Positioning in Ad Hoc Sensor Networks

Dragoş Niculescu, Rutgers University

Abstract

Position and orientation of individual nodes in ad hoc sensor networks are 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 may have local capabilities such as the possibility of measuring ranges to neighbors, angle of arrival, or global capabilities, such as GPS and digital compasses. This article surveys methods used to infer locations in a multihop fashion in networks with or without the mentioned capabilities.

With the ever decreasing cost and size of sensors, the instrumentation of the world becomes possible, with a wide range of meteorological, commercial, and personal applications. Sensor networks, when not micromanaged in terms of topology, are actually ad hoc networks, in most cases not mobile or with occasional mobility. What they share with mobile ad hoc networks is the probabilistic nature of the graph, the problems of connectivity and density control, medium sharing, and scalability. They may not face the mobility problem, but they have a host of new ones. Sensor nodes have much lower processing capabilities than current cell phones or PDAs; for very large scales of deployment, energy consumption is a factor that affects the trade-off between node power and the cost of replacing batteries; configuration and maintenance on a per node basis are not possible.

One lesson that has been learned from the work on more general, and possibly mobile, ad hoc networks is that solutions from the wired world do not usually apply directly to large ad hoc topologies. If we only look at the example of routing, link state is out of the question if we deal with hundreds of thousands of weak nodes, whereas distance vector, although more scalable, has disadvantages with respect to mobility. Both link state and distance vector try to infer data about the entire topology at each node. This makes them undesirable for very large ad hoc networks. Sensor network nodes are expected to have a variable duty cycle, meaning that in order to conserve energy, nodes will sleep most of the time and wake up periodically during flurries of activity or querying. This on and off behavior will incur too large costs on all algorithms that aim to mirror the state of entire networks at each node. A more insidious problem is that the duty cycle affects on demand schemes as well, which otherwise do not face the network representation problems of link state and distance vector.

Is Positioning Necessary?

The short answer is yes. The detailed answer depends on the class of networks we envision. For personal mobile computing, positioning has been a research focus for years [1]. For ad hoc networks used by military or rescue applications, or for sensor networks, positioning is indispensable. In all these cases the availability of position enables more efficient protocols and a number of new applications.

Position of nodes or of a phenomenon can be either the mean or the goal of a sensing application. In applications such as meteorological and environmental monitoring, package tracking, and library archiving, position is the main aspect of the problem. In a sensor network, positions of the nodes is required to make sense of the reported data and associate each reading with a geographical map. Position and orientation sensing enable tracking a mobile through a sensor field in military applications.

At the service level, availability of position enables implementation of algorithms with better scalability. Position-centric addressing was first proposed in the 1970s, but has regained attention recently with the advent of very large networks, such as sensor networks. Here, nodes are named by their position in Euclidean space, and only one node can have a given position. From this bijective relation, it follows that there is no separate job to be performed to support routing. This means that the current state of the packet (e.g., its position) and the position of the destination are enough to determine the process of forwarding. The simplest example of position-centric addressing and routing is Cartesian routing [2], which greedily decides that the next node to receive the packet is the neighbor geographically closest to the destination. While this may sound simple, there are in fact several possible strategies for position-centric forwarding (angle-based, progress-based, etc.), each facing some logistical problems related to the localized greedy nature of the forwarding algorithm. A recent survey by Stojmenović [3] reviews many of the strategies and solutions related to position-centric forwarding and routing.

Position-centric routing has a number of advantages over node- and data-centric: there are no routing tables to be maintained, it has good resilience in the face of mobility, and does not incur high overhead like on-demand routing schemes. On the other hand, we have to acknowledge that position-centric routing may require maintenance of a position database in order to support node-centric applications. This database performs translation from node IP to Euclidean positions so that at the application layer one may use node-centric (IP-based) applications, while lower service layers may use position-centric approaches.

Being convinced that positioning capability is necessary in the new ad hoc and sensor networks, one may argue that with the general availability of the Global Positioning System (GPS) and the proliferation of commercial devices, the problem is almost solved. But depending on the application/network, there are cases when GPS may be undesirable or not usable at all. The technical aspect most relevant for many applications is the line of sight (LOS) requirement of GPS. The positioning service is available only when at least four satellites are visible, which means networks that are indoors, under foliage, or obscured by buildings will not be able to get a position. Even when nodes are deployed in GPS-friendly conditions, there are issues related to the power, form factor, or cost that may make the solution undesirable. If future sensor nodes are to be very small (e.g., embedded in materials), size and cost my prohibit inclusion of a GPS receiver with each node. If a low-power node can function on energy drawn directly from the environment, such as surrounding light, the included hardware will be kept to a minimum. Finally, even leaving aside all the technical arguments, political considerations of guarantees of availability may require a positioning solution independent of external factors.

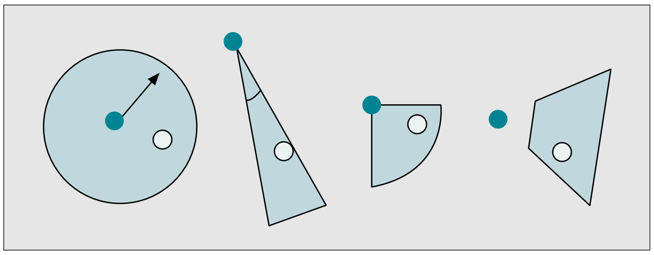
Classification

The positioning problem has many engineering aspects besides networking, and is a hard problem that has been tackled in its different aspects by several research communities including vision, robotics, signal processing, and networking. The type of problem we focus on in this survey is large ad hoc networks of static nodes that generally have similar power and capabilities. We assume that the probabilistic nature of the deployment prohibits solutions based on powerful landmarks or satellites, like GPS. Several solutions proposed in the literature are solely based on the availability of such powerful landmarks that provide angle, or range readings to individual nodes, which are then able to individually perform trilaterations, just as in the case of GPS. We call these solutions *one-hop*, because the node that receives the positioning service is within one hop of the landmark or satellite providing the service. The more appropriate class of algorithms for ad hoc applications are *multihop*, meaning that regular nodes may not be in direct contact or may not directly measure ranges to the landmark.

The first classification of such algorithms is based on whether they use sensing hardware or not. Sensing in this context refers to the ability to measure the placement of neighboring nodes in relation to itself (e.g., range). The alternative is to only make use of connectivity derived from the topology of the graph associated with the ad hoc network.

If sensing is to be used, nodes may make use of measurements with respect to neighbors, such as ranges and angles, or measurements with respect to a global reference system, such as accelerometers and compasses. In a radio-based network a node may use signal strength to infer range to neighbors, but this method is known to be very imprecise. Another method to obtain range is to measure time of flight for sound. This assumes that nodes are equipped with ultrasound beepers and microphones, but the method provides centimeter accuracy. The drawback of this method is that it requires LOS, similar to GPS requiring LOS for measuring the time of flight for the radio signal from the satellite.

Another way to look at positioning algorithms is whether they provide local, relative, or absolute coordinate systems. An *absolute* coordinate system has global coherence and is desirable for most situations, being aligned to popular coordinate systems used in commercial and military references, such as GPS. These are also the most expensive in terms of communication cost and are usually based on landmarks that have known positions. Possible mobility in the network is supported at great cost, since all new positions must be coherent. *Relative* positioning establishes positions that are relative to a system that is local to the network and can possibly be arbitrary, but still provides network-wide coherence. A *local* coordinate system has only local coherence and might be used by a position-centric scheme in which only the communicating parties position themselves with respect to each other.


■ Figure 1. *Convex optimization (adapted from [4]).*

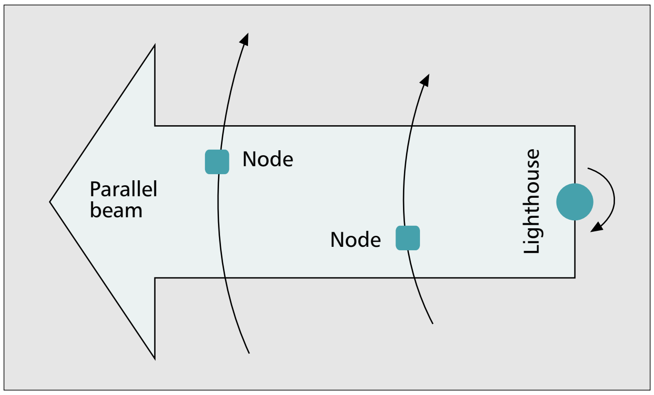
A criterion that has a direct impact on the efficiency and applicability of the algorithm is that of the algorithm being centralized, distributed, or localized. A *centralized* algorithm has the option of using global information that can potentially improve the quality of position estimates. The associated cost is that the entire topology of the ad hoc network must be collected at a central node that optimizes for the positions. Other potential problems include the need to deal with large data structures associated with a large network, requiring a powerful central computing node, and a large communication cost to transfer the description of the topology over to the central point. A *distributed* algorithm makes use of several computing and communication capabilities, with the usual advantages of not relying on a single failure point, not requiring a specialized central node, and naturally load balancing by making use of a geographically distributed resource. A *localized* algorithm is not only distributed, but only makes use of local data, having communication limited to comparatively small regions.

Centralized Methods

Convex Optimization

Doherty *et al.* [4] approach the positioning problem using linear programming and semidefinite programming. If a linear program (LP) optimizes for a linear objective over a set of linear constraints, a semidefinite program (SDP) is a generalization that, in addition to LP, accepts linear matrix inequalities (LMIs). LMIs are of interest here because a two-dimensional quadratic inequality can be cast as an LMI. In Fig. 1 the dark node represents a landmark, and the white node one whose position is constrained by some sensing from the landmark. For a fixed radius model, when nodes are assumed to communicate within a perfect circle, the acceptable region is convex, and cannot be described by a set of linear inequations, but as an LMI. All the other examples in the figure can be described as intersections of either half-planes or quadratic constraints that can be reduced to LMIs.

The advantage of the methods based on convex optimization are that it is simple to model both hardware that provides ranges or angles and simple connectivity, and there are efficient computational methods available for most convex programming problems. Also, they provide optimal solutions, provided the model can be cast as a set convex constraints. However, convex optimization provides this optimality at the cost of centralization and the need to handle large data structures. The complexities of the LP and SDP problems are experimentally shown in [4] to be quadratic, respectively cubic in the number of connections. This number is the product between the number of nodes and the number of neighbors, which means large and dense networks might be harder to solve.



■ Figure 2. *Top view of an idealistic lighthouse with a parallel light beam.*

MDS-MAP

MDS-MAP [5] is a method that only makes use of connectivity to provide positions in a network with or without landmarks. It is based on classic multidimensional scaling (MDS), a method that finds an embedding in a lower-dimensional space for a set of objects characterized by pairwise distances between them.

The proposed method operates in three stages. The first stage computes the shortest paths between all pairs of nodes in the network. These distances are used to initialize a distance matrix for MDS. All pairs shortest path algorithms have a complexity cubic in the number of nodes. The second stage is to apply classical MDS on this matrix, and retain the two largest eigenvalues and eigenvectors in order to construct a 2D map. This, being based on SVD, is also cubic in the number of nodes. The last stage is the conversion to an absolute map if three or more landmarks are available. This last stage has a complexity linear in the number of nodes.

The advantage of MDS-MAP is that it has a wide range of applicability, having the ability to work with both simple connectivity and range measurements to provide both absolute and relative positioning. Also, unlike the case of convex optimization, the complexity of MDS-MAP has a theoretical bound.

One-Hop Positioning

This class of positioning methods has as a main characteristic the possibility for nodes to directly contact the landmarks. The most successful and widely deployed representative of this class is, of course, GPS. Many other solutions in this class have the same disadvantage of requiring LOS to the landmark, but there are cases when this limitation is inherent to the capability of the nodes, such as when low-power nodes have only optical communication capability.

The Lighthouse Location System

An elegant way to exploit LOS communication is used in the Lighthouse project [6]. Positioning of an entire field of small sensors is achieved here with the use of a single lighthouse base station that has to "see" all sensors anyway to collect sensed data. A view of the idealized principle is illustrated in Fig. 2. Using a parallel beam that rotates at a constant speed, the base station sweeps over the entire field of nodes. The observer has to be equipped with a clock and a photo detector. By knowing the rotational speed and width of the beam, and measuring the time it sees the light, the observer is able to independently estimate its range to the base station, effectively placing itself on a cylinder around the lighthouse. Using three such ranging systems, it is possible to position points in a plane based on the procedure of trilateration. The authors, however, aimed to have a unique positioning device, so they used three mutually perpendicular lighthouses (Fig. 3) in order to measure distances to the three axes. A node can obtain its 3D position as an intersection of three cylinders with the support of a single base station. The 2D prototype system can position nodes at a distance of 14 m, with a relative accuracy of 2.2 percent and a relative standard deviation of 0.68 percent.

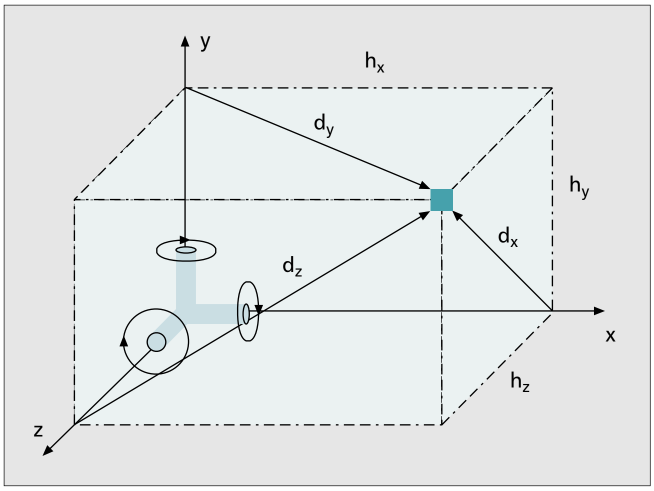
Distributed Methods

The Ad Hoc Localization System

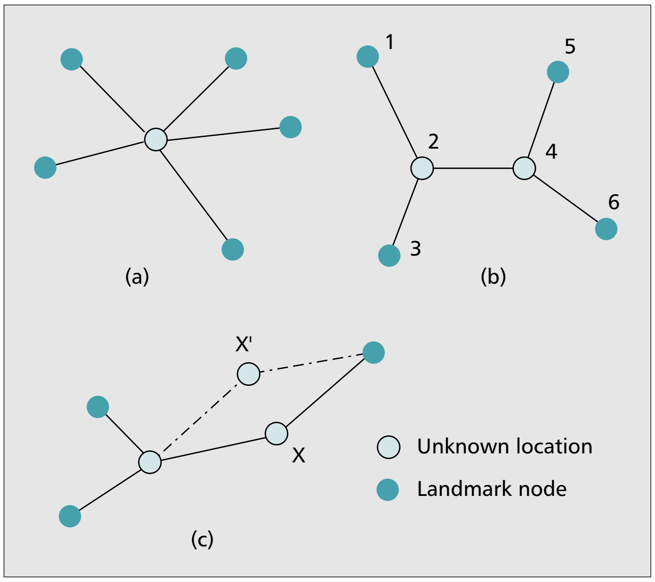
The ad hoc localiztion system (AhLOS) [7] defines several types of multilateration: atomic, iterative, and collaborative. In atomic multilateration (Fig. 4a) landmark density is high enough that a node has enough neighbors to apply basic trilateration. Once at least three distances to three known points are known, a node may compute its own location. Using iterative multilateration, nodes that manage to obtain a location behave as landmarks for other nodes, albeit with reduced accuracy. But even after applying these two methods there are nodes unable to find a position, and an example is shown in Fig. 4b: neither node 2 or 4 may become a landmark, and the problem must be considered in a collaborative fashion. The main contribution of the method is that it identifies such groups of nodes whose only method of getting a location is group collaboration. In this particular example, since distances indicated by lines are known, the group is able to build a nonlinear system using an equation for each edge in the graph having as unknowns the four coordinates corresponding to nodes 2 and 4.

The number of equations and number of unknowns are not always good indicators of the feasibility of the system. For example, in Fig. 4c even if four equations are available to solve for four unknowns, node x is not able to resolve the ambiguity. Algorithm 1 is used by AhLOS to decide the feasibility of collaborative multilateration by having non-landmark nodes call the algorithm with parameters *node* as the node's own identifier, *callerID* as the identifier of the previous node in the recursion, and *isInitiator* set to **true** if the node initiates the recursive process. *beaconCount* is a function that returns the number of neighboring beacons of a node. The algorithm is designed to catch situations like that in Fig. 4b, but may not be able to identify larger groups of nodes.

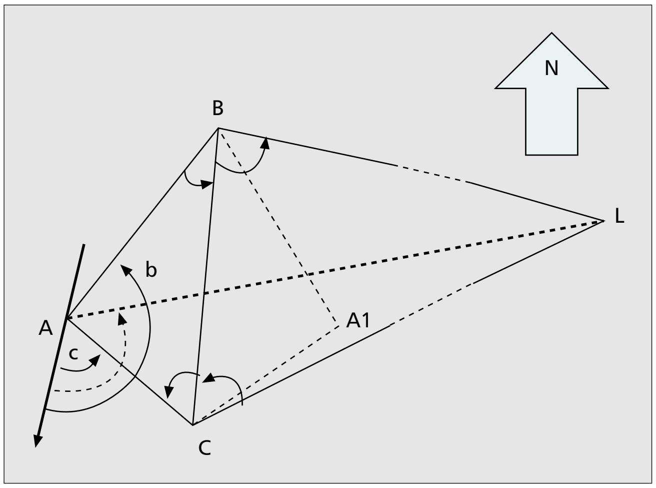
The disadvantage of AhLOS is that it requires a rather high



■ Figure 3. *A 3D base station using three mutually perpendicular lighthouses (adapted from [6]).*



■ Figure 4. *AhLOS multilateration (adapted from [7]).*



■ Figure 5. *APS propagation using local capabilities — Euclidean uses internode range measurements; DV-bearing uses AOA measurements (adapted from [8]).*

percentage of landmarks in order to achieve a high percentage of resolved nodes. For example, with an average node degree of 6.28, to resolve 90 percent of the regular nodes requires a density of 45 percent landmarks. The advantage is that given a good ranging method, it is likely to produce high-quality positions.

Ad Hoc Positioning Systems

The ad hoc positioning system (APS) [8] is a conceptual hybrid between two major ideas: distance vector (DV) routing and beacon based positioning (GPS). What makes it similar to DV routing is the fact that information is forwarded hop by hop, independent 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 APS concept has been shown to work using range and angle measurements, and with multimodal and heterogenous capabilities. While an arbitrary combination of capabilities may not guarantee support for a positioning scheme, there are a number of positioning schemes that are appropriate for certain combinations of capabilities. For example, if ranging or angle of arrival (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.

APS implements 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 or angle between node's axis and another object, and *range* means straight line distance or an estimation of it.

All propagations work very much like a mathematical induction proof. The fixed point: nodes immediately adjacent to a landmark get their orientations/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. APS defines several propagation methods to compute this induction step for any combination of local capabilities: none (mere connectivity), 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 its 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 to be beneficial in certain cases.

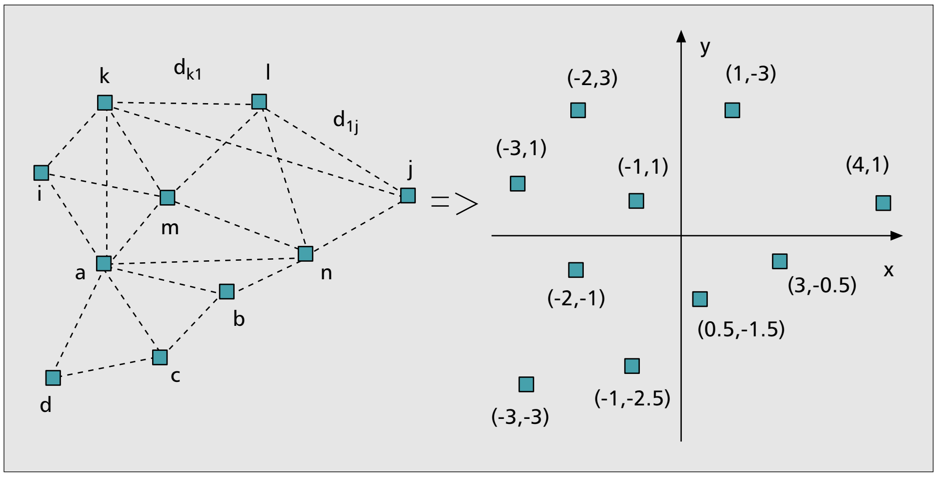
The most basic method, *DV-hop*, uses mere connectivity to infer estimates of ranges to landmarks. It employs two stages, one in which nodes obtain shortest paths in hops to a number of landmarks, and a second one in which the average size of a hop is estimated by a landmark and distributed to nearby nodes. Distance in hops to a landmark is then multiplied by the estimated size of a hop to produce an estimate of the Euclidean distance to the landmark. *DV-distance* uses a similar procedure, but the shortest distance between nodes and landmarks uses sums of measured (sensed) distances instead of hops.

Euclidean is the method by which nodes are able to estimate exact Euclidean distance to landmarks when exact measurements are available. In Fig. 5 node A has distances to immediate neighbors B and C, which have distances to each other and also to faraway landmark L. In the quadrilateral ACLB all sides and one diagonal are known, so node A can infer the second diagonal, AL, which is its range to L, once local ambiguities (A1) are resolved. After ranges to landmarks are obtained using *DV-hop*, *DV-distance*, or *Euclidean*, trilateration is applied independently by each node in order to obtain a position.

If only AOA measurements and compasses are available, *DV-bearing* uses similar logic to *Euclidean* to infer and propagate bearings (angles) to landmarks. If AOA provides bear-

```
boolean isCollaborative (node, callerID, isInitiator)
if isInitiator == true
  limit ← 3
else
  limit ← 2
count ← beaconCount(node)
if count ≥ limit return true
foreach unknown neighbor i not previously visited
  if isCollaborative(i, node, false) count++
  if count ≥ limit return true
endfor
return false
```

■ Algorithm 1. *The feasibility of collaborative multilateration in AhLOS (adapted from [7]).*



■ Figure 6. *SPA: nodes build local coordinate systems based on distances to neighbors (adapted from [9]).*

ings to neighbors, indicated by continuous arrows in Fig. 5, node A is able to infer its bearing to L, shown by a dashed arrow, without needing any range measurement. The actual geometric formulas are different for the propagation of the angle, but the induction argument is the same. Once a node has at least two or three bearings to landmarks, depending on whether it has a compass or not, it may infer its own location by means of triangulation (a procedure similar to trilateration used by GPS, except that it only uses angles to landmarks).

The advantage of APS methods is that they are distributed and localized, support some limited mobility and variable duty cycles, and can accommodate a wide degree of capabilities, from mere connectivity to a multimodal combination of ranges, angles, and compasses. The disadvantages are that they require a somewhat uniform distribution of landmarks, and their DV nature will face increasing costs in the face of high mobility.

Relative Positioning

The Self Positioning Algorithm

The Self Positioning Algorithm (SPA) [9] finds positions in a coordinate system determined by a group called the *location reference group* (LRG). The obtained positions are coherent across the entire network, and thus may be used as support for position-based services. In a first step each node exchanges with its neighbors a table containing all its incident edges. This amounts to each hop having access to second-hop information. This is then processed to produce a local coordinate system, as shown in Fig. 6. Choosing three nodes as descriptors for the origin and the two axes, the algorithm proceeds to insert all the additional one- or two-hop neighbors such that they respect the measured ranges between them. This is in fact an iterative procedure with the same goal as MDS-MAP, but its scope limited to the neighborhood of each node. Once each node independently establishes its local coordinate system, an alignment procedure initiated by the LRG aligns all the coordinate systems to the coordinate systems of the reference group.

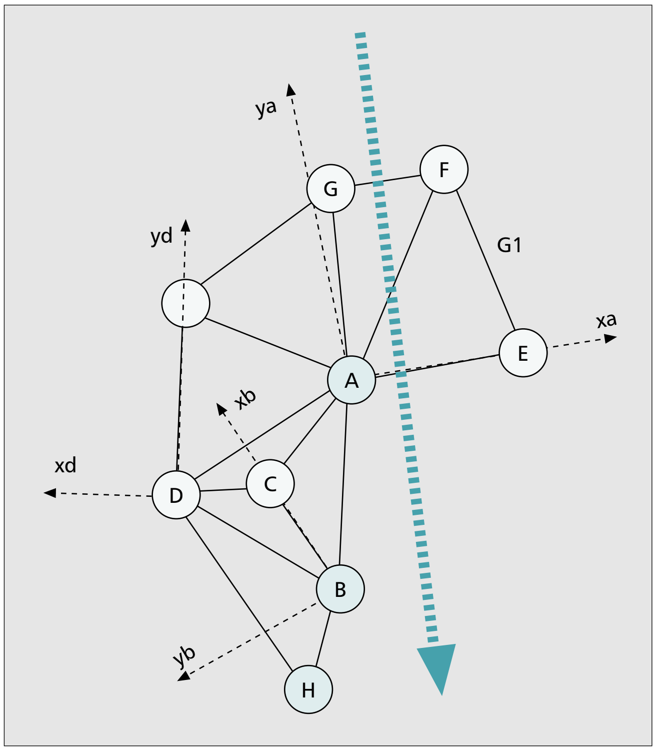
The advantages of SPA are that it provides network-wide coherence in positions without the need for landmarks. But this reliance on the LRG can be costly in front of mobility, even if the reference group is based on nodes with limited mobility and decreased probability of disconnection.

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)

■ Table 1. *Hardware capabilities and associated transformations; T: translation; R: rotation; S: scaling; M: mirroring.*

The Local Positioning System

Extending the ideas used by SPA, [10] develops 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. The local positioning system (LPS) is a method to achieve positioning *only for the nodes participating in forwarding*, with minor increase in communication cost, as if all node positions were known. Instead, each node touched by a trajectory (Fig. 7) spends some computation to position itself in the coordinate system of the source of the packet. Localized positioning can be used in any position-centric network when a communicating group agrees on a common coordinate system. For example, Cartesian forwarding [2] — the simplest position-centric scheme — greedily chooses the next hop clos-



■ Figure 7. *LPS only positions nodes involved in communication.*

est to the destination, approximating a straight line from source to destination. Only nodes on this line are actually positioned in the coordinate system of the source, trading off communication spent during discovery for in-node computation to translate coordinates. The latter is more economical energy-wise than the communication spent by network-wide positioning algorithms like APS or AhLOS.

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. When mirroring is indicated in parentheses, 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 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, a situation that can be robustly detected by a digital accelerometer.

The advantage of the method is the reduced load it places on the network, more appropriate for highly mobile scenarios when a globally coherent coordinate system is unnecessary or expensive to maintain.

Summary

Positioning in ad hoc sensor networks is an essential service, important as both a goal and a mean. It is a goal for most sensor networks in order to label reported data and a mean for most ad hoc networks to implement network management services, such as routing and querying. The ad hoc nature of these networks, their large size, and the diversity of their deployment conditions make line of sight solutions such as GPS less desirable. As in most cases in sensor networks, the design choices are closely linked to the particular problems, and there is no "one size fits all" positioning algorithm for all applications. The multihop algorithms surveyed in this article offer different trade-offs in terms of capabilities required, quality of positions obtained, and load on the network, but a comprehensive quantitative comparison is necessary to better understand these trade-offs.

References

[1] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *IEEE Comp.*, vol. 34, no. 8, 2001, pp. 57–66.

[2] G. Finn, "Routing and Addressing Problems in Large Metropolitan-scale Internetworks," Tech. rep. ISI/RR-87-180, Univ. of Southern CA, Mar. 1987.

[3] I. Stojmenović, "Position-based Routing in Ad Hoc Networks," *IEEE Commun. Mag.*, vol. 40, no. 7, July 2002, pp. 128–34.

[4] L. Doherty, L. E. Ghaoui, and K. S. J. Pister, "Convex Position Estimation in Wireless Sensor Networks," *IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[5] Y. Shang et al., "Localization from Mere Connectivity," *ACM MOBIHOC*, Annapolis, MD, June 1–3 2003.

[6] K. Römer, "The Lighthouse Location System for Smart Dust," *ACM/USENIX Conf. Mobile Sys., Apps., and Svcs.*, San Francisco, CA, May 2003, pp. 15–30.

[7] A. Savvides, C.-C. Han, and M. Srivastava, "Dynamic Fine-grained Localization in Ad-hoc Networks of Sensors," *ACM MOBICOM*, Rome, Italy, 2001.

[8] D. Niculescu and B. Nath, "Position and Orientation in Ad Hoc Networks," *Elsevier Ad Hoc Networks*, vol. 2, no. 2, Apr. 2004, pp. 133–51.

[9] S. Čapkun, M. Hamdi, and J.-P. Hubaux, "GPS-free Positioning in Mobile Ad-hoc Networks," *Hawaii Int'l. Conf. Sys. Scis.*, Outrigger Wailea Resort, Jan. 3–6 2001, HICSS-34.

[10] D.Niculescu and B. Nath, "Localized Positioning in Ad Hoc Networks," *Elsevier Ad Hoc Networks*, vol. 1, no. 2–3, Sept. 2003, pp. 247–59.

Biography

DRAGOŞ NICULESCU (dnicules@cs.rutgers.edu) received a B.S. degree in computer engineering from University Politehnica of Bucharest in 1994, and a Ph.D. degree in computer science from Rutgers University in 2004. His research interests are in mobile computing and networking.