ARTICLE IN PRESS

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Ad Hoc Networks xxx (2003) xxx–xxx

Ad Hoc Networks

www.elsevier.com/locate/adhoc

Position and orientation in ad hoc networks

Dragoş Niculescu *, Badri Nath

Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA

Received 14 June 2003; accepted 20 July 2003

Abstract

Position and orientation information of individual nodes in ad hoc networks is useful for both service and application implementation. Services that can be enabled by availability of position include routing and querying. At application level, position is required in order to label the reported data in a sensor network, whereas position and orientation enable tracking.

Nodes in an ad hoc network may have local capabilities such as the possibility of measuring ranges to neighbors, angle of arrival (AOA), or global capabilities, such as GPS and digital compasses. This article investigates the possibility of using local capabilities to export global capabilities using a distributed, localized, hop by hop method. We show how position and orientation of all the nodes in a connected ad hoc network can be determined with a small fraction of landmarks that can position/orient themselves, given that all nodes have some combination of local capabilities.

Keywords: Ad hoc networks; Positioning; Orientation; Multimodal; AOA; APS; Exporting capabilities

1. Introduction

The main features of new adhoc networks are the large number of unattended nodes with varying capabilities, lack or impracticality of deploying supporting infrastructure, and high cost of human supervised maintenance. What is necessary for these types of networks is a class of algorithms which are scalable, tunable, distributed, and easy to deploy. With recent advances in small device architectures [1], it can be foreseen that cheap, or even disposable nodes, will be available in the future, enabling an array of new agricultural, meteorological and military applications. These large networks of low power nodes face a number of challenges: cost of deployment, capability and complexity of nodes, routing without the use of large conventional routing tables, adaptability in front of intermittent functioning regimes, network partitioning and survivability. It is a given that in many of these networks, due to considerations of cost, size, and power requirements, individual nodes will not have full position and orientation capabilities. A general question is how to export capabilities to various nodes in the network so that the overall capability can be increased in the network. For example, many ad hoc network applications and protocols assume the knowledge of geographic position of nodes for routing and sensing. However, not all nodes have the capability of locally determining their position by means

* Corresponding author.
E-mail addresses: dnicules@cs.rutgers.edu (D. Niculescu), badri@cs.rutgers.edu (B. Nath).

1570-8705/$ - see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/S1570-8705(03)00051-9

of GPS. Finding position without the aid of GPS in each node of an ad hoc network is important in cases where the GPS service is either not accessible, or not practical to use due to power, form factor or line of sight conditions such as indoor sensors, sensors hidden under foliage, etc. A similar argument holds for orientation as compasses face erratical behavior in the vicinity of large metal objects or electrical fields. Orientation, or heading, is used in remote navigation, or remote control of specialized sensors, such as directional microphones or cameras. In this article, we address the problem of self positioning and orientation of the nodes in the field, which may provide a general framework for exporting capabilities in a network where more capable nodes cooperate in dispersing information to less capable nodes.

What is necessary for ad hoc deployment of temporary networks is a method similar in capability to GPS and magnetic compasses, without requiring extra infrastructure, or extensive processing capabilities. What we propose is a method by which nodes in an ad hoc networks collaborate in finding their position and orientation under the assumptions that a small fraction of the network has only the position capability. A compass is not necessary in any node, but if it is available, either at the landmarks, or everywhere, it will enhance the accuracy of the positioning algorithm. Previous positioning methods used so far used either TDOA based ranging, like in Cricket [2] and AhLOS [3], signal strength (RADAR [4], APS [5]), and angle of arrival (Cricket [6], APS/AOA [7]).

One scenario involving sensor networks frequently mentioned in literature is that of aircraft deployment of sensors followed by in flight collection of data by simply cruising the sensor field. This and other meteorological applications are implicitly assuming that the data provided by the sensor is accompanied by the sensor's position. It is thus possible to attach the sensed information to a geographical map of the monitored region. If this is an absolute necessity for making sense of the observed data, accurate position might also be useful for routing and coordination purposes. For some ad hoc networks, algorithms such as Cartesian routing [8], geocast [9], or TBF [10] enable routing with reduced or no routing tables at all, and are appropriate for devices like the Rene mote [1], with only half a kilobyte of RAM. An improvement that can be applied to some ad hoc routing schemes when position is available, Location Aided Routing [11] limits the search for a new route to a smaller destination zone. Our positioning and orientation algorithm is appropriate for indoor location aware applications, when the network's main feature is not the unpredictable, highly mobile topology, but rather temporary and ad hoc deployment. These networks would not justify the cost of setting up an infrastructure to support positioning, like proposed in [12,4], or [2].

The orientation and positioning problems have been extensively studied in the context of mobile robot navigation, however, many methods proposed by the robotics community make extensive use of image processing and preset infrastructure, such as "recognizable" landmarks. Our aim is a positioning method that is robust, but relies on less computational resources and on less infrastructure.

The rest of the paper is organized as follows: the next section describes the assumptions of the problem and the assumed capabilities of the nodes. Section 3 reviews some popular methods used in positioning. Section 4 presents our approach, reviewing several forwarding schemes specific to various hardware combinations, and Section 5 discusses some error control issues. Sections 6 and 7 present simulation results of some of the schemes, and discuss some mobility related issues, and Section 8 summarizes with some concluding remarks.

2. Node capabilities

The network is a collection of ad hoc deployed nodes such that any node can only communicate directly with its immediate neighboring nodes within radio range. In the ideal case, when radio coverage of a node is circular, these networks are modeled as fixed radius random graphs.

Four capabilities are considered: self positioning, self orienting, ranging and AOA. Self positioning can be achieved by the means of GPS, or, if the application permits, by manual input. Orientation is given by a digital compass, which is available in small form factors with high precision

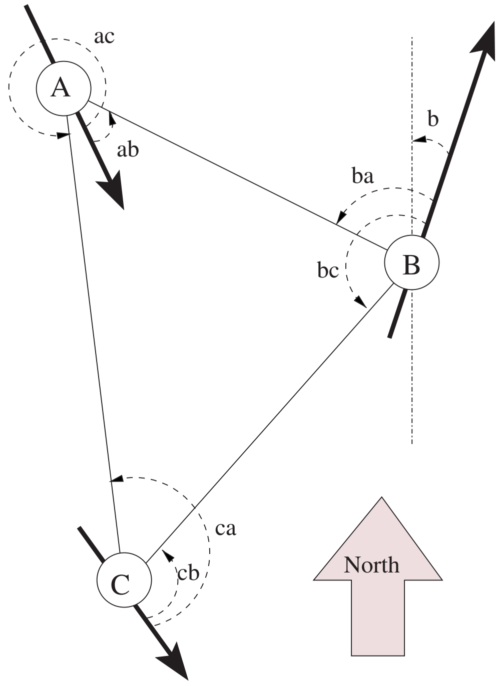

Fig. 1. Nodes with AOA capability.

(<1°). Range measurements provide the possibility for a node to measure distance to neighbors, for example, in Fig. 1, node A could measure distances AB and AC, but it could also obtain BC, after local conversation with B or C. For AOA capability, each node in the network is assumed to have one main axis against which all angles are reported and the capacity to estimate with a given precision the direction from which a neighbor is sending data. We assume that after the deployment, the axis of the node has an arbitrary, unknown heading, represented in Fig. 1 by a thick black arrow. The term *bearing* refers to an angle measurement with respect to another object. In our case, the AOA capability provides for each node bearings to neighboring nodes with respect to a node's own axis. A *radial* is a reverse bearing, or the angle under which an object is seen from another point. We will use the term *heading* with the meaning of bearing to north, that is, the absolute orientation of the main axis of each node. In Fig. 1, for node B, bearing to A is $\widehat{ba}$, radial from A is $\widehat{ab}$, and heading is $\hat{b}$. Node A "sees" its neighbors at angles $\widehat{ac}$ and $\widehat{ab}$, and has the possibility of inferring one angle of the triangle, $\widehat{CAB} = \widehat{ac} - \widehat{ab}$. For consistency all angles are assumed to be measured in trigonometric direction. Node A can also infer its heading, if heading of one of the neighbors, say B, is known. If node B knows its heading (angle to the north) to be $\hat{b}$, then A may infer its heading to be $\hat{a} = 2\pi - (\widehat{ba} + \pi - \widehat{ab}) + \hat{b}$. This in fact a way to export the compass capability from B to A. If however, no compass is available in any node, but each node knows its position, heading can still be found because the orientations of the sides of the triangle can be found from positions of the vertices. For example, if node B knows its own position, and obtains the bearing $\widehat{ba}$ to another known point A, it may obtain the angle AB makes with the vertical, and the orientation $\hat{b}$.

Any combination of the mentioned capabilities may be available in every node, but we will generally assume that there is only a small fraction of nodes that can position themselves, called landmarks. The other global capability, orientation can be reasonably assumed at all nodes, but it is known to suffer interference from electrical and magnetic fields, as well as large metallic objects. Here we make two opposite assumptions: In the first, the compasses are available at all nodes, with good accuracy, which can greatly help the positioning algorithms. In the second, compasses are either not available, or biased by local conditions, and the AOA measurements may provide better orientation accuracy. In this case, orientation can either be a byproduct of a positioning algorithm, or the goal of an orientation algorithm, both which have to be based on AOA.

2.1. AOA capable nodes

AOA capability is achieved by various technologies, some of which might be prohibitive in size and power consumption. A small form factor node that satisfies conditions outlined in the previous section has been developed at MIT by the Cricket Compass project [6]. Its theory of operation is based on both time difference of arrival (TDoA) and phase difference of arrival. Time difference is used in a similar manner in other projects, such as AhLOS [3] and Cricket Location [2],

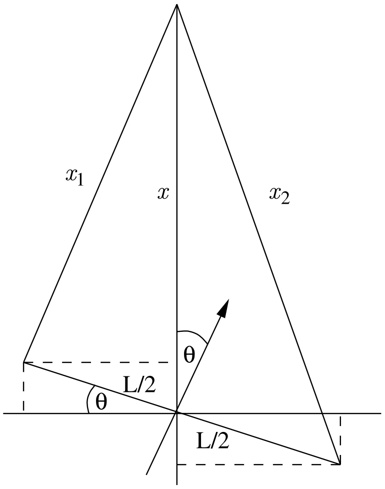

Fig. 2. Basic principle of Cricket Compass (adapted from [6]).

and is based on the six orders of magnitude difference between the speeds of sound and light. If a node sends a RF signal and an ultrasound signal at the same time, the destination node might infer the range to the originating node based on the time difference in arrivals. In order to get the angle of arrival, each node may use two ultrasound receivers placed at a known distance from each other, L (Fig. 2). By knowing ranges x_1, x_2, and distance L, the node is able to infer the orientation θ, with an accuracy of 5° when the angle lies between ±40°. Medusa, used in AhLOS project [3] from UCLA, is another wireless networked node with small size which makes use of several ultrasound receivers, but without actually employing them to detect angle of arrival. These incipient realizations prove that it is feasible to get AOA capability in a small package that would be appropriate for future pervasive ad hoc networks.

3. Triangulation, trilateration and VOR

3.1. Trilateration

In global positioning system (GPS [13]), *trilateration* (in reality multilateration) uses ranges to at least three known satellites to find the coordinates of the receiver, and four satellites to also find the clock bias of the receiver. For our ad hoc positioning purposes, we are using a simplified version of the GPS trilateration, as we only deal with distances, and there is no need for clock synchronization. The trilateration procedure starts with an a priori estimated position that is later corrected towards the true position. Let $\hat{r}_u$ be the estimated position, r_u the real position, $\rho_i = |r_i - r_u| + \epsilon_i$ and $\hat{\rho}_i = |r_i - \hat{r}_u| + \hat{\epsilon}_i$ the respective ranges to the GPS i. The distance equation to each satellite is $\rho_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2}$. The correction of the range, $\Delta\rho$ is approximated linearly using Taylor expansion. If $\widehat{J}_i$ is the unit vector of $\hat{\rho}_i$, $\widehat{J}_i = (r_i - \hat{r}_u)/|r_i - \hat{r}_u|$ and $\Delta r = \hat{r}_u - r_u$, then the approximate of the correction in the range is $\Delta\rho = \hat{\rho}_i - \rho_i \simeq \widehat{J}_i \cdot \Delta r + \Delta\epsilon$. Performing the above approximation for each satellite independently leads to a linear system in which the unknown is the position correction $\Delta r = [\,\Delta x \quad \Delta y\,]$:

$$\Delta\rho = J\Delta r,$$

$$\begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \Delta\rho_3 \\ \cdots \\ \Delta\rho_n \end{bmatrix} = \begin{bmatrix} \widehat{J}_{1x} & \widehat{J}_{1y} \\ \widehat{J}_{2x} & \widehat{J}_{2y} \\ \widehat{J}_{3x} & \widehat{J}_{3y} \\ \cdots & \cdots \\ \widehat{J}_{nx} & \widehat{J}_{ny} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \tag{1}$$

If an uncertainty σ_i is available for each range estimate, the above system is affected by the weights $W = \mathrm{diag}\{1/\sigma_i^2\}$, and its solution is

$$\Delta r = (J^T W J)^{-1} J^{\mathrm{T}} W \Delta\rho. \tag{2}$$

After each iteration, the corrections Δx and Δy are applied to the current position estimate. The iteration process stops when the correction in position is below a chosen threshold.

3.2. Triangulation

Triangulation is a positioning procedure that relies on angle measurements with respect to known landmarks. Historically, triangulation has been used more than trilateration for both navigation and geodetic purposes, because angles are more easily and precisely measured with simple methods.

The central observation suggesting that positioning using angles is possible is that the follow-

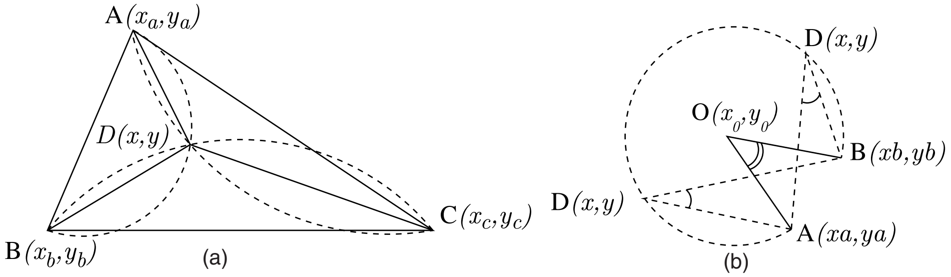

Fig. 3. Positioning using angles or ranges to landmarks. (a) Triangulation and trilateration. (b) Geometric locus of a point seeing two known landmarks at a given angle, is a circle. $\widehat{AOB} = 2\widehat{ADB}$.

ing: if we know the positions for the vertices of a triangle, and the angles at which an interior point "sees" the vertices, we can determine the position of the interior point. The difference from trilateration is that the interior point knows angles towards triangle sides instead of distances to vertices. In Fig. 3a, if beside coordinates of A, B and C, node D knows distances DA, DB and DC, it can use *trilateration* to infer its position. On the other hand, if it knows the angles $\widehat{BDA}$, $\widehat{ADC}$, and $\widehat{CDB}$, it can find its position using *triangulation*. This is done by finding the intersection of the three circles determined by the landmarks and the known angles. Information from several landmarks can be used to get a least square error solution, because in the general case, AOA measurements do not have perfect accuracy. There are several possibilities to compute this estimated point of intersection. The triangulation problem can be reduced to trilateration by some simple transformations. If, for example, a node D knows the angle to a pair of landmarks A and B, it may infer that its position is somewhere on the circle determined by the angle and the position of the two landmarks (Fig. 3b). What is fixed in this picture is the center of the circle, O, whose position may be determined when x_a, y_a, x_b, y_b and angle $\widehat{ADB}$ are known. This may help in transforming a triangulation problem of size n into a trilateration problem of size $\binom{n}{2}$ if for each pair of landmarks observed by a node we create an trilateration equation using x, y, x_0, y_0 and the radius of the circle as the distance. Another possibility is to form all triplets of obtained landmarks and find the center of the circumscribed circle for each such triplet and the unknown point D. This leads to the solving of $\binom{n}{3}$ trilateration problems of size 3, one for each circle, with $\binom{n}{3}$ solution points. For small numbers of landmarks, the $\binom{n}{3}$ has a similar CPU complexity as the $\binom{n}{2}$ one. However, the problems solved are all of size 3, thus requiring much less memory, whereas the $\binom{n}{2}$ approach needs to handle $n^2/2 \times 2$ sized matrices. A solution linear in the number of landmarks n, proposed in [14], makes efficient use of the representation of landmarks as complex numbers. In our simulation we used the simple $\binom{n}{3}$ implementation, as it gives the same quality estimates as the linear solution presented in [14], but it is much more simple to implement and has a low penalty for small n.

3.3. VOR

Another method of positioning using angles is VOR (VHS Omni-directional Ranging), which is currently still the main aid for aircraft navigation. Its principle is very simple: a landmark sends two signals, one that is periodic and omni-directional, while the another one is directional and rotating about the landmark. The airborne equipment receives both signals, and interprets the difference between the signals as a radial from the station. The coordinates of the landmark are known, therefore placing the mobile anywhere on a given line. A second VOR reading provides a second line to be intersected with the first. Given (x_i, y_i, r_i) the coordinates and the radial to the landmark i, a node can build the equation of the line $a_ix + b_iy = c_i$ on which it places itself.

$$\begin{aligned}
&\textit{if}\ \cos(r_i) = 0\\
&a_i = 1; \quad b_i = 0; \quad c_i = x_i\\
&\textit{else}\\
&a_i = \tan(r_i); b_i = -1; \quad c_i = -y_i + x_i \tan(r_i)
\end{aligned}$$

Combining all such lines to landmarks, the linear system to be solved for a location is

$$[a^{\mathrm{T}}|b^{\mathrm{T}}]\begin{bmatrix} x \\ y \end{bmatrix} = [c^{\mathrm{T}}]. \tag{3}$$

This approach is less expensive computationally, for n landmarks requiring just a weighted least square linear system solving. Also, it does not depend on an initial good guess of the solution that is necessary in iterative solving of nonlinear systems. What makes it slightly different form the previous one is the fact that the landmark should be equipped with a compass, so that it reports all radials against a well known direction, such as north. Trilateration and triangulation do not require any compass at all, but still provide positioning (and orientation if AOA is used) for all the nodes.

4. Ad hoc positioning system (APS) algorithm

The problem to be solved is: *given imprecise bearing/range measurements to neighbors in a connected ad hoc network where a small fraction of the nodes have self positioning capability, find orientation and position for all nodes in the network*. The difficulty of the problem stems from the fact that the capable nodes (landmarks) comprise only a small fraction of the network, and most regular nodes are nodes are not in direct contact with enough landmarks. What we are looking for is a hop by hop method to export capabilities from the capable nodes to the regular ones.

APS [5,7] is a hybrid between two major concepts: distance vector (DV) routing, and beacon based positioning (GPS). What makes it similar to DV routing is the fact that information is forwarded in a hop by hop fashion, independently with respect to each landmark. What makes it similar to GPS is that eventually each node estimates its own position, based on the landmark readings it gets. The original APS concept has been shown to work using range measurements [5], and angle measurements [7]. We extend the proposed methods by considering multimodal and heterogenous capabilities. While an arbitrary combination of capabilities may not guarantee support for a positioning scheme, we identify a number of positioning schemes that are appropriate for certain combinations of capabilities. For example if ranging or AOA capabilities are available, we expect them to be present at all nodes of the network. Compasses may also be present throughout the network, but it is always reasonable to assume they are at least present at landmarks.

What we propose is a method to forward orientation/range so that nodes which are not in direct contact with the landmarks can still infer their orientation/range with respect to the landmark. Here, *orientation* means bearing, radial, or both, and *range* means straight line distance, or an estimation of it. First, we shortly review the most basic method, *DV-hop* [5], that does not rely on any capability except connectivity. Then, the range based propagation methods, *DV-distance* [5], *Euclidean* [5], and *DV-coordinate* [15]. The AOA based propagation methods are *DV-bearing* [7], which allows each node to get a bearing to a landmark, and *DV-radial* [7], which allows a node to get a bearing *and* a radial to a landmark. When ranging and AOA are available together at all nodes, we propose *DV-position*, a method that positions nodes in a single propagation stage.

All propagations work very much like a mathematical induction proof. The fixed point: nodes immediately adjacent to a landmark get their bearings/radials/ranges directly from the landmark. The induction step: assuming that a node has some neighbors with orientation/range for a landmark, it will be able to compute its own orientation/range with respect to that landmark, and forward it further into the network. What remains to be found is a method to compute this induction step, for any combination of local capabilities: none, ranging, AOA, AOA + compass, AOA + ranging, AOA + ranging + compass.

If for some reason a node does not get enough ranges/orientations in order to triangulate/trilaterate, it could wait for the neighbors to successfully position themselves and either use local measurements in order to get a position, or simply use a weighted average with the positions of those neighbors. Even if position is available at a node, smoothing with the positions of the neighbors has been reported beneficial in certain cases [3,16,17].

4.1. *DV-hop propagation method*

This is the most basic scheme, and it comprises of three non-overlapping stages.

1. First, it employs a classical distance vector exchange so that all nodes in the network get shortest paths, in hops, to the landmarks. Each node maintains a table $\{X_i, Y_i, h_i\}$ and exchanges updates only with its neighbors.
2. In the second stage, after it cumulates distances to other landmarks, a landmark estimates an average size for one hop, which is then deployed as a correction to the nodes in its neighborhood.
3. When receiving the correction, an arbitrary node may then have estimate distances to landmarks, in meters, which can be used to perform the trilateration, which constitutes the third phase of the method. The correction a landmark (X_i, Y_i) computes is

$$c_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_i}, \quad i \neq j, \text{ all landmarks } j. \tag{4}$$

The drawbacks of *DV-hop* are that it will only work for isotropic networks, that is, when the properties of the graph are the same in all directions, so that the corrections that are deployed reasonably estimate the distances between hops. For uniform networks, [18] shows that the expected progress in the MFR geographic forwarding policy can be estimated locally based only on the N_i, the degree of the nodes as

$$\hat{c}_i = r\left(1 + \mathrm{e}^{-N_i} - \int_{-1}^{1} \mathrm{e}^{-\frac{N_i}{\pi}(\arccos t - t\sqrt{1-t^2})}\,\mathrm{d}t\right). \tag{5}$$

This value is a good approximation of our correction in uniform networks, eliminating the need for second stage, as each node estimates c_i based on its number of neighbors only (N_i). The *DV-hop* idea was independently explored in the context of amorphous computing by Nagpal [16], who has also given an upper bound on the accuracy as $\frac{1}{4}\pi r/N_{\mathrm{avg}}$.

The advantages of the *DV-hop* method are its simplicity and the fact that it does not depend on range or angle measurement error. Measuring ranges to neighbors however, turns out to be of some advantage for the next propagation schemes, which employs another stage of trilateration for the unsuccessful nodes, using successful neighbors as landmarks.

4.2. *DV-distance propagation method*

This method is similar with the previous one with the difference that distance(range) between neighboring nodes is measured in meters rather than in hops. As a metric, the distance vector algorithm is now using the cumulative traveling distance, in meters. On one hand the method is less coarse than *DV-hop*, because not all hops have the same size, but it is sensitive to measurement errors.

4.3. *Euclidean propagation method*

The third scheme works by propagating the estimated *euclidean* distance to the landmark, so this method is the closest to the nature of GPS.

(1) An arbitrary node A needs to have at least two neighbors B and C which have estimates for the landmark L (Fig. 4). A also has measured estimates of distances for AB, AC, and BC, so there is the condition that: either B and C, besides being neighbors of A, are neighbors of each other, or A knows distance BC, from being able to map all its neighbors in a local coordinate system.

In any case, for the quadrilateral $ABCL$, all the sides are known, and one of the diagonals, BC is also known. This allows node A to compute the second diagonal AL, which in fact is the euclidean

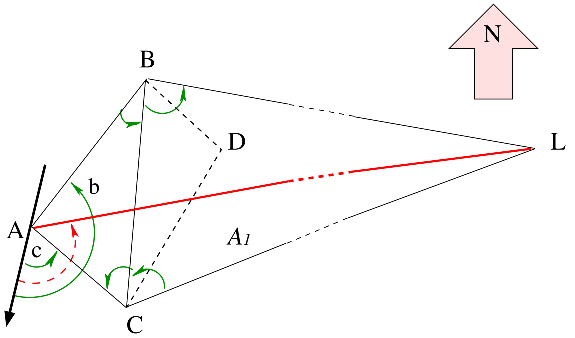

Fig. 4. Node A estimates its bearing/radial/range to L by corroborating B's and C's estimates.

distance from A to the landmark L. It is possible that A is on the same side of BC as L—shown as A_1 in the figure—case in which the distance to L is different. The choice between the two possibilities is made locally by A either by voting, when A has several pairs of immediate neighbors with estimates for L, or by examining relation with other common neighbors of B and C. The latter choice requires the (strong) assumption that if two nodes are in proximity they should be able to communicate. When node A has to decide whether or not is at A_1, it may enlist the help of another common neighbor of B and C—D, that may already have an estimate to L, but does not have a direct link to A. This lets A decide that, since is separated from D by BC, it must also be separated from L. From the simulations, we found that each method solves about the same amount of ambiguities in an uniform density network.

If it cannot be chosen clearly between A and A_1, an estimate distance to L won't be available for A until either more neighbors have estimates for L that will suit voting, or more second hop neighbors have estimates for L, so a clear choice can be made. Once the proper choice for A is available, the actual estimate is obtained by applying Pithagora's generalized theorem in triangles ACB, BCL, and ACL, to find the length of AL. An error reduction improvement applicable for the *Euclidean*, but not for the other methods is for a landmark to correct all the estimates it forwards. It uses the true, GPS obtained coordinates, instead of relying on the measurement based received values. Another advantage is that from the estimation of AL, if uncertainties of all other ranges are known, the uncertainty in AL can also be computed at the time of forwarding, and thus provide the GPS trilateration with weights that increase accuracy. Having an estimate for the obtained range may, in certain cases, reduce the average positioning error by up to 50% (when compared with positioning error obtained in [5]).

$$\cos(\alpha) = \frac{AB^2 - AC^2 - BC^2}{2 \cdot AC \cdot BC},$$

$$\cos(\beta) = \frac{BL^2 - BC^2 - CL^2}{2 \cdot CL \cdot BC},$$

$$AL^2 = AC^2 + CL^2 - 2 \cdot AC \cdot CL \cos(\beta \pm \alpha),$$

$$\sigma_{AL}^2 = \sum \left(\frac{\partial AL}{\partial e}\right)^2 \sigma_e^2,$$
$$e = AC, CL, LB, BA, BC. \tag{6}$$

The uncertainty σ_{AL} is then propagated together with the actual length AL to nodes which are farther from the landmark L. The advantage of this method is that it provides better accuracy under certain conditions, and there are no correction to be deployed later.

(2) Once a node has ranges to three landmarks, it may, by itself, estimate its position, taking into consideration the uncertainty associated with each range.

4.4. DV-bearing and DV-radial propagation methods

(1) Assume node A (Fig. 4) knows its bearings to immediate neighbors B and C (angles $\hat{b}$ and $\hat{c}$), which in turn know their bearings to a faraway landmark L. The problem is for A to find its bearing to L (dashed arrow). If B and C are neighbors of each other, then A has the possibility to find all the angles in triangles ΔABC and ΔBCL. But this would allow A to find the angle $\widehat{LAC}$, which yields the bearing of A with respect to L, as $\hat{c} + \widehat{LAC}$. Node A might accept another bearing to L from another pair of neighbors, if it involves less hops than the pair B–C. A then continues the process by forwarding its estimated bearing to L to

its neighbors which will help farther away nodes get their estimates for L. Forwarding orientations is done in a fashion similar to distance vector routing algorithms. In our case, the landmarks are the ones starting the update messages that are propagated throughout the network, for each landmark independently. Once node A finds its bearings to at least three landmarks that are not on the same line or on the same circle with A, it can infer its position using one of the methods outlined in Section 3.

If the radial method is to be used, a similar argument holds, with the difference that now A needs to know, besides bearings of B and C to L, the radials of B and C from L. If the angle $\widehat{BLN}$ (radial at B) is also known, then the angle $\widehat{ALN}$ (radial at A) can also be found since all angles in both triangles are known. The actual downside for this method is in the increased amount of signaling—nodes B and C forward two values per landmark (bearing and radial) instead of just one, as in the bearing based method. If a compass would be available in every node, the two methods would in fact become identical because when all angles are measured against the same reference direction (north, for example), bearing $= 2\pi -$ radial.

The algorithm has similar signalling overhead behavior with *Euclidean*, and is roughly a TTL limited flooding per landmark. The following table summarizes for each method the required node capabilities and associated signaling-accuracy tradeoffs. "More" signalling refers to the fact that two values are needed per landmark, whereas "less" sends only one. In the case of a large existing packet overhead, one extra value may be of diminished importance. The accuracy of the two propagated methods will be quantified more precisely in the simulation section (6).

Compass	Method	Signaling	Accuracy
Nowhere	*DV-bearing*	Less	Less
Only at Landmarks	*DV-bearing* *DV-radial*	Less More	Less More
All nodes	*DV-radial*	Less	More

(2) The positioning stage uses triangulation for *DV-bearing*, and VOR for *DV-radial*. Both methods allow for the use of uncertainties, in a similar fashion with *Euclidean*.

4.5. *DV*-coordinate propagation method

The fifth method considered, suggested by an idea proposed in [19], was explored in [15] for ranges only, and can be extended for the multimodal case (AOA, compass, accelerometer, and their combinations). The method also aims at propagating ranges, bearings or orientations, but it requires two preprocessing stages that have to complete before the DV propagation starts.

(1) Assuming that second hop information is available, as in case of *Euclidean*, it is possible for a node to establish a local coordinate system for which the node itself is the origin. In Fig. 5, node A, based on the ranges from itself to its neighbors and the ranges between those neighbors, can choose some set of axes x_a, y_a and locally place all the immediate neighbors. The system may be built by solving a nonlinear optimization problem to find all nodes positions given all the ranges and the fact that $A = (0, 0)$, or by choosing two neighbors as indicators for axes of coordinates, and incrementally add nodes. We chose the second approach, since the nonlinear optimization might not scale to higher degrees and needs a good starting

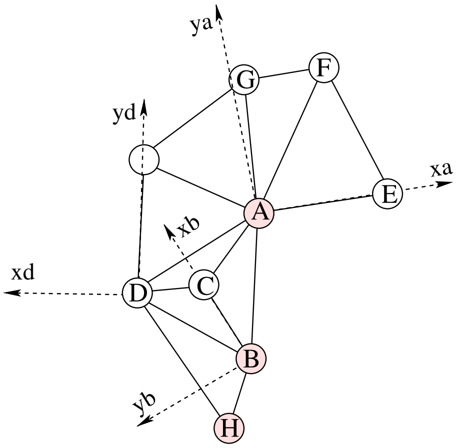

Fig. 5. Local coordinate systems.

point not to fall in local minima. In case of A, E is chosen as an indicator for x_a axis and F for y_a. Using known ranges to eliminate ambiguities, all immediate neighbors of A are added to the local coordinate system. In a similar fashion, if bearings to neighbors are available, the local system can be built based on an initial arbitrary scale. A chooses for example the distance AE to be of unit size, and by knowing all the angles in triangles around itself, it may position all the nodes in an incremental fashion. Availability of AOA and ranging makes the setup of the coordinate system even simpler, since angles and sides in all the triangles around A are known. Every node in the network independently builds its own coordinate system centered at itself, with a procedure that is specific to the hardware available.

(2) The next preprocessing step is registration with the neighbors. If node A sends the coordinates of G to B, B has to translate those coordinates in its own system. The transformation matrix that achieves this translation is obtained through the process of registration [20]. Nodes A and B each have coordinates of nodes A, B, C and D in both coordinate systems, which are used to compute associate transformation matrices used for translation from one system to the other. The registration process is specific to the local capabilities available. Table 1 indicates all the possible combinations of node capabilities, and the transformations involved in the registration process (T = translation, R = rotation, S = scaling, M = mirroring). When mirroring is indicated in parenthesis, it can only happen as a result of a node being deployed upside down, not from the randomness in starting the local coordinate system. When only ranging is used, mirroring is possible regardless of the pose of the node, depending on the nodes chosen as indicators for local axes. In all the other cases, since AoA is assumed to report angles in the same (trigonometric) direction for all nodes, mirroring between two local coordinate system appears only when one node is flipped, situation which can be robustly detected by a digital accelerometer.

Table 1
Hardware capabilities and associated transformations

Capability	Transformations
Range	T, R, M
AoA	T, R, S, (M)
AoA + Compass	T, S, (M)
AoA + Range	T, R, (M)
AoA + Range + Compass	T, (M)

Each coordinate received from a neighbor needs to be translated by a node in order to be consistent with its own coordinate system. Computing this matrices before hand is just an optimization choice applicable to static networks, otherwise registration can be performed on the fly, whenever communication between two neighbor occurs. The complexity of registration is linear in the number of common neighbors used for registration, 4 in this example, and cubic in the number of coordinates, 2 in the plane. This preprocessing step insures that when a node has some estimate for a landmark, it can immediately be translated by the neighboring nodes in their own coordinate systems.

(3) Getting back to DV propagation, instead of propagating the actual euclidean distance to the landmark, two coordinates are sent, designating the *coordinates of the landmark* in the coordinate system of the sending node. If node B receives coordinates of some landmark from A, it first translates those coordinates in its own system using the appropriate translation matrix, computed in the preprocessing step.

(4) A node that gathers a number of landmarks in its own coordinate system now has two possibilities of positioning itself. First, it can simply compute the ranges in its own coordinate system and use them in the global system to solve the trilateration problem. Second, since it has coordinates for the landmarks both in its own local system and in the global system, it may use the registration procedure to find a transformation matrix from the local system to the global one. The projection of 0,0 through this matrix would yield global coordinates for the node. In our simulations, we found these two methods to yield similar performance.

DV-coordinate propagation method may use the registration residual error as a measure of uncertainty. At each propagation step, uncertainty in obtained coordinates is amplified by the local uncertainty resulted from the registration to produce a new uncertainty for the coordinates that get propagated to further nodes.

4.6. DV-position and DV-compass propagation methods

If ranging and AOA are both available at all nodes, *DV-bearing* can be used in parallel with any of the range based methods, and the final position estimation can be inferred from the two estimates. *DV-coordinate* allows the use of several capabilities in building local coordinate systems. *DV-position* is another method that makes simultaneous use of both capabilities for increased accuracy. *DV-compass* is a simplification of *DV-position*, that only exports the orientation capability by assuming AOA throughout the network, and compasses only at the landmarks.

If compasses are available at landmarks, neighbors of landmarks can position themselves in one step using *DV-position*. In Fig. 6, if landmark B has a compass and can measure angle $\hat{b}$, and also AOA capability to measure the angle $\widehat{ba}$ at which A is seen, then it is possible to compute the equation line on which A is placed. If the range AB is known, node A can find its own position. The absolute orientation of A can also be found, as $2\pi - (\widehat{ba} + \pi - \widehat{ab}) + \hat{b}$. Using further propagation, node A can then behave as a less accurate landmark for further away nodes.

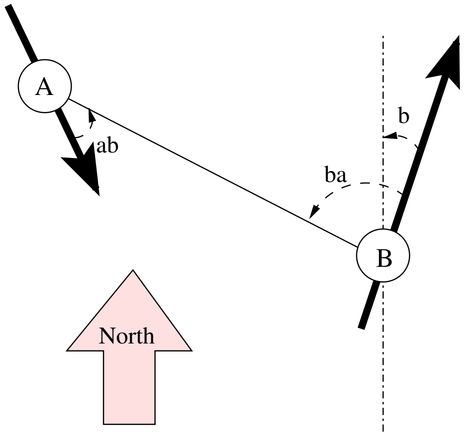

Fig. 6. *DV-position* propagation method.

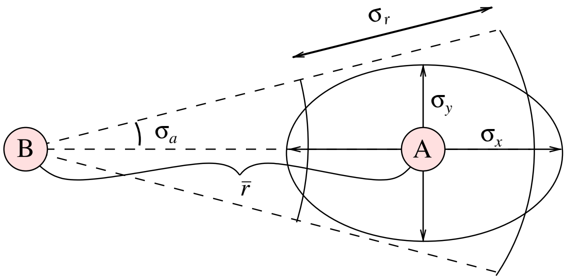

Fig. 7. Normal approximation of uncertainty.

In order to propagate the uncertainty at each step, we assume that each node's position has an associated uncertainty covariance matrix. In order to assume normal spatial errors, AOA errors and range errors are approximated, assuming their independence. In Fig. 7, assume the uncertainty at node B is U_B, and node A, using an AOA reading with uncertainty σ_a^2 and a range reading with uncertainty σ_r^2 infers its position with uncertainty U_A.

$$U_A = U_B + \begin{bmatrix} \overline{r_x} \\ \overline{r_y} \end{bmatrix}^\top \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \begin{bmatrix} \overline{r_x} \\ \overline{r_y} \end{bmatrix}. \tag{7}$$

When angular errors are small, the eigenvalues of the covariance matrix are given by $\sigma_x = \sigma_r$ and $\sigma_y = \|\bar{r}\| \sin \sigma_a$, where σ_a is the uncertainty in angle measurement, σ_r the uncertainty in range measurement, and $\bar{r}$ is the actual range measured. The eigenvectors $\overline{r_x}$ and $\overline{r_y}$ are given by the direction of the vector $\bar{r}$, respectively its normal. The advantage of expressing the uncertainty using a covariance matrix is that it can easily be cumulated as the position is propagated. The second advantage is that positions obtained from different landmarks can be merged in order to get the best position estimate, as well as an estimate of its uncertainty. If $[x_i \quad y_i]$ is the estimate from landmark i, and U_i is the uncertainty accumulated along the path from landmark i, the final uncertainty is

$$U = \left(\sum U_i^{-1}\right)^{-1} \tag{8}$$

and the position estimate:

$$[x \quad y] = \left(\sum [x_i \quad y_i] U_i^{-1}\right) U. \tag{9}$$

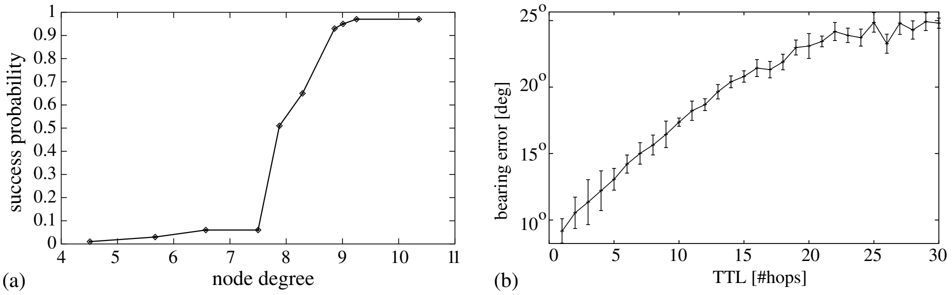

Fig. 8. (a) Probability for a node to satisfy conditions necessary for forwarding. (b) Error is compounded with propagation.

Intuitively, the uncertainty grows with each error affected measurement (7), but can be reduced if several landmarks are used (8). The actual position estimate is in fact a weighted estimate considering all the partial estimates and their uncertainties (9). *DV-compass* is the method of forwarding orientations, as a part of, or separately from *DV-position*. Similarly, the uncertainty in orientation cumulates as

$$D_A = D_B + \sigma_a^2.$$

If compasses are not available at landmarks, the position/orientation/uncertainty propagations all work in a similar fashion, but each in coordinate systems that are specific to individual landmarks. Each landmark works with a coordinate system indicated by its random direction obtained at deployment. The method becomes similar to the *DV-coordinate*, with the difference that at each step, regular nodes place themselves in coordinate systems that are landmark specific. In this situation, a node has to use its positions/orientations in various coordinate systems, in order to find its global estimate.

4.7. Network density

The question that arises in deployment of the network is what kind of node density is needed in order to achieve a certain condition with high probability. It has been conjectured [21] that many random graph properties exhibit phase transitions—sharp increases in the probability when the density increases beyond a certain point. For example, it has been proven that a random network in the plane needs a degree of about 6 in order to have complete connectivity with high probability. We expect the degree requirement to be higher for *DV-bearing*, *DV-radial*, *and Euclidean*, since more than simple connectivity is needed for two neighbors that are also neighbors of each other to be present for any given node. In Fig. 8a, we can see that when the mean degree of a node increases beyond 9, with a very high probability it will meet the conditions to forward orientation using *DV-bearing* or *DV-radial*. This data is empirically obtained by running our forwarding policy in a network of 1000 nodes with a single landmark and then count the number of nodes which get a bearing to it. Variation of the average degree is achieved by increasing the radio range of the nodes. In the case of a sensor network, it is often envisioned that the deployed density is higher than needed to allow for extension in battery life by tuning the duty cycle. This means that an initial degree of 9 might be tolerable (50% more nodes have to be deployed), as the normal functioning regime can be later lowered to 6, which has been shown to be good practical minimum for connectivity [21].

5. Error control

Being that all measurements are affected by errors, the forwarding actually amplifies and compounds smaller errors into larger errors. While some of the methods presented forward uncertainties along with estimates, some generic error control methods may be used to eliminate outlier measurements or aberrations produced by ambiguities. A number of simple techniques may be employed to reduce the propagation of such errors, including: avoiding inference based on small angles or on degenerate triangles (for *Euclidean, DV-bearing* and *DV-radial*), limiting the propagation of DV packets with a simple TTL scheme, and elimination of the outliers in position estimation. The fact that range and angle measurements are affected by error greatly influences the very core of the methods that are based on two common neighbors. As the environment we envision for this positioning algorithm is a low power, low communication capacity node, error control methods employed have to be lightweight. Together, the three mentioned methods achieve an error reduction of about 50%.

The first intuitive remark is that error cumulates with distance, because of the way bearings and ranges are propagated. We verified this fact for *DV-bearing* in a network of 200 nodes, by plotting the average bearing error as a function of distance in hops to the respective landmark (Fig. 8b). For *DV-position*, the covariance matrices are cumulated at each step, leading to uncertainty growing with the number of hops. Limiting the propagation of the DV packets using a TTL scheme is therefore a good idea not only for error control reasons, but also for reducing communication complexity. If TTL is infinite, each landmark is flooding the entire network with its coordinates, thus triggering propagation specific computation (bearing, radial or Euclidean) at every other node. TTL is the main feature that makes the proposed algorithms scalable. As long as enough landmarks can be acquired from the area allowed by the TTL, *the total size of the network does not influence per node communication overhead or the quality of the estimates*.

The next key observation is that small angles are more error prone than large angles. It is preferable to deal with equilateral triangles than with triangles that have two very acute angles and one obtuse angle. For example, the same angle error of 5° will make much more difference in a triangle with a true angle of 10° than in one with 60°. It is an analogue situation with the geometric dilution of precision (GDOP) in the GPS, in which the error amplification depends on the landmark constellation. To address this problem we use a threshold value to eliminate triangles in which small angles are involved. In Fig. 4, if any of the two triangles ΔABC and ΔBCL have small angles, A won't get to propagate orientation or range to L, to avoid a large amplification of possible small errors. There is a tradeoff between coverage and positioning error, and this results from the orientation range forwarding policy. A conservative policy would use a high threshold, limiting the computations with small angles but also limiting the propagation of ranges and orientations, and finally reducing coverage. A relaxed policy would propagate almost all angles, involving more errors, but would improve coverage. In Fig. 9a, when using *DV-bearing*, it is shown how varying from a very conservative forwarding policy (threshold $= 0.5 \simeq 28°$) to a very relaxed one (threshold $= 0.01 \simeq 0.5°$) achieves different levels of coverage (success rate) with different amounts of error. The positioning error represents the average

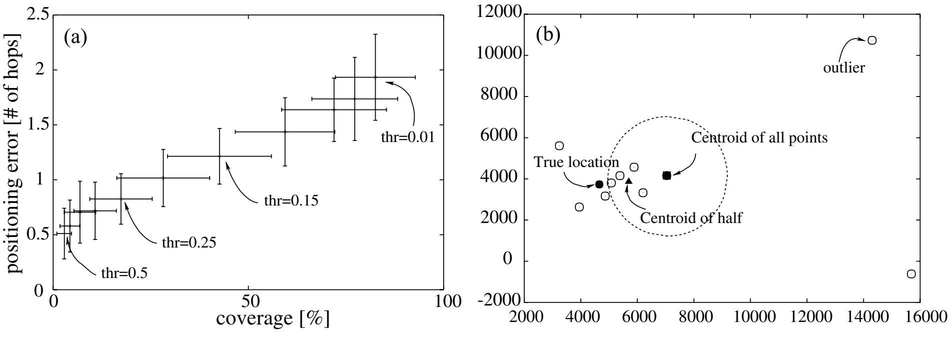

Fig. 9. (a) Angle threshold: tradeoff error-coverage. (b) Removing outliers from centroid computation.

distance in hops from the true position, obtained after propagation (Section 4.4) and triangulation (Section 3). The coverage represents the fraction of nodes successfully obtaining a position. The data is obtained by positioning with different thresholds, in the configuration shown in Fig. 10, with a TTL of 6, 20% landmarks and a high AOA measurement error ($\text{stddev} = 0.4 \simeq 23°$). This angular control of the error can be applied to all the methods that implicitly use triangles: *DV-bearing, DV-radial, Euclidean*, and *DV-coordinate*.

Another error control method, suggested in [14], refers to the position estimations obtained from the triplets of landmarks. In all the mentioned methods, several position estimations may be obtained, leading to the problem of combining them into one single estimate. While this can simply be the weighted centroid of the estimates, in practice it has been observed that large errors are clustered together. This is caused by propagation paths that pass through common points affected either by ambiguities (*Euclidean*), or large range/AOA errors. A robust method suggested in [14] is to first compute the centroid and then remove the outliers before recomputing a new centroid with the remaining points (Fig. 9b). There are more powerful methods available, such as data clustering and k-smallest enclosing circle, but they involve higher computational and memory complexities, which may not be applicable to most small networked nodes, such as sensor nodes.

6. Simulation

Simulations included here are only for *DV-bearing* and *DV-radial*, while *DV-hop, DV-dis-*

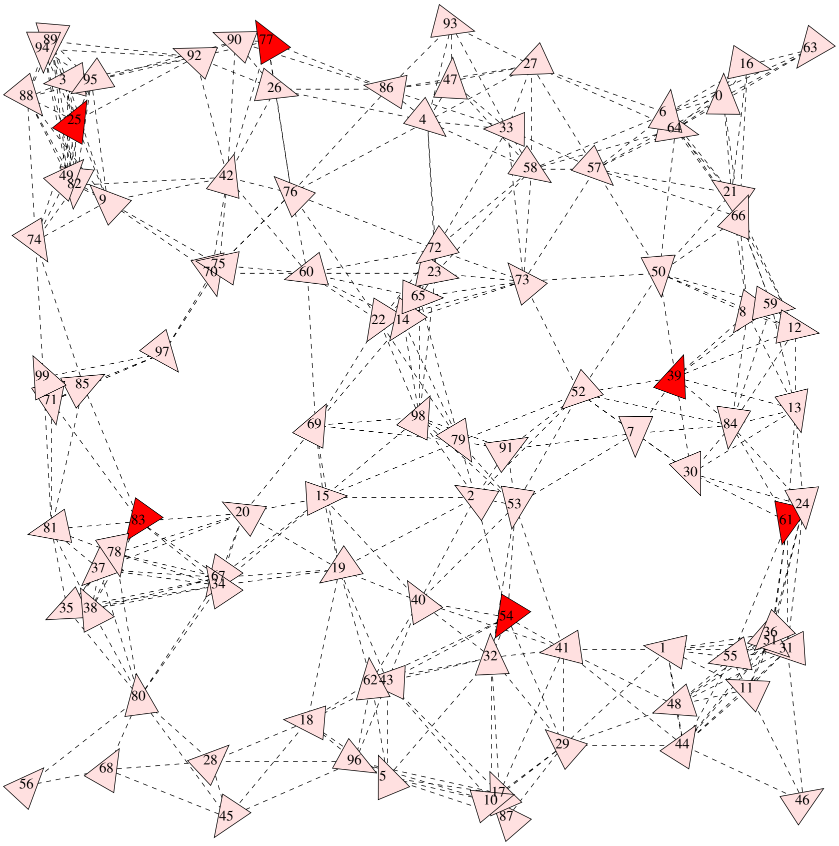

Fig. 10. Isotropic topology.

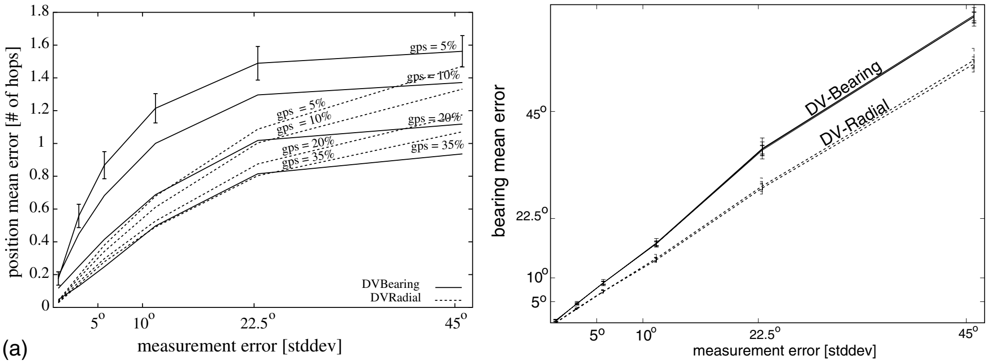

Fig. 11. (a) Positioning error. (b) Bearing error.

tance, and *Euclidean* can be found in [5], and *DV-coordinate* using ranges in [15]. We simulated an isotropic [1] map similar with the one in Fig. 10 (average degree = 10.5, diameter = 32), but with 1000 nodes, each having a random, but unknown heading. A fraction of nodes are landmarks, meaning that they have self positioning capability by an external method such as GPS. Gaussian noise is added to each AOA estimation to simulate measurement errors. Gaussian distribution has the property that 95% of the samples lie within 1.96 standard deviations from the mean. What this means for angle measurements is that if the standard deviation of the noise is for example $\frac{\pi}{8}$, then 95% of the measurements will be in the interval $\left(-\frac{\pi}{4};\frac{\pi}{4}\right)$ of the true bearing, thus giving a total spread of $\frac{\pi}{2}$ for bearing measurements. Performance will be evaluated based on the accuracy of positioning for non-landmark nodes, accuracy of heading, and percentage of the regular nodes which succeed the solving for a position (coverage). All the results presented in this section are averaged from 100 runs with different randomly distributed landmark configurations over the same network. Due to the fact that the proposed algorithms provide different tradeoffs, in order to produce comparable coverage we ran *DV-Bearing* with a TTL of 5 and *DV-radial* with a TTL of 4. In both cases the angle threshold was 0.35 (≃20°). All performance graphs indicate the standard deviation in selected points.

Positioning error (Fig. 11a) is represented relative to the maximum communication range of a node. An error of 1.0 means that the position resulted from the positioning algorithm is one (maximum sized) radio hop away from its true position. For *DV-Bearing*, this position is obtained from the bearings to landmarks, applying the triangulation method mentioned in Section 3. For *DV-Radial*, position is obtained from the radials by solving a linear system. On the horizontal axis of the graphs the standard deviation of the measurement noise is varied from 0 to $\frac{\pi}{4}$, and the several curves on each graph correspond to different landmark ratios. A larger number of landmarks improves both accuracy and precision, by solving a larger system for each positioning problem. For reasonable errors *DV-Radial* provides better positioning accuracy, and exhibits less dependence on the percentage of landmarks.

Bearing error (Fig. 11b) is the average error of the bearing to landmarks obtained by regular nodes after the orientation propagation phase stops. This is a primary measure of how the forwarding method compounds and propagates error. Because each landmark is treated independently, bearing errors are not affected by the number of landmarks available in the network. As expected, *DV-Radial* exhibits lower error, mainly because of the extra value that is forwarded.

[1] isotropic = having the same physical properties in all directions (connectivity, density, node degree, landmark distribution).

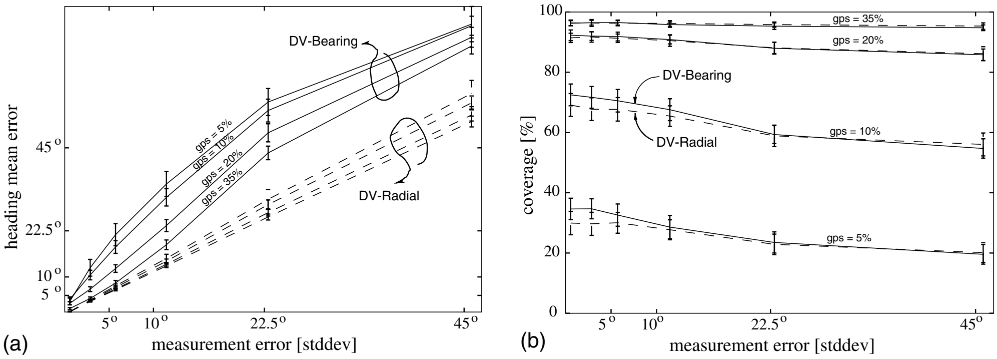

Fig. 12. (a) Heading error. (b) Coverage.

Heading is the angle between nodes axis and the north, as would be given by a compass. Heading error is therefore the error in the absolute orientation averaged over all nodes. In our simulation, it is obtained by each node after estimating a position. Heading error (Fig. 12a) is about double the bearing error, which is consistent with the results presented in [14].

Coverage (Fig. 12b) represents the percentage of non landmark nodes which are able to resolve for a position. The reasons for which a node does not get a position are: fewer than three landmarks accumulated (due to propagation errors), collinear or co-circular landmarks, or numerical instability in the system solving. We aimed for similar coverage for the two algorithms in order to compare the other performance metrics. Even if positioning is theoretically possible with two landmarks for *DV-Radial* and with three landmarks for *DV-Bearing*, in practice, due to angle errors compounding, a much higher number of landmarks might be needed.

The main observations to draw from simulations are the following: accuracy can be traded off for coverage by tuning the TTL and the threshold value. The TTL tradeoff is also between energy and coverage, as its reduction would lead to less energy spending but also to less coverage. Positions obtained are usable for applications such as Caresian routing [8], as it is showed in [5], with errors of similar scale. Bearing errors follow closely the measurement noise, but they can be further decreased using more sophisticated correction methods.

In order to evaluate the accuracy of positions and orientations for a realistic application, we devised a simple example in which a mobile traverses a fixed network and is sensed by nodes within a certain distance. Nodes are initially deployed randomly, with a fraction of them (20%) having the self positioning capability. After running the APS algorithm to infer their position and orientation, the nodes sensing the mobile report their position and the direction in which the mobile was observed. At a central location, reports from various nodes are aggregated to produce an estimate position of the mobile. Since both positions and directions reported by nodes are based on APS produced positions/orientations, and therefore affected by errors, and because there may be more than two reporting nodes, the estimate position of the mobile is obtained by solving an over-determined linear system, in order to minimize the square error. In Fig. 13, the original trajectory is shown with a dashed line, and the restored one with a solid line. Standard deviations are indicated for each sample point. While more complicated data fusion/prediction techniques (such as Kalman filters) may be used here to improve the estimated trajectory, the purpose of the example is merely to quantify the APS produced error in the position and orientation of the nodes, with no additional processing. The network used (Fig. 10) was an isotropic topology with 100 nodes, mean degree 8.18, 20 nodes of which have self positioning capability. The measurement error considered was white gaussian noise with a standard deviation of 0.08 radians, which is about

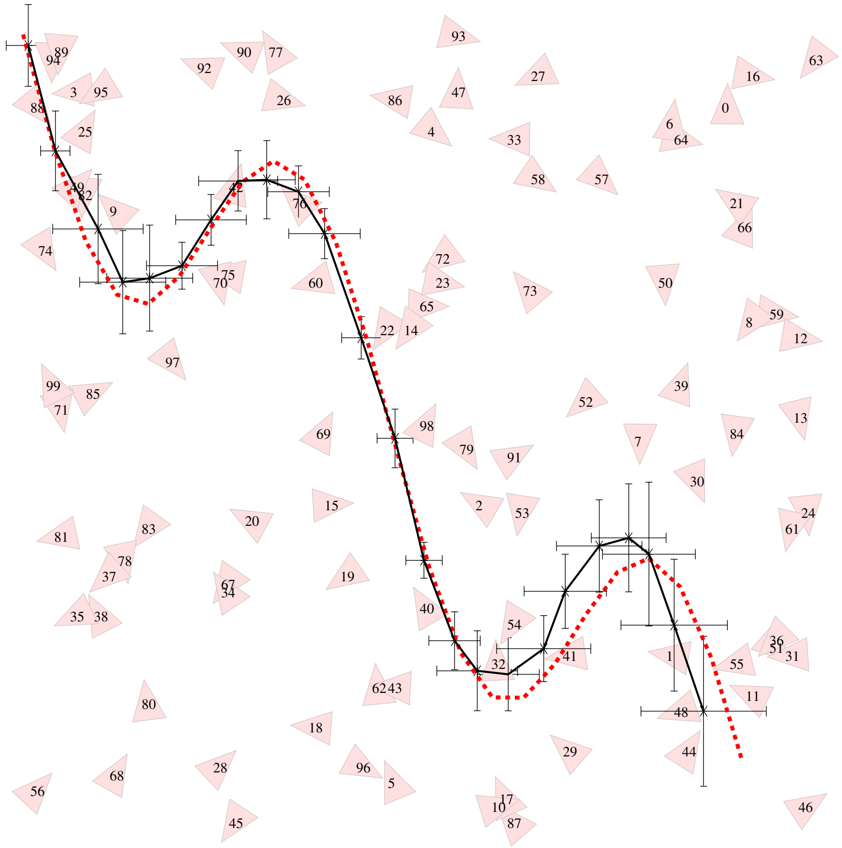

Fig. 13. Tracking example isotropic topology.

double the error of 5° achieved by the AOA nodes realized by the Cricket compass project [6]. The algorithm used to infer position and orientation is *DV-bearing*, which trades off some accuracy in order to work with less signaling and fewer capabilities (no compasses anywhere in the network). We assumed that the sensing distance is equal to communication radius, so that for each point we get about 6 or 7 readings. The sensing angle error is assumed to be 0, so that all the errors in the restored trajectory quantify the errors in our positioning algorithm (*DV-bearing*). It is interesting to note that estimations in the middle of the network are much more accurate that the ones at the edge (and this was verified with various other trajectories). The main cause for this is that an observation at the edge is obtained from angles which are clustered together in a small zone of the trigonometric circle—for example, a corner estimation would have all the angles in one single quadrant. In fact this is true about positions obtained by both algorithms. This would suggest that this class of algorithms (positioning, orientation, tracking) would run better when the border of the network is reduced in size, or is directly supported by preferential landmark placement.

7. Node mobility

Our current simulation of APS only considers static topologies. While highly mobile topologies, usually associated with ad hoc networks, would require a great deal of communication to maintain up to date location, we envision ad hoc topologies that do not change often, such as sensor networks, and indoor or outdoor temporary infrastructures, like disposable networks. APS aims to keep a low signaling complexity in the event network topology changes slightly. When a node moves, it will be able to get distance vector updates from its new neighbors and triangulate to get its new position, therefore communication remains localized to nodes that are actually mobile. Not even moving landmarks would cause a communication surge in our approach because the only things that identify a landmark are its coordinates. In fact, a moving landmark would provide *more information* to the positioning algorithm, as the new position of the landmark acts as a new landmark for both mobile and fixed nodes. With a mobile "landmark", we can envision a case when a single, fly-over GPS enabled node is in fact enough to initialize an entire static network. Subsequent mobility of the network is supported as long as a sufficient fraction of nodes remains fixed at any one time to serve updates for the mobile nodes. While APS would perform well for limited mobility, it is very likely that its DV nature would incur high signalling costs in highly mobile scenarios. Drawing from experience in ad hoc routing, we may infer that an on-demand positioning scheme would be more appropriate for these cases.

An avenue that is explored extensively in mobile robotics research, involves usage of accelerometers and gyroscopes. Situations may arise when either a node does not have enough neighbors to get sufficient orientation readings, or the node wishes to stay in an inactive state for security or power conservation reasons. In this cases dead reckoning could be used to infer an estimate of current position based on the last triangulated position. This capability is given by accelerometers, which can provide relative positioning after a double integration of acceleration readings. Heading can be inferred in a similar manner when gyroscopes is available.

8. Conclusions

Multimodal sensing can enhance the performance of positioning algorithms. AOA and ranging, possibly enhanced with compasses and accelerometers, have the possibility to provide better positioning than any of them taken separately. Both AOA and ranging are or can be currently achieved using common hardware—time difference of arrival (TDoA), based on ultrasound transmitters/receivers. Not requiring significant additional hardware makes multimodal based sensing a viable approach for positioning. The APS methods we proposed infer position and orientation in an ad hoc network where nodes have local capabilities—ranging and AOA, possibly enhanced by compasses. The assumption is that all nodes have local capabilities, while only a fraction have global capabilities, such as self positioning. Several algorithms were examined, each providing different signaling-accuracy-coverage-capabilities tradeoffs. The advantages of the APS method are that it provides absolute coordinates and absolute orientation, that it works well for disconnected networks, and does not require any additional deployed infrastructure. What makes the algorithm scalable to very large networks is that the communication protocol is localized. Simulations showed that resulted positions have an accuracy comparable to the radio range between nodes, and resulted orientations are usable for navigational or tracking purposes.

Acknowledgements

This research work was supported in part by DARPA under contract number N-666001-00-1-8953.

References

[1] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for networked sensors, in: ASPLOS-IX, Cambridge, MA, 2000.

[2] N. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location-support system, in: ACM MOBICOM, Boston, MA, 2000.

[3] A. Savvides, C.-C. Han, M. Srivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, in: ACM MOBICOM, Rome, Italy, 2001.

[4] P. Bahl, V.N. Padmanabhan, RADAR: an in-building RF-based user location and tracking system, in: INFOCOM, Tel Aviv, Israel, 2000.

[5] D. Niculescu, B. Nath, Ad hoc positioning system (APS), in: GLOBECOM, San Antonio, 2001.

[6] N. Priyantha, A. Miu, H. Balakrishnan, S. Teller, The cricket compass for context-aware mobile applications, in: ACM MOBICOM, Rome, Italy, 2001.

[7] D. Niculescu, B. Nath, Ad hoc positioning system (APS) using AoA, in: INFOCOM, San Francisco, CA, 2003.

[8] G. Finn, Routing and addressing problems in large metropolitan-scale internetworks, Tech. Rep. ISI Research Report ISI/RR-87-180, University of Southern California, March 1987.

[9] J.C. Navas, T. Imielinski, Geographic addressing and routing, in: ACM MOBICOM, Budapest, Hungary, 1997.

[10] D. Niculescu, B. Nath, Trajectory based forwarding and its applications, Tech. Rep. DCS-TR-488, Rutgers University, DATAMAN Lab, Department of Computer Science, May 2002.

[11] Y.-B. Ko, N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, in: ACM MOBICOM, 1998.

[12] N. Bulusu, J. Heidemann, D. Estrin, GPS-less low cost outdoor localization for very small devices, IEEE Personal Communications (Special Issue on Smart Spaces and Environments) 7 (5) (2000) 28–34.

[13] B. Parkinson, J. Spilker, Global Positioning System: Theory and Application, American Institute of Astronautics and Aeronautics, 1996.

[14] M. Betke, L. Gurvitis, Mobile robot localization using landmarks, in: IEEE International Conference on Robotics and Automation, vol. 2, 1994, pp. 135–142.

[15] D. Niculescu, B. Nath, DV based positioning in ad hoc networks, Telecommunication Systems 22 (1–4) (2003) 267–280.

[16] R. Nagpal, Organizing a global coordinate system from local information on an amorphous computer, Tech. Rep. 1666, MIT AI Lab, 1999.

[17] K. Langendoen, N. Reijers, Distributed localization in wireless sensor networks a quantitative comparison, Tech. Rep. PDS-2002-3, Delft University of Technology, The Netherlands, 2002.

[18] L. Kleinrock, J. Silvester, Optimum transmission radii for packet radio networks or why six is a magic number, in: IEEE National Telecommunications Conference, Birmingham, Alabama, 1978, pp. 4.3.1–4.3.5.

[19] S. Capkun, M. Hamdi, J.-P. Hubaux, GPS-free positioning in mobile ad-hoc networks, Cluster Computing 5 (2) (2002) 157–167.

[20] B. Horn, H.M. Hilden, S. Negahdaripour, Closed form solution of absolute orientation using orthonormal matrices, Journal of the Optical Society of America 5 (7) (1998) 1127–1135.

[21] B. Krishnamachari, S. Wicker, R. Bejar, Phase transition phenomena in wireless ad-hoc networks, in: GLOBECOM, San Antonio, TX, 2001.

Dragoş Niculescu has a computer science degree from University "Politehnica" of Bucharest, in 1994, and is currently a Ph.D. candidate in computer science at Rutgers University, in DATAMAN Lab. His research interests include mobile, ad hoc and pervasive networking.

Badri Nath is a professor of computer science at Rutgers University. His research interests are in the area of mobile computing and sensor networks. As part of his research, he is developing protocols and services suited for large scale dense networks. His current focus is on developing protocols and services for an information architecture on top of sensor networks. In particular, he is investigating new paradigms for routing, localization, and data management in sensor networks. He is also the recipient of the ten year best paper award at VLDB-2002. He has served on several program committees of conferences in the area of mobile computing, data management and sensor networks. Currently, he serves on the editorial board of WINET, wireless communications and mobile computing, and IEEE pervasive computing. He has a PhD from the University of Massachusetts, Amherst.