

GPS-free positioning in mobile Ad-Hoc networks

Srdan Čapkun, Maher Hamdi, Jean-Pierre Hubaux

EPFL - Communication Systems Department (DSC)

Ecole Polytechnique Federale de Lausanne (EPFL)

CH-1015 Lausanne, Switzerland

{srdan.capkun, maher.hamdi, jean-pierre.hubaux}@epfl.ch

Abstract

In this paper we consider the problem of node positioning in ad-hoc networks. We propose a distributed, infrastructure-free positioning algorithm that does not rely on Global Positioning System (GPS). The algorithm uses the distances between the nodes to build a relative coordinate system in which the node positions are computed in two dimensions. 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.

1 Introduction

The presented work is a part of the Terminode project, a 10 year ongoing project at Swiss Federal Institute of Technology (EPFL). The project investigates large area, wireless, mobile networks referred to as mobile ad-hoc wide area networks [1]. 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 propose an algorithm for GPS-free positioning of the nodes in an ad-hoc network. Our goal is to show that in the scenarios where an infrastructure does not exist and GPS cannot be used, there is a way to obtain positions of the nodes by distributed processing. GPS-free positioning is desirable, notably when the GPS signal is too weak (e.g., indoor), or when it is jammed, or if for cost or integration reasons the inclusion of a GPS receiver has to be avoided.

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. The Time of Arrival (TOA) method is used to obtain the range between two mobile devices. We show that despite the range measurement errors, and the motion of the nodes, the algorithm provides enough stability and location accuracy to sustain basic network functions. The basic idea of the algorithm is illustrated in Fig. 1. The nodes in the ad-hoc networks are not usually aware of their geographical positions. As GPS is not used in our algorithm, the algorithm provides relative positions of the nodes in respect to the network topology.

The algorithm can be useful in the scenarios where the location information is used to support basic network functions. Two examples of this kind of application scenario are Location Aided Routing [8] and Geodesic Packet Forwarding [7]. In Geodesic Packet Forwarding algorithm, the source sends packets in the physical direction of the destination node. When the omni-directional antennas are used, Self-Positioning Algorithm provides enough information to every node to support Geodesic Packet Forwarding. Given that the node knows its position and the positions of the destination node in the relative coordinate system, it is able to compute in which direction (to which next hop node) to send packets.

This paper is organized as follows. Section 2 describes the related work in the field of radio-location techniques. 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 US FCC regulations for locating E911 callers, positioning services in mobile systems have

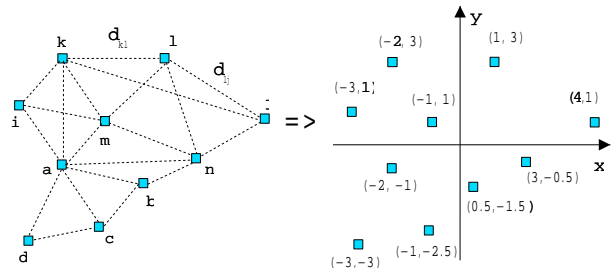


Figure 1: The algorithm uses the distances between the nodes and builds the relative coordinate system.

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 Weighting 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.

Recently, some position-based routing and packet forwarding algorithms for the ad-hoc networks have been proposed in [8, 7, 14, 15]. 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

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.

The node j is called a one-hop neighbor of node i if nodes i and j can communicate directly (in one hop). We define $\forall i \in N$, a set of nodes K_i such that $\forall j \in K_i, j \neq i, j$ is a one hop neighbor of i , where N is a set of all the nodes in the network. We call K_i the set of one-hop neighbors of node i . We define $\forall i \in N$ the set D_i as a set of distances measured from the node i to the nodes $j \in K_i$. The neighbors can be detected by using beacons. After the absence of 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 can not 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 .

The coordinates of the nodes i, p and q are:

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

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

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

The positions of the nodes $j \in K_i, j \neq p, q$, for which the distances d_{ij}, d_{qj}, d_{pj} are known, are computed by triangulation. Therefore, we obtain

$$\begin{aligned} j_x &= d_{ij} \cos \alpha_j \\ \text{if } \beta_j &= |\alpha_j - \gamma| \Rightarrow j_y = d_{ij} \sin \alpha_j \\ \text{else } &\Rightarrow j_y = -d_{ij} \sin \alpha_j, \end{aligned} \quad (3)$$

where, α_j is the angle $\angle(p, i, j)$ in the triangle (p, i, j) , β_j is the angle $\angle(j, i, q)$ in the triangle (j, i, q) and γ is the angle $\angle(p, i, q)$ in the triangle (p, i, q) . We obtain the values of α_j and β_j by using the cosine rule

$$\alpha = \arccos \frac{d_{ij}^2 + d_{ip}^2 - d_{pj}^2}{2d_{ij}d_{ip}} \quad (4)$$

$$\beta = \arccos \frac{d_{iq}^2 + d_{ij}^2 - d_{qj}^2}{2d_{iq}d_{ij}} \quad (5)$$

Fig. 3 shows the example of this computation for node j .

The positions of the nodes $k \in K_i, j \neq p, q$, 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 a *Local View Set (LVS)* for node i as a 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 maximally $2 \binom{|K_i|}{2}$ different pairs of ps and qs , where $|K_i|$ is the cardinality of the set K_i . We denote the set of all possible combinations of p and q for node i as a set C_i .

$$\begin{aligned} C_i &= \{(p, q) \in K_i \text{ such that } p \in K_q\} \\ 0 &\leq |C_i| \leq 2 \binom{|K_i|}{2} \end{aligned} \quad (6)$$

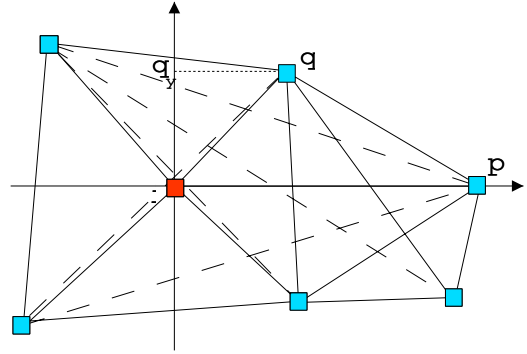


Figure 2: The coordinate system of node i is defined by choosing nodes p and q .

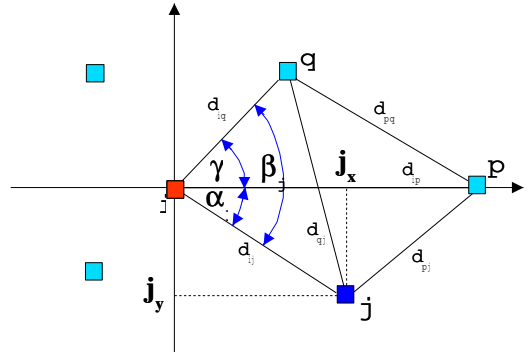


Figure 3: Example illustrates the way to obtain the position of node j in the coordinate system of node i .

By choosing different ps and qs for the same node i , we obtain maximally $|C_i|$ 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. Therefore, we have:

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

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 di-

rection 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 obtain the correction angle 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 $j \in LVS_i$

We differ two possible scenarios. In first scenario, 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 scenario the rotation of the coordinate system of k is not enough to have the same direction of the coordinate systems, but the coordinate system of k needs to be mirrored around one of its axes after the rotation. These two scenarios are illustrated in Fig. 4.

The correction angle for node k in the first scenario 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 in 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 done 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 scenario that it is in. Node j is used for this detection.

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

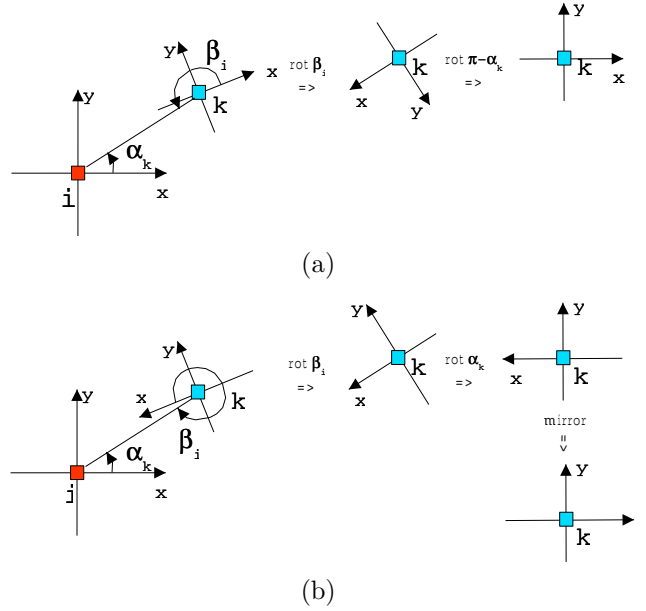


Figure 4: Example illustrating two possible scenarios of correction of the coordinate system of node k .

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 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 will detect if the mirroring is necessary or not. This is illustrated in Fig. 6.

Once that node k rotates its local coordinate system by correction angle and mirrors it if necessary, nodes i

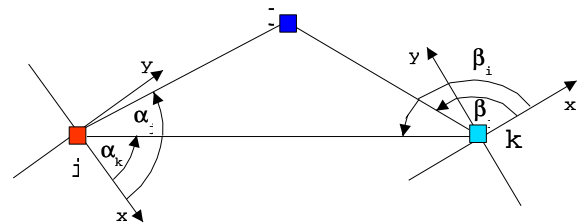


Figure 5: Position of the node j in the local coordinate systems of i and k .

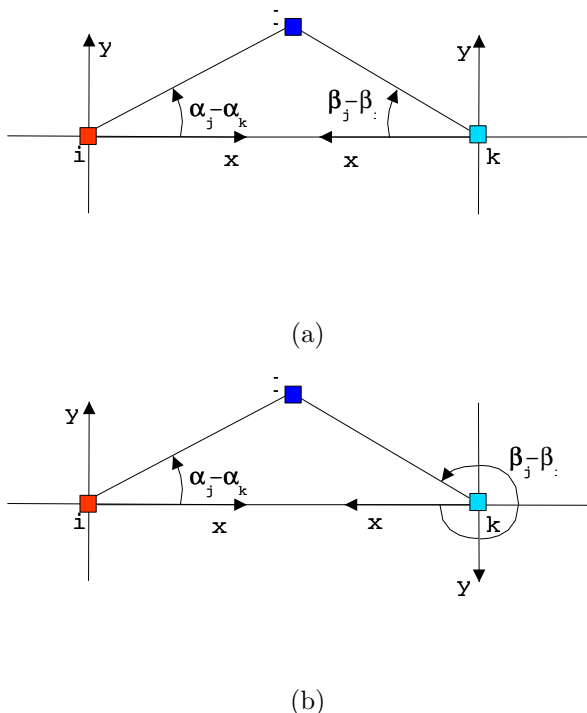


Figure 6: The position of the node j in the local coordinate systems of i and k gives to node k the information if its coordinate systems needs to be rotated (a) or rotated and mirrored (b).

and k have the same direction of their local coordinate systems. The same procedure can be repeated for all the nodes in the network.

4.2 Position computing

Our goal is to achieve that all the nodes obtain their positions within one coordinate system. If the coordinate system of node i is chosen to be the referent coordinate system, 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 the previous section, we explained how a node can adjust its coordinate system to have the same direction as one of its neighbors. Here 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 position, 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

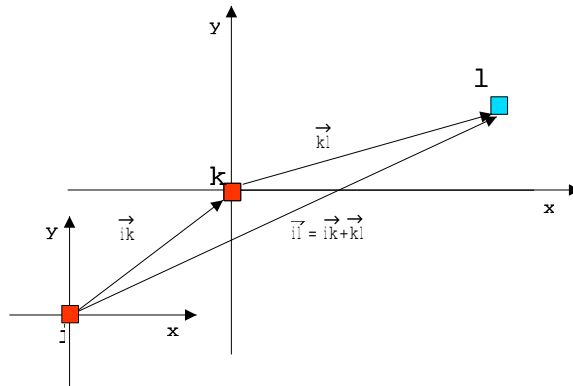


Figure 7: Position computing when the local coordinate systems have the same direction.

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.

$$\vec{il} = \vec{ik} + \vec{kl} \quad (8)$$

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 referent coordinate system by triangulation.

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 the disconnection of the nodes is not expected. A more stable approach, but in terms of a message broadcast, a very costly approach 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.

We propose the following approach. We define a set of nodes called *Location Reference Group* $LRG \subseteq N$ chosen to be stable and less likely to disappear from the network. For example, we chose it such that the density of the nodes in the LRG 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 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, more difficult it becomes to maintain and more costly to compute the center and the direction of the network coordinate system.

4.3.1 Location Reference Group initialization

Every node 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:

$$\begin{aligned} c_x &= \frac{\sum j_x}{m} \\ c_y &= \frac{\sum j_y}{m}, \end{aligned} \quad (9)$$

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. The density factor is the ratio between the number of nodes and the size of the observed area.

Once the node computes 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 a 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 this group. The size of the LRG can be modified by changing the n factor. This factor can be tuned according to the network size.

4.3.2 Location Reference Group maintenance

Due to the mobility of the nodes, the LRG members will change their position and 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 the highest number of the LRG nodes still 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.

If the node does not have the LRG master in its n-hop neighborhood, it starts an initialization timer. If within certain time the node does not receive the

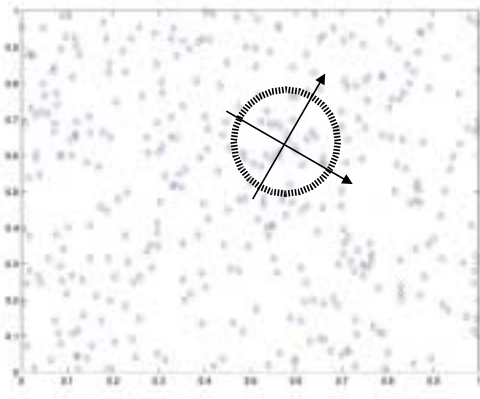


Figure 8: The Location Reference Group.

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 2-hop LRG in a network of 400 nodes.

4.3.3 Network coordinate system direction

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, which is performed at each node that belongs to the Location Reference Group:

- The node initially chooses the direction of the coordinate system, by choosing its (p, q) pair. We note this coordinate system as C_1 .
- When rerunning the local coordinate system algorithm, the node chooses the new (p, q) pair. We note 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.
- It compares the positions of the nodes in two coordinate systems and searches for the maximum

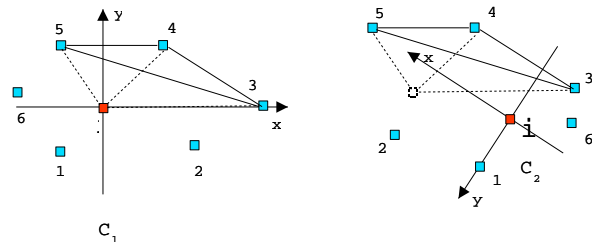


Figure 9: An example illustrating the reconstruction of the coordinate system C_1 in the coordinate system C_2 .

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 can not 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 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.

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 measurement 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]. We believe that a range estimation errors will be corrupted by the same types of errors in mobile ad-hoc networks as in cellular systems. To the best of our knowledge, no measurements have been made to give the overall distribution of the range error in mobile ad-hoc networks.

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 the ad-hoc mobile environment, where there are no base stations to rely on. This is due to the fact that it is irrelevant whether one station is moving or both as their relative movement will produce some errors. Additionally, the errors will be caused by the environment, which does not depend on the motion of the stations. 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:

- 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 neighbours
- 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 minimal 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 is much higher. However, in cellular system base stations have fixed positions, and their mutual distances do not introduce any error.

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, 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. 10 shows that the average number of the nodes for which the position can be obtained, $|LVS|$, is always lower than the average number of neighbors $|K|$. Fig. 11 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 in respect to the power range is shown in Fig. 11. This illustrates that the full LVS connectivity will be achieved at 260 m power range, while the node connectivity is achieved at 170 m. Therefore, the positions will be computed for all the nodes if the nodes have 260 m power range. For 225 m, the positions of the nodes will be computed within the dominant set that contains 90 percent of the nodes.

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. 12 shows that if we choose a larger (3-hop) neighborhood instead of a 2-hop neighborhood,

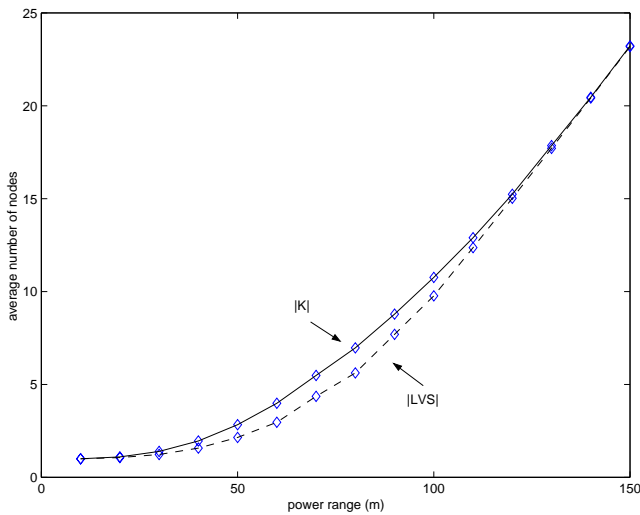


Figure 10: Average number of neighbors $|K|$ (solid line) and average number of nodes for which the positions can be obtained $|LVS|$ (dashed line).

the mobility of the center of the network decreases accordingly. Fig. 13 illustrates the influence of the average node speed increase on the changes of the network coordinate system direction.

7 Conclusions

In this paper we showed that:

- It is possible to achieve a unique coordinate system by self-organization of the nodes.
- The algorithm imposes requirements on the node connectivity.
- The angle and the center of the network coordinate system can be stabilized using simple heuristics.
- The algorithm provides enough information to support network functions such as Location Aided Routing and Geodesic Packet Forwarding.

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

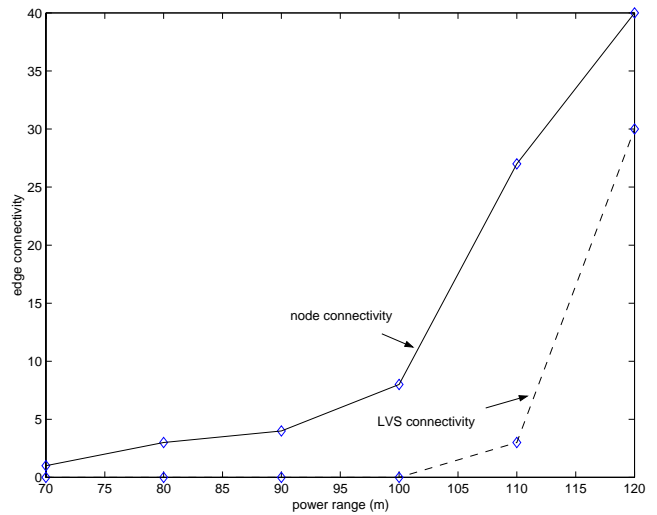


Figure 11: Node and LVS connectivity.

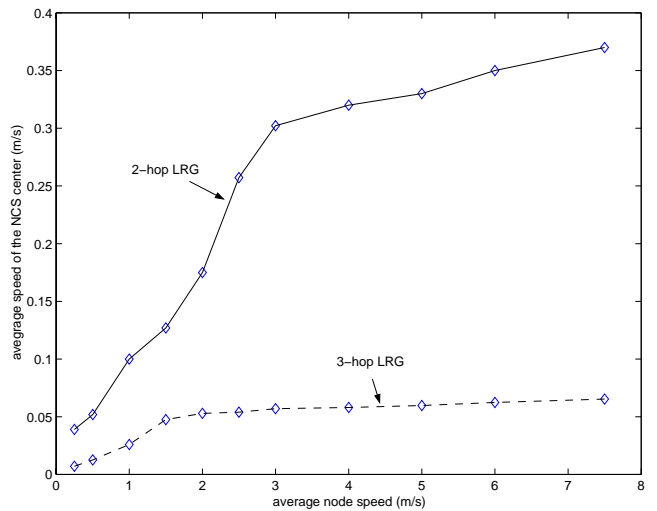


Figure 12: Speed of the network coordinate system center for 2-hop and 3-hop LRG.

the inconsistency between the computed and the real position of the center.

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 the centralized knowledge about the network topology. One major drawback 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

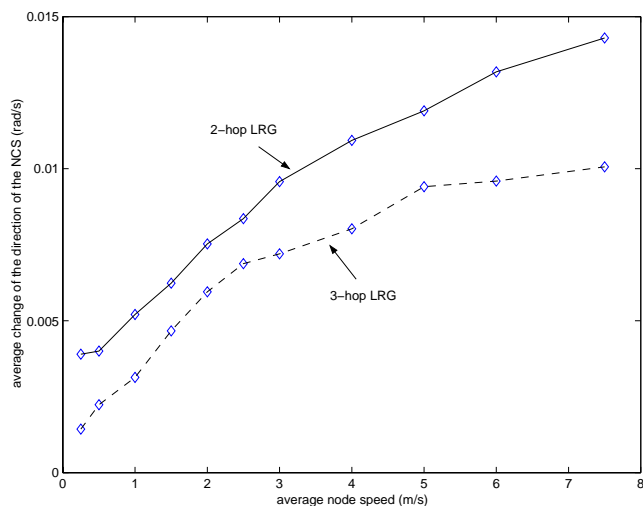


Figure 13: Comparison of speed of change of the direction of the network coordinate system center for 2-hop and 3-hop Location Reference Groups.

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.

Future work in GPS-free positioning in ad-hoc networks would be to improve the accuracy of the range measurements and therefore to reduce the position error. An additional improvement would be to improve the center and direction stability heuristics and to extend the algorithm for three dimensional models. Finally we envision to test the algorithm and its performance in the real world application.

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*, PentechPress, 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] I. Stojmenovic and X. Lin, *A loop-free routing for wireless networks*, Proc. IASTED Conf. On parallel and Distributed systems, 1998.
- [15] E. Kranakis, H. Singh and J. Urrutia, *Compass routing in geometric networks*, Proc. Canadian Conference on Computational Geometry, 1999.