GPS-free positioning in mobile ad hoc networks

Srdjan Čapkun, Maher Hamdi, Jean-Pierre Hubaux

Institute for computer Communication and Applications (ICA)
Communication Systems Department (DSC)
Swiss Federal Institute of Technology Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
(srdan.capkun, maher.hamdi, jean-pierre.hubaux)@epfl.ch

We consider the problem of node positioning in ad hoc networks. We propose a distributed, infrastructure-free positioning algorithm that does not rely on GPS (Global Positioning System). Instead, the algorithm uses the distances between the nodes to build a relative coordinate system in which the node positions are computed in two dimensions. Despite the distance measurement errors and the motion of the nodes, the algorithm provides sufficient location information and accuracy to support basic network functions. Examples of applications where this algorithm can be used include Location Aided Routing [8] and Geodesic Packet Forwarding [7]. Another example are sensor networks, where mobility is less of a problem. The main contribution of this work is to define and compute relative positions of the nodes in an ad hoc network without using GPS. We further explain how the proposed approach can be applied to wide area ad hoc networks.

Keywords: ad hoc, mobile, distributed, wireless networks, personal communication system, self-locating, radio-location, GPS-free, self-organization.

1. Introduction

The presented work is a part of the Terminode project, a 10 year ongoing project that investigates large area, wireless, mobile ad hoc networks [1,19,20]. The main design points of the project are to eliminate any infrastructure and to build a decentralized, self-organized and scalable network where nodes perform all networking functions (traditionally implemented in backbone switches/routers and servers).

In this paper, we describe an algorithm for the positioning of nodes in an ad hoc network that does not use GPS. The algorithm provides a position information to the nodes in the scenarios where an infrastructure does not exist and GPS cannot be used. GPS-free positioning is also desirable, when the GPS signal is too weak (e.g., indoor), when it is jammed, or when a GPS receiver has to be avoided for cost or integration reasons.

We introduce a distributed algorithm that enables the nodes to find their positions within the network area using only their local information. The algorithm is referred to as the Self-Positioning Algorithm (SPA). It uses range measurements between the nodes to build a Network Coordinate System (Fig 1.). The Time of Arrival (TOA) method is used to obtain the range between two mobile devices.

Despite the distance measurement errors and the motion of the nodes, the algorithm provides sufficient location information and accuracy to support basic network functions. Examples of applications where this algorithm can be used include Location Aided Routing [8] and Geodesic Packet Forwarding [7], both in ad hoc and sensor networks. In the Geodesic Packet Forwarding algorithm, the source sends packets in the physical direction of the destination node. Given that the node knows its position and the positions of the destination node in the relative coordinate system, it is able to com-

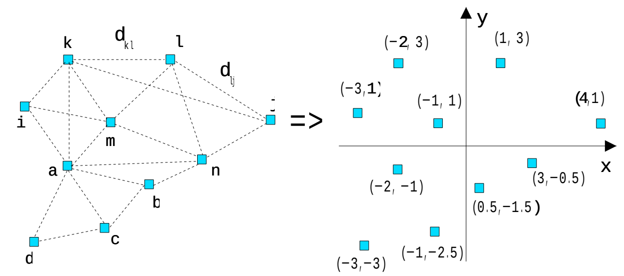


Figure 1. The algorithm uses the distances between the nodes and builds the relative coordinate system.

pute in which direction (to which next hop node) to send packets.

The nodes in the ad hoc networks are usually not aware of their geographical positions. As GPS is not used in our algorithm, we provide relative positions of the nodes with respect to the network topology.

For the sake of simplicity, we present the algorithm in two dimensions, but it can be easily extended to provide position information in three dimensions.

The paper is organized as follows. In section 2 related work in the field of radio-location techniques is described. In section 3 we present the algorithm for building a Local Coordinate System at each node. In section 4 we describe the means to define the center and the direction of the Network Coordinate System. In section 5 we discuss the influence of the range errors on the accuracy of the position estimation. We present the simulation results in section 6.

2. Related work

Following the release of US FCC regulations for locating E911 callers, positioning services in mobile systems have drawn much attention recently. The new regulations introduce stringent demands on the accuracy of mobile phone location. The FCC requires that by October 2001, the wireless operators locate the position of emergency callers with a root mean square error below 125m [9].

Several radio-location methods are proposed for locating the Mobile Stations (MSs) in cellular systems [2]: the Signal Strength method, the Angle of Arrival (AOA) method, the Time of Arrival (TOA) and Time Difference of Arrival (TDOA) methods. The Time of Arrival and Signal Strength methods use range measurements from the mobile device to several base stations to obtain its position. Thus, the accuracy of the estimated position depends on the accuracy of the range measurements. Distance measurements are corrupted by two types of errors: Non-Line of Sight (NLOS) error and measuring error. The measurements in cellular systems, taken by Nokia [3], show that NLOS error dominates the standard measurement noise, and tends to be the main cause of the error in range estimation. They also show that the location estimation error linearly increases with the distance error. Following these measurements, Wylie and Holtzman propose a method for the detection and correction of NLOS errors [5]. They show that it is possible to detect a NLOS environment by using the standard deviation of the measurement noise and the history of the range measurements. They propose a method for LOS reconstruction and they show that the correction is only possible if the standard measurement noise dominates the NLOS error. A different approach is presented in [4] by Chen. Chen shows that if the NLOS measurements are unrecognizable, it is still possible to correct the location estimation errors, if the number of range measurements is greater than the minimum required. The algorithm is referred to as the Residual Weighing Algorithm (Rwgh).

In [6] Rose and Yates give a theoretical framework of the mobility and location tracking in mobile systems. They present a study of mobility tracking based on user/service/host location probability distribution.

A significant amount of work has been reported considering context-aware computer systems. In these systems, mobile computers automatically configure themselves based on what was happening in the environment around them. Examples of these systems include Active Badge [14] and RADAR [15].

Recently, some position-based routing and packet forwarding algorithms for the ad hoc networks have been proposed in [7,8,16–18]. In all these algorithms it is assumed that the positions of the nodes are obtained through GPS.

To the best of our knowledge, no algorithms have been proposed for positioning of the nodes without GPS in ad hoc networks.

3. Local Coordinate System

We make the following assumptions on our system model:

- the observed network is an infrastructure-free network of mobile and wireless devices
- all the devices (nodes) have the same technical characteristics
- all the wireless links between the nodes are bidirectional
- the nodes use omnidirectional antennae
- the maximum speed of the movement of nodes is limited to 20 m/s

In this section we show how every node builds its Local Coordinate System. The node becomes the center of its own coordinate system with the position $(0, 0)$ and the positions of its neighbors are computed accordingly. In section 4 we further show how the agreement on a common coordinate system (Network Coordinate System) center and direction is achieved.

If a node j can communicate directly (in one hop) with node i, j is called a one-hop neighbor of i. Let N be the set of all the nodes in the network. $\forall i \in N$, we define K_i as the set of one-hop neighbors of i. Likewise, $\forall i \in N$, we define D_i, as a set of distances between i and each node $j \in K_i$. The neighbors can be detected by using beacons. After the absence of a certain number of successive beacons, it is concluded that the node is no longer a neighbor. The distances between the nodes are measured by some means, e.g. the Time of Arrival method.

The following procedure is performed at every node i:

- detect one-hop neighbors (K_i)
- measure the distances to one-hop neighbors (D_i)
- send the K_i and D_i to all one-hop neighbors

Thus, every node knows its two-hop neighbors and some of the distances between its one-hop and two-hop neighbors. A number of distances cannot be obtained due to the power range limitations or the obstacles between the nodes. Fig. 2 shows node i and its one-hop neighbors. Continuous lines represent the known distances between the nodes, while dashed lines represent the distances that cannot be obtained.

By choosing two nodes $p, q \in K_i$ such that the distance between p and q (d_{pq}) is known and larger than zero and such that the nodes i, p and q do not lie on the same line, node i defines its Local Coordinate System. The latter is defined such that node p lies on the positive x axis of the coordinate system and node q has a positive q_y component. In this way, the Local Coordinate System of i is uniquely defined as a function of i, p and q.
Thus, the coordinates of the nodes i, p and q are:

$$\begin{aligned} &i_x = 0; \;\; i_y = 0; \\ &p_x = d_{ip}; \;\; p_y = 0; \\ &q_x = d_{iq}\cos\gamma; \;\; q_y = d_{iq}\sin\gamma, \end{aligned} \tag{1}$$

where γ is the angle $\angle(p, i, q)$ and it is obtained by using a cosines rule for triangles:

$$\gamma = \arccos \frac{d_{iq}^2 + d_{ip}^2 - d_{pq}^2}{2 d_{iq} d_{ip}} \tag{2}$$

The positions of the nodes $j \in K_i$, $j \neq p, q$, for which the distances $d_{ij}, d_{qj} and d_{pj}$ are known, are

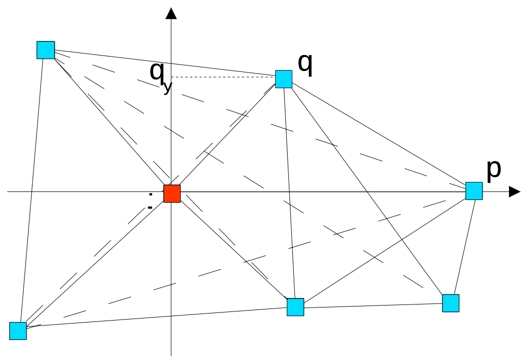


Figure 2. The coordinate system of node i is defined by choosing nodes p and q.

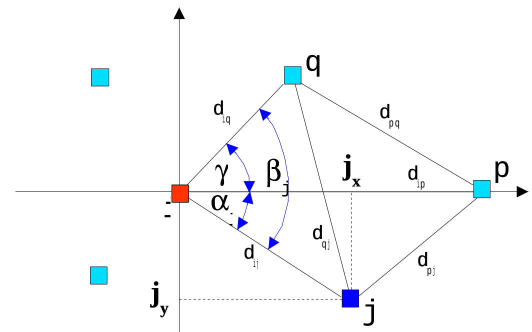


Figure 3. An example illustrating the way to obtain the position of node j in the coordinate system of node i.

computed by triangulisation. Therefore, we obtain

$$\begin{array}{l} j_x = d_{ij} \cos \alpha_j \\ if \quad \beta_j = |\alpha_j - \gamma| \;\; => \;\; j_y = d_{ij} \sin \alpha_j \\ else \;\; => \;\; j_y = -d_{ij} \sin \alpha_j, \end{array} \tag{3}$$

where α_j is the angle $\angle(p,i,j)$, and β_j is the angle $\angle(j,i,q)$. In practice, β_j will never be exactly equal to $|\alpha_j - \gamma|$ due to the errors in distance measurements. The purpose of this exercise is to find on witch side of the x axis node j is located and some difference between these two values needs to be tolerated.

We obtain the values of α_j and β_j by using the cosine rule

$$\alpha_j = \arccos \frac{d_{ij}^2 + d_{ip}^2 - d_{pj}^2}{2 d_{ij} d_{ip}} \tag{4}$$

$$\beta_j = \arccos \frac{d_{iq}^2 + d_{ij}^2 - d_{qj}^2}{2 d_{iq} d_{ij}} \tag{5}$$

The angles α_j, β_j and γ are placed within the triangles (p,i,j), (j,i,q) and (p,i,q), respectively and thus we observe just their absolute values and not their directions.

Fig. 3 shows the example of this computation for node j. The positions of the nodes $k \in K_i$, $k \neq p, q$, which are not the neighbors of nodes p and q, can be computed by using the positions of the node i and at least two other nodes for which the positions are already obtained, if the distance from the node k to these nodes is known.

Limited power ranges of the nodes reduce the number of one-hop neighbors for which the node i is able to compute the position. We define the *Local View Set (LVS)* for node i as the set of nodes $LVS_i(p,q) \subseteq K_i$ such that $\forall j \in LVS_i$, node i can compute the location of the node j, in the Local Coordinate System of node i. Out of $|K_i|$ neighbors, node i can choose $2\binom{|K_i|}{2}$ different couples of ps and qs. We denote the set of all possible combinations of p and q for node i as a set C_i.

$$\begin{array}{l} C_i = \{(p,q) \in K_i \;\; such \;\; that \;\; p \in K_q\} \\ 0 \leq |C_i| \leq 2\binom{|K_i|}{2} \end{array} \tag{6}$$

By choosing different ps and qs for the same node i, we obtain $|C_i|$ different Local View Sets, where $|C_i|$ is the cardinality of the set C_i. The choice of p and q should maximize the number of the nodes for which we can compute the position:

$$(p,q) = arg \max_{(p_k,q_k) \in C_i} |LVS_i(p_k,q_k)| \tag{7}$$

4. Network Coordinate System

After the nodes build their Local Coordinate Systems, their positions are set to (0,0) and their coordinate systems have different directions. We say that two coordinate systems have the *same direction* if the directions of their x and y axes are the same. In this section we describe how to adjust the directions of the Local Coordinate Systems of the nodes to obtain the same direction for all the nodes in the network. We call this direction, *the direction of the Network Coordinate System*. We further explain the algorithm for electing the center of the Network Coordinate System. Finally, we show the way to compute the positions of the nodes in the Network Coordinate System.

4.1. Coordinate system direction

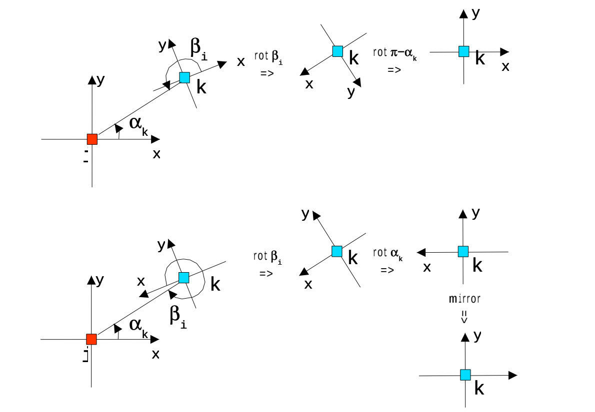
We observe two nodes, i and k. To adjust the direction of the coordinate system of the node k to have the same direction as the coordinate system of the node i, node k has to rotate and possibly mirror its coordinate system. We denote this rotation angle as the *correction angle* for the node k. To perform the angle correction operation, two conditions have to be met

- $i \in LVS_k$ and $k \in LVS_i$
- $\exists j \neq i, k$ such that $j \in LVS_k$ and $j \in LVS_i$

There are two possible situations. In the first situation, the directions of the coordinate systems of i and k are such that to have the coordinate system of k equally directed as the coordinate system of i, the coordinate system of k needs to be rotated by some rotation angle. In the second situation, the rotation of the coordinate system of k is not enough to have the same direction of the coordinate systems; in addition, the coordinate system of k needs to be mirrored around one of its axes after the rotation. These two situations are illustrated in Fig. 4.

The correction angle for node k in the first situation is $\beta_i - \alpha_k + \pi$. In the second scenario, the correction angle for node k is $\beta_i + \alpha_k$, and the mirroring is done with respect to the y axes. Here, α_k is the angle of the vector $\vec{ik}$ in the coordinate system of the node i and β_i is the angle of the vector $\vec{ki}$ in the coordinate system of the node k. All the rotations at node k are in the positive direction of its Local Coordinate System.

Before the correction of the direction of its coordinate system, node k uses the following procedure to detect the situation that it is in. Node j is used



(a) (b)

Figure 4. An example illustrating two possible situations of correction of the coordinate system of node k.

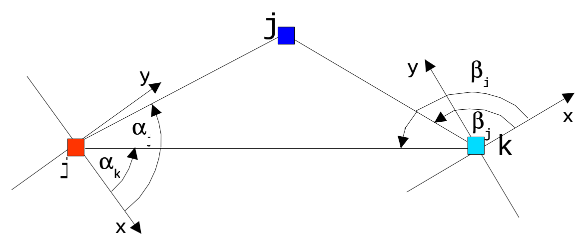


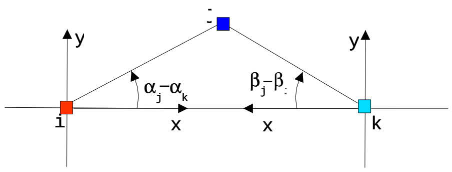
Figure 5. Position of the node j in the Local Coordinate Systems of i and k.

for this detection.

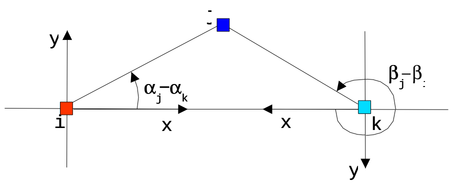
if $\alpha_j - \alpha_k < \pi$ *and* $\beta_j - \beta_i > \pi$
or $\alpha_j - \alpha_k > \pi$ *and* $\beta_j - \beta_i < \pi$
=> mirroring is not necessary
=> the correction angle is $\beta_i - \alpha_k + \pi$

if $\alpha_j - \alpha_k < \pi$ *and* $\beta_j - \beta_i < \pi$
or $\alpha_j - \alpha_k > \pi$ *and* $\beta_j - \beta_i > \pi$
=> mirroring is necessary
=> the correction angle is $\beta_i + \alpha_k$

This procedure is explained as follows. We observe the position of the node j in the coordinate systems of the nodes i and k. The angle of the vector $\vec{ij}$ in the coordinate system of i is α_j and the angle of the vector $\vec{kj}$ in the coordinate system of k is β_j. This is illustrated in Fig. 5. If the coordinate



(a)



(b)

Figure 6. The position of the node j in the Local Coordinate Systems of i and k gives to node k the information whether its coordinate systems needs to be rotated (a) or rotated and mirrored (b).

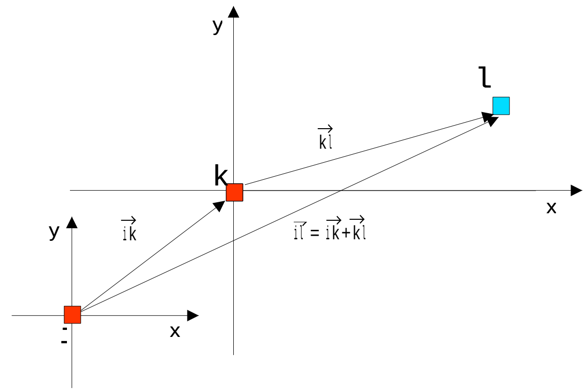


Figure 7. Position computing when the Local Coordinate Systems have the same direction.

systems of nodes i and k are rotated, by α_k and β_i respectively, the angles of the vectors $\vec{ij}$ and $\vec{kj}$ change to $\alpha_j - \alpha_k$ and $\beta_j - \beta_i$. We observe that the position of the node j makes it possible to detect whether the mirroring is necessary or not. This is illustrated in Fig. 6.

Once node k has rotated its local coordinate system by the correction angle and has mirrored it if necessary, nodes i and k have the same direction of their Local Coordinate Systems. The same procedure can be repeated for all the nodes in the network, in an appropriate order.

4.2. Position computing

We showed that the nodes can compute their local coordinate systems, and are able to adjust their coordinate systems with respect to their neighbors. Our goal is that all the nodes in the network compute their position in the Network Coordinate System. So far we have not defined the Network Coordinate System. For now, we will choose the Network Coordinate System as the Local Coordinate System of one of the nodes in the network (i.e. node i). If the Network Coordinate System is chosen in this way, all the nodes in the network have to adjust the directions of their coordinate systems to the direction of the coordinate system of node i and every node has to compute its position in the coordinate system of the node i. In this section we explain how nodes can compute their positions in the coordinate system of node i. All the nodes that belong to the local view set of the node i know their positions, as it is computed directly by node i. Therefore, node k knows its position in the coordinate system of node i. We now observe node l, which is a two-hop neighbor of the node i and belongs to the local view set of node k. Node i knows its position in the coordinate system of node k, and knows the position of node k in the coordinate system of node i. As the coordinate systems of nodes k and i have the same directions, the position of the node l in the coordinate system of the node i is simply obtained as a sum of two vectors.

$$\vec{il} = \vec{ik} + \vec{kl} \tag{8}$$

This is illustrated in Fig. 7.

The same is applied to the 3-hop neighbors of node i that belong to the local view set of node l, if the coordinate system of l has the same direction as the coordinate systems of i and k. These nodes will receive the position of node l in the coordinate system of node i and add this vector to their vector in the coordinate system of node l. In this way, the nodes obtain their position in the coordinate system of node i. The procedure is repeated for all the nodes in the network, and all the nodes in the network will compute their positions in the coordinate system of node i.

The nodes that are not able to build their local coordinate system, but communicate with three nodes that already computed their positions in the referent coordinate system, can obtain their position in the Network Coordinate System by triangulisation.

4.3. Location Reference Group

As described in the previous section, the Local Coordinate System of node i becomes the Network Coordinate System and all the nodes adjust the directions of their coordinate systems to the direction of the coordinate system of node i and compute their position in the coordinate system of node i. However, the motion of node i will cause that all the nodes have to recompute their positions in the Network Coordinate System. This will cause a large inconsistency between the real and computed positions of the nodes. This approach can be used in small area networks where the nodes have low mobility and where disconnection of the nodes is not expected. A more stable approach, albeit expensive in terms of exchanged messages is to compute the center of the coordinate system as a function of the positions of all the nodes in the network. In this case, the Network Coordinate System center would be the geometrical center of the network topology and the direction of the coordinate system would be the mean value of the directions of the Local Coordinate Systems of the nodes. The algorithms proposed hereafter can also be used in the networks where some fixed nodes are introduced and then used to stabilize the Network Coordinate System. This would ease the assumptions on mobility of all the nodes in the network. However, in the sequel we assume that all nodes are potentially mobile.

We propose the following approach. We define a set of nodes called the *Location Reference Group* $LRG \subseteq N$ chosen to be stable and less likely to disappear from the network. For example, we choose it such that its density of nodes is the highest in the network. The network center is not a particular node, but a relative position dependent on the topology of the Location Reference Group. Within the Location Reference Group, a broadcast is used to obtain its own topology. When the nodes are moving, the LRG center is recomputed accordingly. We expect the average speed of the LRG center to be much smaller than the average speed of the nodes. In this way, we stabilize the center of the network and reduce the inconsistency. The direction of the Network Coordinate System is computed as the mean value of the directions of the Local Coordinate Systems of the nodes in the LRG. The larger the LRG, the more stable it is, but the more difficult it becomes to maintain and the more costly to compute the center and the direction of the Network Coordinate System.

4.3.1. Location Reference Group initialization

We want to identify the nodes belonging to the LRG. For this purpose, every node of N performs the following operations:

- broadcast the hello packet to its n-hop neighborhood to obtain the node IDs, their mutual distances and the directions of their coordinate systems
- compute the positions of the n-hop neighbors in its Local Coordinate System
- compute the n-hop neighborhood center as:

$$c_x = \frac{\Sigma j_x}{m} \qquad c_y = \frac{\Sigma j_y}{m}, \tag{9}$$

  where m is the number of nodes in the n-hop neighborhood and j_x and j_y are the x and y coordinates of the nodes, respectively.
- compute the n-hop neighborhood direction as the average of the Local Coordinate System directions of the nodes that belong to its n-hop neighborhood, and for which it can obtain the positions

- compute the *density factor* as a function of the number of nodes and the distances to the nodes in its n-hop neighborhood

Once the node has computed these parameters, it broadcasts the density factor, the information about the center and the direction of the n-hop neighborhood to its neighbors. The nodes with the lower density factor will be slaved by the nodes with a higher density factor and will adjust the directions of their coordinate systems accordingly. The nodes in the network will then compute their positions in the coordinate system of the n-hop neighborhood of the node with the highest density factor. The node with the highest density factor in the network is called the *initial location reference group master* and the nodes in its n-hop neighborhood, for which it can obtain their positions are called the *initial location reference group*. The nodes belonging to the Location Reference Group maintain the list of nodes in the group. The size of the LRG can be modified according to the expected network size.

4.3.2. Location Reference Group maintenance

Because of the mobility of the nodes, the LRG members will change their position and then the center of the LRG will change. To update this change regularly, we introduce the following algorithm performed by the members of the LRG:

- broadcast the hello packet to its n-hop neighborhood to obtain the node IDs, their mutual distances and the directions of their coordinate systems
- compare the n-hop neighbors list with the list of the LRG members.

The node that is a n-hop neighbor of the LRG master and has the highest number of the LRG nodes in its n-hop neighborhood is elected to be the new LRG master and its n-hop neighbors, for which it can obtain the position information become the new Location Reference Group. The LRG maintenance procedure is repeated periodically every fixed time period.

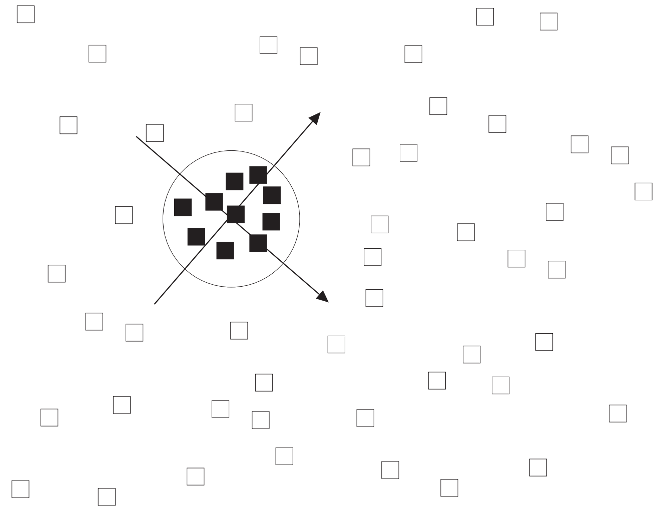


Figure 8. The Location Reference Group.

Every node that was in the LRG and no longer has the LRG master in its n-hop neighborhood, starts an initialization timer. If within a certain time the node does not receive the new position information issued by the LRG master, it starts the initialization procedure described above. This procedure can be initialized by any node, if after some fixed time period, the node does not receive the information about the new Location Reference Group master. Using the LRG maintenance algorithm, the network center moves at a much smaller speed than the nodes in the network. In this way, the inconsistency due to the movement of the center is reduced. Fig. 8 shows the example of the 1-hop LRG.

4.3.3. Network Coordinate System direction

We have shown how the nodes build their Local Coordinate Systems and how the Network Coordinate System is built. Furthermore, we have shown the means of stabilizing the center of the coordinate system. In this section we show how to stabilize the direction of the Network Coordinate System.

The choice of the nodes p and q and thus the directions of the Local Coordinate Systems is random. This makes the direction of the Network Coordinate System random, as it depends only on the directions of Local Coordinate Systems of the nodes in the Location Reference Group. We propose the following algorithm to stabilize the direction of the Network Coordinate System, which is performed at each node that belongs to the Location Reference Group:

- The node initially chooses the direction of its coordinate system, by choosing its (p, q) pair. We denote this coordinate system as C_1.
- When rerunning the Local Coordinate System algorithm, the node chooses the new (p, q) pair. We denote this coordinate system as C_2. The positions of the nodes changed due to their motion, and the choice of p and q may change.
- The node compares the positions of the nodes in two coordinate systems and searches for the maximum set of nodes (at least 3) that have the same topology in both C_1 and C_2. From this, we conclude that the nodes belonging to this set did not move during the time between two runs of the algorithm. This conclusion is not certain, but it has a very high probability of being true.
- The node uses this set of nodes and their distances to reconstruct the center of C_1 in the coordinate system C_2. This allows the node to adjust the direction of C_2 to the direction of C_1. If the node cannot reconstruct the coordinate system C_1, then it keeps the direction of C_2 as the direction of its new local coordinate system.

This algorithm allows every node that belongs to the LRG to introduce direction stability in its Local Coordinate System. The LRG master computes the direction of the Network Coordinate System as the average direction of the nodes in the LRG. Therefore, this algorithm stabilizes the direction of the network coordinate system. In a high density area, such as the LRG, we expect to have a low mobility set that will enable this algorithm to be used. The example of the coordinate system reconstruction is shown in Fig. 9.

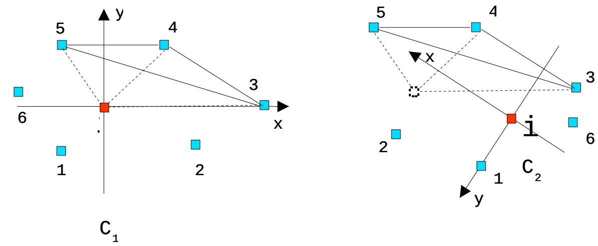


Figure 9. An example illustrating the reconstruction of the coordinate system C_1 in the coordinate system C_2.

4.4. Local View Set connectivity

If the node cannot exchange the information about its coordinate system with other nodes, it cannot compute its position in the Network Coordinate System. We say that the nodes are connected in terms of a coordinate system connectivity, if the information about the Network Coordinate System can be propagated to all the nodes in N. We refer to the coordinate system connectivity as the *Local View Set connectivity.* We observe two graphs:

- The connectivity graph $G(N, E)$ is the graph with set of vertices N and edges E where $\forall i, j \in N$ such that $e_{ij} \in E$ nodes i and j can communicate directly (in one hop).
- The Network Coordinate System graph $G_{NCS}(N, E_{NCS})$ is the graph with set of vertices N and edges E_{NCS} where $\forall i, j \in N$ such that $e_{ij} \in E_{NCS}$, node $i \in LVS_j$, node $j \in LVS_i$ and $\exists m \in N$ such that $m \in LVS_i \cap LVS_j$. The connectivity graph therefore shows between which pairs of nodes the adjustment of the Local Coordinate System can be made and thus the Network Coordinate System propagated.

The *edge connectivity* $\lambda(G)$ is the minimum number of edges whose deletion from graph G disconnects G [10]. Not all neighbors are able to propagate the information about the Network Coordinate System and therefore $LVS_i \subseteq K_i$, $\forall i \in N$. For this reason the edge connectivity of G_{NCS} will

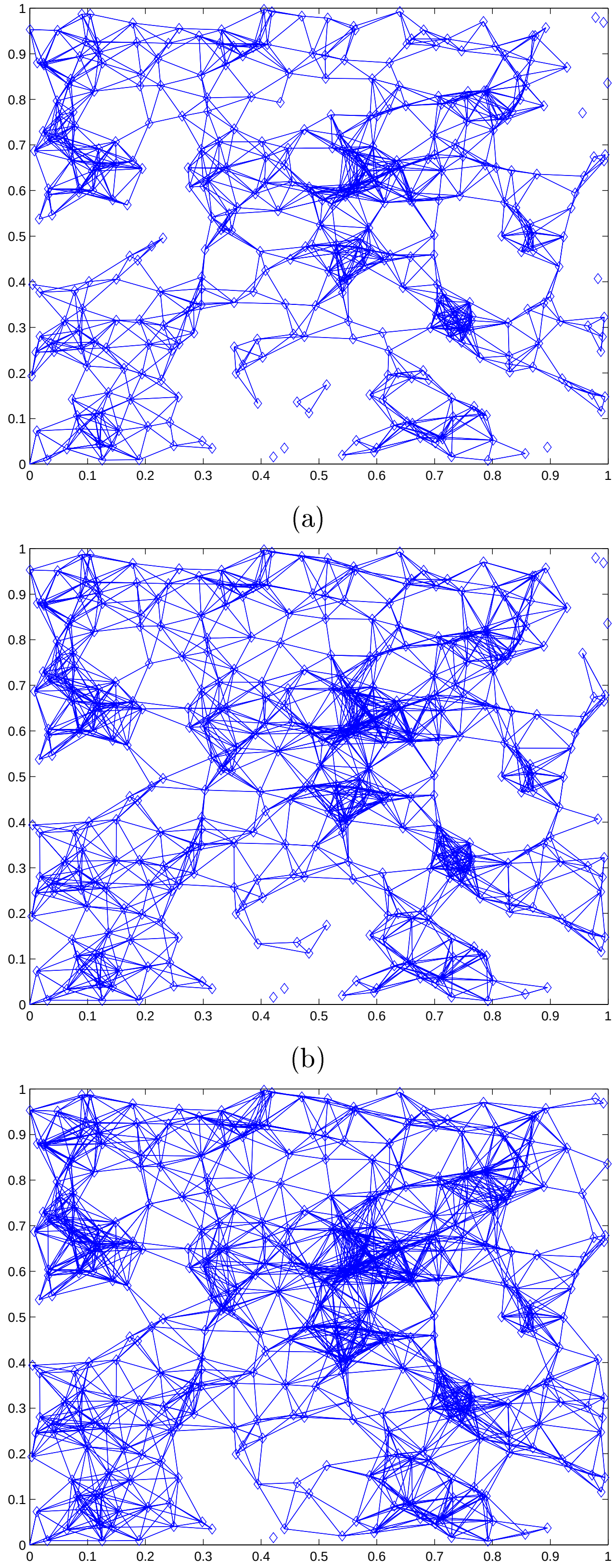


Figure 10. An example of node connectivity for different power ranges (a) 90 m, (b) 100 m, (c) 110 m. The node density is 400 nodes/km^2.

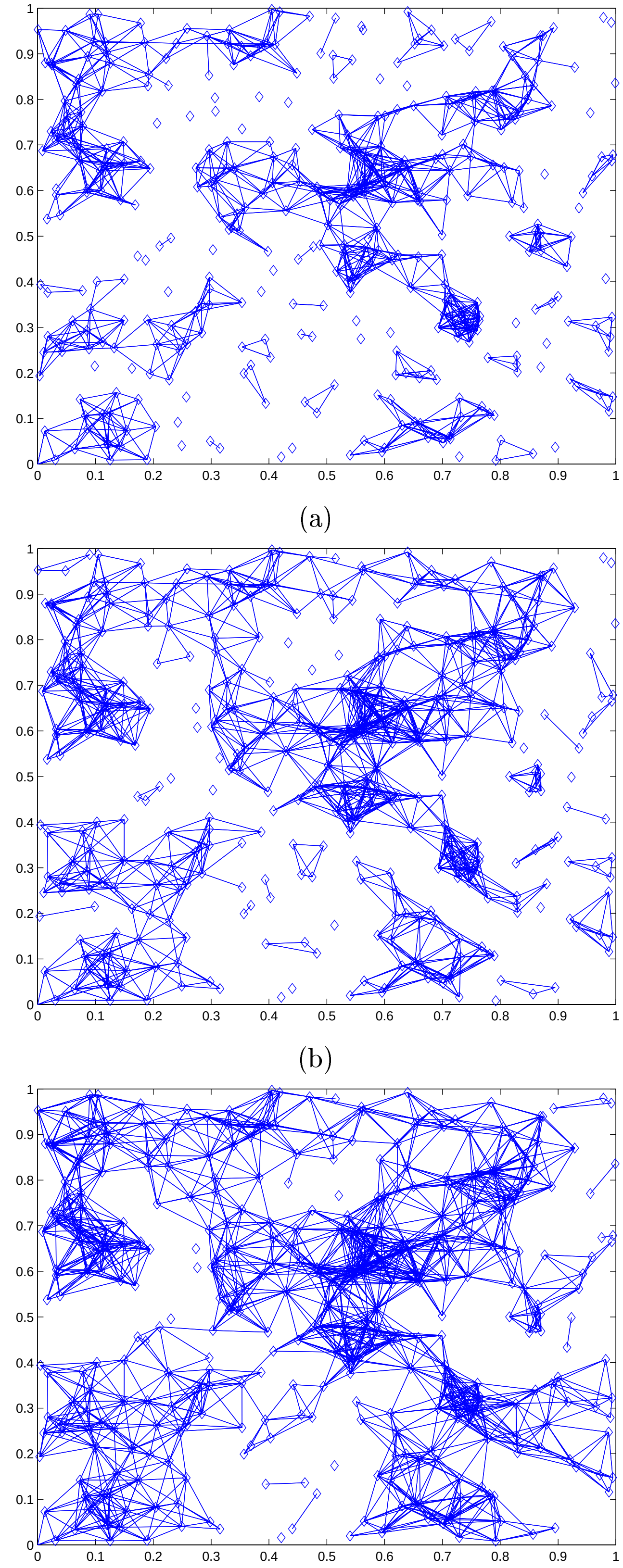


Figure 11. An example of Local View Set connectivity for different power ranges (a) 90 m, (b) 100 m, (c) 110 m. The node density is 400 nodes/km^2.

always be less or equal than the edge connectivity of G.

$$\lambda(G_{NCS}) \leq \lambda(G) \tag{10}$$

This relation shows that in order to achieve a high LVS connectivity, we need to have a high node connectivity. The connectivity of both graphs depends on the power ranges of the nodes and on the density of nodes in the region. The high connectivity ensures that when the nodes are moving, the nodes will remain LVS connected. Fig. 10 and 11 show examples of node and Local View Set connectivity on the same topology of the nodes. (400 nodes uniformly distributed over a 1 km^2 plane).

5. Location estimation error

The algorithms described in the previous sections use the ranges between nodes to build a global coordinate system. Therefore, the accuracy of the range measurements will influence position accuracy. In radio-location methods for cellular systems, two methods are provided that can be used for distance measurements: Time of Arrival and Signal Strength measurements. These measurements are corrupted by two types of errors: measuring errors and Non-Line of Sight (NLOS) errors. Several models have been proposed to model both measuring error [13] and NLOS error [11] [12]. To the best of our knowledge, no measurements have been published to give the overall distribution of the range error in mobile ad hoc networks. As a first approximation we will assume that the errors are similar to the ones encountered in cellular systems.

5.1. NLOS mitigation

In [4], the mitigation was performed in the infrastructure-based environment, where the positions of the base stations are known. We believe that error mitigation is still possible in an ad hoc mobile environment, where there are no base stations to rely on. We observe the location accuracy within the Local Coordinate System. The node location model is formulated as an estimation model. To estimate the position of the node, the following algorithm is used:

- the positions of the nodes in the Local Coordinate System are computed without using the observed node
- the position of the observed node is estimated using the positions of at least three of its neighbors
- the residual weighting algorithm is applied to mitigate the error.

The detailed description of the residual weighing algorithm can be found in [4].

The analogy between the error mitigation in cellular systems, and the error mitigation in ad hoc networks exists because, in both cases, the range measurements are used to obtain the positions of the nodes and if the number of distances is larger than the minimum required, the error can be mitigated. In cellular systems, we expect a smaller number of range measurements than in ad hoc networks because the mobile station is usually covered by a relatively small number of base stations; whereas in ad hoc networks, the average number of neighbors can be higher. However, in cellular systems, base stations have fixed positions, and their mutual distances do not introduce any error.

5.2. Error propagation

In this section we observe the influence of error cumulation on the node position estimation accuracy. We observe how the position estimation error increases with the distance of the node from the Network Coordinate System center. If the node is n-hop distant from the center of the coordinate system, its position estimation will be the sum of the position vectors along the path from the center to the observed node. Therefore, the position estimation error will be a sum of all error vectors along the same path. We assume that the distribution of the direction of the error vectors is uniform, as the range estimation errors can produce the error vector with the same probability in every direction. We further assume that the length of the error vector and the direction of the error vector are mutually independent. The power ranges of the nodes are assumed to be the same for all the nodes and thus we expect the lengths of the error vectors to be equally distributed. Therefore, the expected value of the total error vector is

$$E(\vec{X}) = E(\vec{x}_1)+,...,+E(\vec{x}_{n-1})$$
$$= (n-1)E(arg(\vec{x}_1))E(cos\alpha_1 + isin\alpha_1) \quad (11)$$
$$= 0,$$

where $\vec{x}_1, ..., \vec{x}_{n-1}$ are the error vectors in the Local Coordinate Systems used to compute the position of the observed node, and α_1 is the direction of the error vector of the one hop neighbor of the Network Coordinate System center.

This shows that the expected value of the error vector in the node that is n-hops distant from the center will be zero. Nevertheless, the standard deviation of the error vector is expected to increase when the node is more remote from the center of the network. This points to a limitation in the scalability of the algorithm.

6. Simulation results

In this section we present the simulation results and we show the performance of the algorithm. The results are divided into two parts. In the first part we show the influence of the power range on node and LVS connectivity. In the second part we present the results that illustrate the motion of the center and the changes in the direction of the Network Coordinate System due to the mobility of the nodes.

The system model is the following. We model the positions of the nodes according to the Poissonian distribution: When a set of nodes is generated (400 nodes in our simulation), the points are distributed from a center point on the plane, the distances between the nodes are distributed according to the exponential distribution, and the angle is distributed uniformly. The motion of the nodes is random. The nodes choose randomly a point on the plane, and the speed required to arrive to that point. The maximum and the minimum traveling speed is defined. When the nodes arrive at the chosen point, they wait for a fixed time, and then another random pair (speed, point) is chosen. We assume that all the nodes have the same power range. The performance of the algorithm is observed with respect to the power range.

6.1. Local View Set connectivity

In this section we present the results regarding the connectivity of the nodes and the LVS connectivity. Fig. 12 shows that the average number of the nodes for which the position can be obtained, $|LVS|$, is lower than the average number of neighbors $|K|$. Fig. 12 shows that as the power range increases, the difference between the neighbor set K and the set LVS becomes smaller. This convergence is due to the increasing node connectivity and LVS connectivity as the power range increases. The edge connectivity with respect to the power range is shown in Fig. 13. This illustrates that the LVS connectivity larger than zero will be achieved at 110 m power range, while the node connectivity is larger than zero already at 90 m power range. Therefore, the positions will be computed in the Network Coordinate System for all the nodes if the nodes have 110 m power range.

6.2. Center and direction stability

In this section we illustrate the movement of the center and the change in the direction of the Network Coordinate System. Fig. 14 shows that if we choose a larger (3-hop) neighborhood instead of a 2-hop neighborhood, the mobility of the center of the network decreases accordingly. Fig. 15 illustrates the influence of the average node speed increase on the changes of the network coordinate system direction.

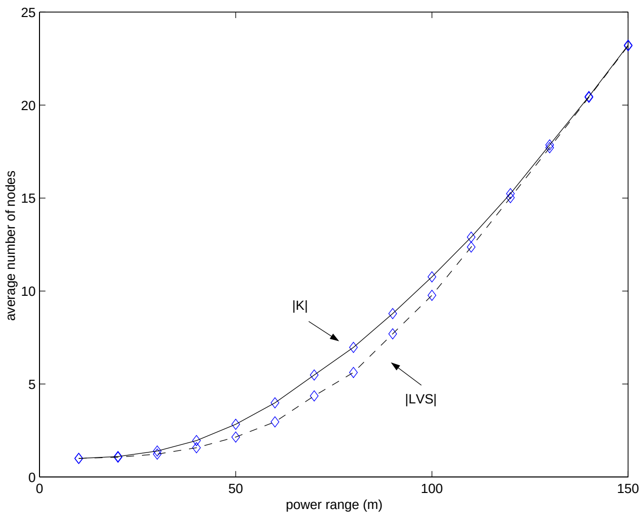


Figure 12. Average number of neighbors $|K|$ (solid line) and average number of the nodes for which the positions can be obtained $|LVS|$ (dashed line).

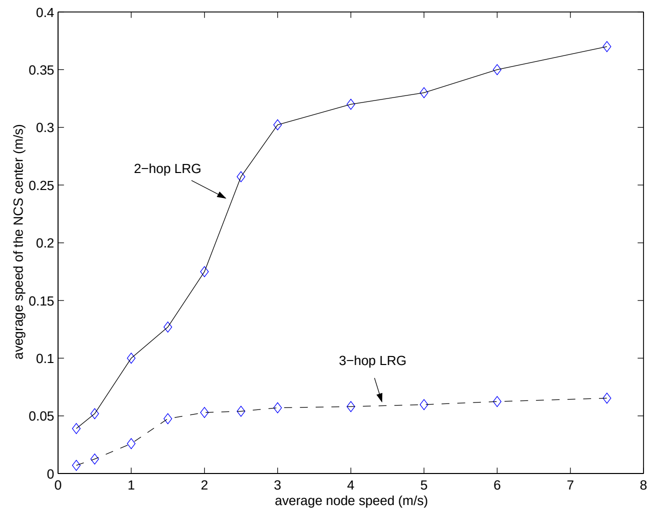


Figure 14. Speed of the Network Coordinate System center for 2-hop and 3-hop LRG.

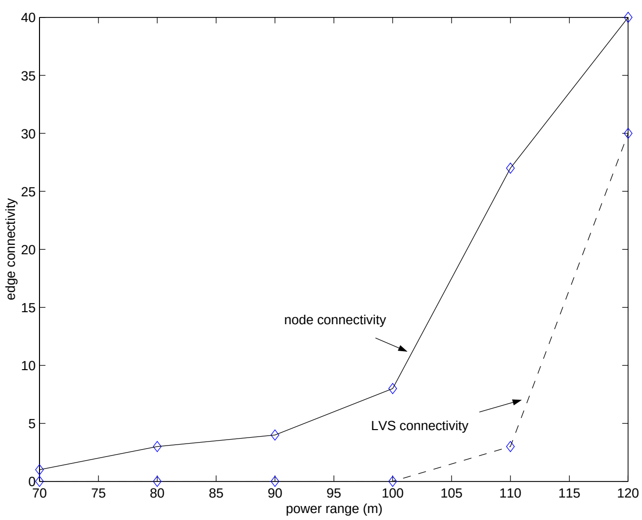


Figure 13. Node and LVS connectivity.

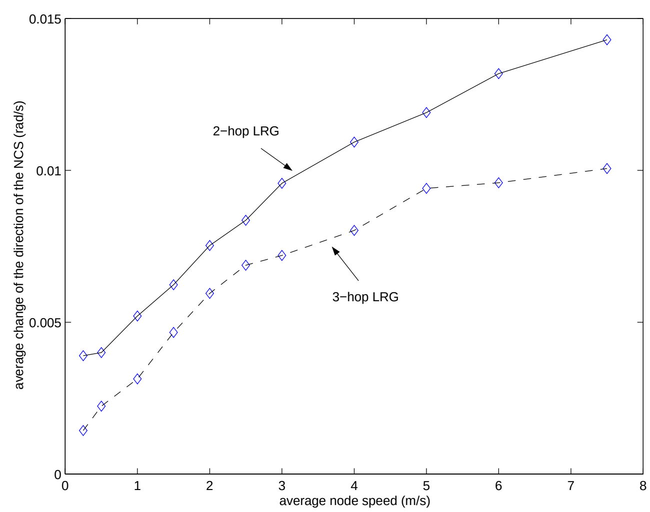


Figure 15. Comparation of speed of change of the direction of the Network Coordinate System center for 2-hop and 3-hop Location Reference Groups.

6.3. Communication cost

In this section we observe the algorithm in terms of number of messages that need to be exchanged between the nodes to keep the algorithm running. Here it is important to remember that the nodes are using omnidirectional antennae and that we consider each broadcast message as one message sent to all the neighbors and not as k messages sent to k neighbors.

The average number of messages that needs to be sent per node in order to build a Local Coordinate Systems is in the order of k, where k is the average number of one-hop neighbors in the network. k messages are exchanged to measure the distances between the node and its neighbors. One additional message is used by each node to broadcast this information to its one-hop neighbors. This makes in total $k+1$ messages sent per node to build Local Coordinate Systems.

After building their Local Coordinate Systems, the nodes compute their density factors (as a function of the number of nodes and the distances to nodes in the n-hop neighborhood). The number of messages that need to be sent per node to compute the density factor depends on the size of the region (neighborhood) for which the density factor is computed. The average number of messages sent per node to compute the density factor is in the order of k^{n-1}, where k is the average number of one-hop neighbors in the network and n is the neighborhood size for which the node is computing the density factor (n-hop neighborhood).

The final step of the algorithm supposes a broadcast to all the nodes in the network. This operation is very costly in terms of number of messages. The number of messages sent per node is in the order of k^{l-1}, where l is the average number of hops needed to reach the border of the graph (averaged over all the nodes in the network).

Building a Local Coordinate System and computing the density factor are not very costly operations, but they are performed at each node frequently, whereas building the Network coordinate system is a very costly operation. How often the Network Coordinate System has to be rebuild depends on the algorithms for maintaining the center and the direction of the NCS, and on the algorithms within the nodes which stabilize the LCS's of the nodes.

7. Conclusions

We have proposed algorithms that:

- It is possible to achieve a relative coordinate system by self-organization of the nodes.
- The power range should be chosen carefully to ensure high LVS connectivity and high algorithm coverage.
- The algorithm imposes low requirements on the node connectivity.
- The angle and the center of the Network Coordinate System can be stabilized using simple heuristics.
- The distance measurement errors will introduce an error in position estimation.
- Despite the distance measurement errors and the motion of the nodes, the algorithm provides sufficient location information and accuracy to support basic network functions.

Several issues need to be addressed when implementing the algorithm. First, the power range must be large enough to ensure LVS connectivity (simple node connectivity does not guarantee that the positions of all the nodes will be computed). Second, the size of the Location Reference Group must be chosen such that it increases the stability of the center and the direction of the Network Coordinate System. This will reduce the inconsistency between the computed and the real position of the center.

One major characteristic of this approach is that the nodes do not know the physical direction of the coordinate system. The nodes know where their neighbors are placed in the coordinate system, but they have no way to associate the Network Coordinate System with the geographic coordinate system. This is only possible if the algorithm is used along with some GPS-capable devices. However, the algorithm can be used without the use of GPS for geodesic packet forwarding and location dependent routing.

The presented algorithm provides position information to the nodes, based only on the local view of each node and using the local processing capabilities of the nodes. We showed that it is possible to build a coordinate system without centralized knowledge of the network topology.

Future work in GPS-free positioning includes the improvement of the accuracy of the range measurements and therefore to reduce the position error. Additionally, to improve the center and direction stability heuristics and to optimize the algorithm and to extend it for three dimensional models. Finally we envision testing the algorithm and its performance in real world applications.

Acknowledgments

We thank Martin Vetterli, Raymond Knopp, John Farserotu and Milan Vojnovic for their valuable suggestions and the stimulating discussions.

References

[1] J.-P. Hubaux, J.-Y. Le Boudec, S. Giordano, M. Hamdi, Lj. Blazevic, L. Buttyan and M. Vojnovic *Towards Mobile Ad-Hoc WANs: Terminodes*, IEEE WCNC, September 2000.

[2] J.J. Caffery and G.L. Stuber, *Overview of Radiolocation CDMA Cellular Systems*, IEEE Communications Magazine, April 1998.

[3] M.I. Silventoinen and T. Rantalainen, *Mobile Station Emergency Locating in GSM*, Personal Wireless Communications, IEEE International Conference on 1996.

[4] P.-C. Chen, *A non-line-of-sight error mitigation algorithm in location estimation*, IEEE Wireless Communications and Networking Conference, 1999.

[5] M.P. Wylie and J. Holtzman, *The non-line of sight problem in mobile location estimation*, 5th IEEE International Conference on Universal Personal Communications, 1996.

[6] C. Rose and R. Yates, *Location Uncertainty in Mobile Networks: a theoretical framework*, D. of Electrical engineering WINLAB, November 1996.

[7] Lj. Blazevic, S. Giordano and J. Y. Le Boudec, *Self-Organizing Wide-Area routing*, SCI 2000/ISAS 2000, Orlando, July 2000.

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

[9] FCC RM-8143. *Revision of the commission rules to ensure compatibility with enhanced 911 emergency calling system*, RM 8143, FCC, October 1994.

[10] H. Whitney, *Congruent Graphs and the Connectivity of Graphs*, Amer. J. Math. 54, 1932.

[11] R. Steele, *Mobile Radio Communications*, Pentech-Press, 1992.

[12] W. C.Y. Lee, *Mobile Communication Engineering*, McGraw-Hall, 1993.

[13] W. Zhang, J. Luo and N. Mandayam, *Mobile location estimation using time of arrivals in cdma systems*, Technical report 166, WINLAB, Rutgers University, 1998.

[14] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward and Paul Webster, *The Anatomy of a Context-Aware Application*, Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MOBICOM'99, Seattle, Washington, USA, August 1999, pp. 59-68.

[15] P. Bahl and V.N. Padmanabhan, *RADAR: an in-building RF-based user location and tracking system*, INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , Volume: 2 , 2000.

[16] E. Kranakis, H. Singh and J. Urrutia, *Compass routing in geometric networks*, Proc. Canadian Conference on Computational Geometry, 1999.

[17] I. Stojmenovic and X. Lin, *A loop-free routing for wireless networks*, Proc. IASTED Conf. On parallel and Distributed systems, 1998.

[18] P.Bose, P. Morin, I. Stojmenovic and J. Urrutia, *Routing with guaranteed delivery in ad hoc wireless networks*, Proc. of 3rd ACM Int. Workshop on Discrete Algorithms and methods for monile computing and Communications DIAL M99, Seattle, August 20, 1999, 48-55.

[19] *www.terminodes.org*,

[20] J.-P. Hubaux, T. Gross, J.-Y. Boudec and M. Vetterli *Towards self-organized mobile ad hoc networks: the Terminode project*, IEEE Communications Magazine, January 2001