Telecommunication Systems 22:1–4, 267–280, 2003


# DV Based Positioning in Ad Hoc Networks

DRAGOŞ NICULESCU * and BADRI NATH dnicules@cs.rutgers.edu
*Division of Computer and Information Sciences, Rutgers, The State University of New Jersey, 110 Frelinghuysen Road Piscataway, NJ 08854-8019, USA*

**Abstract.** Many ad hoc network protocols and applications assume the knowledge of geographic location of nodes. The absolute position of each networked node is an assumed fact by most sensor networks which can then present the sensed information on a geographical map. Finding position without the aid of GPS in each node of an ad hoc network is important in cases where GPS is either not accessible, or not practical to use due to power, form factor or line of sight conditions. Position would also enable routing in sufficiently isotropic large networks, without the use of large routing tables. We are proposing APS – a localized, distributed, hop by hop positioning algorithm, that works as an extension of both distance vector routing and GPS positioning in order to provide approximate position for all nodes in a network where only a limited fraction of nodes have self positioning capability.

**Keywords:** positioning, ad hoc networks, APS

## Introduction

Ad hoc networks have mostly been studied in the context of high mobility, high power nodes, and moderate network sizes. Sensor networks, while typically having low powered nodes, low mobility and large sizes, classify as ad hoc networks in many cases, when deterministic placement of nodes is not possible. With recent advances in sensing device architectures [Hill et al., 7], 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: routing without the use of large conventional routing tables, adaptability in front of intermittent functioning regime, network partitioning and survivability. In this paper, we address the problem of self locating the nodes in the field, which may provide a solution to the first challenge, and solve other practical problems as well. 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, which makes it possible to attach this information to a geographical map of the monitored region. If this is an absolute necessity in order to make sense of the observed data, accurate position might also be useful for routing and coordination purposes. Algorithms such as GEDIR [Stojmenovic and Lin, 16], or GEOCAST [Navas and Imielinski, 10], en-

* Corresponding author.

able routing with reduced or no routing tables at all, which are appropriate for devices like the Rene mote [Hill et al., 7], with only half a kilobyte of RAM. An improvement that can be applied to some ad hoc routing schemes, Location Aided Routing [Ko and Vaidya, 9] limits the search for a new route to a smaller request zone. Also, APS is appropriate for indoor location aware applications, when the network's main feature is not the unpredictable, highly mobile topology, but rather deployment that is temporary, and ad hoc. These networks would not justify the cost of setting up an infrastructure to support positioning, like proposed in [Bahl and Padmanabhan, 1; Bulusu et al., 3; Priyantha et al., 13].

GPS, which is a public service, can satisfy some of the above requirements. However, attaching a GPS receiver to each node is not always the preferred solution for several reasons: *cost* – if we are envisioning networks of thousands, or tens of thousands of nodes, (this factor might be of diminished importance in the future); *limited power* – battery capacities are increasing much slower than, say Moore's law; *inaccessibility* – nodes may be deployed indoors, or GPS reception might be obstructed by climatic conditions; *imprecision* – even with the selective availability recently turned off (May 2000), the positioning error might still be of 10–20 m, which might be larger the hop size of some networks; *form factor* – a Rene board [Hill et al., 7] is currently the size of a small coin.

There are several requirements a positioning algorithm has to satisfy. First, it has to be distributed: in a very large network of low memory and low bandwidth nodes, designed for intermittent operation, even shuttling the entire topology to a server in a hop by hop manner would put too high a strain on the nodes close to the basestation/server. Partitioned areas would make centralization impossible, and anisotropic networks would put more strain on some nodes that have to support more forwarding traffic than others. Changing topologies would also make the centralized solution undesirable. Second, it has to minimize the amount of node to node communication and computation power, as the radio and the processor are the main sources of draining battery life. Also, it is desirable to have a low signaling complexity in the event a part of the network changes topology. Third, the positioning system should work even if the network becomes disconnected – in the context of sensor networks, the data can be later collected by a fly-over basestation. Finally, our aim is to provide absolute positioning, in the global coordinate system of the GPS, as opposed to relative coordinates, for the following reasons: relative positioning might incur a higher signaling cost in the case the network topology changes, and absolute positioning enables a unique namespace, that of GPS coordinates.

The rest of the paper is organized as follows: the next section summarizes similar efforts in current research, section 2 presents a short GPS review, as its principles are central to our approach. Section 3 explains the APS approach, with the proposed propagation methods, section 4 presents simulation results and we conclude with some considerations about node mobility effects on APS.

## 1. Related work

Doherty et al. [5] propose a positioning scheme that works in a centralized manner by collecting the entire topology in a server and then solving a large optimization problem that minimizes positioning errors for each node. Capkun et al. [4] presents a relative positioning system, without the use of GPS, in which the origin of the coordinate system is voted by a collection of nodes called reference group. The disadvantages, besides the ones stemming from the relative positioning versus absolute, are that when the reference moves, positions have to be recomputed for nodes that have not moved, and if intermediate nodes move, fixed nodes depending on them also have to recompute position (not knowing if the reference has moved). However, the coordinate system propagation is appropriate for hop by hop dissemination of distances to landmarks, and is applicable with our distance based scheme. Bulusu et al. [3] present a position system based on an uniform grid of powerful (compared to the nodes) basestations, which serves as landmark mesh. A random node in the network will be able to localize itself by estimating its distance to the well-known positions of closest basestations. RADAR [Bahl and Padmanabhan, 1] is a scheme in which the entire map is in advance measured for its radio propagation properties, and positioning is achieved by recognizing fingerprints of previously mapped positions. The Cricket location system [Priyantha et al., 13] uses radio and ultrasound signals to estimate Euclidean distances to well-known beacons, which are then used to perform triangulation. The key features of our proposed approach, in contrast with the ones mentioned above, are that it is decentralized, it does not need special infrastructure, and provides absolute positioning. AhLOS [Savvides et al., 15], which is the approach most similar to ours, groups of nodes which can collaborate in resolving their positions but involves solving of large nonlinear systems, depending on the sizes of those groups.

## 2. GPS review

In Global Positioning System (GPS) [Parkinson and Spilker, 12], trilateration uses ranges to at least four known satellites to find the coordinates of the receiver, and the clock bias of the receiver. For our node 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 triangulation procedure starts with an apriori estimated position that is later corrected towards the true position. Let $\hat{r}_u$ be the estimated position, $r_u$ the real position, $\rho_i = |r_i - r_u| + \varepsilon_i$ and $\hat{\rho}_i = |r_i - \hat{r}_u| + \hat{\varepsilon}_i$ the respective ranges to the GPS $i$. The distance equation to each satellite is $\rho_i^2 = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2}$. The correction of the range, $\Delta\rho$ is approximated linearly using Taylor expansion. If $\hat{J}_i$ is the unit vector of $\hat{\rho}_i$, $\hat{J}_i = (r_i - \hat{r}_u)/|r_i - \hat{r}_u|$ and $\Delta r = \hat{r}_u - r_u$, then the approximate of the correction in the range is: $\Delta\rho = \hat{\rho}_i - \rho_i \simeq \hat{J}_i \cdot \Delta r + \Delta\varepsilon$. Performing the above approximation for each satellite independently leads to a linear system in which the unknown is the position correction $\Delta r = [\Delta x \ \ \Delta y]$:

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

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

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

$$\Delta r = \left(J^T W J\right)^{-1} J^T W \Delta\rho.$$

After each iteration, the corrections $\Delta x$ and $\Delta y$ are applied to the current position estimate. The iteration process stops when the correction in position is below a chosen threshold.

## 3. Ad hoc Positioning System (APS)

If a graph is sufficiently connected, and the lengths of its edges are all known, then its plane topology may be reconstructed. But what is a *sufficient* degree of connectivity? If we assimilate the graph with a wire frame, where nodes act as hinges, our goal is to determine which part of the graph has non-moving parts, and those will be the nodes which can determine their position. Once such a wire-frame is fixed, it will have a reference system of its own, that eventually has to be aligned to the global coordinate system of the GPS. In order to fix this wire frame somewhere on the global plane, at least three nodes (called landmarks), that are GPS enhanced, or know their position by some other means, have to be present in the connected graph. We assume that landmarks are deployed randomly and uniformly across the network.

Devices as simple as the Rene motes have software access to the signal strength of the radio signal, thus offering a way to estimate distance to immediate neighbors. Other methods, such as TDOA, with accuracy higher than signal strength estimation, may also be used to infer range. One of the aims of our positioning system is to enhance position accuracy as the fraction of landmarks of the entire population increases. Even if it is theoretically sufficient to have three landmarks, the presence of measurement errors will demand higher fractions of landmarks, depending on the requirements of the application.

### 3.1. APS algorithm

It is not desirable to have the landmarks emit with large power to cover the entire network for several reasons: collisions in local communication, high power usage, coverage problems when moving. Also, it is not acceptable to assume some fixed positions for the landmarks, as the applications we envision are either in flight deployments over inaccessible areas, or possibly involving movement and reconfiguration of the network. In this case, one option is to use hop by hop propagation capability of the network to forward distances to landmarks. In general, we aim for the same principle as GPS, with the difference that the landmarks are contacted in a hop by hop fashion, rather than directly, as ephemerides are. In what follows we will refer to one landmark only, as the algorithm behaves identically and independently for all the landmarks in the network. It is clear that the immediate neighbors of the landmark can estimate the distance to the landmark by direct signal strength measurement. Using some propagation method, the second hop neighbors then are able to infer their distance to the landmark, and the rest of the network follows, in a controlled flood manner, initiated at the landmark. Complexity of signaling is therefore driven by the total number of landmarks, and by the average degree of each node.

What makes this method similar with the distance vector routing, is that at any time, each node only communicates with its immediate neighbors, and in each message exchange it communicates its available estimates to landmarks acquired so far. This is appropriate for nodes with limited capabilities, which do not need, and cannot handle the image of the entire, possible moving, network. We are exploring several methods of hop to hop distance propagation and examine advantages and drawbacks for each of them. Each propagation method is appropriate for a certain class of problems as it influences the amount of signaling, power consumption, and position accuracy achieved.

Once an arbitrary node has range estimates to a number ($\geqslant 3$) of landmarks, it can compute its own position in the plane, using a similar procedure with the one used in GPS position calculation described in the previous section. The estimate we start with is the centroid of the landmarks collected by the node. A node might not succeed in getting a position for a variety of reasons. It might not have enough landmarks collected, or the ranges are faulty and produce an unsolvable system, or the starting point is too far off from the true position. In such cases, when trilateration fails, a node uses more successful neighbors as additional landmarks and trilaterate to obtain a position. This method is only available if the distance propagation method uses signal strength. Theoretically, this process may continue until all nodes get some position estimate [Savarese et al., 14]. The convergence of this process, and therefore the amount of signalling, depends on the fraction of nodes that are successfully position after each step.

### *3.2. "DV-hop" propagation method*

This is the most basic scheme, and it comprises of three non-overlapping stages. First, it employs a classical distance vector exchange so that all nodes in the network get distances, in hops, to the landmarks. Each node maintains a table $\{X_i, Y_i, h_i\}$ and exchanges updates only with its neighbors. In the second stage, a landmark, after it cumulates distances to other landmarks, it estimates an average size for one hop, which is then deployed as a correction to the nodes in its neighborhood. 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

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

In the example in figure 1, nodes $L_1$, $L_2$ and $L_3$ are landmarks, and node $L_1$ has both the Euclidean distance to $L_2$ and $L_3$ , and the path length of 2 hops and 6 hops, respectively. $L_1$ computes the correction $(100 + 40)/(6 + 2) = 17.5$, which is, in fact, the estimated average size of one hop, in meters. $L_1$ has then the choice of either computing a single correction to be broadcasted into the network, or preferentially send different corrections along different directions. In our experiments we are using the first option. In a similar manner, $L_2$ computes a correction of $(40 + 75)/(2 + 5) = 16.42$ and $L_3$ a correction of $(75 + 100)/(6 + 5) = 15.90$. A regular node gets an update from one of the landmarks, and it is usually the closest one, depending on the deployment policy and the time the correction phase of APS starts at each landmark. Corrections are distributed by controlled flooding, meaning that once a node gets and forwards a correction, it will drop all the subsequent ones. This policy ensures that most nodes will receive only one correction, from the closest landmark. When networks are large, a method to reduce signaling would be to set a TTL field for propagation packets, which would limit the number of landmarks acquired by a node. Here, controlled flooding helps keeping the corrections localized in the neighborhood of the landmarks they were generated from, thus accounting for nonisotropies across the network. In the above example, assume $A$ gets its correction from $L_2$ – its estimate distances to the three landmarks would be: to $L_1$ – $3 \cdot 16.42$, to $L_2$ – $2 \cdot 16.42$, and to $L_3$, $3 \cdot 16.42$. This values are then plugged into the triangulation procedure described in the previous section, for $A$ to get an estimate position.

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. The advantages are its simplicity and the fact that it does not depend on range measurement error. Measuring

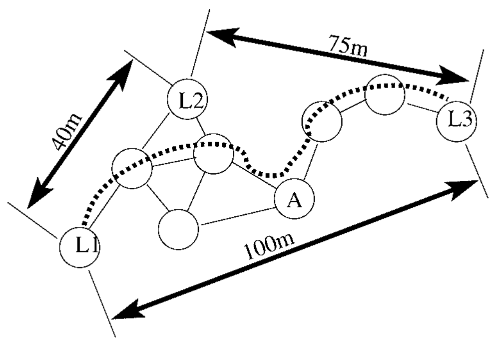



Figure 1. "*DV-hop*" correction example.

ranges to neighbors however, turns out to be of some advantage for the next propagation schemes, which employ another stage of trilateration for the unsuccessful nodes, using successful neighbors as landmarks.

### 3.3. *"DV-distance" propagation method*

This method is similar with the previous one with the difference that distance between neighboring nodes is measured using radio signal strength and is propagated 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, on the other hand it is sensitive to measurement errors.

### 3.4. *"Euclidean" propagation method*

The third scheme works by propagating the true *Euclidean* distance to the landmark, so this method is the closest to the nature of GPS. An arbitrary node $A$ needs to have at least two neighbors $B$ and $C$ which have estimates for the landmark $L$ (figure 2). $A$ 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 distance from $A$ to the landmark $L$. It is possible that $A$ is on the same side of $BC$ as $L$ – shown as $A'$ 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 $C$. If it cannot be chosen clearly between $A$ and $A'$, an estimate distance to $L$ will not 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 Pythagoras' generalized theorem in triangles $ACB$, $BCL$, and $ACL$, to find the length of $AL$. An error reduction improvement

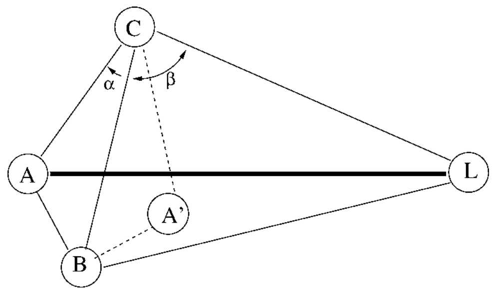



Figure 2. *Euclidean* propagation method.

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%.

$$\begin{aligned}
\cos(\alpha) &= \frac{AB^2 - AC^2 - BC^2}{2 \cdot AC \cdot BC}, \\
\cos(\beta) &= \frac{BL^2 - BC^2 - CL^2}{2 \cdot CL \cdot BC}, \\
AL^2 &= AC^2 + CL^2 - 2 \cdot AC \cdot CL \cos(\beta \pm \alpha), \\
\sigma_{AL}^2 &= \sum \left( \frac{\partial AL}{\partial e} \right)^2 \sigma_e^2, \quad e = AC, CL, LB, BA, BC.
\end{aligned}$$

The uncertainty $\sigma_{AL}$ 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. Once a node has ranges to three landmarks, it may, by itself, estimate its position.

### *3.5. "DV-coordinate" propagation method*

The fourth method we consider, is similar to the idea proposed in [Capkun et al., 4]. This method requires some preprocessing stage that has to complete before the DV propagation starts. 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 figure 3, node $A$, based on the ranges from itself to its neighbors and the ranges between those neighbors, can chooses 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 incrementally by choosing two neighbors as indicators for axes of coordinates. We chose the second approach, since the nonlinear optimization might not scale to higher degrees and needs a good starting 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. Every node in the network independently builds its own coordinate system centered at itself.

The next preprocessing step is registration with the neighbors. If, for example, 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 obtain through the process of registration [Horn et al., 8]. 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. Each coordinate

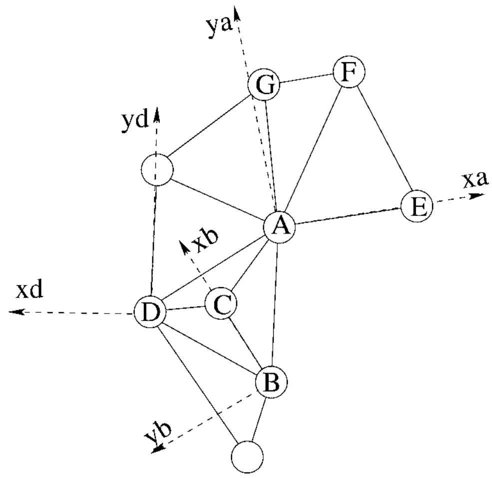



Figure 3. Local coordinate systems.

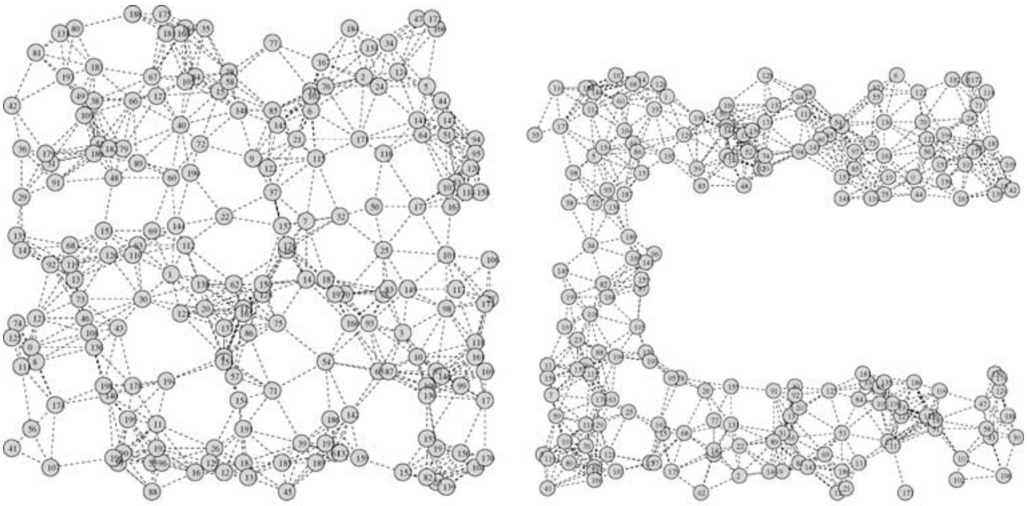
received from a neighbor needs to be translated by a node in order to be consistent with its own coordinate system. Computing this matrices before hand is just an optimization choice applicable to static networks, as registration can be performed on the fly, whenever communication between two neighbor occurs. Note, however, that the complexity of registration is linear in the number of neighbors used for registration (4 in the above example), and cubic in the number of coordinates.

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. 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 in the 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.

Error control for "*DV-coordinate*" propagation method could use the registration residual error as a measure of uncertainty. Uncertainty in obtained ranges is then amplified by the local uncertainty resulted from the registration to produce a new uncertainty for the range that will be propagated to other nodes.

## 4. Simulation results

We simulated APS with the proposed propagation methods, with randomly generated topologies of 200 nodes, with average degree 9. Two topologies are considered (fig-

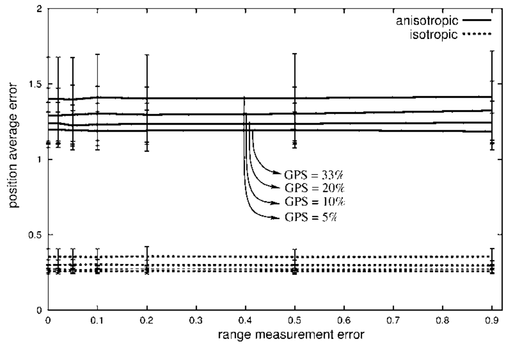


(a) isotropic (b) anisotropic

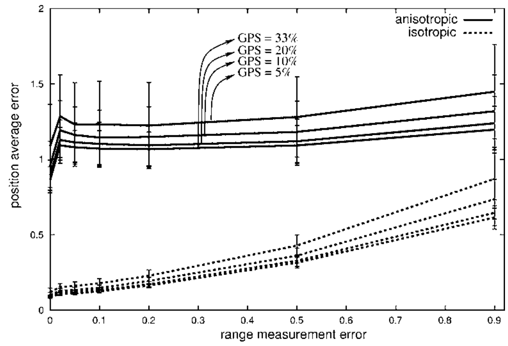
Figure 4. Test topologies.

ure 4): an isotropic one, where nodes are placed in a random uniform manner, so that density, connectivity and communication range are approximately the same throughout the network, an anisotropic one, with nonuniform connectivity – it has the shape of letter $C$, so that number of hops between the north and south branches is not a correct indication of geometric distance. All the performance graphs presented have the range measurement error on the horizontal axis and several curves corresponding to percentage of landmarks in the network. The measurement error is considered to be in the range 2–90% of the nominal value of the range, uniformly distributed. Each point indicates the standard deviation obtained after 400 runs with random errors in the ranges used throughout the network. All results (except "*DV-Hop*") are obtained after running the required phases for each algorithm, followed by a one time step in which unsuccessful nodes employ their neighbors in triangulation.

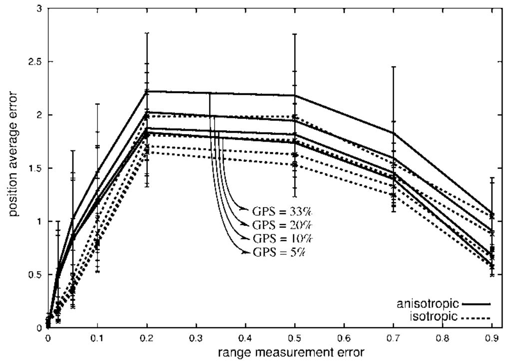
Figures 5(a) and 5(b) show position error, relative to the hop size for "*DV-hop*" and "*DV-distance*" in both topologies. *Since* "*DV-hop*" *does* not make any use of ranging, its performance is insensitive to error. "*DV-distance*" on the other hand shows degradation in positioning as ranging error increases. These two methods exhibit high variance across topologies – the anisotropic case performing worse, due to the assumption that distance in hops is an indication of true distance. Fraction of nodes that obtain a position is above 97% for isotropic and 88% for anisotropic for both cases (not shown on the figure). "*DV-coordinate*" (figure 5(c)) and "*Euclidean*" (figure 5(d)) behave more consistently across topologies, but have different performance tradeoffs. "*Euclidean*" performs better in the anisotropic topology and competes fairly in the isotropic one with a high landmark ratio. Success rate, however, for these two (figures 6(a), (b)) is not as good as for "*DV-hop*" and "*DV-distance*". In many cases, "*Euclidean*" may either



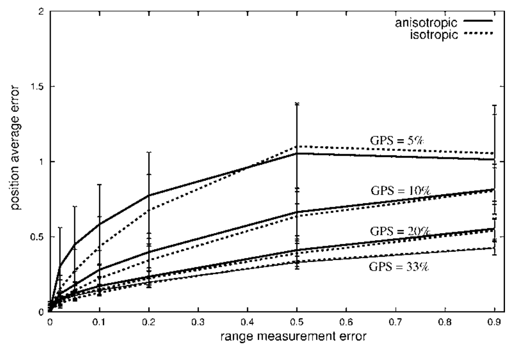

(a) DV-hop




(b) DV-distance

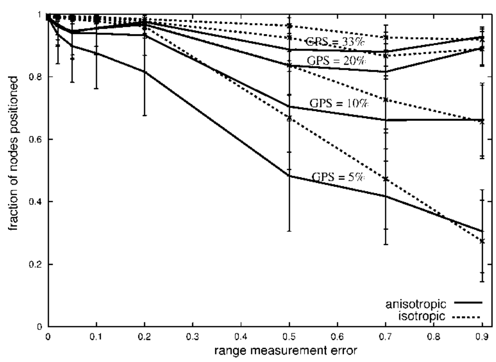



(c) DV-coordinate

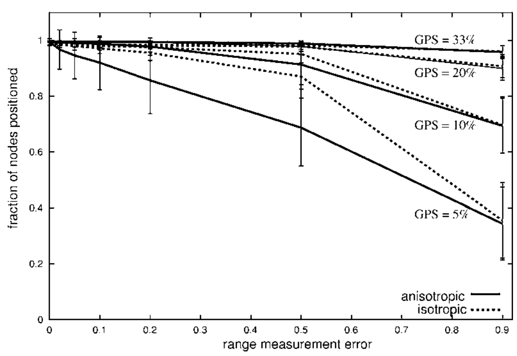



(d) Euclidean

Figure 5. Positioning error.




(a) DV-coordinate




(b) Euclidean

Figure 6. Positioning success rate.

not be able to resolve ambiguities, or obtain faulty ranges to $L$ (figure 2), for example, ranges that do not satisfy the triangle inequality. In those cases, nodes are not able to propagate ranges for certain landmarks, thus reducing the number of landmarks that are finally collected. In the case of "*DV-coordinate*", the registration process may fail when there is no sufficient overlapping, or faulty overlapping between neighboring nodes. This method, however, does not have an error control method similar to the uncertainty used for "*Euclidean*". What this means is that ranges are propagated with errors, and the trilateration phase may fail due to the impossible constellation obtained, which manifests as a numerical instability. This explains the peak in degradation of "*DV-coordinate*" – as error increases, the registration process is not able to produce translation matrices for many neighbors, therefore reducing the propagation of landmark information, and eventually reducing the success rate of the final trilateration. Higher measurement errors in fact reduce the amount of DV propagation, and ranges obtained are generally shorter than in the case where registration is more permissive, as in the case with lower measurement errors.

The way in which errors are propagating is the factor which determines which nodes can successfully estimate their position. Some nodes may not have an estimate due to not having at least three estimates to three non-collinear landmarks, or not attaining convergence during the iterative system solving. In practice, successful nodes may become landmarks, albeit imprecise ones, which may help in positioning nodes that are left. Therefore, APS could work in an iterative manner until all nodes are positioned.

Message complexity is relevant because usually nodes communicate over a shared medium, and a high density of nodes, coupled with a high messaging complexity, leads to a high collision rate and ultimately to lower throughput and higher power consumption. Although having one extra stage for the propagation of corrections, "*DV-hop*" and "*DV-distance*" use less signaling than "*Euclidean*", that needs second hop information, which depends on the square of the degree of the network. "*DV-coordinate*" uses even more communication, making use not only of second hop information for registration, but sends two coordinates for each range, instead of one.

To evaluate how effective the APS estimated positions are for purposes of routing, we implemented a simple, greedy version of cartesian routing [Finn, 6]. Having the coordinates $(X, Y)$ of the packet destination, a forwarding node will choose as the next hop the neighbor that estimates the least Euclidean distance to $(X, Y)$. There are no routing loops because when all neighbors declare a larger distance than the forwarding node, the packet is dropped. This obviously works better for isotropic networks and this is the case that we simulated. The algorithm does not guarantee delivery, such algorithms are described elsewhere in the literature [Bose et al., 2]. Simulation showed that when using APS, path lengths increase on average with no more than 6% over the case of geodesic routing when using true positions, unaffected by errors. For these, and other simulation results comparing three of the mentioned propagation methods, see [Niculescu and Nath, 11].

## 5. Node mobility

Although we have not explicitly modeled mobility, APS aims to keep a low signaling complexity in the event network topology changes. While highly mobile topologies, usually associated with ad hoc networks, would require a great deal of communication to maintain up to date position, we envision ad hoc topologies that do not change often, such as sensor networks, indoor or outdoor temporary infrastructures. When a node moves, it will be able to get *DV* updates from its new neighbors and triangulate to get its new position, therefore communication remains localized to nodes that are actually mobile. This is in contrast with previously proposed solutions [Capkun et al., 4], which rely on a reference group that would prompt reevaluations in the entire network in case of movement of the reference group. 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 APS algorithm, as the new position of the landmark acts as a new landmark for both mobile and fixed nodes. To refer again to the sensor network example, we can envision a case when a single, fly-over GPS enabled node is in fact enough for an entire network. Later 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.

## 6. Conclusion

We presented APS (Ad hoc Positioning System), a method to extend the capabilities of GPS to non-GPS enabled nodes in a hop by hop fashion in an ad hoc network. Positioning is based on a hybrid method combining distance vector like propagation and GPS triangulation to estimate position in presence of signal strength measurement errors. APS has the following properties: is localized and distributed, does not require special infrastructure or setup, provides global coordinates and requires recomputation only for moving nodes. Several propagation methods were investigated, each providing a different tradeoff between accuracy, signaling complexity, coverage and the isotropy of the network. "*DV-hop*" and "*DV-distance*" algorithms behave well for most purposes and have a low signaling complexity. "*Euclidean*" provides better accuracy for non-isotropic topologies, and is generally more predictable in performance, at the cost of more communication. "*DV-coordinate*", although not as competitive, provides an interesting alternative which can be improved upon. Actual positions obtained by APS are on average less than one radio hop from the true position for all but one of our proposed propagation methods. Positions produced by APS are usable by cartesian and geographic routing algorithms, producing paths with less than 6% overhead over the paths produced with the true positions.

## References

[1] P. Bahl and V.N. Padmanabhan, Radar: An in-building rf-based user location and tracking system, in: *IEEE INFOCOM*, Tel Aviv, Israel, 2000.

[2] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, in: *3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Seattle, WA, 1999.

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

[4] S. Capkun, M. Hamdi and J. Hubaux, GPS-free positioning in mobile ad-hoc networks, in: *Hawaii International Conference on System Sciences, HICSS-34*, Outrigger Wailea Resort, 2001.

[5] L. Doherty, L.E. Ghaoui and 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, Technical Report, ISI Research Report ISI/RR-87-180, University of Southern California (1987).

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

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

[9] Y.-B. Ko and N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, in: *MobiCom'98*, 1998.

[10] J.C. Navas and T. Imielinski, Geographic addressing and routing, in: *MobiCom'97*, Budapest, Hungary, 1997.

[11] D. Niculescu and B. Nath, Ad hoc positioning system (APS), Technical Report DCS-TR-435, Department of Computer Science, Rutgers University (2001).

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

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

[14] C. Savarese, J. Rabaey and K. Langendoen, Robust positioning algorithms for distributed ad-hoc wireless sensor networks, Technical Report, Delft University of Technology (2001).

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

[16] I. Stojmenovic and 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.