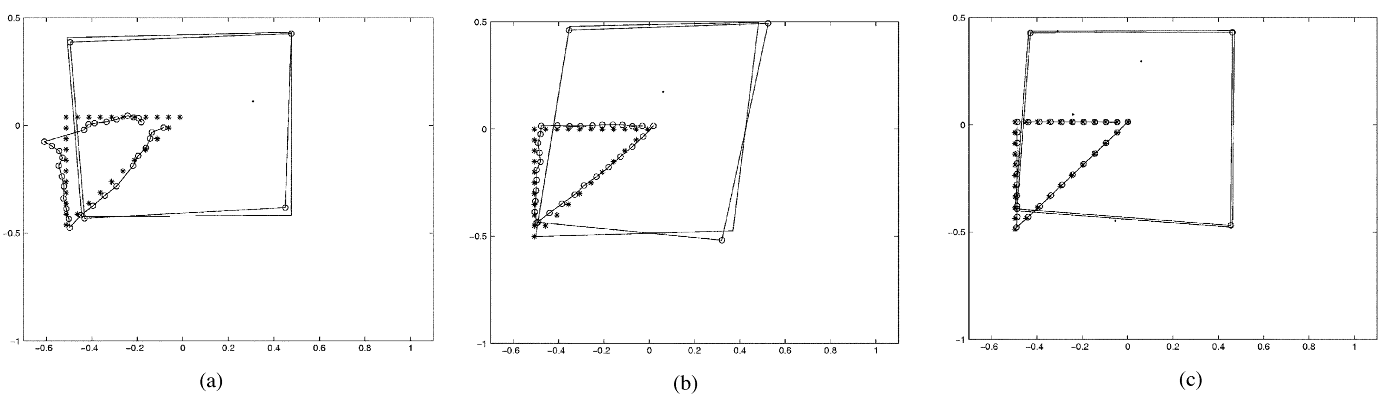


Fig. 11. Center match. (a) $N = 50$. (b) $N = 100$. (c) $N = 400$.

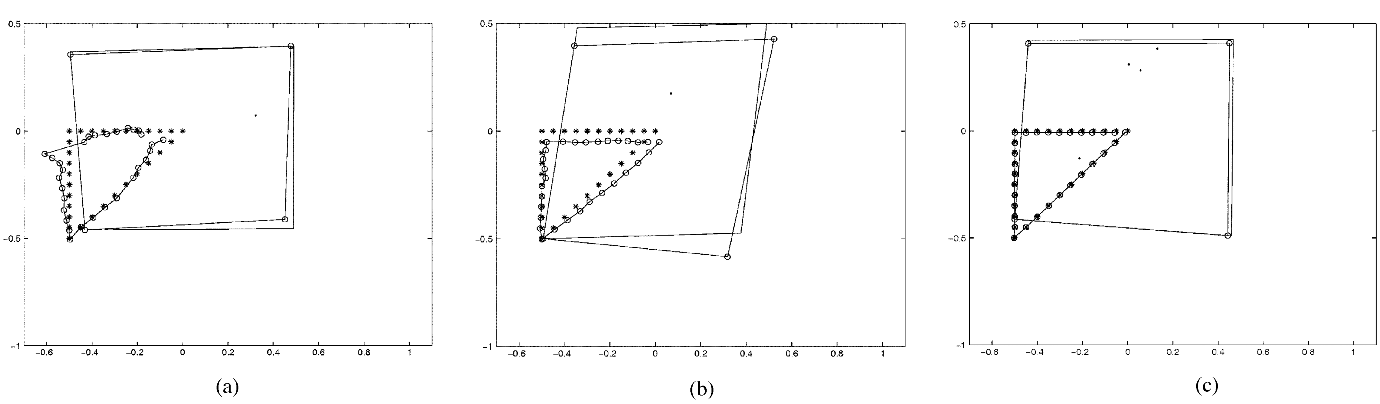


Fig. 12. GPS tuning. (a) $N = 50$. (b) $N = 100$. (c) $N = 400$.

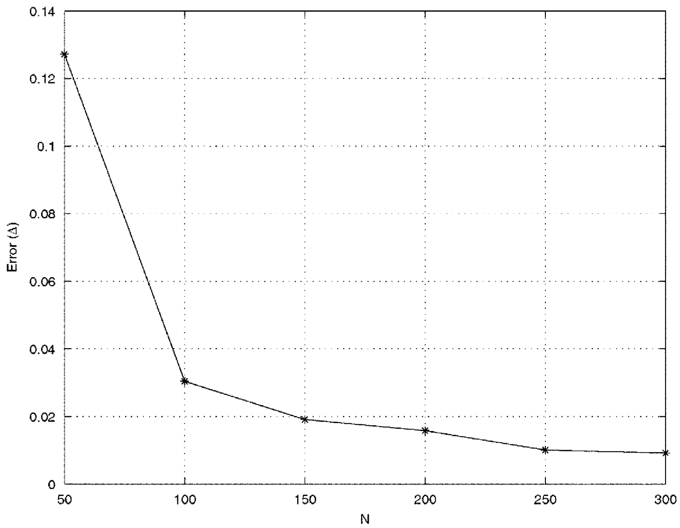


Fig. 13. Coordinates error versus node density.

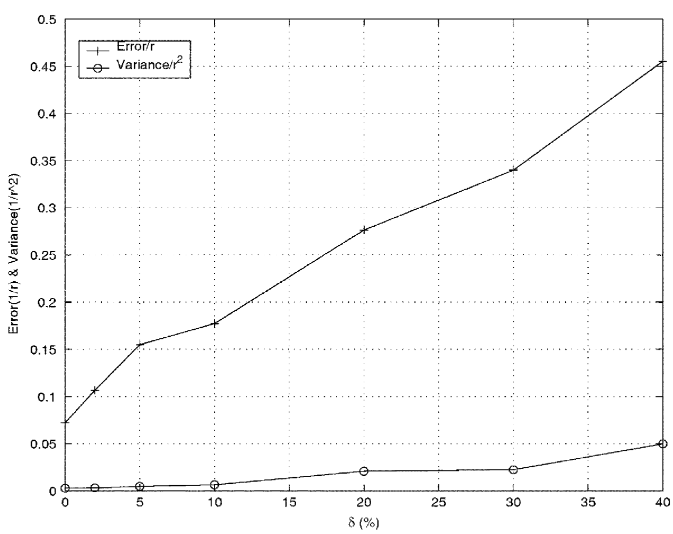


Fig. 14. Coordinates error versus one-hop measurement inaccuracy.

actual distance, thus degrading the coordinates accuracy and swaying the robustness of the proposed positioning technique. In this subsection, we assume that the measurement error at each hop is uniformly distributed between $\pm\delta$ (%), and study the fault-tolerance of the self-configurable positioning technique.

Fig. 15 shows the established coordinates systems with different one-hop distance measurement errors. As expected, the degree of distortion of the established coordinates becomes more and more pronounced with the increase of δ. When $\delta > 20\%$, the distortion degree is so large that the established coordinates are quite different from the "real" coordinates. These results, however, may still be useful to identify the rough direction of a node. Fig. 14 illustrates the relative average coordinates error and variance (as a fraction of r) versus the inaccuracy of one-hop distance measurements. N is 100. When measurement inaccuracy $\delta < 5\%$, the average coordinates error Δ is less than $0.15r$. Apparently, Δ increases with δ. But even when δ is as high as 40%, the average coordinates error is still less than about $0.45r$. In addition, the proposed positioning technique exhibits small variance throughout the range of δ examined, signifying its stability and consistence.

Different applications have different accuracy requirements. For example, in location-aided routing, the source sends the packets in a direction (i.e., in a corn) toward the destination, where the exact node position is usually not necessary. On the other hand, some sensor applications need more precise location information to enable effective data collection. Since the proposed positioning technique is generally applicable to any multihop wireless networks, different applications may choose appropriate hardware/software to provide the one-hop distance that meets the required accuracy.

C. The Number of Landmarks

Fig. 16 shows the effect of the number of landmarks on the performance of the positioning technique. With more landmarks, the Simplex method considers more constraints (i.e., more number of distances between the landmarks or between the regular nodes and the landmarks), thus improving the accuracy. At the same time, however, more landmarks result in significantly increased computing time. As we can see in Fig. 16, the coordinates error (Δ) decreases with the increase of the landmarks, and becomes converged after 5 landmarks have been deployed. In the network with lower node density (e.g., $N = 100$), the number of landmarks has a stronger impact on Δ. When the node density is high, Δ is small even with only three landmarks. In regular ad hoc networks and sensor networks that consist of hundreds of nodes, the proposed positioning technique can usually yield results acceptable by most applications with four to seven landmarks. Nevertheless, more landmarks or semi-landmarks can be deployed to compensate for the inaccuracy introduced by one-hop measurements, helping to achieve the required accuracy level.

D. Control Overhead

In this subsection, we evaluate the control overhead resulted from the proposed self-configurable positioning technique. The overhead for initial landmark discovery is relatively high because flooding is used to locate the landmarks. However, since it happens only during system initialization and has not lasting impact on the system, we ignore the overhead in the initial stage and focus on the overhead for coordinates update only. In order to update the coordinates, a regular node sends a control packet to all landmarks and receives their replies as we have discussed in Section III. In a multihop network, the control packets are forwarded by the intermediate nodes. We measure the average number of control packets transmitted in the network for each coordinates update. The results are illustrated in Fig. 17. As we can see, the total control overhead increases with the number of nodes. But note that, the average number of control packets carried by each node is very low (less than 0.4). In fact, since the average number of intermediate nodes along the path between a regular node and a landmark is proportional to node density, the total control overhead is proportional to $\sqrt{N}$, while the average number of control packets is proportional to $1/\sqrt{N}$.

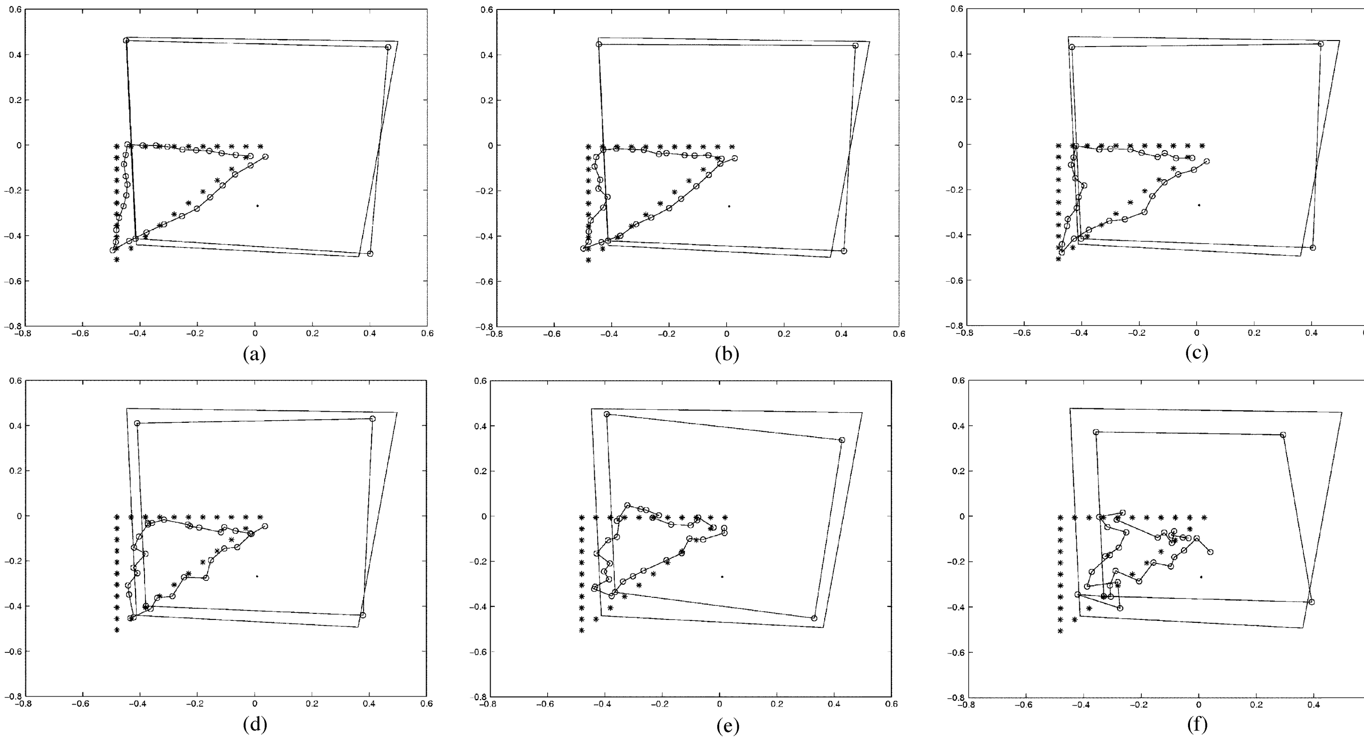


Fig. 15. $N = 100$. (a) $\delta = 2\%$. (b) $\delta = 5\%$. (c) $\delta = 10\%$. (d) $\delta = 20\%$. (e) $\delta = 30\%$. (f) $\delta = 40\%$.

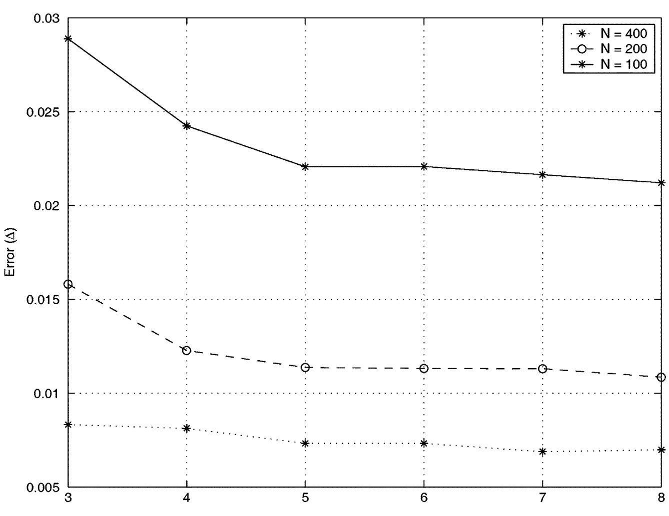


Fig. 16. Coordinates error versus the number of landmarks.

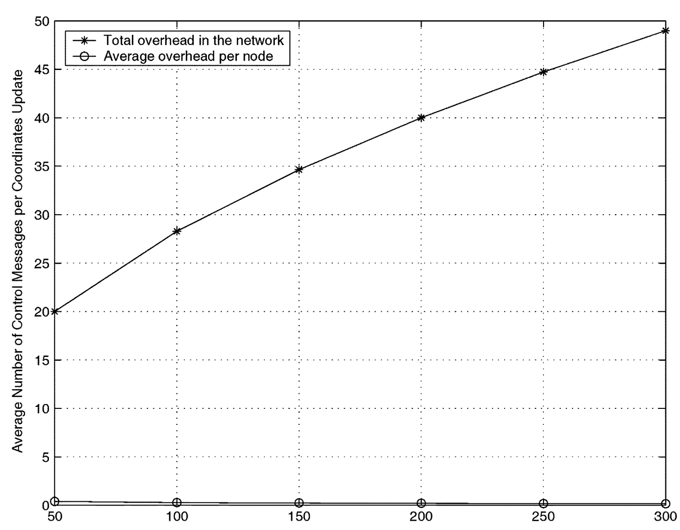


Fig. 17. Average control overhead.

E. Node Mobility

The delay in establishing the coordinates system is short. It includes the transmission delay of the control packets for Euclidean distance estimation and the time for some local computation involved in the Simplex method. The former depends on the networks size and is usually in the order of milliseconds. The latter is determined by the number of landmarks. For example, in our simulation running on a 2.66 GHz Pentium IV processor, the computational delay of the Simplex method is summarized in Table I. In order to establish the coordinates of the landmarks, the Simplex method minimizes the sum of the error of the distance between *any* two landmarks. Thus, the computing time increases exponentially with the number of landmarks (K). But note that, the computing time does not exceed 900 ms even with eight landmarks, and moreover, it occurs only during the system

TABLE I
COMPUTATIONAL TIME INVOLVED IN THE SIMPLEX METHOD FOR THE COORDINATES ESTABLISHMENT OF LANDMARKS AND REGULAR NODES (MILLISECONDS)

K	Landmarks	Regular Nodes
3	41.7	6.3
4	52.0	8.8
5	104.7	9.8
6	234.3	10.3
7	437.0	10.9
8	864.0	14.4

initialization. In contrast to the landmark coordinates establishment, where K pairs of coordinates variables, i.e., $(x_i, y_i), 1 \leq i \leq K$, need to be adjusted in the Simplex method, the computing time for the coordinates of a regular node (with only two coordinates variables) is much shorter (around 10 ms) and increases linearly with the number of landmarks. Hence, the node mobility (with a moderate moving speed) has no significant affect on the coordinates determination of the regular nodes, because the computation can be finished before the node's position has a significant change, except more frequent calculation may be needed to obtain the up-to-date location information. Thus, the results are omitted here.

V. Conclusion and Future Work

We have proposed a self-configurable positioning technique for multihop wireless networks. A number of nodes at the "corners" of the network serve as landmarks for estimating the distances by a Euclidean distance estimation model and establishing the coordinates themselves by minimizing an error objective function. Other nodes calculate their coordinates according to the landmarks. The proposed positioning technique is self-configurable and independent of global position information. Extensive simulations have been carried out to evaluate its performance in terms of accuracy, robustness, and overhead. Our results indicate that the coordinates error is determined by node density, one-hop distance measurement inaccuracy, and the number of landmarks. With a node density of ten neighbors per node, the proposed self-configurable positioning technique yields useful results, despite with noticeable errors. When node density is high and one-hop distance measurement inaccuracy is low, the established coordinates are quite accurate. The proposed positioning technique has exhibited high fault-tolerance and can obtain useful results even with 20% or more measurement inaccuracy. More landmarks result in a better positioning system, but at the same time, the computing complexity increases accordingly. Acceptable results can usually be obtained using four to seven landmarks, depending on the accuracy requirement. The computing time for coordinates establishment is in the order of milliseconds, capable for most applications in the mobile ad hoc networks as well as the sensor networks. In our future work, we plan to implement the proposed positioning technique in a wireless network testbed and study its performance with real measurements.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments that helped to improve this paper.

References

[1] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *Proc. ACM/IEEE 4th Annu. Int. Conf. Mobile Computing and Networking (MobiCom'98)*, 1998, pp. 66–75.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM Wireless Netw.*, vol. 7, no. 6, pp. 609–616, 2001.

[3] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM/IEEE 6th Annu. Int. Conf. Mobile Computing and Networking (MobiCom'00)*, 2000, pp. 243–254.

[4] C. K. Toh, *Wireless ATM and Ad hoc Networks: Protocols and Architectures*. Boston, MA: Kluwer, 1996.

[5] C. E. Perkins, *Ad Hoc Networking*. Reading, MA: Addison Wesley, 2000.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, pp. 393–422, 2002.

[7] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Pers. Commun. Mag.*, vol. 7, no. 5, pp. 28–34, May 2000.

[8] M. S. Khan, "Political and economic dimensions of global navigation satellite system (GNSS)," in *Proc. IEEE Aerospace Conf.*, 2001, pp. 1271–1276.

[9] T. Rappaport, J. Reed, and B. Woerner, "Position location using wireless communications on highways of the future," *IEEE Commun. Mag.*, vol. 34, no. 10, pp. 33–41, Oct. 1996.

[10] B. Parkinson and S. Gilbert, "NAVSTAR: Global positioning system—Ten years later," *Proc. IEEE*, vol. 71, pp. 1177–1186, 1983.

[11] K. Pahlavan, X. Li, and J.-P. Makela, "Indoor geolocation science and technology," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 112–118, Feb. 2002.

[12] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE Annu. Joint Conf. IEEE Computer and Communications Societies (INFOCOM'00)*, 2000, pp. 775–784.

[13] S. Capkun, M. Hamdi, and J. P. Hubaux, "GPS-free positioning in mobile ad hoc networks," in *Proc. 34th Annu. Hawaii Int. Conf. System Sciences*, 2001, pp. 3481–3490.

[14] W. Ruml, Y. Shang, and Y. Zhang, "Location from mere connectivity," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHOC'03)*, 2003, pp. 201–212.

[15] A. Savvides, C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad hoc networks of sensors," in *Proc. ACM/IEEE 7th Annu. Int. Conf. Mobile Computing and Networking (MobiCom'01)*, 2001, pp. 166–179.

[16] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," in *Proc. IEEE Int. Conf. Network Protocols (ICNP'01)*, 2001, pp. 35–41.

[17] L. Doherty, L. Ghaoui, and K. Pister, "Convex position estimation in wireless sensor networks," in *Proc. IEEE Annu. Joint Conf. IEEE Computer and Communications Societies (INFOCOM'01)*, 2001, pp. 1655–1663.

[18] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes in large scale sensor networks," in *Proc. ACM/IEEE 9th Annu. Int. Conf. Mobile Computing and Networking (MobiCom'03)*, 2003, pp. 81–95.

[19] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proc. IEEE Global Communications Conf. (GLOBECOM'01)*, 2001, pp. 2926–2931.

[20] C. Savarese and J. Rabaey, "Robust positioning algorithms for distributed ad hoc wireless sensor networks," presented at the USENIX Annu. Technical Conf., Monteray, CA, Jun. 2002.

[21] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *Proc. ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, 2002, pp. 105–111.

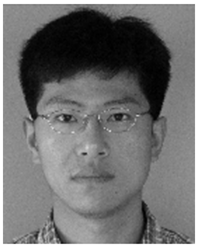
[22] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proc. IEEE Annu. Joint Conf. IEEE Computer and Communications Societies (INFOCOM'03)*, 2003, pp. 1734–1743.

[23] T. Ng and H. Zhang, "Predicting the Internet network distance with coordinates-based approaches," in *Proc. IEEE Annu. Joint Conf. IEEE Computer and Communications Societies (INFOCOM'02)*, 2002, pp. 170–179.

[24] J. Nelder and R. Mead, "A Simplex method for function minimization," *Comput. J.*, vol. 7, pp. 308–313, 1965.

[25] Matlab [Online]. Available: http://www.mathworks.com/

[26] Intel [Online]. Available: http://www.intel.com/research/exploratory/heterogeneous.htm



Hongyi Wu (M'02) received the B.S. degree in scientific instruments from Zhejiang University in 1996, the M.S. degree in electrical engineering and the Ph.D. degree in computer science from State University of New York (SUNY) at Buffalo in 2000 and 2002, respectively.

He is currently an Assistant Professor at the Center for Advanced Computer Studies (CACS), University of Louisiana (UL) at Lafayette. His research interests include wireless mobile ad hoc networks, wireless sensor networks, next generation cellular systems, and integrated heterogeneous wireless systems. He has published more than two dozens of technical papers in leading journals and conference proceedings.

Dr. Wu has served as chair and technical committee member of several IEEE conferences, and a guest editor of ACM MONET special issue on Integration of Heterogeneous Wireless Technologies. He received an NSF CAREER Award in 2004. He has been a member of the Association for Computing Machinery (ACM) since 2002.

Chong Wang (S'00) received the B.Sc. degree in physics from Fudan University, China, and the M.Sc. degree in telecommunications from the University of Louisiana at Lafayette. He is currently pursuing the Ph.D. degree in computer science at the University of Louisiana at Lafayette, where he works in the Mobile Computing and Wireless Communication Systems Laboratory.

Nian-Feng Tzeng (S'85–M'86–SM'92) received the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign.

Since 1987, he has been with Center for Advanced Computer Studies, the University of Louisiana at Lafayette, where he is currently a Professor. His current research interest is in the areas of computer communications and networks, high-performance computer systems, parallel and distributed processing, and fault-tolerant computing.

Dr. Tzeng was on the editorial board of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 1998–2001, and on the editorial board of the IEEE TRANSACTIONS ON COMPUTERS, 1994–1998. He served as a Distinguished Visitor of the IEEE Computer Society, 1994–1997, and was the Chair of Technical Committee on Distributed Processing of the the IEEE Computer Society from 1999 until 2002. He was on the technical program committees of various conferences and served as the Technical Program Chair of the 10th International Conference on Parallel and Distributed Systems, July 2004. He is the recipient of the Outstanding Paper Award of the 10th International Conference on Distributed Computing Systems, May 1990. He received the University Foundation Distinguished Professor Award in 1997.