

Beyond Trilateration: On the Localizability of Wireless Ad-hoc Networks

Zheng Yang, Yunhao Liu
 Hong Kong University of Science and Technology
 {yangzh, liu}@cse.ust.hk

Xiang-Yang Li
 Illinois Institute of Technology
 xli@cs.iit.edu

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 non-localizable. In this study, we analyze the limitation of trilateration based approaches and propose a novel approach which inherits the simplicity and efficiency of trilateration, while 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 1-hop localizable nodes. To validate this approach, a prototype system with 19 wireless sensors is deployed. Intensive and large-scale simulations are further conducted to evaluate the scalability and efficiency of our design.

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 medicare, smart space, wireless sensor network surveillance, mobile peer-to-peer computing, etc., location is one of the most essential contexts.

Many methods have been proposed in the literature and used in practice to localize wireless devices. One method to determine the location of a device is manual configuration, which may not be feasible for large-scale deployments or mobile systems. Another popular system, Global Positioning System (GPS), is not suitable for indoor environments and suffers high hardware cost.

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 ones determine their locations by measuring the Euclidean distances to their neighbors. Several distance ranging methods, such as Radio Signal Strength (RSS) [22] and Time Difference of Arrival (TDoA) [19], 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 $G = (V, E, d)$, where V is the set of wireless nodes, E is the set of links, and $d(u, v)$ denotes the distance measurements between a pair of nodes u and v . 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 some beacon nodes) is called *localizable* if there

is a unique location v' of every node v such that the distance $d(u, v) = d(u', v')$ for all links in E and constraint I is preserved.

Finding the location of all nodes that respect the given distance measurements d and constraint I is also called *graph embedding* in the Euclidean space. Previous studies have shown that localizability problem is closely related to the graph rigidity [4, 8, 9, 12]. 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 [12]. 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 [8]. Indeed, the solvability of network localization is equivalent to the global rigidity property of graphs [4, 8]. Certainly, rigidity is a necessary condition of global rigidity. Figure 1 shows an example of rigid graphs which is not globally rigid by having two distinct embeddings in 2D plane. In this case, node 4 can “flip” above nodes 1 and 2 while still preserving distance.

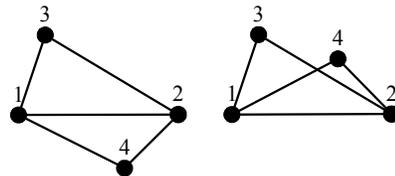


Figure 1: A rigid graph with two distinct embeddings.

Jackson et al. [9] 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. This statement provides a sufficient and necessary condition for testing global rigidity. Hence, localizability of a graph can be answered in polynomial time in a centralized manner by testing the 3-connectivity and redundant rigidity. Designing an efficient distributed algorithm for global rigidity, however, is non-trivial as neither connectivity nor rigidity can be tested locally by nature. For example, Figure 2 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) 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.

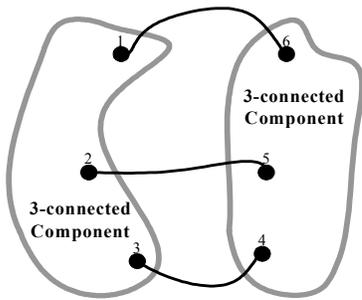
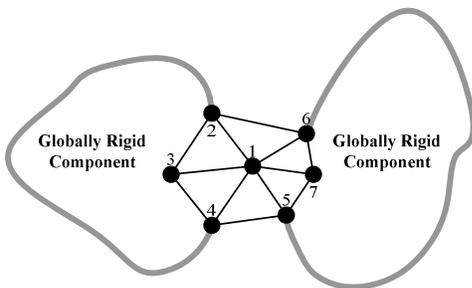


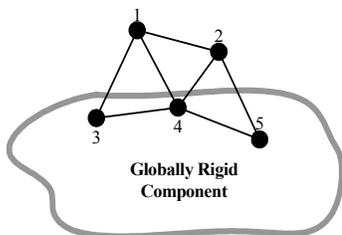
Figure 2: Global information is needed to test connectivity.

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 [16, 17, 20] as it is fully distributed, easy to implement, and efficient in terms of communication and computation.

Trilateration based approaches, however, recognize only a subset (called trilateration extension) of globally rigid graphs. In Figure 3(a), two globally rigid components are connected by nodes i ($i=1,2,\dots,7$). Suppose the nodes 1, 2, 3, and 4 in the left component are known as localizable. The localizability information, however, 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) Geographical gap.



(b) Border nodes.

Figure 3: The deficiency of trilateration.

A similar situation is recurrent for the border nodes, as illustrated in Figure 3(b). In this case, nodes 1 and 2 cannot be localized by trilateration even though nodes 3, 4, and 5 are localized. Unfortunately, the entire graph in Figure 3(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 testing that can recognize as many globally rigid networks as possible, or as many localizable nodes in a partially localizable network as possible. Our study shows that trilateration is actually a special case, the simplest with 4 nodes, of *wheel* graphs [23], 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 improves the performance by identifying more localizable nodes. We prove the optimality of this design: it is able to recognize all 1-hop localizable nodes using only local information.

We validate this design by deploying a prototype system with 19 TelosB sensor nodes. 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 literatures in Section V and conclude the work in Section VI.

II. NODE LOCALIZABILITY

A. The 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*. Figure 4 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.

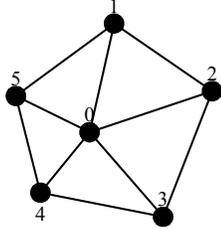


Figure 4: A wheel graph W_6 .

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

Lemma 1. [4] The wheel graph W_n is globally rigid.

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

Thus, all vertices in a wheel structure with 3 beacons are uniquely localizable, which indicates an approach to identify localizable vertices. Note that realizing graphs is still NP-hard even for globally rigid graphs [4, 21].

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 un-weighted edge connecting two vertices if the distance between them can be measured or 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 3 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. Without loss of generality, suppose 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 3 vertices (x , v_1 , and v_2) are on a cycle in $N(v)$. According to Dirac [3], 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.

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 release the connectivity requirement to 2-connectivity.

A 2-connected component in a graph G is a maximal subgraph of G without any *articulation* vertex whose removal will disconnects G . For simplicity, we use *blocks* to denote 2-connected components henceforth if no confusion caused.

Lemma 2. In a graph G with an edge (v_1, v_2) , any other 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 Figure 5, 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 Figure 6. 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 another cycle containing x , v_1 , and v_2 in B .

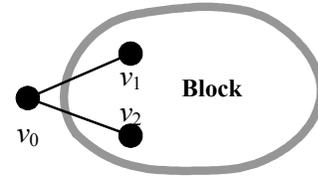


Figure 5: The construction of B' .

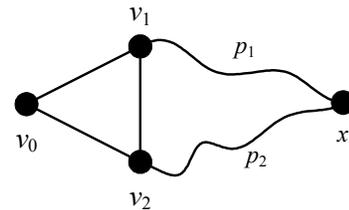


Figure 6: x has 2 disjoint paths to v_0 .

Necessity. Suppose to the contrary that a vertex x is on a cycle containing v_1 and v_2 but 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 Figure 7. 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. ■

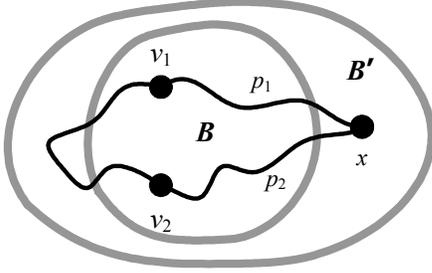


Figure 7: B is properly included in B' .

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

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 Lemma 2 and Lemma 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 Figure 8, 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 unique block. Based on this, we propose a sufficient and necessary condition to find wheel vertices.

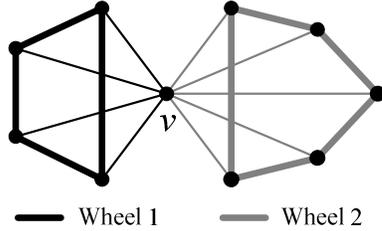


Figure 8: Two wheels centered at v .

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

Proof: Sufficiency. If a vertex x belongs to a wheel with 3 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 3 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 3 references are the rim vertices. Trilateration is actually the minimum wheel graph with 4 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

```

1: if  $k \geq 3$ , then
2:   find all blocks in  $N(v)$ ,
   denoted by  $B_i, i=1, \dots, m$ ;
   let  $B_1$  be the unique one of
   localizable nodes;
3:   for each vertex  $x$  not being
   marked in  $B_1$ 
4:     mark  $x$  localizable;
5:     connect  $x$  to all other
   localizable ones;
6:   end for
7: 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 that does Algorithm 1 find 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. [6] (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 3 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 Figure 9, 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

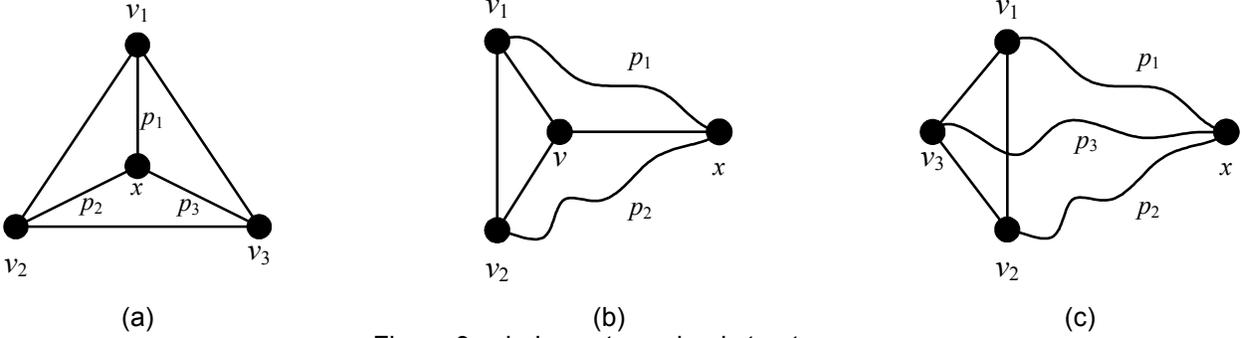


Figure 9: x belongs to a wheel structure.

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 node localizability in neighborhood graphs. Now, we consider the localizability for entire networks. We call this problem the network-wide localizability test so as to distinguish with the case of a single node.

A. The 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

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

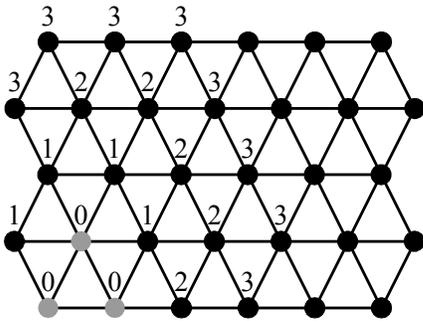


Figure 10: A wheel extension graph. Here the grey nodes are beacons; an edge denotes that the distance between the two end-nodes is known.

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. Figure 10 shows an example which is a wheel extension but not a trilateration extension. The node deployment in Figure 10 is classical and often used to analyze coverage and connectivity problems in which location is critical.

B. The Localizability Protocol

For localizability, it is important to know whether a graph is a wheel extension. In this section, we present a distributed protocol which 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 Figure 10. 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 3 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 non-marked Bi has >=2
     localizable nodes
13:     mark Bi localizable;
14:     update N(v);
15:     inform Bi 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 line 3 to 11, at most $d(v)$ neighbors are marked and informed. Hence, the total running time of Algorithm 2 are $\sum_{v \in G} O(d(v)^2 + d(v)) = O(n^3)$ 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, 1-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 marked by Algorithm 2 if and only if it is 1-hop localizable in G .

Proof: Sufficiency. This part holds because Algorithm 2 marks a vertex if it is in a 1-hop wheel with 3 localizable nodes.

Necessity. If a vertex x is 1-hop localizable, it is included in a wheel with 3 localizable nodes by Theorem 2. The hub vertex, may be x or not, certainly knows these 3 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 lies in:

- (1) Capability: recognizing a superset of localizable nodes.
- (2) Efficiency: taking $O(n)$ running time for sparse graphs and $O(n^3)$ for dense ones.
- (3) Low cost: introducing no extra wireless communication cost by using only localized information.

IV. PERFORMANCE EVALUATION

A. Prototype Implementation

The localizability protocol is implemented on the hardware platform of the OceanSense project [1, 25]. In this project, wireless sensors are deployed off the seashore. In our experiment, 19 TelosB nodes are distributed in a 200m*300m area, floating on the sea surface. Sensor movements, however, are restricted in a disk area because of anchors [25]. The network continuously reports back data as well as the network topology every 30 minutes.

In our experiments, we employ radio signal strength for distance ranging, such that the communication links can be viewed as a distance graph. Four out of 19 nodes are set as beacons and the remaining 15 nodes test localizability by carrying on the proposed protocol (WHEEL). We collect a number of instances of the network topology from 4-hour data. For comparison, we also calculate the theoretical upper bound (TRI) of all trilateration based approaches.

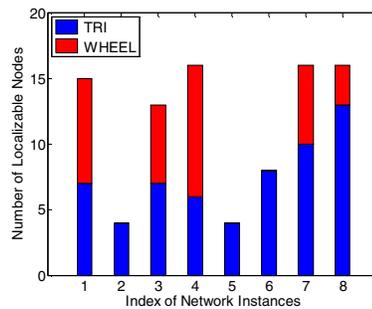


Figure 11: Prototype Performance.

The results are plotted in Figure 11, in which the blue bars denote the number of nodes localized by TRI; while red ones denote the nodes which can be identified by WHEEL but not TRI. Among all 8 network topologies, 5 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.

Figure 12 studies the relationship between connectivity and rigidity. The curve r_i denotes the percentage of i -connected networks in varied radius while r_g denotes globally rigid networks. Like many other properties for random geometric graphs, both connectivity and rigidity have transition phenomena. It can be seen that r_g lies between r_3 and r_6 and is closer to r_3 . This observation reflects the theoretical conclusion that 3-connectivity is a necessary condition while 6-connectivity is a sufficient one for global rigidity.

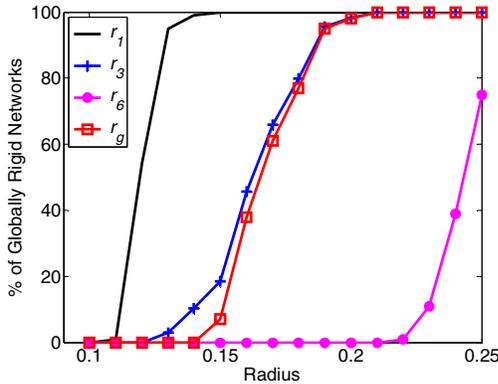
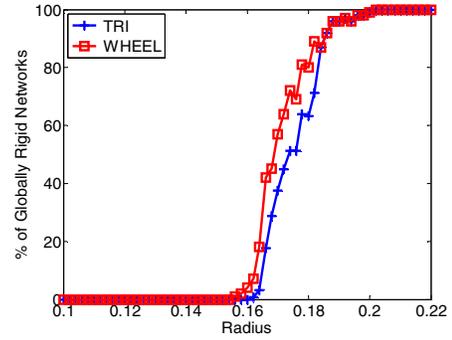


Figure 12: The relationship between connectivity and rigidity.

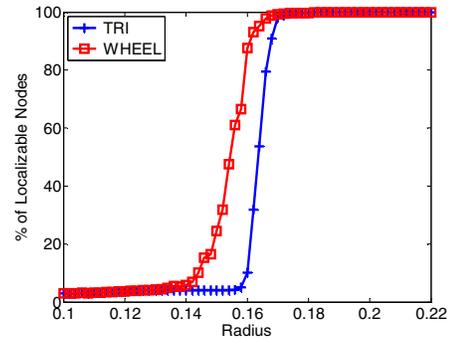
We then explore the impact of network topology on localizability. As shown in Figure 13 (a), for both strategies, the percentage of globally rigid networks grows along with the increasing of communication radius. Note that the transition phenomena appear again at the radii around 0.16. It can be seen 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 conclusion becomes obvious for the number of localizable nodes in partially localizable networks, as shown in Figure 13 (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, a 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 Figure 14(a) and (b) with radius $r = 0.15$ and $r = 0.16$, respectively. As shown in Figure 14(a), WHEEL identifies 27% 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.

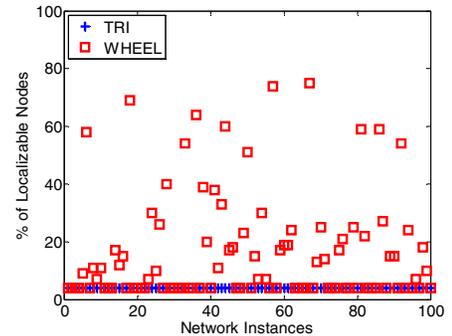


(a) Percentage of globally rigid graphs

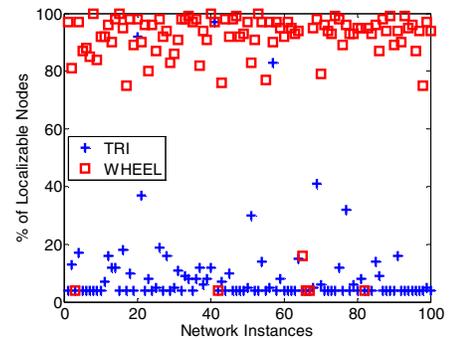


(b) Percentage of localizable nodes

Figure 13: Comparison of TRI and WHEEL.

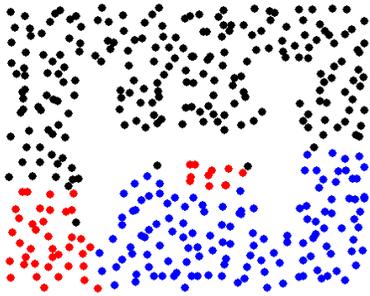


(a) Radius $r = 0.15$

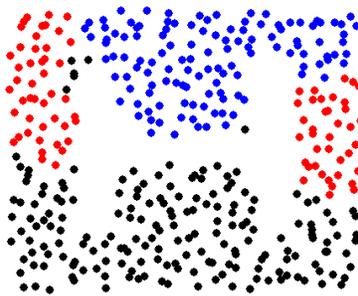


(b) Radius $r = 0.16$

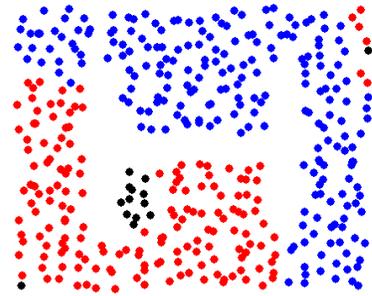
Figure 14: Comparison of TRI and WHEEL (2).



(a) Case 1

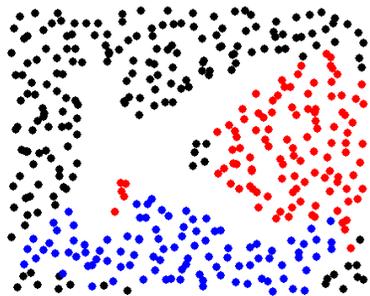


(b) Case 2

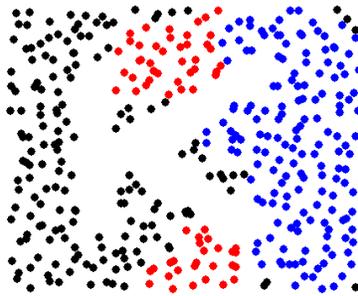


(c) Case 3

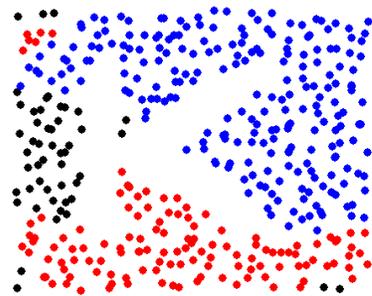
Figure 15: Networks with “H” holes.



(a) Case 1



(b) Case 2



(c) Case 3

Figure 16: Networks with “K” holes.

We further provide two examples to show how WHEEL outperforms TRI. In Figure 15, a particular network with an “H” hole is generated in which 400 nodes are randomly distributed. The blue dots denote the nodes marked by TRI by given three beacons, while reds denote the nodes marked by WHEEL but not by TRI. Neither TRI nor WHEEL can mark the remaining blacks. WHEEL can easily step over gaps, such as borders or barriers, and recognize more nodes than TRI. The same phenomenon recurs in the second network instance with a “K” hole, shown in Figure 16. We conducted more simulations and the results are consistent.

V. RELATED WORK

A. Localization Literature

Many localization algorithms adopt distance ranging techniques (called range-based), including Radio Signal Strength (RSS) [22] and Time Difference of Arrival (TDoA) [19]. RSS maps received signal strength to distance according to a signal attenuation model, while TDoA measures the signal propagation time for distance calculation. Based on them, localization is conducted by exploring rigid graph structures. Some works [4, 6] study the relationship between network localization and rigidity properties of ground truth graphs.

The majority of localization algorithms [2, 19, 20] assume a dense network such that iterative trilateration (or multilateration) can be carried out. Other methods [5] record all possible locations in each positioning step and prune

incompatible ones whenever possible, which, in the worst case, can result in an exponential space requirement. Recently, a method [13] of exploring rigid topology structure without distance is proposed which provides a novel view for range-free localization.

B. Graph Rigidity Literature

In graph rigidity literature, many efforts have been made to explore the combinatorial conditions for rigidity. Laman [12] 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 [11] for rigidity test. For global rigidity, a sufficient and necessary condition [9] is presented based on the results in [8] by combining both redundant rigidity and 3-connectivity. Recently, Jackson and Jordan [10] prove a sufficient condition of 6 mixed connectivity, which improves a previous result of 6-connectivity by [15].

There are also some results for random geometric graphs. Assuming the unit disk model, many researchers [7, 14, 18, 24] considered critical conditions for graph connectivity. Simulation results [4] ensure that the hitting radius of global rigidity is between 3- and 6-connectivity in probability sense. In addition, the asymptotic hitting radius for trilateration graphs is also given in [4].

VI. CONCLUSION

Trilateration, as a basic building block of many existing localization approaches, often wrongly recognize localizable

graphs as non-localizable. 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 improves the performance by identifying more localizable nodes. To validate this approach, a prototype system with 19 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 lack of theoretical analyses of the gap between WHEEL and trilateration, as well as the gap between WHEEL and the theoretical upper bound, which is a direction of our future studies.

ACKNOWLEDGMENT

This work is supported in part by the Hong Kong RGC grant HKUST6169/07E, the NSFC/RGC Joint Research Scheme N_HKUST 602/08, the National Basic Research Program of China (973 Program) under grant No.2006CB303000, the National High Technology Research and Development Program of China (863 Program) under grant No. 2007AA01Z180, and NSFC Key Project grant No. 60533110. The research of Xiang-Yang Li is also supported by NSF CNS-0832120, NSF CCF-0515088, National Natural Science Foundation of China under Grant No. 60828003, RGC under Grant HKBU 2104/06E, and CERG under Grant PolyU-5232/07E.

REFERENCES

- [1] "OceanSense Project," <http://www.cse.ust.hk/~liu/Ocean/index.html>, 2008.
- [2] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in Proceedings of IEEE INFOCOM, 2000.
- [3] G. A. Dirac, "In abstrakten Graphen vorhandene vollstandige 4-Graphen undihre Unterteilungen," *Math Nachrichten*, vol. 22, pp. 61-85, 1960.
- [4] 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 Proceedings of INFOCOM, 2004.
- [5] D. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A. S. Morse, and Y. R. Yang, "Localization in sparse networks using sweeps," in Proceedings of ACM MobiCom, 2006.
- [6] 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 Proceedings of IEEE INFOCOM, 2005.
- [7] P. Gupta and P. Kumar, "Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications*, 1998.
- [8] B. Hendrickson, "Conditions for unique graph realizations," *SIAM Journal of Computing*, vol. 21, pp. 65-84, 1992.
- [9] B. Jackson and T. Jordan, "Connected rigidity matroids and unique realizations of graphs," *Journal of Combinatorial Theory Series B*, vol. 94, pp. 1-29, 2005.
- [10] B. Jackson and T. Jordan, "A sufficient connectivity condition for generic rigidity in the plane," Operations Research Department, Eotvos University, Budapest TR-2008-01, 2008.
- [11] D. J. Jacobs and B. Hendrickson, "An algorithm for two-dimensional rigidity percolation: The pebble game," *Journal of Computational Physics*, vol. 137, pp. 346-365, 1997.
- [12] G. Laman, "On graphs and rigidity of plane skeletal structures," *Journal of Engineering Mathematics*, vol. 4, pp. 331-340, 1970.
- [13] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based localization of large scale sensor networks with complex shape," in Proceedings of IEEE INFOCOM, 2008.
- [14] X.-Y. Li, Y. Wang, P.-J. Wan, and C.-W. Yi, "Fault tolerant deployment and topology control for wireless ad hoc networks," in Proceedings of ACM MobiHoc, 2003.
- [15] L. Lovasz and Y. Yemini, "On generic rigidity in the plane," *SIAM Journal on Algebraic and Discrete Methods*, vol. 3, pp. 91-98, 1982.
- [16] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in Proceedings of ACM SenSys, 2004.
- [17] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Journal of Telecommunication Systems*, 2003.
- [18] M. Penrose, "On K-connectivity for a geometric random graph," *Random Structures and Algorithms*, 1999.
- [19] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in Proceedings of ACM MobiCom, 2000.
- [20] A. Savvides, C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in Proceedings of ACM MobiCom, 2001.
- [21] J. B. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," in Proceedings of Allerton Conference Communication, Control and Computing, 1979.
- [22] S. Y. Seidel and T. S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Transactions on Antennas and Propagation*, vol. 40, pp. 209-217, 1992.
- [23] D. B. West, *Introduction to graph theory*, pp. 174, 2 ed. Pearson Education, 2001.
- [24] F. Xue and P. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, pp. 169-181, 2004.
- [25] Z. Yang, M. Li, and Y. Liu, "Sea depth measurement with restricted floating sensors," in Proceedings of IEEE RTSS, 2007.