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

Ad Hoc Networks 1 (2003) 247–259

Ad Hoc Networks

www.elsevier.com/locate/adhoc

Localized positioning in ad hoc networks

Dragoş Niculescu *, Badri Nath

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

Abstract

Position centric approaches, such as Cartesian routing, geographic routing, and the recently proposed trajectory based forwarding, address scalability issues in large ad hoc networks by using Euclidean space as a complementary name space. These approaches require that nodes know their position in a common coordinate system. While a GPS receiver in each node would be ideal, in many cases an approximation algorithm is necessary for networks with only a few GPS enabled nodes. These algorithms however require collaboration of large portions of the network, thus imposing an overhead for nodes which do not need positioning, or are mobile. We propose local positioning system, a method that makes use of local node capabilities—angle of arrival, range estimations, compasses and accelerometers, in order to internally position only the groups of nodes involved in particular conversations. Localized positioning enables position centric uses, like discovery, flooding and routing in networks where global positioning is not available.

Keywords: Positioning; Position centric routing; Local positioning system; Trajectory based forwarding

1. Introduction

The main features of new ad hoc networks include 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, easy to deploy, and most importantly easy to maintain. 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 regime, network partitioning and survivability. In all these networks, both basic network operations (routing, forwarding), and higher level functions (multicast, resource discovery) impose a tradeoff between communication overhead and infrastructure support. For example, a network may have high powered basestations to hold large routing tables, or it has to use flooding to discover routes on demand; it may have a GPS [1] receiver in each node, or it has to spend some energy running a positioning algorithm [2–5]. Another tradeoff is encountered in route management—either proactively maintain routes to all possible destinations, or reactively discover them when needed. Both approaches prove better than the other one under different mobility and communication conditions.

* 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)00005-2

The scalability problem of very large ad hoc networks of small nodes, such as sensor networks, is that traditional routing schemes are either table based, or source based, both of which incur high costs on the network. A table based routing scheme can require a high amount of memory at each node, depending on the number of possible destinations. It can require a large amount of signaling to maintain the table in case of mobility. A source based, on demand scheme shifts the scalability problem to the discovery process, based on flooding, and to the packet overhead, sensitive to path length. A class of routing schemes that addresses these problems is using the Euclidean space as a complementary namespace for routing [6–10]. These are position centric approaches, because the position of the destination functions as a delivery address, thus eliminating both the need of routing tables at forwarding nodes, and the need for complete path indication in each packet.

Position centric approaches normally assume that positions of nodes are known in the network, as node positions are used for both naming and forwarding decisions. In this paper we argue that Cartesian routing, trajectory based forwarding (TBF), and the applications they enable would work even in a network in which positions are not globally known, provided that some localized sensing of neighboring nodes is available. The central idea of localized positioning is that only nodes maintaining active communications need to be positioned with respect to each other. Positioning can be relative, that is, each group of communicating nodes maintains a different coordinate system for the purpose of that communication. A group usually means two end points and the intermediate nodes in the case of routing, but may also comprise of nodes involved in discovery, or flooding. A point to multipoint communication group for example, can use a coordinate system belonging to the source, provided that the intermediate nodes and the receivers are all registered to that coordinate system.

The necessary node functionality to support localized positioning is also required by most positioning algorithms, and includes the ability to measure ranges to neighbors, angle of arrival (AoA), or orientation of the node with respect to the North. Ranging is usually provided by estimating the distance to a neighbor by measuring the strength of the radio signal from that neighbor, or by time difference of arrival (TDoA), a method that sometimes employs radio and ultrasound signals. AoA is a method to estimate the direction from which a neighbor is sending data. It can be implemented either using an antenna array, or a combination of radio and ultrasound receivers. Digital compasses may be used in combination with AoA measurement to provide more information about a node's orientation. All these methods have been previously used to provide or extend positioning capabilities in ad hoc networks [2–4,11–13].

If positions are not available to all nodes in a consistent fashion, due to lack or obstruction of positioning infrastructure, such as GPS satellites, the only option left is to run a distributed positioning algorithm. These algorithms however, come with their own disadvantages—they may either require separate infrastructure, like in the case of Cricket [12], requiring ultrasound beacons, or may require a preprocessing phase in which large portions of the network collaborates, thus imposing a high communication cost on all nodes, every time the network changes topology, like in the case of APS [2] or AhLOS [3].

Instead of positioning all the nodes in the network, which may be either globally inconsistent due to mobility, or expensive due to high signaling, we propose a method to locally position only the nodes involved in point to point communication, in a coordinate system relative to the originator of the packet. This enables some position centric applications—flooding, discovery, or source–sink communication in networks in which global positioning is not available, or necessary.

The main sections of the paper are organized as follows: the next section reviews related work, Section 3 is a short review of trajectory based forwarding, Section 4 presents local positioning system (LPS), our method to position only the nodes involved in communication (forwarding the trajectory). Section 5 discusses simulation results, and we summarize with some concluding remarks in Section 6.

2. Related work

Cartesian routing [6] assumes that when all nodes know their positions, and the position of the destination is known, intermediate nodes can forward using greedy decisions, for example choosing the node that provides most progress towards destination. Geocasting [7] sends messages to all nodes contained in a designated polygon, but was designed for IP networks in which nodes are GPS enabled. A recently proposed scheme, trajectory based forwarding [14], generalizes the Cartesian scheme by allowing packets to travel along arbitrary trajectories. Global, precise locations like those available from GPS [1] are highly desirable for all these methods. When GPS cannot provide full coverage, extension or replacement systems have been designed. RADAR [11] is a positioning system that maps many signal strength readings at different locations of a building and stores them in a signal strength readings database. A location query provides the readings to its visible beacons, thus choosing a most likely location from the database. The system has good performance, but has the disadvantage of requiring the beacon infrastructure and manual mapping of the building. Cricket [12,13] is an MIT project employing radio and ultrasound beacons in order to implement ranging and orientation capabilities for roaming mobiles. APS [2,4] and AhLOS [3] are positioning algorithms which do not need in advance setting of infrastructure, but require a small percentage of the nodes in the ad hoc network to have self positioning capability. The schemes extend the positioning capability to the rest of the nodes by the means of network wide collaboration algorithms. Self positioning algorithm [15] proposes relative positioning, in which the coordinate system is established by a reference group of nodes. Their approach inspired our present work on LPS, by suggesting that adjacent nodes can establish local coordinate systems and register them.

In order to use locations as a naming space, a location service is necessary to translate node addresses in coordinates. Grid location service (GLS) [16] implements a naming service that allows node centric applications to run on top of Cartesian, geographic or trajectory based routing. A source can find the coordinates of the destination node from the location service and then use geographic or Cartesian routing to route to that destination. While the location service is not only useful, but necessary in a global positioning network, it is still a research issue what similar mechanism would be necessary in a network in which communication is based on relative coordinate systems.

3. Trajectory based forwarding review

TBF [14,10], is a paradigm which aims to provide solutions that require neither infrastructure support, nor communication overhead for route maintenance. The idea is to embed the trajectory $X(t)$, $Y(t)$ in each packet and let the intermediate nodes take the forwarding decisions. TBF is a generalization of source based routing [17] and Cartesian forwarding [6]. Like in source based routing, the path is indicated by the source, but without actually specifying all the intermediate nodes. Like in Cartesian forwarding, decisions taken at each node are greedy, but are not based on distance to destination—the measure is the distance to the desired trajectory.

TBF combines the best of the two methods: packets follow a trajectory established at the source, but each forwarding node takes a greedy decision to infer the next hop based on local position information, while the overhead of representing the trajectory does not depend on path length. In a network where node positions are known, the packet may be forwarded to the neighbor that is geographically closest to the desired trajectory indicated by the source node. If the destination node is known, the trajectory followed by the packet might be a line, and the method reduces to Cartesian forwarding. In the general case, however, one can envision a larger array of applications (Fig. 1), including broadcasting replacement (a), disjoint (b) and braided (c) multipaths, discovery (d), geocasting (e) and multicasting (f). In most of these cases, it is only necessary for the source, destinations and the intermediate nodes to agree on a positioning system in order to support the position centric service.

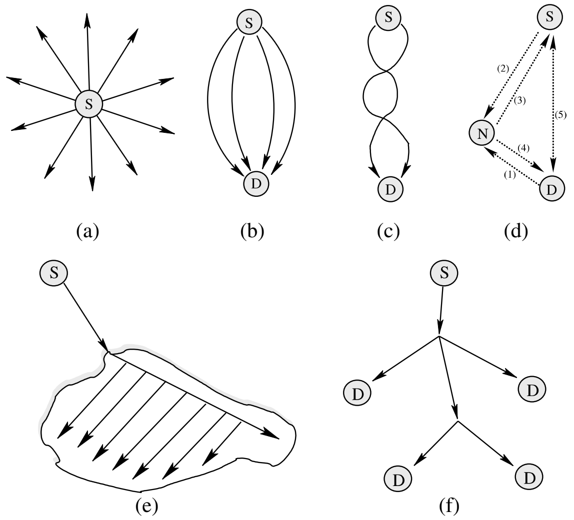

Fig. 1. Trajectory based flooding, routing and discovery.

4. Local positioning system

Localized positioning can be used in any node centric network, when a communicating group agrees on a common coordinate system. We describe LPS in the context of trajectory based forwarding because TBF naturally creates communication groups of nodes which are touched by the trajectory. The trajectory may be a curve describing a path, or a tree, describing a distribution path.

We extend an idea proposed in [15], and develop a method for nodes to use some capabilities (ranging, AoA, compasses) to establish local coordinate systems in which all immediate neighbors are placed. It is then possible to register all these coordinate systems with the coordinate system of the source of the packet. LPS is a method to achieve positioning *only for the nodes along the trajectory*, with minor increase in communication cost, as if all node positions were known. Instead, each node touched by trajectory spends some computation to position itself in the coordinate system of the source of the packet. The tradeoff achieved by the method is to spend computation in order to save communication. The computation spent for registration is more economical energy-wise compared to the communication spent by a network wide positioning algorithm (for current hardware, such as the Berkeley mote [18]).

4.1. Node capabilities

The network model considered is a large scale, dense, ad hoc set of 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 and symmetrical, these networks are modeled as fixed radius random graphs, or unit graphs.

Range based estimation of distance between two nodes has been previously used for positioning purposes [2,3,11], even with the high measurement error involved. In Fig. 2, node A would have estimations of distances to its neighbors AB and AC, but could also find BC after communicating with B, or C. In most implementations, ranging is achieved either by using an estimate from the strength of the signal (unreliable), or using TDoA.

In AoA approach, 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. After the deployment, the axis of the node has an arbitrary, unknown heading, represented in Fig. 2 by a thick black arrow. When interacting with two neighbors, a node can

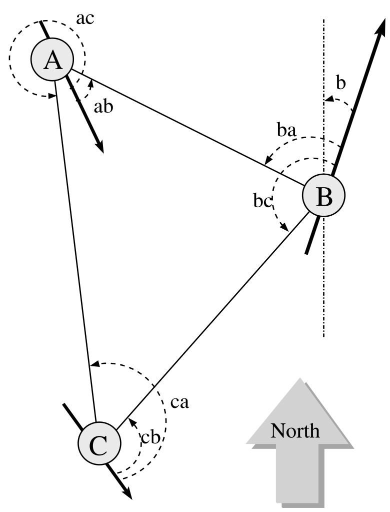

Fig. 2. Nodes capabilities: measuring angles, ranges, orientation.

find out the angle between its own axis and the direction the signal comes from. 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. AoA capability is usually achieved by using an antenna array, which might be prohibitive in size and power consumption. A small form factor node that satisfies conditions outlined has been developed at MIT by the Cricket compass project [13]. Its principle of operation is based on both TDoA and phase difference of arrival. Other node capabilities that might be available in small form factors include accelerometers and compasses. The accelerometer's main use is to indicate pose, while compass indicates the absolute orientation of each node.

4.2. Local coordinate systems

Denote with $\mathscr{C}_i$ the coordinate system of node i. The source node of a trajectory has the coordinate system $\mathscr{C}_0$. In Fig. 3, if node A is able to measure distances to all its neighbors, via a ranging method, it can compute the sides and the angles for all triangles created with all pairs of neighbors which are also neighbors of each other. This would enable A to place itself in 0,0 of its local coordinate system $\mathscr{C}_a$, and all its immediate neighbors at positions that satisfy all the range requirements known to A, ranges indicated by continuous lines in the figure. To establish its coordinate system, A randomly chooses E as an indicator for its x^a axis and F as an indicator for its y^a axis. All the other neighbors are then placed in this coordinate system such that all the range readings are respected. Most nodes can be unambiguously positioned if they are neighbors with two previously positioned nodes. Node G for example is positioned immediately after E and F are positioned by choosing between two possibilities. It can be placed where it is on the figure, or on the position of G_1. But if G were G_1, it would have been close enough to E to have a direct link, so this possibility is eliminated. In the same fashion, all nodes around $A(0,0)$ are positioned. Node B sets up a similar local coordinate system $\mathscr{C}_b$, by initiating its axes using neighbors C and D.

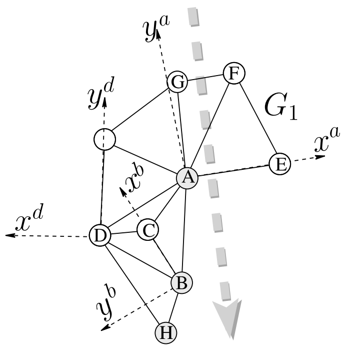

Fig. 3. Local coordinate systems.

Since a condition of global connectivity in a random ad hoc network is for the average degree to be at least 6 [19], in higher density networks most nodes will succeed in placing all their neighbors in their local coordinate systems.

In the example in Fig. 3, a translation and a rotation are enough to overlap the local system of B, $\mathscr{C}_b$ over the local system of A, $\mathscr{C}_a$. In the case of D, however, the first two neighbors randomly chosen as axes indicators produce a coordinate system that cannot be overlapped over $\mathscr{C}_a$ using only translation and rotation. In this case, due to D's original, localized and independent choice of axes, a mirroring transformation will also be necessary. The main cause for these mirroring transformations is the fact that ranging information does not provide a sense of direction (at least in the first phase, before the local coordinate systems are set).

A similar procedure to establish a local coordinate system is possible when using AoA measurements. In the same Fig. 3, node A knows all angles to neighbors reported against its main axis (see Section 4.1), and therefore has already an established set of axes. In triangle AFE for example, A knows all angles but no sides. Assuming that the range $AE = 1$, it can then find the sizes AF and FE which can then be propagated to all other triangles in order to get coordinates for all points. This coordinate system, $\mathscr{C}_a$ can be registered with $\mathscr{C}_b$ using a scale transformation (in addition to translation and rotation), because B will probably choose a different edge as a unit for its local coordinate system. If a compass is available in each

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)

node, all the reference systems have parallel axes, and the rotation is not necessary to achieve coordinate system alignment. A mirroring transformation will occur when using AoA only in the case in which a node is deployed upside down.

It is possible to use both AoA and ranging in creating local coordinate systems, possibly enhanced with local compasses. Table 1 indicates all the possible combinations of node capabilities, and the transformations involved in the alignment 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. The general transformation matrix to translate a point from $\mathscr{C}_b$ to $\mathscr{C}_a$ is

$$M_{b,a} = \begin{bmatrix} sr_1 & sr_2 & t_x \\ sr_3 & sr_4 & t_y \\ 0 & 0 & 1 \end{bmatrix},$$

where $\{r_*\}$ is the rotation transformation and possibly mirroring, s the scaling factor, $\{t_*\}$ the translation, $\{r_*\}$ is an orthonormal matrix with the following properties:

$$\begin{cases} |r_1| = |r_4|, \\ |r_2| = |r_3|, \\ \det\{r_*\} = -1 \text{ for mirroring}, \quad 1 \text{ otherwise.} \end{cases}$$

If a transformation is not present, its parameters will be set to neutral values to preserve the properties of M. For example, when ranging is available and no scaling is involved, $s = 1$, when compasses are available and there is no rotation, $\{r_*\} = I_2$ (the identity matrix).

If for example, $C_a = [x_a^C y_a^C 1]^{\mathrm{T}}$ designates the position of C in $\mathscr{C}_a$, and C_b its position in $\mathscr{C}_b$, then

$$C_a = M_{b,a} C_b. \tag{1}$$

4.3. Registration

The registration between two local coordinate systems is the process that computes a transformation matrix which will overlap any point from one coordinate system to the other. The input to this process are points for which the coordinates are known in both coordinate systems with some accuracy. If perfect ranging were used in creating the local coordinate systems, the registration would produce a rigid transformation.

If two nodes, such as A and B in Fig. 3 wish to agree their coordinate systems, they must implement the registration procedure. They reach this agreement by using points that are common in the two coordinate systems, in our example A, B, C and D. In practice, since ranging does not provide perfect distances and AoA perfect angles, the two coordinate systems will not overlap in all points if we restrict M to the type of transformation described above. The error in registration is defined as the sum of squared distances between corresponding points after registration. If B has the two sets of coordinates of nodes A, B, C and D in both coordinate systems, it computes the transformation matrix $M_{b,a}$ that minimizes the error:

$$\sum \|X_a - M_{b,a} X_b\|^2, \quad X = A, B, C, D. \tag{2}$$

Fortunately, there is a closed form solution to this optimization, described in [20], that has a running time linear in the number of common points. The solution for the rotation R involves computing of eigenvectors and eigenvalues for a 2×2 matrix (for a 2D problem). The scaling transformation s is obtained as the square root of the ratio of the variances of the two sets around their centroids. The translation t is the difference of centroid of set a and scaled and rotated centroid of set b.

These methods of establishing local coordinate systems and registering them holds for the idealized model of networks as unit graphs. In practice, more robust methods can be used, for example using median instead of the mean for the error minimized, or formulate the local coordinate system also as an optimization problem.

4.4. Forwarding using LPS

The aim for LPS is to make position centric functions behave as in a network where node positions are available. To make the explanation more specific, we refer here to linear trajectories, but the discussion may be extrapolated to other spatial patterns, trees, or general trajectories. The key idea is that the only nodes that are positioned are the ones involved in forwarding along the trajectory. Positioning is done in a hop by hop fashion, in the coordinate system chosen by the initiating node—the source of the packet. The forwarding procedure works with a node selecting the next hop based on the proximity to the desired trajectory, or any of the other possible policies. In Fig. 3, the ideal trajectory is shown as a thick dashed arrow. Assume that A is already registered with $\mathscr{C}_0$, that is, it has computed the matrix $M_{a,0}$. If the next node to be selected along the trajectory is B, it will receive from node A the position of common nodes in $\mathscr{C}_0$, $\{A, B, C, D\}_0$, so that node B, can register its own coordinate system to $\mathscr{C}_0$ by solving the optimization problem (2). Once $M_{b,0}$ is obtained at B, all the neighbors of B are evaluated in $\mathscr{C}_0$, by using transformation (1). Node B is then able to select one of its own neighbors that is closer to the trajectory, in order to continue the process.

What is in fact achieved by LPS is the registration of all coordinate systems of visited nodes to the coordinate system of the initiating node, which achieves positioning of all these nodes in the coordinate system of the source. This positioning system has a number of advantages:

- It is localized to the nodes actually involved in communication. Unlike a network wide positioning algorithm, such as [2,3], which involves collaboration and coordination of a large number of nodes, LPS involves only the nodes "touched" by the desired trajectory.
- The size of packet only depends on the number of common neighbors two consecutive nodes have, and this is upper bounded by the maximum degree of the nodes. In the forwarding packet, the equation of the trajectory is also included, but this is a fixed size, so the packet does not increase with the length of the trajectory.
- It can make use of any of the two previously used localization capabilities—AoA (possibly enhanced with local compasses and accelerometers), or range estimation.
- It can be used for broadcasting, discovery, and position centric routing between static endpoints, because all may rely on relative positioning.

4.5. Reducing the overhead for static networks

Registration at system i produces a matrix

$$M_{i,0} = \text{register}(\mathscr{C}_i, \mathscr{C}_0)$$

which can translate local coordinates $(x, y)_i$ in $\mathscr{C}_0$ coordinates:

$$(x, y)_0 = M_{i,0}(x, y)_i.$$

The amount of *packet overhead* to support LPS comprises:

1. $(X(t), Y(t))_0$—the parametric representation of the trajectory. These functions are meaningful in $\mathscr{C}_0$ only;
2. t_{crt}—the point of the curve closest to the current node;
3. coordinates of common nodes—$(A, B, C, D)_0$ in coordinate system $\mathscr{C}_0$.

For arbitrary trajectories, TBF uses (1) and (2) to support forwarding. If Cartesian routing is used, the line between the current node and the destination describes the trajectory, therefore only (3) is an overhead. When global coordinates are used, there is no packet overhead for Cartesian, and TBF specific overhead is (1) and (2).

The amount of *CPU overhead* for forwarding at node i:

1. Register $\mathscr{C}_i$ to $\mathscr{C}_0$:

   $M_{i,0} = \text{register}(\{A,B,C,D\}_i, \{A,B,C,D\}_0)$;

2. Translate all neighbors of node i in C_0:

   $H_0 = M_{i,0}H_i$, $\forall H$ neighbor of i;

3. Evaluate $X(t)$, $Y(t)$ in the local neighborhood of node i, to find the next hop, in $\mathscr{C}_0$.

Out of these, (1) and (2) are specific to LPS, and (3) is specific to TBF. If used with Cartesian forwarding, LPS would use (1) and (2) only. In a global coordinate system, TBF would only use (3), and Cartesian would incur no overhead, besides choosing the best next node.

By examining the two categories of overhead—computation and communication, we can see that they are dominated by registration related items: (3) for packet overhead, (1) and (2) for CPU overhead. Registration is what makes all the nodes operate in a common coordinate system, the core of LPS. The only problem is that it claims a per packet cost in terms of both CPU and packet overhead, a cost caused by the absence of a global coordinate system.

In static networks, this problem can be addressed by preprocessing. The procedure of successively aligning coordinate systems as the trajectory goes is important for discovery purposes, or for the cases when topology changes. When the network is static, per packet registration of coordinate systems for each packet may be avoided by running a preprocessing procedure in which all nodes only register once with each neighbor, and cache the transformation matrix for that neighbor for all subsequent communication, thus reducing both CPU and packet overhead. Instead of registering with the origin of communication, which is only known at forwarding time, register with the immediate neighbor, and use the fact that the registration matrix $M_{i,0}$ is in fact the product of all the registration matrices from the source of the packet to the current node.

$$M_{i,0} = M_{i,i-1}M_{i-1,0},$$

$$M_{i,0} = \prod_{j=i}^{1} M_{j,j-1}.$$

This means that in fact node i only needs to register with the neighbor labeled $i-1$ ($M_{i,i-1}$), in order to handle all communication forwarded from $i-1$, which is independent of packet's coordinate system. What is specific to a particular source with coordinate system $\mathscr{C}_0$ at node i is $M_{i-1,0}$, which can be forwarded from the node $i-1$, together with the trajectory. The matrix is usually small in size, depending on the hardware used, see Table 1. Translation requires two numbers t_x and t_y, rotation is an angle, and the scale is also a number. These values replace the coordinates of common nodes (item (3) of packet overhead), whose number grows with the node degree, and achieve the registration step (item (1) of CPU overhead) with a simple matrix multiplication. Therefore the per packet cost imposed by LPS to support TBF in a static network without global positioning is reduced to four numbers as packet overhead. The per packet computation overhead imposed by LPS is one matrix multiplication per neighbor. The preprocessing cost involves establishing local coordinate systems, registering individually with each neighbor, and storing the transformation matrices for subsequent communication.

5. Simulation

We simulated an isotropic [1] map (average degree = 9.5), with 200 nodes each having a random, but unknown heading. In order to evaluate LPS for the purposes of forwarding, routing, and discovery, we tested various linear trajectories, and used four metrics to compare the performance of forwarding in the presence of ranging errors and angular errors. For each of 30 pairs of source–destination nodes situated near the sides of the map (North–South and East–West), we compared the trajectory obtained when ranging or angle measurement is imperfect with the trajectory obtained when these measurement are perfect. In the latter case all nodes register to their globally coherent positions. Therefore, the "perfect trajec-

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

tory" is the one obtained with perfect locations, which is still different from the ideal trajectory, as described by the parametric curves. Note that globally coherent positions all differ from true GPS positions by that same transformation.

- *Deviation* of a trajectory is computed as the mean distance between a point on the obtained trajectory and the closest point on the perfect trajectory. The value of this deviation is then normalized to the maximum hop size to be expressed as a hop count.
- *Path length* is the number of hops achieved by a trajectory, relative to the perfect trajectory. The forwarding process may be stopped when the trajectory exits the map, or when packets cannot be forwarded anymore due to local aberrations in registration, obstacles, faulty or inexistent local coordinate systems.
- *Minimum distance to destination* shows how close a trajectory gets to the desired destination.
- *Probability of reaching destination* expresses the chances of the trajectory passing within one hop of the desired target. In this set of experiments, when the trajectory reaches the destination, or an immediate neighbor of it, the forwarding is stopped.

For the AoA, we model normally distributed errors whose standard deviation is shown on the horizontal axis. Corresponding to an error of ±5°, achieved by the Cricket compass project corresponds a standard deviation of 2.5° on our graphs. For ranging measurement, we assume a linear error model where the actual measurement is within a fixed fraction of the true range (0–100%).

Simulations are divided in two groups, based on the node capabilities used by LPS. The first group

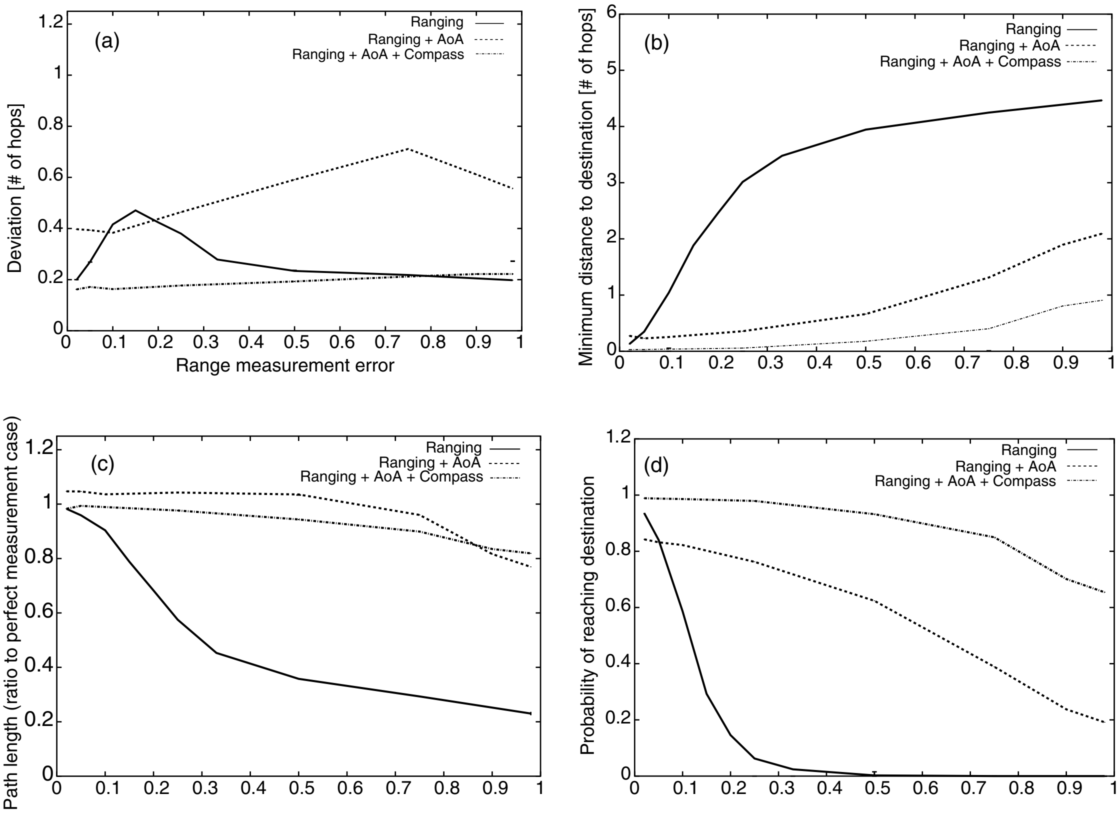

Fig. 4. Range, range + AoA, range + AoA + compass. (a) Deviation from 'perfect' trajectory, (b) minimum distance to destination, (c) path length—relative to 'perfect' trajectory, (d) probability of reaching the destination.

(Fig. 4) compares ranging alone with ranging and AoA, possibly enhanced with a compass. The AoA deviation is fixed in these experiments to 0.15 (about 8.59°). The experiments reveal that ranging is less robust in forwarding the coordinate system, being greatly enhanced by the presence of angular measurements, even with high errors. The reason for this difference stems from the way local coordinate systems are built. Using ranging with errors, the coordinate systems are prone to false mirroring, large aberrations produced by working with small or obtuse angles, bias depending on the nodes initializing the coordinate system. We experimented with various thresholds to limit the shape of triangles resulted from ranging—eliminating very obtuse or very acute angles from being used in inferring local coordinates. Such thresholds provide a tradeoff between the amount of possible misleading information that is accepted in the registration and the number of nodes which successfully get a local coordinate system. For this set of experiments, we used an angular threshold of 0.6 (~36°), thus eliminating from the inference the triangles with smaller angles, at the price of dropping around 3% of the nodes which were not able to establish local coordinate systems. When AoA is available, the local coordinate system building relies on the inherent order in which angles are reported, completely eliminating false mirroring, and reducing range aberrations. Range based measurement shows a maximum in the deviation (Fig. 4a), explained by the sharp decrease in the path length (Fig. 4b) following the maximum, which produces shorter paths with a relatively good start.

The second group of experiments (Fig. 5) studies the behavior of angular measurements, possibly enhanced with a compass. The absence of ranging

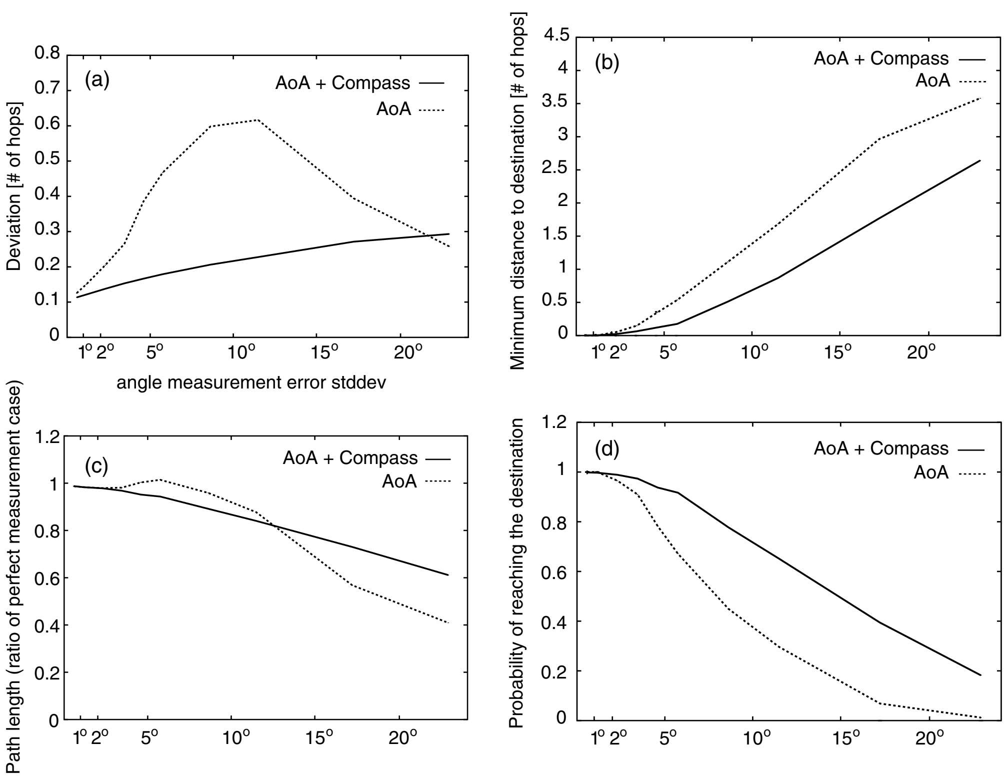

Fig. 5. AoA, AoA + compass. (a) Deviation from 'perfect' trajectory, (b) minimum distance to destination, (c) path length—relative to 'perfect' trajectory, (d) probability of reaching the destination.

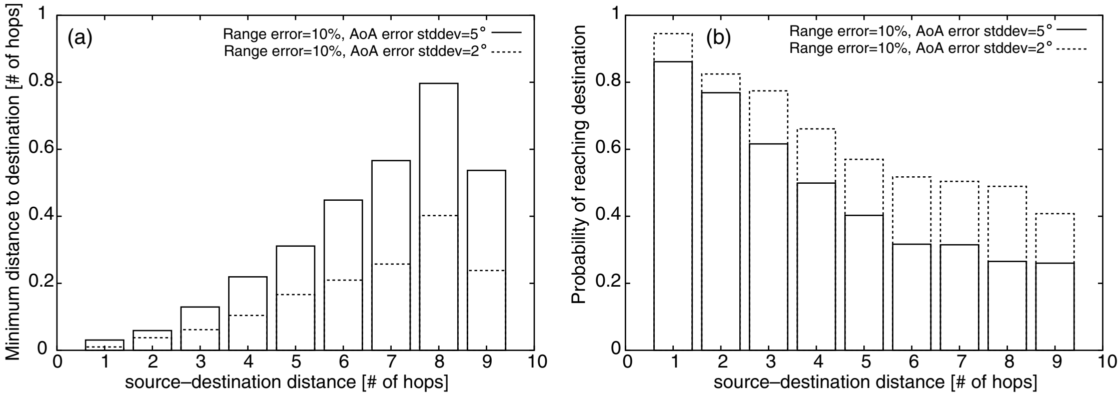

Fig. 6. Errors compound with distance. (a) Minimum distance to destination, (b) probability of reaching the destination.

in these experiments brings the need for a scaling transformation in the registration process. On the horizontal axis of all these graphs, the standard deviation of the angular measurement error is indicated. The deviation when using simple AoA (Fig. 5a) shows a decreasing trend that is caused by the paths getting shorter with increasing errors, but still having a good start, which yields a low deviation. The same type of behavior was encountered when a single type of sensing was employed (ranging), and is a symptom indicating short paths. For the region where the deviation is high, path length can be even longer than the perfect path, because, once it misses the destination, it can wander for 1–2 more hops before it reaches the edge of the map. Both groups of experiments show that a compass in each node brings a substantial improvement in all metrics. One of the main causes for this improvement is that rotation is eliminated from the registration process ($\{r_*\} = I_2$).

These two sets of simulations showed that multimodal sensing (AoA + ranging) performs much better than its components taken separately, under the same error conditions. A compass attached to each node brings significant improvement in all the metrics considered.

In order to evaluate the amount of degradation that occurs in the iterative registration process, we conducted a set of experiments in which source and destination nodes are chosen randomly on the map, with uniform probabilities for X and Y coordinates. In Fig. 6, minimum distance and probability of reaching the destination are plotted as a function of the initial distance between source and destination, normalized by the maximum hop size. LPS employed ranging with an error of 10% and AoA measurements with deviations of 2° and 5°. As expected, the error is compounding with distance yielding decreasing probabilities of reaching the destination and increasing distances to destination. An interesting side effect is that for far apart source–destination pairs, which will usually be placed on the opposite sides of the map, the minimum distance between the trajectory and destination decreases, as the trajectory may shortly wander left or right, searching for a possible continuation of the map, and thus reducing the distance to the destination.

The main conclusions revealed by the simulations are that, although error compounds with distance, it can be countered efficiently by using additional hardware available in small form factors. An accelerometer present in each node could detect node flipping, eliminating false mirroring, while a digital compass could eliminate rotations from registrations process, when angular measurements are used. With AoA error range in the range 2–5°, which has already been realized by other projects [13], LPS maintains a low deviation, making it usable for discovery purposes, and reaches the destination with high probability, making it usable for routing.

6. Conclusions

We presented LPS, a method to achieve localized positioning in ad hoc networks where certain capabilities (AoA, ranging, compasses, accelerometers) are present. LPS requires only the collaboration of nodes involved in a communication group, be it a trajectory, or a distribution tree. LPS does not incur the communication overhead of a network wide positioning algorithm by registering only the coordinate systems that are in actual communication.

When the network is static, per packet registration of coordinate systems may be avoided by running a preprocessing procedure: all nodes only register once with each neighbor, and cache the transformation matrix for that neighbor for all subsequent communication, thus reducing both CPU and packet overhead. Simulations showed that trajectories positioned by LPS have a low deviation, making them usable for discovery purposes, and reach the destination with high probability, making them usable for routing.

Acknowledgements

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

References

[1] B. Parkinson, J. Spilker, Global Positioning System: Theory and Application, American Institute of Astronautics and Aeronautics, 1996.
[2] D. Niculescu, B. Nath, Ad hoc positioning system (APS), in: GLOBECOM, San Antonio, 2001.
[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] D. Niculescu, B. Nath, Ad hoc positioning system (APS) using AoA, in: INFOCOM, San Francisco, CA, 2003.
[5] L. Doherty, L.E. Ghaoui, K.S.J. Pister, Convex position estimation in wireless sensor networks, in: IEEE INFOCOM, Anchorage, AK, 2001.
[6] G. Finn, Routing and addressing problems in large metropolitan-scale internetworks, Tech. Rep. ISI Research Report ISI/R.R-87-180, University of Southern California, March 1987.
[7] J.C. Navas, T. Imielinski, Geographic addressing and routing, in: ACM MOBICOM, Budapest, Hungary, 1997.
[8] Y.-B. Ko, N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, in: ACM MOBICOM, 1998.
[9] I. Stojmenovic, X. Lin, Gedir: loop-free location based routing in wireless networks, in: International Conference on Parallel and Distributed Computing and Systems, Boston, MA, 1999, pp. 1025–1028.
[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] P. Bahl, V.N. Padmanabhan, RADAR: an in-building RF-based user location and tracking system, in: INFOCOM, Tel Aviv, Israel, 2000.
[12] N. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location-support system, in: ACM MOBICOM, Boston, MA, 2000.
[13] N. Priyantha, A. Miu, H. Balakrishnan, S. Teller, The cricket compass for context-aware mobile applications, in: ACM MOBICOM, Rome, Italy, 2001.
[14] B. Nath, D. Niculescu, Routing on a curve, in: HOTNETS I, Princeton, NJ, 2002.
[15] S. Capkun, M. Hamdi, J.-P. Hubaux, GPS-free positioning in mobile ad-hoc networks, Cluster Computing 5 (2) (2001) 157–167.
[16] J. Li, J. Jannotti, D. De Couto, D. Karger, R. Morris, A scalable location service for geographic ad hoc routing, in: ACM MOBICOM, Boston, MA, 2000, pp. 120–130.
[17] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, vol. 353, Kluwer Academic Publishers, Dordrecht, 1996, pp. 153–181.
[18] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for networked sensors, in: ASPLOS-1X, Cambridge, MA, 2000.
[19] B. Krishnamachari, S. Wicker, R. Bejar, Phase transition phenomena in wireless ad-hoc networks, in: GLOBECOM, San Antonio, TX. 2001.
[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.

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 net works. 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 Ph.D. from the University of Massachusetts, Amherst.