Summary for MTNS 2006

Sequential Localization of Networks [*]

J. Fang M. Cao A. S. Morse
Yale University

B. D. O. Anderson
Australian National University
and National ICT Australia Ltd.

The sensor network localization problem with distance information is to determine the positions of all sensors in a network given the positions of some sensors and the distances between some pairs of sensors. Knowing the positions of sensors is essential in many network algorithms such as geographic routing and coverage. The sensor network localization problem is solvable if and only if the network is "localizable". A network is said to be *localizable* if there exists exactly one position corresponding to each sensor such that the given inter-sensor distances are satisfied. The authors of [1] use rigidity theory to give the necessary and sufficient conditions for a network to be localizable. However, the process of localizing a network has been shown to be NP-hard even when the network is known to be localizable [2]. This leaves us to consider the more refined question of what kinds of localizable networks can we efficiently localize? This has been investigated in [3] and we extend the results of that paper. While some ingenious heuristics-based schemes have been proposed [4, 5, 6, 7], we are interested in provably correct localization algorithms and the kinds of networks that can be efficiently localized by them. In the following we will give a localization algorithm that consists of a finite number of steps to be carried out sequentially. We will also give some classes of networks that can be efficiently localized by the algorithm. In the process, we will raise a number of graph theoretic questions related to rigidity theory. The resolution of these questions can in turn help us shed light on which networks can be efficiently and provably correctly localized. This is another indication that the sensor network localization problem and rigidity theory are intricately related.

We begin by giving some terms and definitions to be used in the exposition which follows. A *configuration* $p = \{p_1, \ldots, p_m\}$ in d-dimensional space is a set of m points in $\mathbb{R}^d$, and p is said to be *generic* if the coordinates of points in p are algebraically independent. Two configurations p and q of m points are *congruent* if for all $i, j \in \{1, \ldots, m\}$, the distance between the ith and jth points of p is equal to the distance between the ith and jth points of q. A *point formation* of m points at a configuration p consists of configuration p and a simple undirected graph $\mathbb{G}$ with vertex set $\mathcal{V} = \{1, \ldots, m\}$, and is denoted by $(\mathbb{G}, p)$. If (i, j) is an edge in $\mathbb{G}$, then we say the *length* of edge (i, j) in point formation $(\mathbb{G}, p)$ is the distance between the ith and jth points of p. A point formation $(\mathbb{G}, p)$ is *globally rigid* in $\mathbb{R}^d$ if p and q are congruent configurations in $\mathbb{R}^d$ whenever $(\mathbb{G}, p)$ and $(\mathbb{G}, q)$ have the same edge lengths. A graph $\mathbb{G}$ is said to be *generically globally rigid* in $\mathbb{R}^d$ if

[*] The work of Fang, Cao and Morse was supported in part, by grants from the U.S. Army Research Office and the U.S. National Science Foundation and by a gift from the Xerox Corporation. The work of Anderson was supported by an Australian Research Council Discovery Projects Grant and by National ICT Australia, which is funded by the Australian Government's Department of Communications, Information Technology and the Arts and Australian Research Council through the Backing Australia's Ability initiative and the ICT Center of Excellence Program.

$(\mathbb{G}, p)$ is globally rigid in $\mathbb{R}^d$ whenever p in $\mathbb{R}^d$ is generic. Henceforth we will fix $d = 2$ and omit any mention of $\mathbb{R}^d$. Since *almost all* configurations are generic, we will also ignore the non-generic configurations and consider the generic global rigidity of a graph. A graph is *minimally generically globally rigid* if the removal of any edge causes the graph to not be generically globally rigid. A sensor network is modelled by a point formation $(\mathbb{G}, p)$ where p is the set of sensor positions, and each sensor corresponds to exactly one vertex of $\mathbb{G}$, and vice versa, with (i, j) being an edge of $\mathbb{G}$ if the distance between the corresponding sensors is known. We say that $\mathbb{G}$ is the graph of the network. A vertex v of $\mathbb{G}$ is called an *anchor vertex* if the position of sensor v is given and a *sensor vertex* if it is not. It is known that if a network is modelled by point formation $(\mathbb{G}, p)$ and p is generic, then the network is localizable if and only if it has at least 3 non-collinear anchors and $\mathbb{G}$ is generically globally rigid. The *second power* of $\mathbb{G} = (\mathcal{V}, \mathcal{E})$, written $\mathbb{G}^2$, is the graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \cup \mathcal{E}^2$, where $(i, j) \in \mathcal{E}^2$ just in case $i, j \in \mathcal{V}$ and there exists $k \in \mathcal{V}$ such that $(i, k), (k, j) \in \mathcal{E}$.

A graph is said to be a *trilateration graph with trilateration ordering* $\{v_1, \dots, v_n\}$ if its vertices can be relabelled as $\{v_1, \dots, v_n\}$ so that the subgraph induced by v_1, v_2, v_3 is complete and each v_i with $i > 3$ is adjacent to at least three distinct vertices v_j which "precede" it in the ordering, where by *precede* we mean $j < i$ ([1]). It is shown in [1] that trilateration graphs are generically globally rigid. A graph is a *bilateration graph with bilateration ordering* $\{v_1, \dots, v_n\}$ if its vertices can be relabelled as $\{v_1, \dots, v_n\}$ so that (1) the subgraph induced by v_1, v_2, v_3 is complete, (2) each v_i, $3 < i < n$, is adjacent to at least two distinct vertices v_j which precede it in the ordering and one vertex v_k which "follows" it in the ordering, where by *follows* we mean $k > i$, and (3) vertex v_n is adjacent to at least three distinct vertices preceding it in the ordering. We show that bilateration graphs are 2-connected but not necessarily 3-connected. We define the *maximal anchor-free subgraph* of the graph $\mathbb{G}$ of a network to be the maximal subgraph of $\mathbb{G}$ containing only sensor vertices. We say that $\mathbb{G}$ is *partially acyclic* if its maximal anchor-free subgraph is acyclic. A graph is *edge 2-connected* if there exists two paths with no edge in common between any two vertices. It is known that the second power of an edge 2-connected graph is generically globally rigid in $\mathbb{R}^2$ [3]. We show that the second power of an edge 2-connected graph is also a bilateration graph.

In the following, we will describe a localization algorithm that consists of a finite number of steps to be carried out sequentially. We will then use trilateration, bilateration and edge 2-connected graphs to identify and construct some networks that can be efficiently localized by this algorithm.

Let $\mathbb{N}$ be a network of n sensors labelled 1 through n and positioned at distinct points $\pi(1), \pi(2), \dots, \pi(n)$ in $\mathbb{R}^2$. Suppose that $1, 2, \dots, m$ are the labels of $\mathbb{N}$'s anchors and that $m \geq 3$. Let $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ be the network's graph. For each vertex $v \in \mathcal{V}$, let $\mathcal{N}(v)$ denote the set consisting of v and all vertices in $\mathbb{G}$ which are adjacent to v. For each such vertex $u \in \mathcal{N}(v)$ write d_{uv} for the distance between sensors u and v.

Let $2^{\mathbb{R}^2}$ be the power set of $\mathbb{R}^2$ and write $\mathbb{R}_+$ for the set of positive real numbers. Let $f : 2^{\mathbb{R}^2} \times \mathbb{R}_+ \to 2^{\mathbb{R}^2}$ denote the function $(\mathcal{S}, d) \longmapsto \mathcal{S}'$ where $\mathcal{S}'$ is the set of $p \in \mathbb{R}^2$ such that $||p - q|| = d$ for some $q \in \mathcal{S}$. If $\mathcal{S}$ is not empty, then geometrically $f(\mathcal{S}, d)$ is the union of all points in the plane which lie on circles with the same radius d centered at the points in $\mathcal{S}$. Of course if $\mathcal{S}$ is empty then so is $f(\mathcal{S}, d)$ and conversely. We will be especially interested in the case when $\mathcal{S}$ is a non-empty "finite set" and $d > 0$, where by *finite set* we mean a set with a finite number of points in $\mathbb{R}^2$. In this case $f(\mathcal{S}, d)$ is simply the union of a finite number of circles in the plane which all have radius d. An easily verified property of f is that if vertex u is adjacent to vertex $v \in \mathcal{V}$, and $\mathcal{S}(u)$ is a set for which $\pi(u) \in \mathcal{S}(u)$, then $\pi(v) \in f(\mathcal{S}(u), d_{uv})$. We call this the *position keeping* property of f.

Let $\mathbb{S}$ denote the set of all non-empty subsets of $\mathbb{R}^2$ with finitely many elements. Let q be a positive integer no smaller than 2 and write $\mathbb{S}^q$ for the q-fold Cartesian product of $\mathbb{S}$ with itself. Similarly, let $(\mathbb{R}_+)^q$ denote the q-fold Cartesian product of $\mathbb{R}_+$ with itself. Our aim is to define a function $g_q : \mathbb{S}^q \times (\mathbb{R}_+)^q \rightarrow 2^{\mathbb{R}^2}$ in such a way so that for each $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q\} \in \mathbb{S}^q$ and $\{d_1, d_2, \dots, d_q\} \in (\mathbb{R}_+)^q$, $g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q)$ is at most a finite set. Furthermore, we shall require the definition of g_q to be such that whenever there are distinct points $u_i \in \mathcal{S}_i$, $i \in \{1, 2, \dots, q\}$, if $v \in \mathbb{R}^2$ satisfies $\|v - u_i\| = d_i$, $i \in \{1, 2, \dots, q\}$, then v must be a point in $g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q)$. Defining $g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q)$ in the most obvious way as the intersection of the sets $f(\mathcal{S}_i, d_i)$, $i \in \{1, 2, \dots, q\}$, will not be adequate for it may be the case that the resulting intersection is a continuous circle of points in the plane rather than a finite set.

Let $\mathcal{I} = \bigcap_{j=1}^{q} \mathcal{S}_j$, and let $p_1, p_2, \dots, p_k$ denote the elements of $\mathcal{I}$. For any set $\mathcal{S} \in \mathbb{S}$, and any subset $\mathcal{T} \subset \mathcal{S}$, let $\mathcal{S} \backslash \mathcal{T}$ denote the complement of $\mathcal{T}$ in $\mathcal{S}$. The following definition of g_q satisfies both of the requirements listed above:

$$g_q(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q, d_1, d_2, \dots, d_q) = \Big(f(\mathcal{S}_1 \backslash \mathcal{I}, d_1) \cap f(\mathcal{S}_2, d_2) \cap \dots \cap f(\mathcal{S}_q, d_q) \Big) \bigcup \left(\bigcup_{i=1}^{k} f(\{p_i\}, d_1) \cap f(\mathcal{S}_2 \backslash \{p_i\}, d_2) \cap \dots \cap f(\mathcal{S}_q \backslash \{p_i\}, d_q) \right) \tag{1}$$

An ordering $v_1, v_2, \dots, v_n$ of the vertices in $\mathcal{V}$ for which $v_i = i$, $i \in \{1, 2, \dots, m\}$, and the sets

$$\mathcal{M}(v_i) = \mathcal{N}(v_i) \cap \{v_1, v_2, \dots, v_{i-1}\}, \quad i \in \{m+1, m+2, \dots, n\}$$

are all non-empty is called a *sweep* of $\mathbb{N}$ and is denoted by $[v]$. Note that an ordering $v_1, v_2, \dots, v_n$ is a sweep just in case that each v_i, $i > m$, is adjacent to at least one vertex preceding it. We shall require the networks we consider to have at least one "finite position generating" sweep $v_1, v_2, \dots, v_n$, where by *finite position generating* we mean that each vertex v_i, $i > m$, is adjacent to at least two distinct vertices preceding it. If $v_1, v_2, \dots, v_n$ is a finite position generating sweep, then each of the $\mathcal{M}(v_i)$, $i > m$, is a set of at least two elements.

In the following, we will describe a recursive procedure for computing a sequence of finite sets for each vertex $v \in \mathcal{V}$, i.e. $\mathcal{S}(v, 1)$, $\mathcal{S}(v, 2), \cdots \mathcal{S}(v, i), \cdots$, such that each $\mathcal{S}(v, i)$ is a finite set, $\pi(v) \in \mathcal{S}(v, i)$, and if $i < j$, then $\mathcal{S}(v, i) \supset \mathcal{S}(v, j)$. We begin by choosing a finite position generating sweep $[v]$ of $\mathbb{N}$. For $i \in \{m+1, m+2, \dots, n\}$, let

$$\mathcal{M}(v_i) = \mathcal{N}(v_i) \cap \{v_1, v_2, \dots, v_{i-1}\}$$

We denote the cardinality of $\mathcal{M}(v_i)$ by q_i and the elements of $\mathcal{M}(v_i)$ by $u_{i1}, u_{i2}, \dots, u_{iq_i}$. We define the sets $\mathcal{S}(v_i, 1)$, $i \in \{1, 2, \dots, n\}$ as follows. For $i \in \{1, 2, \dots, m\}$, we define

$$\mathcal{S}(v_i, 1) = \pi(v_i) \tag{2}$$

and for $i \in \{m+1, m+2, \dots, n\}$, we define

$$\mathcal{S}(v_i, 1) = g_{q_i}(\mathcal{S}(u_{i1}, 1), \mathcal{S}(u_{i2}, 1), \dots, \mathcal{S}(u_{iq_i}, 1), d_{u_{i1}v_i}, d_{u_{i2}v_i}, \dots, d_{u_{iq_i}v_i}) \tag{3}$$

Note that since $[v]$ is assumed to be a finite position generating sweep, each $\mathcal{M}(v_i)$ has at least 2 elements and $q_i \geq 2$. Hence, for $i \in \{m+1, m+2, \ldots, n\}$, g_{q_i} is defined and $\mathcal{S}(v_i, 1)$ is a finite set because the image of g_{q_i} consists only of finite sets. Since $\mathcal{S}(v_i, 1)$, $i \in \{1, 2, \ldots, m\}$, are also finite sets because of (2), we have that $\mathcal{S}(v, 1)$ is a finite set for each $v \in \mathcal{V}$. Note also that $\pi(v_i) \in \mathcal{S}(v_i, 1)$, $v_i \in \mathcal{V}$. This is clearly true for $i \in \{1, 2, \ldots, m\}$ because of (2). The assumption that $\pi(v)$, $v \in \mathcal{V}$, are distinct together with the definition of g_{q_i} and the position keeping property of f imply that $\pi(v_i) \in \mathcal{S}(v_i, 1)$ for $i \in \{m+1, m+2, \ldots, n\}$. We call the computation of $\mathcal{S}(v, 1)$, $v \in \mathcal{V}$, the *computation of the initial sweep* of $\mathbb{N}$.

Now suppose that the initial sweep of $\mathbb{N}$ has been computed. The sets $\mathcal{S}(v, 2)$, $v \in \mathcal{V}$, are computed as follows. Let $[u] = u_1, u_2, \ldots, u_n$ be a sweep, and let $\mathcal{M}(u_i) = \mathcal{N}(u_i) \cap \{u_1, u_2, \ldots, u_{i-1}\}$ for $i \in \{m+1, m+2, \ldots, n\}$. Note that $[u]$ need not be a finite position generating sweep. For $i \in \{1, 2, \ldots, m\}$ we define

$$\mathcal{S}(u_i, 2) = \pi(u_i) \tag{4}$$

and for $i \in \{m+1, m+2, \ldots, n\}$ we define

$$\mathcal{S}(u_i, 2) = \mathcal{S}(u_i, 1) \bigcap_{w \in \mathcal{M}(u_i)} f(\mathcal{S}(w, 2), d_{wu_i}) \tag{5}$$

For each u_i, $i \in \{m+1, m+2, \ldots, n\}$, (5) implies that $\mathcal{S}(u_i, 2)$ must be a finite set since $\mathcal{S}(u_i, 1)$ is a finite set. Moreover, since $\mathcal{S}(u_i, 2)$, $i \in \{1, 2, \ldots, m\}$, are all finite sets because of (4), we have that $\mathcal{S}(v, 2)$ is a finite set for each $v \in \mathcal{V}$. Also, $\pi(v) \in \mathcal{S}(v, 2)$ for all $v \in \mathcal{V}$. This is clearly true for u_i, $i \in \{1, 2, \ldots, m\}$ because of (4). For $i \in \{m+1, m+2, \ldots, n\}$, that $\pi(u_i) \in \mathcal{S}(u_i, 1)$ and the position keeping property of f imply $\pi(u_i) \in \mathcal{S}(u_i, 2)$. We call the computation of $\mathcal{S}(v, 2)$, $v \in \mathcal{V}$, the computation of the second sweep of $\mathbb{N}$. It is obvious from (4) and (5) that $\mathcal{S}(v, 2) \subset \mathcal{S}(v, 1)$ for all $v \in \mathcal{V}$.

Now suppose the kth sweep of $\mathbb{N}$ has been computed, and that for each $v \in \mathcal{V}$, $\pi(v) \in \mathcal{S}(v, k)$ and $\mathcal{S}(v, k)$ is a finite set. The $(k+1)$th sweep is computed as follows. Let $[x] = x_1, x_2, \ldots, x_n$ be a sweep, and let $\mathcal{M}(x_i) = \mathcal{N}(x_i) \cap \{x_1, x_2, \ldots, x_{i-1}\}$ for $i \in \{m+1, m+2, \ldots, n\}$. For $i \in \{1, 2, \ldots, m\}$, we define

$$\mathcal{S}(x_i, k+1) = \pi(x_i) \tag{6}$$

and for $i \in \{m+1, m+2, \ldots, n\}$ we define

$$\mathcal{S}(x_i, k+1) = \mathcal{S}(x_i, k) \bigcap_{w \in \mathcal{M}(x_i)} f(\mathcal{S}(w, k+1), d_{wx_i}) \tag{7}$$

For each $v \in \mathcal{V}$, we have that $\mathcal{S}(v, k+1)$ is a finite set , $\pi(v) \in \mathcal{S}(v, k+1)$, and $\mathcal{S}(v, k+1) \subset \mathcal{S}(v, k)$ by the same reasoning as before.

The preceding shows that if we compute a sequence of sweeps starting with one which is finite position generating, we can generate a sequence of finite sets for each $v \in \mathcal{V}$, i.e. $\mathcal{S}(v, 1), \mathcal{S}(v, 2), \cdots$, $\mathcal{S}(v, i), \cdots$, where each set is obtained by means of a finite number of computations and

$$\mathcal{S}(v, 1) \supset \mathcal{S}(v, 2) \supset \cdots \supset \mathcal{S}(v, i) \cdots$$

and $\pi(v) \in \mathcal{S}(v, i)$ for each i. Thus if we can select a finite number of sweeps, say k, such that for all $v \in \mathcal{V}$, each $\mathcal{S}(v, k)$ will contain just the position $\pi(v)$ of sensor v, then localization will be complete. We call this the *sequential localization* of the network, and we say that the network is *sequentially localizable* in k sweeps. Hence, sequential localization of a network is carried out in a finite number of steps, each of which is solvable in a straightforward manner. This is in sharp contrast to a direct assault on the localization problem by attempting to solve a large number of simultaneous quadratic equations in $2(n-m)$ variables. In the exposition which follows, we will give the graph properties of some networks that are sequentially localizable in just one or two sweeps. The sweeps are selected by considering properties of the network's graph, hence localizing the network in as few sweeps as possible.

The network $\mathbb{N}$ is said to be *easily localizable* if the vertices of $\mathbb{G}$ can be relabelled as $\{v_1, \ldots, v_n\}$ so that the position of v_i, $i > 3$, can be uniquely determined from just the positions of the vertices in $\mathcal{N}(v_i)$ which precede it together with the distances from all of these vertices to v_i [1, 3]. We show that $\mathbb{N}$ is easily localizable if and only if $\mathbb{G}$ is a trilateration graph. Obviously, easily localizable networks are also sequentially localizable in one sweep.

Suppose the set of sensor positions of $\mathbb{N}$ is a generic configuration in $\mathbb{R}^2$. We show that $\mathbb{N}$ is sequentially localizable in two sweeps if $\mathbb{G}$ is a generically globally rigid partially acyclic bilateration graph. A number of generically globally rigid partially acyclic bilateration graphs are also minimally generically globally rigid graphs. This implies there are localizable networks with just enough edges to ensure localizability that are also sequentially localizable. Many practical networks are such that two vertices of its graph are adjacent if the corresponding sensors are within sensing radius of each other. Suppose $\mathbb{N}$ is such a network. We show that if $\mathbb{G}$ is a "ring graph", then $\mathbb{N}$ is sequentially localizable in two sweeps after doubling the sensing radius of each sensor. Where by a *ring graph* we mean a graph whose vertices can be labelled as $\{1, \ldots, n\}$ so that vertex i, $1 < i < n$, is adjacent to vertices $i-1$ and $i+1$, vertex 1 is adjacent to vertices 2 and n, and vertex n is adjacent to vertices $n-1$ and 1. Note that if $\mathbb{G}$ is a ring graph, then the graph of the resulting network after doubling the sensing radius of all the sensors is a bilateration graph. An important consequence of the previous is that if $\mathbb{G}$ is edge 2-connected, then $\mathbb{N}$ is sequentially localizable after doubling the sensing radius of all the sensors. Note that if $\mathbb{G}$ is edge 2-connected, then the graph of the resulting network after doubling the sensing radius of all the sensors is a bilateration graph. We note that rigidity theory, especially as regards to global rigidity, play an important part in the proofs of these assertions.

Sequential localization of a network may take a number of arithmetic operations anywhere from linear to exponential in the number of sensors depending on the structure of the network's graph. However, the complexity of localizing a large network can be made efficient by localizing sections of the network in sequence. For example, suppose graph $\mathbb{G}$ of a network has subgraphs $\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_D$ where each vertex of $\mathbb{G}$ is in at least one of the subgraphs. For $i \in \{1, 2, \ldots, D\}$, let $\mathbb{N}_i$ denote the network modelled by point formation $(\mathbb{G}_i, p_i)$, where p_i is the set of positions of the sensors corresponding to vertices of $\mathbb{G}_i$. If each $\mathbb{N}_i$ is sequentially localizable when all sensors occurring in $\mathbb{N}_j$, $j < i$, are considered anchors of $\mathbb{N}_i$, then the complexity of localizing the entire network is dominated by $max_{i \in \{1,\ldots,D\}} Z(\mathbb{N}_i)$, where $Z(\mathbb{N}_i)$ is the complexity of localizing $\mathbb{N}_i$. If the number of sensors in each $\mathbb{N}_i$ not occurring in any $\mathbb{N}_j$, $j < i$, is small, then $Z(\mathbb{N}_i)$ is not too large even if it is exponential in the number of sensors of $\mathbb{N}_i$. Consequently the complexity of localizing the entire network is rendered acceptable.

It is an open question if all networks with generically globally rigid bilateration graphs are sequentially localizable. And while we know that not all bilateration graphs are generically globally rigid, the necessary and sufficient conditions for a bilateration graph to be generically globally rigid are still unknown.

References

[1] T. Eren, D. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. In *Proc. of IEEE INFOCOM*, 2004.

[2] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur. A theory of network localization. Unpublished manuscript, July 2004.

[3] B.D.O. Anderson, P.N. Belhumeur, T. Eren, D.K. Goldenberg, A.S. Morse, W. Whiteley, and R.Y. Yang. Graphical properties of easily localizable sensor networks.

[4] A. Savvides, H. Park, and M. Srivastava. The n-hop multilateration primitive for node localization problems. *ACM Mibile networks and applications*, 8:443–451, 2003.

[5] Joe Albowicz, Alvin Chen, and Lixia Zhang. Recursive position estimation in sensor networks. In *Proceedings of the 9th International Conference on Network Protocols*, pages 35–41, 2001.

[6] L. Doherty, K. S. J. Pister, and L. E. Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings of IEEE INFOCOM 01*, page 16551633, Anchorage, Ak, 2001.

[7] D. Niculescu and B. Nath. Ad-hoc positioning system. In *Proceedings of IEEE Globecom 2001*, 2001.