Beyond Trilateration: On the Localizability of Wireless Ad Hoc Networks

Zheng Yang, *Student Member, IEEE, Student Member, ACM*, Yunhao Liu, *Senior Member, IEEE, Member, ACM*, and Xiang-Yang Li, *Senior Member, IEEE, Member, ACM*

***Abstract*—The proliferation of wireless and mobile devices has fostered the demand of context-aware applications, in which location is often viewed as one of the most significant contexts. Classically, trilateration is widely employed for testing network localizability; even in many cases, it wrongly recognizes a localizable graph as nonlocalizable. In this study, we analyze the limitation of trilateration-based approaches and propose a novel approach that inherits the simplicity and efficiency of trilateration and, at the same time, improves the performance by identifying more localizable nodes. We prove the correctness and optimality of this design by showing that it is able to locally recognize all one-hop localizable nodes. To validate this approach, a prototype system with 60 wireless sensors is deployed. Intensive and large-scale simulations are further conducted to evaluate the scalability and efficiency of our design.**

***Index Terms*—Localizability, localization, trilateration, wireless ad hoc and sensor networks.**

I. INTRODUCTION

PERVASIVE and mobile systems for context-aware computing are growing at a phenomenal rate. In most of today's applications, such as pervasive medical care, smart space, wireless sensor network surveillance, mobile peer-to-peer computing, etc., location is one of the most essential contexts.

In recent years, a number of schemes have been proposed for in-network localization, in which some special nodes (called beacons or seeds) know their global locations and the rest determine their locations by measuring the Euclidean distances to their neighbors. Several distance-ranging methods, such as Radio Signal Strength (RSS) [24] and Time Difference of Arrival (TDoA) [21], are adopted in practical systems. Based on those approaches, the ground truth of a wireless ad hoc network can be modeled by a distance graph [17] $G=(V,E,d)$, where V is the set of wireless communication devices (e.g., laptop, RFID, or sensor node) and there is an edge $(i,j) \in E$ if the distance between i and j, denoted by $d(i,j)$, can be measured. Beacons are pairwisely connected since the inter-beacon distances can be determined according to beacon locations.

Consequently, an essential question is followed as whether or not a network is localizable by given its distance graph. A graph $G=(V,E,d)$ with possible additional constraint I (such as the known locations of beacon nodes) is called *localizable* if there is a unique location $p(v)$ of every node v such that the distance $d(u,v)=d(p(u),p(v))$ for all links (u,v) in E and constraint I is preserved. A node is localizable if it belongs to a localizable network or subnetwork. Being aware of localizability not only helps localization, but also provides instructive directions to some location-based services, such as topology control, mobility control, and network diagnosis. For example, for those nonlocalizable networks, we expect to make them localizable by adjusting some network parameters. Traditional solutions include augmenting ranging capability, increasing node density, controlling node mobility, or equipping more nodes with GPS. Such measures can be more efficiently conducted with the knowledge of localizability. For example, these adjustments may focus only on nonlocalizable nodes instead of blindly exerting on all nodes.

Previous studies have shown that the localizability problem is closely related to the graph rigidity [3], [6], [7], [10]. A graph is called *generically rigid* (or called rigid) if one cannot continuously deform the graph embedding in the plane while preserving the distance constraints [10]. Here, the word "generically" means the distances are algebraically independent, i.e., no degeneracy. A graph is *generically globally rigid* (or called globally rigid) if there is a unique realization in the plane [6]. Jackson *et al.* [7] prove that a graph is globally rigid if and only if it is 3-connected and redundantly rigid. A graph is *redundantly rigid* if the removal of any edge results in a graph that is still rigid. Accordingly, the localizability of a graph can be answered in polynomial time in a centralized manner by testing the 3-connectivity and redundant rigidity [9].

Designing an efficient distributed algorithm for global rigidity, however, is nontrivial as neither connectivity nor rigidity can be tested locally by nature. For example, Fig. 1 shows a graph consisting of two known 3-connected components and three edges (1, 6), (2, 5), and (3, 4) between them. In this case, three bridge edges are far away from each other. By employing any localized algorithm on this example, a single node (without loss of generality, say node 1), using the information only from neighbors within a constant number of hops, cannot be aware of the existence of edges (2, 5) and (3, 4)

Manuscript received April 13, 2009; revised October 29, 2009 and March 23, 2010; accepted April 14, 2010; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Liu. Date of publication May 20, 2010; date of current version December 17, 2010. This work was supported in part by NSFC/RGC Joint Research Scheme N_HKUST 602/08; NSF CNS-0832120; the NSFC under grants 60828003, 60873262, and 60970118; the program for Zhejiang Provincial Key Innovative Research Team; the program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program); and the National Basic Research Program of China (973 Program) under grants 2010CB328100 and 2010CB334707.

Z. Yang and Y. Liu are with the Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: yangzh@cse.ust.hk; liu@cse.ust.hk).

X.-Y. Li is with the Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: xli@cs.iit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2010.2049578

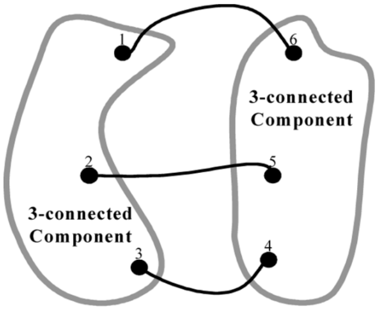


Fig. 1. Global information is needed to test connectivity.

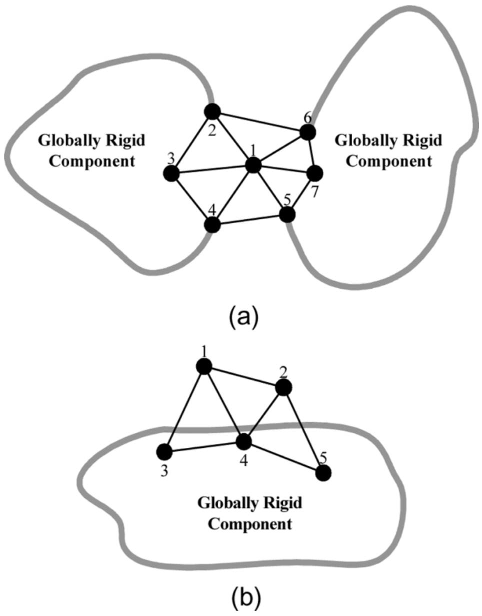


Fig. 2. The deficiency of trilateration. (a) Geographical gap. (b) Border nodes.

that are not incident upon itself. Thus, it fails to identify the entire graph as 3-connected. For rigidity, the situation is the same as connectivity.

As a compromise, trilateration is proposed for testing localizability based on the fact that the location of an object can be determined if the distances to three references are known. Accordingly, it is possible to identify localizable nodes in a network by iteratively applying trilaterations. In practice, trilateration is widely used [18], [20], [22] as it is fully distributed, easy to implement, and efficient in terms of communication and computation.

Trilatertion-based approaches, however, recognize only a subset (called trilateration extension) of globally rigid graphs. In Fig. 2(a), two globally rigid components are connected by nodes $i (i = 1, 2, \ldots, 7)$. Suppose the nodes 1, 2, 3, and 4 in the left component are known as localizable. The localizability information cannot propagate to the other part by trilateration since none of the nodes 5, 6, and 7 connects to three localizable nodes. Obviously, trilateration wrongly reports that nodes in the right component are not localizable, ignoring the fact that the entire graph is globally rigid.

A similar situation occurs again for the border nodes, as illustrated in Fig. 2(b). In this case, nodes 1 and 2 cannot be localized by trilateration even though nodes 3, 4, and 5 are localized. However, the entire graph in Fig. 2(b) is globally rigid and thus localizable. Importantly, border nodes are often more critical in many applications. For example, a sensor network for forbidden region monitoring has special interests on when and where intruders crash into, which are collected by border nodes only.

These observations expose the deficiency of trilateration-based methods. In this study, we address the challenge of designing localized algorithms for localizability. Our study shows that trilateration is actually a special case, the simplest with four nodes, of *wheel* graphs [26], which motivates us to explore the possibility of generalizing the idea of trilateration.

The main contributions of this paper are as follows. Based on the fact that wheel structures are globally rigid, we present a distributed algorithm to find localizable nodes by testing whether they are included in some wheel graphs within their neighborhoods. The algorithm inherits the simplicity and efficiency of trilateration while, at the same time, improving the performance by identifying more localizable nodes. We prove the optimality of this design: Using only local information, it is able to recognize all one-hop localizable nodes. That is, in theory, the proposed algorithm achieves the upper-bound performance of all distributed algorithms.

We validate this design by deploying a prototype system with 60 wireless sensors. The large-scale simulations are further conducted to examine the efficiency and scalability. The results show that our design remarkably outperforms the widely used trilateration.

The rest of the paper is organized as follows. In Section II, we focus on the problem of identifying localizable nodes within neighborhoods. The protocol for network localizability is presented in Section III, as well as the correctness and optimality. Our prototype implementation and simulation are discussed in Section IV. We summarize related work in both localization and graph rigidity literature in Section V and conclude the work in Section VI.

II. Neighborhood Localizability

A. *Wheel Graph*

A wheel graph W_n is a graph with n vertices, formed by connecting a single vertex to all vertices of an $(n-1)$-cycle. The vertices in the cycle will be referred to as *rim vertices*, the central vertex as the *hub*, an edge between the hub and a rim vertex as a *spoke*, and an edge between two rim vertices as a *rim edge*. Fig. 3 shows a particular realization of a wheel graph W_6, in which node 0 is the hub and others are rims.

The wheel graph has many good properties. From the standpoint of the hub vertex, all elements, including vertices and edges, are in its one-hop neighborhood, which indicates that the wheel structure is fully included in the neighborhood graph of the hub vertex.

Furthermore, wheel graphs are important for localizability because they are globally rigid in 2D space.

Lemma 1: The wheel graph W_n is globally rigid.

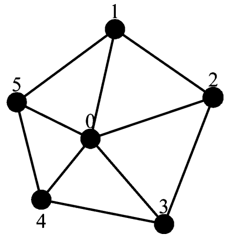


Fig. 3. A wheel graph W_6.

Proof: The graph W_n is redundantly rigid and 3-connected. Accordingly, it is globally rigid. ■

Thus, all vertices in a wheel structure with three beacons are uniquely localizable, which indicates an approach to identify localizable vertices. Realizing nodes in general wheel graphs is NP-hard [3], [23]. However, for sensor networks, the degree d of a node cannot be arbitrarily high since distance measurements, as well as communication links, only exist between nearby nodes. Therefore, the coordinates of nodes can be calculated by bilateration that examines the location space of at most $O(2^d)$ possible locations, where d is bounded by a constant number.

B. Conditions for Node Localizability

In this section, we analyze the conditions for single-node localizability by using localized information. Note that the word "localized" refers to the knowledge of direct neighbors.

We define the distance graph G_N of a wireless ad hoc network. Each wireless communication device (e.g., laptop, RFID, or sensor node) is modeled as a vertex of G_N, and there is an unweighted edge connecting two vertices if the distance between them can be measured or if both of them are in known locations, e.g., beacon nodes.

The *closed neighborhood graph* of a vertex v, denoted by $N[v]$, is a subgraph of G_N containing only v and its one-hop (direct) neighbors and edges between them in G_N. We also define the *open neighborhood graph* $N(v)$, where $N(v)$ is obtained by removing v and all edges incident to v from $N[v]$. Note that $N[v]$ is the local information known by a vertex v.

According to the previous analysis, if a vertex in $N[v]$ is included in a wheel graph centered at v, it is localizable by given three beacons. The localizability issue now can be transformed to finding wheel vertices in $N[v]$ when given a number of known localizable vertices.

We first consider the presence of three localizable vertices in $N[v]$. There are two cases of their distribution: 1) the hub v and two rim vertices; 2) three rim vertices. In the second case, v can be easily localized by trilateration. As a result, this case degenerates to the first one. We thus focus on the first case in the following analysis. Assume the two rim localizable vertices are v_1 and v_2.

To show that a vertex x belongs to a wheel structure in $N[v]$ centered at v and including two vertices v_1 and v_2, it is equivalent to show that x lies on a cycle containing v_1 and v_2 in $N(v)$. Accordingly, we turn to find whether a given group of three vertices (x, v_1, and v_2) are on a cycle in $N(v)$. According to Dirac's result [26], if a graph G is 3-connected, for any three vertices in G, G has a cycle including them. Therefore, if $N(v)$ is 3-connected, all vertices are included in some wheels in $N[v]$. The requirement of 3-connectivity, however, is too critical to be realistic and not necessary indeed.

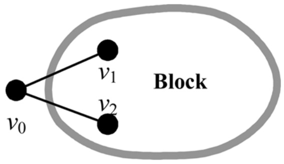


Fig. 4. The construction of B'.

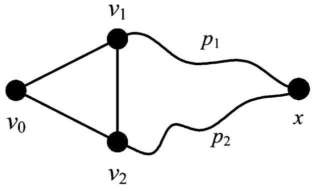


Fig. 5. x has two disjoint paths to v_0.

As we know, $N(v)$ is a distance graph in which there is an edge (x, y) if the distance between two vertices x and y is known. Thus, the edge (v_1, v_2) should exist in $N(v)$ since v_1 and v_2 are known as localizable. This observation helps to relax the connectivity requirement to 2-connectivity. As we know, a 2-connected component in a graph G is a maximal subgraph of G without any *articulation* vertex whose removal will disconnect G. For simplicity, we use *blocks* to denote 2-connected components henceforth so no confusion is caused.

Lemma 2: In a graph G with an edge (v_1, v_2), a vertex x belongs to the block B including v_1 and v_2 if and only if it is on a cycle containing v_1 and v_2.

Proof: Sufficiency. The graph B', as shown in Fig. 4, is constructed by adding a vertex v_0 and two edges (v_0, v_1) and (v_0, v_2) to B. We show that B' is also a block by the fact that the removal of any vertex cannot disconnect B'. There are two cases: 1) if v_0 is removed, the remaining graph, actually B, is connected definitely; 2) if a vertex in B is removed, the remaining vertices originally in B are still connected because B is 2-connected and v_0 is connected by either v_1 or v_2. Thus, B' is a block, and there are at least two vertex-disjoint paths between any two vertices. Suppose the two disjoint paths connecting a vertex x and v_0 are p_1 and p_2, illustrated in Fig. 5. Then, x is on a cycle in B' by simply cascading p_1 and p_2. Due to the construction of B', we can replace two consecutive edges (v_0, v_1) and (v_0, v_2) in the cycle by a shortcut (v_1, v_2), resulting in another cycle containing x, v_1, and v_2 in B.

Necessity. Suppose to the contrary that a vertex x is on a cycle containing v_1 and v_2, but is not included in B. We construct B' by adding the cycle to B; specifically, add all vertices and edges of the cycle to B if they are not in B originally, as illustrated in Fig. 6. There is no articulation vertex in B', and B' is also 2-connected. According to the construction of B', at least x is a newly introduced element, which indicates B is properly included by B', contradicting the maximality assumption of blocks. ■

According to Lemma 2, it follows a more general conclusion.

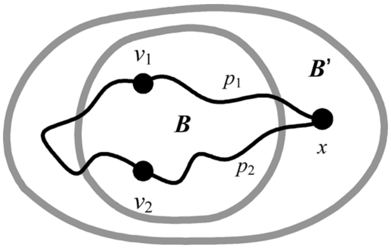


Fig. 6. B is properly included in B'.

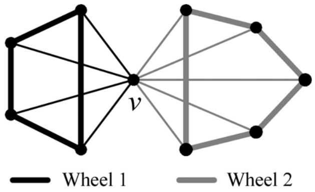


Fig. 7. Two wheels centered at v.

Lemma 3: If a graph G is 2-connected, then G' is globally rigid, where G' is obtained by adding a vertex v_0 and edges between v_0 to all vertices in G.

Proof: We take an arbitrary edge (v_1, v_2) in G. Since G is 2-connected, every other vertex x in G is on a cycle containing v_1 and v_2 by Lemma 2 and further belongs to a globally rigid wheel structure in G' including v_0, v_1, and v_2. Since every wheel in G' shares three vertices, all vertices are actually in the only one globally rigid component. ■

Using Lemmas 2 and 3, the wheel vertices can be identified by calculating blocks in neighborhood graphs. Note that not all blocks in $N(v)$ are localizable. As shown in Fig. 7, two wheels centered at v are not rigid to each other. Indeed, localizability also depends on the distribution of beacons. As we know, beacons are fully connected and entirely included in a block. Based on this, we propose a sufficient and necessary condition to find wheel vertices.

Theorem 1: In a neighborhood graph $N[v]$ with k ($k >= 3$) localizable vertices v_i ($i = 1, \dots, k$ and $v = v_k$), a vertex (other than v_i) belongs to a wheel structure with at least three localizable vertices if and only if it is included in the only (unique) block of $N(v)$ that contains $k - 1$ localizable vertices.

Proof: Sufficiency. If a vertex x belongs to a wheel with three localizable vertices in $N[v]$, it is on a cycle in $N(v)$ containing at least 2 localizable vertices, say v_1 and v_2. According to Lemma 2, x is included in the block of v_1 and v_2, which actually contains all $k - 1$ localizable vertices.

Necessity. If a vertex x is included by the block of localizable vertices in $N(v)$ (let v_1 and v_2 denote two of them), then x, v_1, and v_2 are on a cycle because (v_1, v_2) in $N(v)$. By adding v_k back, x belongs to the corresponding wheel with three localizable vertices in $N[v]$. ■

So far, we achieve a necessary and sufficient condition for finding localizable vertices. In addition, we can see that the trilateration is a special case of wheel graphs. Suppose a vertex v is localized by trilateration based on three reference nodes. In $N[v]$, these reference nodes are pairwise-connected because they are localizable. Thus, v is the hub vertex of the wheel where three references are the rim vertices. Trilateration is actually the minimum wheel graph with four vertices.

C. Algorithm and Correctness

According to Theorem 1, finding wheel vertices can be implemented by calculating blocks. Suppose there are k localizable vertices in a neighborhood graph $N[v]$.

Algorithm 1: Node Localizability

```
if $k >= 3$, then
    find all blocks in $N(v)$, denoted by
        $B_i, i = 1, \dots, m$; let $B_1$ be the unique one
        of localizable nodes;
    for each vertex $x$ not being marked in $B_1$
        mark $x$ localizable;
        connect $x$ to all other localizable ones;
    end for
end if
```

The core part of Algorithm 1 is to find blocks in a graph $G = (V, E)$. This can be done by depth-first search in linear time in terms of the size of graphs. Hence, the time complexity of Algorithm 1 is $O(|V| + |E|)$.

Algorithm 1 is designed to find wheel vertices in $N[v]$ that are localizable by Theorem 1. The remaining question is whether Algorithm 1 finds all localizable vertices in $N[v]$. In other words, is there any localizable vertex that is not included by any wheel in $N[v]$? In the following, we prove that, as expected, Algorithm 1 finds all localizable vertices in $N[v]$.

Lemma 4: [5] (Necessary condition for node localizability) In a graph G, if a vertex is uniquely localizable, it must have three vertex-disjoint paths to three distinct localizable vertices.

Theorem 2: (Correctness) In a neighborhood graph $N[v]$, a vertex is marked by Algorithm 1 if and only if it is uniquely localizable in $N[v]$.

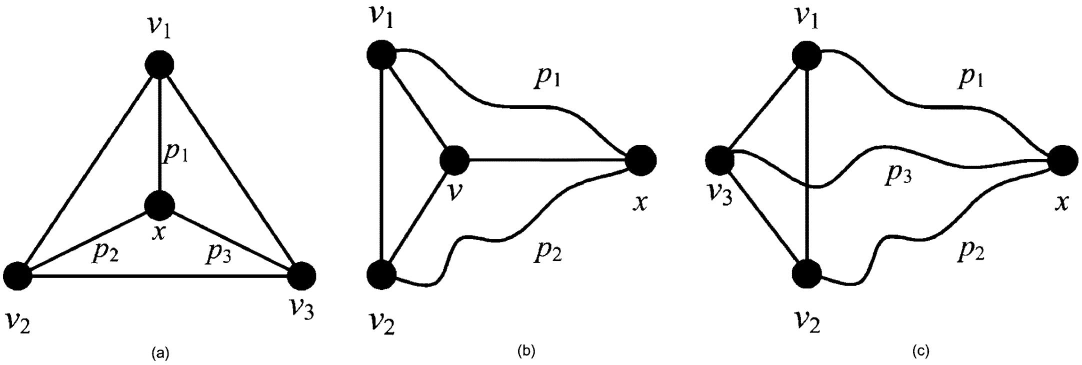
Proof: Sufficiency. Algorithm 1 finds wheel structures with at least three beacons in $N[v]$. According to Lemma 1, all vertices belonging to these wheels are localizable.

Necessity. If a vertex x is localizable in $N[v]$, by Lemma 4, it has three disjoint paths p_i to three distinct known localizable vertices $v_i, i = 1, 2, 3$, respectively. All v_i are connected with each other in $N[v]$. As illustrated in Fig. 8, there are three cases: 1) x is the hub vertex v, then it is in the wheel in which all v_i construct the rim cycle; 2) v is one of v_i (without loss of generality, assume v_3), then x is included in a wheel graph centered at v and having the rim cycle cascading $p_1, (v_1, v_2)$, and p_2; 3) x is on a cycle by cascading $p_1, (v_1, v_3), (v_3, v_2)$, and p_2. This is a simple cycle because v_3 cannot be in p_1 and p_2 due to the separation of p_i. Therefore, in all cases, x is included in a wheel graph in $N[v]$ and marked by Algorithm 1. ■

Theorem 2 also guarantees the optimality of Algorithm 1 since it finds the maximum number of localizable vertices in $N[v]$.

III. Network-Wide Localizability

The previous section discusses the localizability in neighborhood graphs. Now, we consider the localizability for entire net-



(a) (b) (c)

Fig. 8. x belongs to a wheel structure. (a) Case 1. (b) Case 2. (c) Case 3.

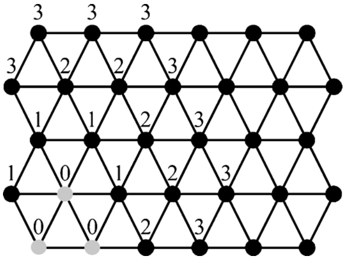


Fig. 9. A wheel extension graph. Here, the gray nodes are beacons. An edge denotes that the distance between the two end-nodes is known.

works. We call this problem the network-wide localizability test so as to distinguish with the case of a single node.

A. Wheel Extension

Similar to the trilateration extension, we first define the wheel extension.

Definition 1: A graph G is a *wheel extension* if there are the following:

a) three pairwise-connected vertices, say v_1, v_2, and v_3; and
b) an ordering of remaining vertices as $v_4, v_5, v_6 \ldots$, such that any v_i is included in a wheel graph (a subgraph of G) containing three early vertices in the sequence.

Lemma 5: The wheel extension is globally rigid.

The proof of Lemma 5 is straightforward, so we skip it. The family of wheel extensions is actually a superset of trilateration extensions. Fig. 9 shows an example that is a wheel extension but not a trilateration extension. The node deployment in Fig. 9 is classical and often used to analyze coverage and connectivity problems in which location is critical.

B. Localizability Protocol

For localizability, it is important to know whether a graph is a wheel extension. In this section, we present a distributed protocol that tests the localizability by marking all localizable nodes in a network. The protocol works in an iterative manner in which a node marked in the current iteration acts as a known localizable one (or beacon) in subsequent iterations. Localizability information diffuses step by step and reaches the entire network after a number of iterations.

A particular iterative process is shown in Fig. 9. First, three beacons are given and marked with 0. In the first iteration, nodes marked 1 are identified because they are included in a wheel graph with three beacons. Such a procedure continues until all localizable nodes are marked.

The localizability protocol is given in Algorithm 2, which is conducted in a distributed manner at each node. If all nodes in a network are marked by Algorithm 2, the network graph is a wheel extension; and vice versa.

Algorithm 2: Network Localizability

```
 1: exchange neighbor list between neighbors;
 2: construct $N[v]$;
 3: if $N[v]$ has $>=$ 3 localizable nodes
 4:     run Algorithm 1 on $N(v)$, obtaining a number of
        blocks $B_i$; (Assume $B_1$ is the unique localizable
        one)
 5:     mark $v$ and $B_1$ localizable;
 6:     inform $B_1$ the change;
 7:     update $N(v)$;
 8: end if;
 9: while(true)
10:     wait for state change of neighbor nodes;
11:     update $N(v)$;
12:     if any nonmarked $B_i$ has $>=$ 2 localizable nodes
13:         mark $B_i$ localizable;
14:         update $N(v)$;
15:         inform $B_i$ the change;
16:     end if
17: end while
```

We now analyze the time complexity of Algorithm 2 running on a graph G with n vertices. Since Algorithm 1 is only executed on the vertices with at least three localizable ones in $N[v]$, these vertices are localizable and will be finally marked by Algorithm 2. Therefore, the running time of Algorithm 2 is output-sensitive. In the worst case, Algorithm 1 will be executed in all vertices in G. Let $d(v)$ denote the degree of a vertex v. In line 2, calculating blocks in $N(v)$ costs $O(d(v)^2)$ time in dense graphs or $O(d(v))$ in sparse graphs. In the while loop between lines 3–11, at most $d(v)$ neighbors are marked and informed. Hence, the total running time of Algorithm 2 is $\sum_{v \in G} O(d(v)^2 + d(v)) = O(n^3)$ in dense graphs and

$\sum_{v \in G} O(d(v)) = O(n)$ in sparse graphs. The bound is tight due to the instance of $G = K_n$, where K_n is the complete graph of n vertices.

In practice, a wireless ad hoc network cannot be excessively dense because the communication links only exist between nearby nodes due to signal attenuation. In addition, the mechanism of topology control reduces redundant links to alleviate collision and interference. Hence, the proposed algorithm is practically efficient.

C. Correctness and Optimality

To analyze the correctness of Algorithm 2, we first define the concept of k-hop localizability.

Definition 2: In a network, a node is k-hop localizable if it can be localized by using only the information of at most k-hop neighbors.

Clearly, one-hop localizable is the most critical condition for all k, and the set of k-hop localizable nodes is monotonically increasing.

Theorem 3: In a graph G, a vertex is marked by Algorithm 2 if and only if it is one-hop localizable in G.

Proof: Sufficiency. This part holds because Algorithm 2 marks a vertex if it is in a one-hop wheel with three localizable nodes.

Necessity. If a vertex x is one-hop localizable, it is included in a wheel with three localizable nodes by Theorem 2. The hub vertex, which may be x or not, certainly knows these three localizable nodes, thus x will be marked by Algorithm 2 when Algorithm 1 is executed on the hub vertex. ■

Theorem 3 not only guarantees the correctness of Algorithm 2, but also indicates the set of localizable nodes is not dependent on the ordering of node processing.

D. Advantages

Compared to the previous trilateration-based methods, the advantages of the proposed protocol lie in the following.

1) Capability: WHEEL can recognize a superset of localizable nodes. Furthermore, it is optimal and achieves the theoretical upper bound.
2) Efficiency: WHEEL takes $O(n)$ running time for sparse graphs and $O(n^3)$ for dense ones.
3) Low cost: Using only localized information, WHEEL introduces no extra wireless communication cost.

IV. Performance Evaluation

A. Prototype Implementation

The localizability protocol is implemented on the hardware platform of the OceanSense project [1], [27], as shown in Fig. 10. We launched a working prototype sensor network consisting of tens of nodes that float on the surface of the sea and collect scientific data such as sea depth, ambient illumination, pollution, etc. Localization is one of the most important issues in the project since sensing data without locations is almost meaningless. The system also collects the network topology that is highly dynamic under natural conditions due to ocean current, wind blow, tide, etc.

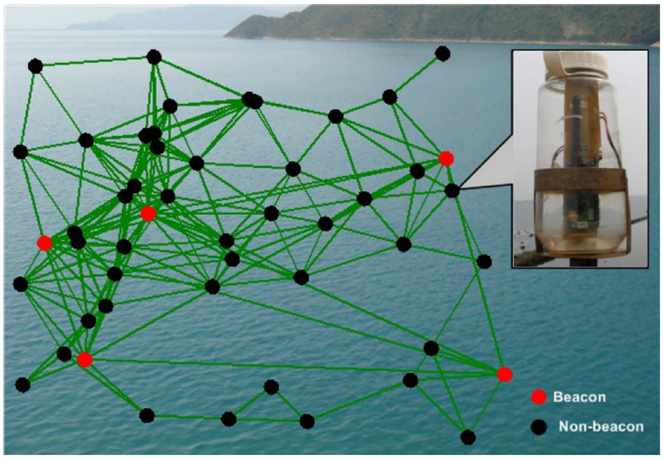


Fig. 10. System deployment. The upper right figure shows an encapsulated waterproof sensor mote.

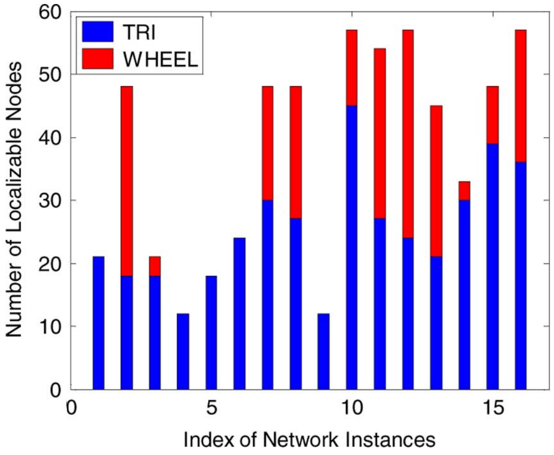


Fig. 11. Prototype Performance. In nearly 70% cases, WHEEL outperforms TRI by recognizing a larger number of localizable nodes.

We equip five out of 60 nodes with GPS receivers and adopt the RSS-based ranging technique. Based on distance ranging, the proposed WHEEL protocol is employed for testing localizability. In fact, WHEEL does not rely on any particular ranging techniques and works properly with RSS, TOA, TDOA, etc. We collect a number of instances of the network topology from 8-h observation. For comparison, we also calculate the theoretical upper bound of all trilateration-based approaches. Many variations of trilateration (a.k.a. multilateration) have been proposed [13], [18], [28], aiming at improving the localization accuracy. They mainly focus on how to deal with ranging noises or systematic errors and select references with many other concerns. That is to say, in some cases, they opt to give up locating some nodes with possibly inaccurate location estimate. Therefore, the basic trilateration can locate the most number of nodes among all its variations. Hence, trilateration (TRI) is chosen as a representative of all trilateration-based approaches.

The experiment results are plotted in Fig. 11, in which the dark bars denote the number of nodes localized by TRI; while lighter ones denote the nodes that can be identified by WHEEL

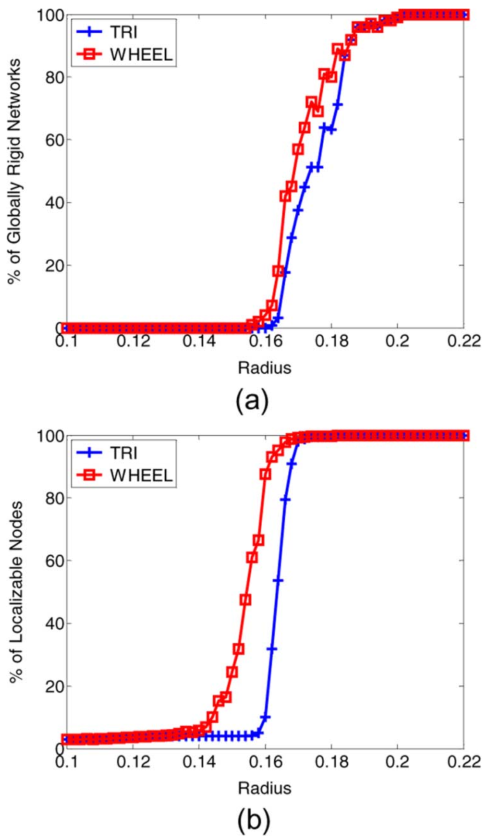


Fig. 12. Comparison of TRI and WHEEL. (a) Percentage of globally rigid graphs. (b) Percentage of localizable nodes.

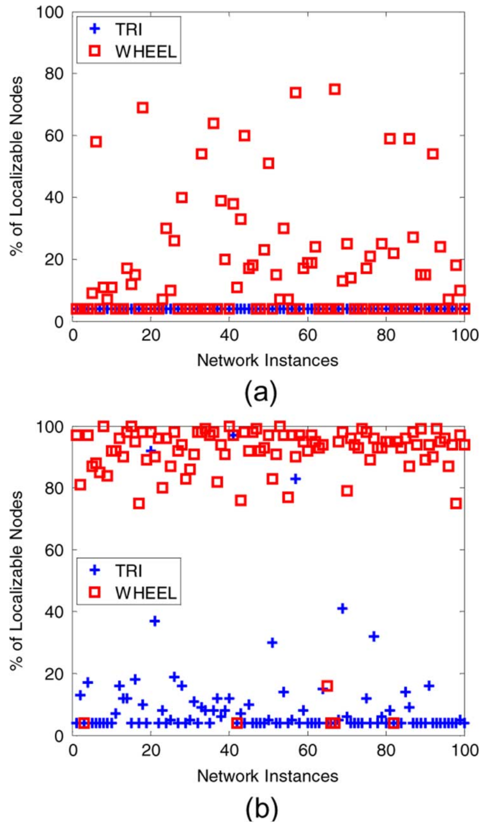


Fig. 13. Comparison of TRI and WHEEL (2). (a) Radius $r = 0.15$. (b) Radius $r = 0.16$.

but not TRI. Among all 16 network topologies, 11 of them obtain notable improvements by using WHEEL to recognize more localizable nodes.

B. Large-Scale Simulation

Large-scale simulations are further conducted to examine the effectiveness and scalability of this design under varied network parameters.

We generate networks of 400 nodes randomly, uniformly deployed in a unit square $[0, 1]^2$. The unit disk model with a radius is adopted for communication and distance ranging. For each evaluation, we integrate results from 100 network instances.

We explore the impact of network topology on localizability. As shown in Fig. 12(a), for both strategies, the percentage of globally rigid networks grows along with the increasing communication radius. Note that the transition phenomena appear again at the radii around 0.16. It can be seen that WHEEL provides a smaller hitting radius than TRI, which exhibits a strong applicability of WHEEL since it can work well in relatively low-density or sparsely connected networks.

Such a conclusion becomes obvious for the number of localizable nodes in partially localizable networks, as shown in Fig. 12(b). It studies the capability of recognizing localizable nodes in a partially localizable network. We can see that WHEEL remarkably surpasses TRI. At radius 0.158, 90% of localizable nodes are identified by WHEEL, while TRI only marks 5% under the same network settings.

We also study the performances of TRI and WHEEL at some specific communication radii. In this evaluation, the number of recognized localizable nodes of 100 network instances is shown in Fig. 13(a) and (b) with radius $r = 0.15$ and $r = 0.16$, respectively. As shown in Fig. 13(a), WHEEL identifies 27% of nodes as localizable, while TRI cannot work at all due to the sparse network connectivity. When $r = 0.16$, WHEEL recognizes more than 90% localizable nodes in 73 cases, while TRI only mark less than 10% localizable nodes in 77 cases. The observation supports the conclusion that at a specific range of communication radius (or connectivity), WHEEL remarkably outperforms TRI.

We further provide two examples to show how WHEEL outperforms TRI. In Fig. 14, a particular network with an "H" hole is generated, in which 400 nodes are randomly distributed. The dark gray dots denote the nodes marked by TRI by given three beacons, while light gray dots denote the nodes marked by WHEEL but not by TRI. Neither TRI nor WHEEL can mark the remaining black dots. WHEEL can easily step over gaps, such as borders or barriers, and recognize more nodes than TRI. The same phenomenon appears in another network instance with a "K" hole, as shown in Fig. 15. We conducted more simulations, and the results are consistent.

V. Related Work

A. Localization Literature

Existing localization approaches for wireless ad hoc networks fall into two categories. Range-based approaches [19], [21], [22]

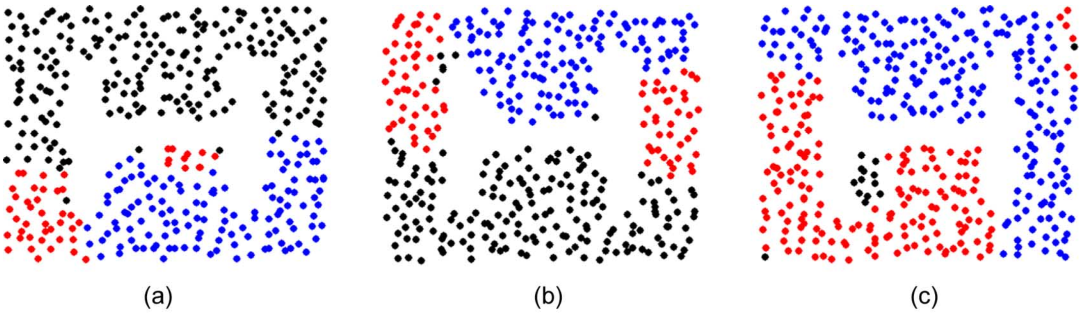


Fig. 14. Networks with "H" holes. (a) Case 1. (b) Case 2. (c) Case 3.

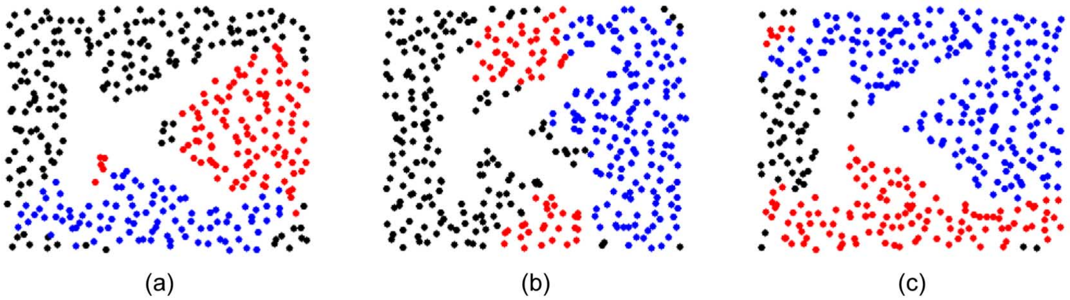


Fig. 15. Networks with "K" holes. (a) Case 1. (b) Case 2. (c) Case 3.

assume that nodes are able to measure the distances or the relative directions of neighbor nodes, while range-free approaches [11], [12], [25] do not assume such special hardware functionality, and each node merely knows the existence of its neighbors.

Many localization algorithms are range-based and adopt distance-ranging techniques, such as RSS [24] and TDoA [21]. RSS maps received signal strength to distance according to a signal attenuation model, while TDoA measures the signal propagation time for distance calculation. In practice, RSS-based ranging measurements contain noise on the order of several meters [2], especially in rigorous environments. On the contrast, TDoA is impressively accurate and obtains close to centimeter accuracy for node separations under several meters in indoor environments [21], [22].

The majority of localization algorithms [13], [18], [22], [28] assume a dense network such that iterative trilateration (or multilateration) can be carried out. To deal with network sparseness, Sweeps [4] record all possible locations in each positioning step and prune incompatible ones whenever possible. It is highly capable and exceeds other localization approaches in terms of the number of nodes that can be located. Similar to Sweeps, WHEEL uses bilateration as the basic positioning technique. The proposed WHEEL differs from Sweeps as follows: 1) Sweeps is a centralized algorithm that requires the entire network topology, while WHEEL utilizes only the distance measurements within a one-hop neighborhood; 2) the computational cost of Sweeps grows exponentially to the network size in worst cases; however, WHEEL is a polynomial-time algorithm that takes $O(n^3)$ time in dense networks ($O(n^2)$ links) and $O(n)$ time in sparse networks ($O(n)$ links); 3) WHEEL is able to locate all one-hop localizable nodes, providing a tight theoretical upper bound of all distributed localization approaches.

The focus of this paper is range-based localization in which the ground truth of network deployments can be modeled by distance graphs.

B. *Graph Rigidity Literature*

In graph rigidity literature, many efforts have been made to explore the combinatorial conditions for rigidity. Laman [10] first pointed out that a graph $G(V, E)$ is generically rigid if it has a induced subgraph in which edges are "independently" distributed. The statement also leads to an $O(|V|^2)$ algorithm [9] for rigidity test. For global rigidity, a sufficient and necessary condition [7] is presented based on the results in [6] by combining both redundant rigidity and 3-connectivity. Recently, Jackson and Jordan [8] prove a sufficient condition of 6-mixed connectivity, which improves a previous result of 6-connectivity by [16].

VI. Conclusion

Trilateration, as a basic building block of many existing localization approaches, often wrongly recognizes localizable graphs as nonlocalizable. To address the issue, we analyze the limitation of trilateration-based approaches and propose a novel approach, called WHEEL, based on globally rigid wheel graphs. This design inherits the simplicity and efficiency of trilateration while, at the same time, significantly improving the performance by identifying more localizable nodes. To validate this approach, a prototype system with tens of wireless sensors is deployed. Large-scale simulations are further conducted to evaluate the scalability and efficiency. Experimental results show that WHEEL greatly outperforms previous approaches. Such improvements, however, are observed from intensive simulations. It is still the lack of theoretical analyses of the gap between WHEEL and trilateration, as well as the gap between WHEEL and the theoretical upper bound with global information, that is a direction of our future studies. We also plan to explore how localizability aids network functions, such as topology control [15], mobility control, network diagnosis [14], etc.

References

[1] OceanSense Project, 2010 [Online]. Available: http://www.cse.ust.hk/~liu/Ocean/index.html

[2] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, 2000, vol. 2, pp. 775–784.

[3] T. Eren, D. K. 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. IEEE INFOCOM*, 2004, vol. 4, pp. 2673–2684.

[4] D. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A. S. Morse, and Y. R. Yang, "Localization in sparse networks using sweeps," in *Proc. ACM MobiCom*, 2006, pp. 110–121.

[5] D. Goldenberg, A. Krishnamurthy, W. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides, and B. Anderson, "Network localization in partially localizable networks," in *Proc. IEEE INFOCOM*, 2005, vol. 1, pp. 313–326.

[6] B. Hendrickson, "Conditions for unique graph realizations," *SIAM J. Comput.*, vol. 21, no. 1, pp. 65–84, 1992.

[7] B. Jackson and T. Jordan, "Connected rigidity matroids and unique realizations of graphs," *J. Combin. Theory Ser. B*, vol. 94, no. 1, pp. 1–29, 2005.

[8] B. Jackson and T. Jordan, "A sufficient connectivity condition for generic rigidity in the plane," Operations Research Department, Eotvos University, Budapest, Turkey, TR-2008-01, 2008.

[9] D. J. Jacobs and B. Hendrickson, "An algorithm for two-dimensional rigidity percolation: The pebble game," *J. Comput. Phys.*, vol. 137, pp. 346–365, 1997.

[10] G. Laman, "On graphs and rigidity of plane skeletal structures," *J. Eng. Math.*, vol. 4, pp. 331–340, 1970.

[11] M. Li and Y. Liu, "Rendered path: Range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 320–332, Feb. 2010.

[12] H. Lim and J. C. Hou, "Distributed localization for anisotropic sensor networks," *ACM Trans. Sensor Netw.*, vol. 5, no. 2, 2009, Article no. 11.

[13] J. Liu, Y. Zhang, and F. Zhao, "Robust distributed node localization with error management," in *Proc. ACM MobiHoc*, 2006, pp. 250–261.

[14] K. Liu, M. Li, Y. Liu, M. Li, Z. Guo, and F. Hong, "Passive diagnosis for wireless sensor networks," in *Proc. ACM SenSys*, 2008, pp. 371–372.

[15] Y. Liu, Q. Zhang, and L. M. Ni, "Opportunity-based topology control in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 3, pp. 405–416, Mar. 2010.

[16] L. Lovasz and Y. Yemini, "On generic rigidity in the plane," *SIAM J. Algebr. Discrete Methods*, vol. 3, no. 1, pp. 91–98, 1982.

[17] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, pp. 2529–2553, 2007.

[18] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. ACM SenSys*, 2004, pp. 50–61.

[19] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proc. IEEE GLOBECOM*, 2001, vol. 5, pp. 2926–2931.

[20] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *J. Telecommun. Syst.*, vol. 22, no. 1–4, pp. 267–280, 2003.

[21] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," in *Proc. ACM MobiCom*, 2000, pp. 32–43.

[22] A. Savvides, C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. ACM MobiCom*, 2001, pp. 166–179.

[23] J. B. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," in *Proc. Allerton Conf. Commun., Control Comput.*, 1979.

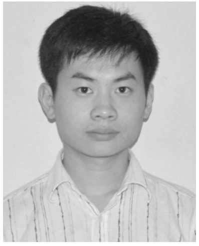
[24] S. Y. Seidel and T. S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Trans. Antennas Propag.*, vol. 40, no. 2, pp. 209–217, Feb. 1992.

[25] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 961–974, Nov. 2004.

[26] D. B. West, *Introduction to Graph Theory*, 2nd ed. Englewood Cliffs, NJ: Pearson, 2001.

[27] Z. Yang, M. Li, and Y. Liu, "Sea depth measurement with restricted floating sensors," in *Proc. IEEE RTSS*, 2007, pp. 469–478.

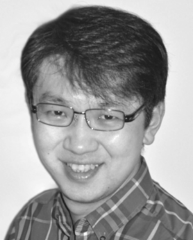
[28] Z. Yang and Y. Liu, "Quality of trilateration: Confidence based iterative localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 631–640, May 2010.



Zheng Yang (S'06) received the B.Eng. degree in computer science from Tsinghua University, Beijing, China, in 2006. He is currently a Ph.D. student with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong.

His main research interests include wireless ad hoc/sensor networks and pervasive computing.

Mr. Yang is a Student Member of the Association for Computing Machinery (ACM).



Yunhao Liu (SM'06) received the B.S. degree in automation from Tsinghua University, Beijing, China, in 1995, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, East Lansing, in 2003 and 2004, respectively.

He is now with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. He is also a member of the Tsinghua EMC Chair Professor Group. His research interests include peer-to-peer computing, pervasive computing, and sensor networks.

Dr. Liu is a Member of the Association for Computing Machinery (ACM).

Xiang-Yang Li (SM'09) received the B.Eng. degree in computer science and the Bachelor's degree in business management from Tsinghua University, Beijing, China, in 1995, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 2000 and 2001, respectively.

He has been an Associate Professor since 2006 and Assistant Professor of Computer Science at the Illinois Institute of Technology from 2000 to 2006. His research interests span wireless ad hoc and sensor networks, noncooperative computing, computational geometry, and algorithms.

Dr. Li is a Member of the Association for Computing Machinery (ACM). He is an Editor of the IEEE Transactions on Parallel and Distributed Systems.