

# ON SELECTING NODES TO IMPROVE ESTIMATED POSITIONS

Erwan Ermel,<sup>1,2</sup> Anne Fladenmuller,<sup>1</sup> Guy Pujolle,<sup>1</sup> and André Cotton,<sup>2</sup>

<sup>1</sup>*Laboratoire d'Informatique de Paris 6  
Université Pierre et Marie Curie  
Paris, France*

firstname.lastname@lip6.fr

<sup>2</sup>*Thales Communication  
BGCOM/TCF/SEA/TAI  
Colombes, France*

firstname.lastname@fr.thalesgroup.com

## Abstract

We consider node localization problems in ad hoc wireless networks in which two types of nodes are considered: nodes with self-locating capability like GPS and nodes with no self-locating capability. Our simple algorithm for improving the position accuracy consists of selecting and processing only the nodes that are likely to enhance the position estimation. We focus our approach on defining a hull of neighboring nodes as key of position accuracy enhancement.

## 1. Introduction

Recent developments of wireless technologies have allowed new types of networks to be envisaged. Mobile *ad hoc* networks (MANETs), which consist of wireless hosts establishing multi-hops communication with each others in the absence of fixed infrastructure, represent one of the current challenge of the networking research community. Many applications are considered for these new mobile ad hoc networks, for instance military ones to establish communications in war theatres, for disasters relief or more basically to set a network in large zones when it is difficult to install a cable-based infrastructure.

Routing in such conditions is quite challenging. For instance, due to the unpredictable mobility of nodes, the network topology is inherently unstable. Important aspects to consider include, the robustness of the solution, its compatibility with existing solutions in wired environments, and its availability of routes. Most existing routing proposals are topology-based routing protocols

such as DSR [Johnson et al., 2001], AODV [Perkins, 2001], OLSR [Clausen and Jacquet, 2003]. However, with these schemes, when the number of nodes increases, the size of the routing tables grows considerably, they become difficult to maintain and therefore their scalability quickly becomes a complex issue. Another approach has been proposed which overcomes this problem: position-based routing protocols, such as GPSR [Karp and Kung, 2000] based on the Greedy-Face-Greedy algorithm [Bose et al., 2001], and Geocasting [Navas and Imelinski, 1997]. Such an approach does not require any routing table as routing is based on the geographical position of nodes. Nevertheless, such an approach requires that each node has its own position coordinates to be part of the routing protocol. This can be achieved, if all nodes are equipped of positioning equipments such as a GPS type systems but this hypothesis can be seen as restrictive. As a matter of fact, for cost reasons, it is more than likely that ad hoc networks will be composed of heterogeneous nodes: some will have self-locating capability such as GPS [Hofmann-Wellenhof et al., 1997] or Galileo [Galileo, ], whereas some others, that we'll refer to as *simple* nodes, will have to estimate their position.

Position estimation methods are mostly based on geometrical computations like triangulation and trilateration. To evaluate the distance between two nodes, four classes of methods can be defined. The first one consists of estimating the distance by estimating the *time-of-flight* of a signal between two anchors. Time Of Arrival (TOA) is used in [Capkun et al., 2002, Werb and Lanzl, 1998] and in all radar systems while the Time Difference Of Arrival (TDOA) technique is used in [Savvides et al., 2001, Ward et al., 1997] (differential time from two anchors or from two different signal like radio and ultra-sound). The second class is based on the strength of the signal. For a given emitting power, the distance can be estimated as the signal strength decreases with the distance. Radars equipments are also based on this technique [Savarese et al., 2001, Beutel, 1999]. The third class is based on triangulation. The Angle of Arrival (AoA) estimates the direction of an incoming signal from several anchors, and then estimates the position. This method is used in VOR systems [vor, ]. The fourth and last class merges all the remaining techniques like connectivity based approach [Doherty et al., 2001][Niculescu and Nath, 2003][Bulusu et al., 2000], Indoor Localization Systems [Want et al., 2000].

In [Ermel et al., 2004], we propose a simple convex hull selection to enhance the accuracy of the position estimation process. We extends in this paper this approach by selecting nodes by their position and also their position accuracy.

In this paper, we propose to enhance the accuracy of the estimated position without adding any new constraint on the environment so this proposal could be used for each of the position estimation method stated above. The structure of this paper is done as follow. We first present the assumptions and the definition made in this paper. Section 3 details two hull methods to select an-

chors within neighbors nodes, followed in Section 4 by our simulation results. Section 5 concludes the paper.

## 2. Assumptions and definitions

We limit our approach to a one-hop anchor selection but the technique is also feasible for  $n$ -hops node selection. No distance measurement is to be used between nodes to estimate the position of a simple node. Therefore a simple node only exploits its own connectivity to other nodes in its direct neighborhood.

Let  $S$  be a *simple* node. Let  $S_{est}$  be the estimated position and  $S_{real}$  be the coordinates of the real location of  $S$ . As  $S_{real}$  is to be estimated by  $S_{est}$ ,  $S_{real}$  information is only used by simulations.

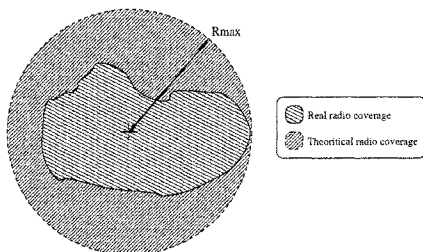


Figure 1. Theoretical and real radio coverage.

Let  $R_{max}$  be the maximum theoretical transmission range of node  $S$  (Fig. 1). We also define the accuracy of the node position  $C_{acc}$ , which is function of the localization error represented by the distance between  $S_{real}$  and  $S_{est}$ :

$$C_{acc} = 1 - \frac{|S_{real}S_{est}|}{R_{max}} \tag{1}$$

where  $|AB|$  is the distance between  $A$  and  $B$ . By construction,  $0 \leq C_{acc} \leq 1$ . The position estimation process may badly estimate  $S_{est}$  and somehow an impossible geometrical case occurs:  $S_{est}$  doesn't lie into the radio coverage area of  $S$  i.e.  $|S_{real}S_{est}| \leq R_{max}$ . Thus a minimum function has to be used to solve this problem:

$$C_{acc} = 1 - \frac{\min(|S_{real}S_{est}|, R_{max})}{R_{max}} \tag{2}$$

Self-locating nodes with accurate position, like position given by a GPS have a position accuracy of 1. Simple nodes which have to estimate their position have a position accuracy  $0 \leq C_{acc} < 1$ . As real positions are unknown

for simple nodes, simple nodes have to estimate their position but also the precision of their position. Several methods to estimate the position accuracy, like statistical approach and area computing approach, are detailed in [Ermel et al., 2003].

In following sections, we assume that the nodes can estimate their position accuracy  $C_{acc}$  as a function of their estimated position and their neighborhood. We also define  $R_{error}$  as the maximum radius of the radio coverage of a node that takes into account the accuracy of its estimated position:

$$R_{error} = R_{max} + R_{max}(1 - C_{acc}). \tag{3}$$

For example, for a self-locating node,  $R_{error} = R_{max}$  while for simple nodes  $R_{max} < R_{error} \leq 2R_{max}$ .

### 3. Anchors selection

Our main goal in this paper is to enhance the accuracy of an estimated position by selecting only anchors that are likely to improve the position estimation process. We detail in this section two hull selections schemes to achieve our goal.

Computational geometry is the study of algorithms for solving geometric problems on a computer. These problems are for example selecting a convex hull among a list of nodes, Voronoï diagrams, geometric searching. These algorithms are well detailed in [Preparata and Shamos, 1991, de Berg et al., 1997, O'Rourke, 1998, Lemaire, 1997].

The main idea of using a convex hull as a selection method among nodes is to choose only nodes that have the greatest distance between anchors. As the position estimation process is based on trilateration, the further the anchors are from each other, the better will be the accuracy of the estimated position.

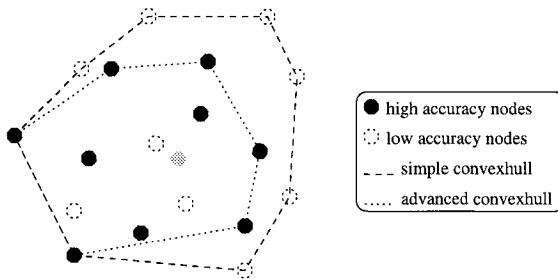


Figure 2. Convex hull: simple convex hull considers only the distance metric to elect hull nodes while in advanced hull distance and position accuracy of the nodes are taken into account.

Only the hull nodes are taking into account in the position estimation process. The remaining nodes are simply discarded. A convex hull example is shown in Fig. 2.

We detail here two hull methods: the simple convex hull and the advanced convex hull.

### 3.1 Simple convex hull

In the plane, the convex hull of  $S$ , a set of points, can be visualized as the shape assumed by a rubber band that has been stretched around the set  $S$  and released to conform as closely as possible to  $S$ .

The simple convex hull selects only the nodes for their physical position whatever their position accuracy  $C_{acc}$  is.

An example of this simple convex hull definition is shown in Fig. 2.

### 3.2 Advanced hull

The simple hull selection phase chooses the border nodes of the neighborhood regardless of their position accuracy  $C_{acc}$ . Thus it seemed interesting to improve this process by selecting the nodes with the highest accuracy the closest to the ones chosen by the simple hull method. While the simple convex hull is defined with a single distance metric, we then defined an advanced hull with two metrics: the position and the position accuracy of the nodes.

As shown in fig. 2, if a simple greedy advanced convex hull is used, the two resulting hulls are quite different. As we worked from the assumption that the further the nodes are, the better the resulting position accuracy is, the use of accurate nodes appears not that far from being the best solution. Nevertheless, the selection of the inner nodes of the hull, but the closer from the hull border, is not a simple task.

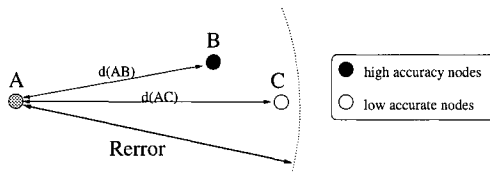


Figure 3. Virtual accuracy definition: accurate nodes may be chosen as an anchor despite their distance from the source  $A$

Therefore, we define a Virtual Accuracy parameter  $V_{acc}$  of a node as:

$$V_{acc}^B = \frac{|AB|}{R_{error}^A} C_{acc}^B \tag{4}$$

where  $|\overline{AB}|$  is the distance between a node  $A$  and an anchor  $B$ ,  $R_{error}^A$  the maximum transmission range of  $A$  defined in Section 2, and  $C_{acc}^B$  the position accuracy of  $B$ .

The Virtual Accuracy  $V_{acc}$  merges the distance and the position accuracy metrics defined in the advanced hull into a simple metric. The advanced hull selection is divided into two steps:

- A simple convex hull selection to determine the nodes belonging to the convex hull.
- The selecting of the appropriate nodes as a function of their virtual accuracy  $V_{acc}$  parameter.

The second step consists of selecting nodes close to the hull with the highest virtual accuracy. Nodes that do not belong to the simple hull compare their virtual position accuracy to their nearest hull nodes; then the highest virtual accuracy node is selected as a member of the advanced hull. If the virtual accuracy of the nodes is the same, the node with the highest accuracy  $C_{acc}$  is selected. Example of this case is shown in Fig. 3.

Note that by selecting nodes with a higher virtual accuracy than hull nodes, the hull is no longer convex.

The simple convex hull selection is based only on a simple distance metric whereas the advanced hull is based not only on a distance metric but also on the position accuracy of the nodes. We will study in Section 4 the performances of these two hull selection methods.

## 4. Simulation Results

The simulations were performed under Java. 50 nodes were randomly placed in a 1000m x 1000m square. Self-locating nodes and simple nodes were also randomly selected. The maximum theoretical transmission range  $R_{max}$  was set to 170m. The mean and the standard deviation are obtained by a maximum likelihood estimation of the mean of the  $\beta$ -distribution. Matlab has been used to solve the  $\beta$ -distribution.

### 4.1 Evaluation of the hull selection

We first evaluate the *fairness* of the selection between the simple nodes (sn) and self-locating nodes (sln). The selection may favor high accurate nodes to the detriment of others. De facto, our selection can be called unfair as the different classes of nodes are not equal to the selection process.

Let  $i$  (respectively  $j$ ) be the number of sln nodes before the selection (respectively sn nodes). We store into a selection matrix  $SM_{sln}$  (respectively

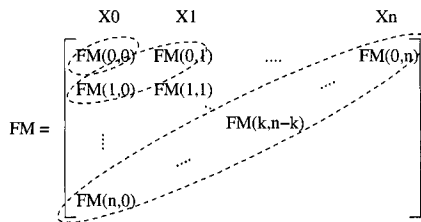
$SM_{sn}$ ) the number of remaining sln nodes (respectively sn nodes) after the selection step as a function of  $i$  and  $j$ .  $SM_{sln}(i, j)$  is the number of remaining sln nodes after the selection. We define also two other matrices  $FM_{sln}$  and  $FM_{sn}$  (Fairness Matrices):

$$FM_{sln}(i, j) = \frac{SM_{sln}(i, j)}{i} \text{ and } FM_{sn}(i, j) = \frac{SM_{sn}(i, j)}{j}$$

$FM(i, j)$  is the fraction of the number of nodes of a class after the selection by the initial number of neighbor nodes of the same class. Let  $X_{sln}^k$  and  $X_{sn}^k$  be a set of value defined as:

$$X_{sln}^k = \{FM_{sln}(i, j) \text{ with } i + j = k; \forall i \neq 0; \forall k \leq n\}$$

$$X_{sn}^k = \{FM_{sn}(i, j) \text{ with } i + j = k; \forall j \neq 0; \forall k \leq n\}$$



with for example

$$X_{sn}^1 = \{FM_{sn}(0, 1)\}$$

$$X_{sln}^n = \{FM_{sln}(1, n) \dots FM_{sln}(k, n - k) \dots FM_{sln}(n, 0)\}$$

At last,  $X_{sln}^k$  and  $X_{sn}^k$  are merged into a single term  $X^k$  as:

$$X^k = X_{sln}^k \cup X_{sn}^k$$

The *fairness index*  $F$  defined in [Jain, 1992] is now used on the  $X^k$  sets to estimate the fairness of the selection among the sln nodes and the sn nodes:

$$F_{index}(X^k) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \text{ with } x_i \in X^k. \tag{5}$$

For all values of  $x_i$ , the *fairness index*  $F$  always remains between 0 and 1. If the selection is fair among the different classes of nodes,  $F$  is equal to 1.

The results of the fairness index are shown in Fig. 4 and Fig. 5

Fig 4 shows the fairness of the hull selection among the self-locating nodes and the simple nodes. From 1 to 3 neighbor nodes, no distinction is made in the selection process between the self-locating nodes and the simple nodes. These results are indeed expected as no selection is done under 3 neighbor nodes.

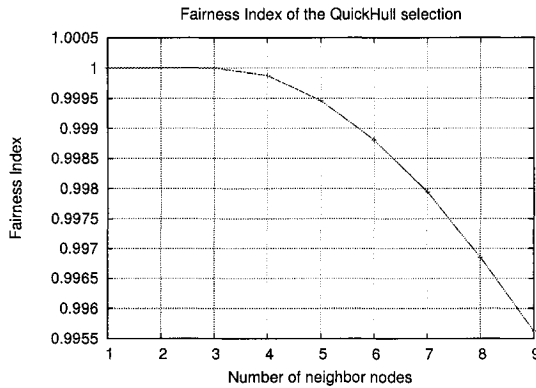


Figure 4. Fairness index of the hull selection among the neighbor nodes.

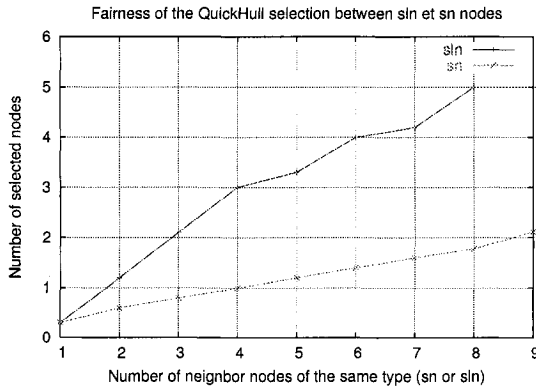


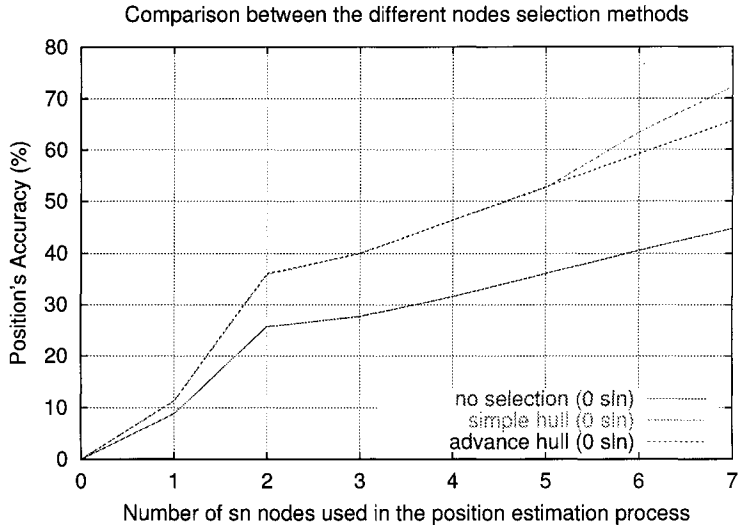
Figure 5. The selection advantages the sln nodes to the detriment of sn nodes.

Above 3 neighbors, the fairness index decreases and proves that our selection is no fairer. However  $F$  only gives an idea of the fairness of the selection between the different nodes classes but gives no clues on which class is privileged i.e. we don't know if the sln or the sn nodes are privileged. As the values used in  $F$  are already averages of data,  $F$  decreases slightly from 1 to 0.9955. These values seem very low, but by construction of the averages, high variations of the fairness index can't be seen in a global view. But if comparisons are made between simple cases, the variations of  $F$  are noticeable.

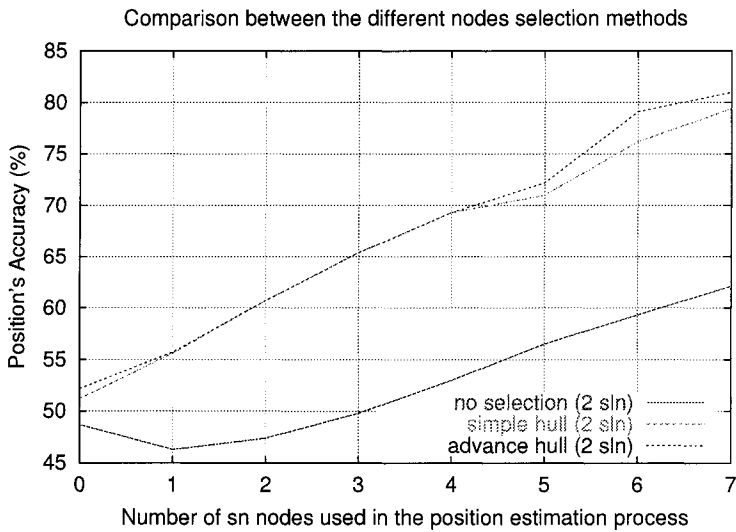
As for Fig 5 the selection ratio between the sln nodes and the sn nodes is shown. For example, with 4 sln neighbor nodes, 3 are kept by the selection



while for 4 sn nodes, only one of them is kept by the selection. Thus our selection really favors sln nodes.



(a)



(b)

Figure 6. Evaluation of the effect of the nodes selection on the accuracy of the estimated position of the nodes.

Fig.6 shows the impact of our selection on the accuracy of an estimated position. To compare our method, we choose as the reference model the greedy scheme: all the neighbor nodes are selected in the position estimation process. None are discarded. The estimated position is obtained by a simple centroid formula, where all the nodes are given the same weight.

Fig. 6(a) shows the resulting accuracy of an estimated method when no self-locating nodes are present in the neighborhood ( $sln=0$ ) while in Fig. 6(b) two  $sln$  nodes are present. In both figures, the simple selection gives in every cases better position accuracy than the greedy approach does. The selection enhances the position accuracy up to 20% for the simple hull selection.

While the simple hull selection enhances the position accuracy of an estimated position for every case, the advanced hull selection gives different results depending on the number of  $sln$  nodes selected in the position estimation process. When no  $sln$  nodes are used, like in Fig. 6(a), the resulting position accuracy is even worse than the one given by the greedy method. These results are due to the fact that the advanced hull selection uses the position accuracy of the nodes. But in that case, there is not trustworthy node. Therefore the results show a *snow ball effect* on the position accuracy: position errors are widely propagated along the position estimation process. Thus the advanced hull selection is not suitable when no trustworthy nodes are present in the neighborhood.

Fig. 6(b) shows that the advanced hull selection gives better results than in Fig. 6(a). In this case, 2  $sln$  nodes are used in the position estimation process. The advanced hull selection do not improve significantly the simple hull selection in low density networks (only 1 to 2 %). In high density cases, like with at least 7 neighbors, the advanced selection improves the position accuracy up to 5%.

The complexity of a simple convex hull is  $o(n \cdot \ln(n))$  of its one step while advance hull selection need two steps, the first is a simple hull selection ,  $o(n \cdot \ln(n))$ , and the second for searching good virtual nodes is performed in  $o(n(n - 1))$ . Thus from the results and the complexity of the different algorithms, only the simple convex hull selection is really suitable as a good position estimation enhancement.

The position accuracy mainly takes advantages of the distance between nodes and of the network density. Using the position accuracy as a selection factor in a node selection process is then not a good way to enhance the position accuracy in low density networks.

The simulation results show that only the distance between nodes have an importance in the accuracy of an estimated position.

## 5. Conclusion

We present and compare in this paper two simple methods to select anchors in a wireless network to enhance the position estimation of simple nodes, with no self-locating capabilities.

The first proposal consists of defining a convex hull among neighbor nodes while our second proposal extends the simple convex hull selection by using in addition to the distance metric the position accuracy of neighbor nodes. Whatever the hull method used to select the nodes, the resulting position accuracy is enhanced from a greedy scheme up to 20%. We also show that the accuracy of an estimated position only takes advantage of the distance between the hull nodes, whatever their position accuracy. Thus the simple hull selection gives better results than the advanced hull selection in low density networks, while in high density networks, the advanced hull selection improves slightly the simple selection.

Our next step consists of implementing such selection algorithms in a global geographical routing protocol in a heterogeneous network.

## References

- [vor, ] *Flight Training Handbook*. Aviation Book Co.
- [Beutel, 1999] Beutel, J. (1999). Geolocalisation in a picoradio environment. Master's thesis, ETH Zurich, Electronics Lab.
- [Bose et al., 2001] Bose, P., Morin, P., Stojmenovic, I., and Urrutia, J. (2001). Routing with guaranteed delivery in ad hoc wireless networks. volume 7, pages 609–616.
- [Bulusu et al., 2000] Bulusu, N., Heidemann, J., and Estrin, D. (2000). Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communication, Special Issue on Smart Spaces and Environment*, 7(5):28–34.
- [Capkun et al., 2002] Capkun, S., Hamdi, M., and Hubaux, J. P. (2002). Gps-free positioning in mobile ad-hoc networks. *Cluster Computing*, 5(2).
- [Clausen and Jacquet, 2003] Clausen, T. and Jacquet, P. (2003). Optimized link state routing (olsr) protocol. *Internet Draft*.
- [de Berg et al., 1997] de Berg, M., Kreveld, M., Overmars, M., and Scharzkopf, O. (1997). *Computational Geometry, Algorithms and Application*. Springer.
- [Doherty et al., 2001] Doherty, L., Pister, K. S. J., and Ghaoui, L. E. (2001). Convex optimization methods for sensor node position estimation. In *Proceedings of IEEE INFOCOM'2001*, Anchorage.
- [Ermel et al., 2003] Ermel, E., Fladenmuller, A., Pujolle, G., and Cotton, A. (2003). Estimation de positions dans des réseaux sans-fil hybrides. In *CFIP 2003*.
- [Ermel et al., 2004] Ermel, E., Fladenmuller, A., Pujolle, G., and Cotton, A. (2004). Improved position estimation in wireless heterogeneous networks. In *Networking 2004*.
- [Galileo, ] Galileo. [http://europa.eu.int/comm/dgs/energy\\_transport/galileo/](http://europa.eu.int/comm/dgs/energy_transport/galileo/).
- [Hofmann-Wellenhof et al., 1997] Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. (1997). *Global Positioning System : Theory and Practice*. Springer-Verlag.

- [Jain, 1992] Jain, R. (1992). *The Art of Computer Systems Performance Analysis*. Wiley.
- [Johnson et al., 2001] Johnson, D. B., Maltz, D. A., Hu, Y.-C., and Jetcheva, J. G. (2001). The dynamic source routing protocol for mobile ad hoc networks. *Internet Draft*.
- [Karp and Kung, 2000] Karp, B. and Kung, H. T. (2000). Gpsr : Greedy perimeter stateless routing for wireless networks. In *Proceedings of ACM/IEEE MOBICOM'00*.
- [Lemaire, 1997] Lemaire, C. (1997). *Triangulation de Delaunay et arbres multidimensionnels*. PhD thesis, Ecole des mines de Saint-Etienne.
- [Navas and Imelinski, 1997] Navas, J. and Imelinski, T. (1997). Geocast - geographic addressing and routing. In *Proceedings of ACM/IEEE MOBICOM'97*, volume 3, pages 66–76.
- [Niculescu and Nath, 2003] Niculescu, D. and Nath, B. (2003). Dv based positioning in ad hoc networks. *Telecommunication Systems*, 22(1):267–280.
- [O'Rourke, 1998] O'Rourke, J. (1998). *Computational Geometry in C*. Cambridge University Press.
- [Perkins, 2001] Perkins, C. E. (2001). Ad hoc on demand distance vector (aodv) routing. *Internet Draft*.
- [Preparata and Shamos, 1991] Preparata, F. P. and Shamos, M. I. (1991). *Computational Geometry: An Introduction*. Springer Verlag.
- [Savarese et al., 2001] Savarese, C., Rabaey, J. M., and Beutel, J. (2001). Localization in distributed ad-hoc wireless sensor networks. *Proceedings of the ICASSP*.
- [Savvides et al., 2001] Savvides, A., Han, C.-C., and Strivastava, M. B. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors.
- [Want et al., 2000] Want, R., Hopper, A., Falcao, V., and Gibbons, J. (2000). The active badge location system. *ACM Transaction on Information Systems*, 10(1):91–102.
- [Ward et al., 1997] Ward, A., Jones, A., and Hopper, A. (1997). A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47.
- [Werb and Lanzl, 1998] Werb, J. and Lanzl, C. (1998). A positioning system for finding things indoors. *IEEE Spectrum*, 35(9):71–78.