Robust Location Detection in Emergency Sensor Networks

Saikat Ray, Rachanee Ungrangsi, Francesco De Pellegrini,
Ari Trachtenberg, David Starobinski*

February 3, 2003

Abstract

We propose a new framework for providing robust location detection in emergency response systems, based on the theory of identifying codes. The key idea of this approach is to allow sensor coverage areas to overlap in such a way that each resolvable position is covered by a unique set of sensors. In this setting, determining a sensor-placement with a minimum number of sensors is equivalent to constructing an optimal identifying code, an NP-complete problem in general. We thus propose and analyze a new polynomial-time algorithm for generating irreducible codes for arbitrary topologies. We also generalize the concept of identifying codes to incorporate robustness properties that are critically needed in emergency networks and provide a polynomial-time algorithm to compute irreducible robust identifying codes. Through analysis and simulation, we show that our approach typically requires significantly fewer sensors than existing proximity-based schemes. Alternatively, for a fixed number of sensors, our scheme can provide robustness in the face of sensor failures or physical damage to the system.

A version of this paper appeard as

S. Ray and R. Ungrangsi and F. De Pellegrinin and A. Trachtenberg and D. Starobinski, "Robust location detection in emergency sensor networks", IEEE INFOCOM, San Francisco, CA, April 2003.

*The authors are with the Electrical and Computer Engineering Department at Boston University. This work was partially supported by the National Science Foundation under NSF Career Grant CCR-0133521, NSF CAREER grant ANI-0132802 and by a SPRInG award from Boston University.

1 Introduction

Communication systems play an essential role in emergency situations such as fires, building collapses or extreme weather phenomena. Unfortunately, existing systems often provide minimal communication infrastructure for supplying information about the nature or the extent of a disaster *in situ*. As a result, first responders typically enter emergency situations with little real-time information about the site, and, should they become trapped, only a haphazard means of rescue are available to them. One promising method for providing real-time feedback from disaster sites involves the use of *sensor networks*.

Recent advances in sensor technologies [1–3] make it possible to install and interconnect tiny devices within existing infrastructure, such as smoke detectors or overhead lighting, for networked use in case of an emergency. These networks could provide emergency control centers with 3D building visualization, real-time monitoring of hot spots or structure failures, and tracking of victims and personnel.

Central to such features is the ability to perform indoor *location detection* in the face of unpredictable reflections (from furniture, people, walls), occlusions (due to smoke, fire), and changing building topology (from falling walls, collapsed ceilings). Indeed, many essential tasks of an advance emergency response system require the following capabilities:

- To enable crew members to identify their own and others' locations.
- To locate victims, potential hazards, or sources of the emergency.
- To identify and rescue trapped personnel.

Though several indoor location detection schemes have been described in the literature (see Section 2), none of them have been designed for the specific working conditions of emergency networks, and are thus generally unsuitable for this purpose. Chiefly, they lack robustness against equipment failures and changing structural topology. In particular, several existing systems are *proximity*-based, in which user location is determined by nearby sensors (also called *beacons*). When sensors fail in such systems, an entire coverage area is lost.

In this paper, we propose to address the issue of robust location detection through a novel framework based on the theory of *identifying codes* [4]. Our approach generalizes existing proximity-based location detection techniques by allowing sensor coverage areas to overlap. Our key idea is to ensure that

each resolvable position be covered by a unique set of sensors, which then serves as its signature.

In general, our approach exhibits two major advantages over existing location detection schemes:

1. For a fixed number of sensors, each with a given coverage area, our scheme can perform location detection at a finer resolution than a scheme which does not allow overlapping coverage areas.

2. Our solution can be designed to function correctly even in the face of corruptions or failures in the system.

These two advantages trade off with one another, meaning that, given a fixed number of sensors, one can design a system with finer resolution at the expense of robustness, or more robustness at the expense of resolution.

The main challenge in designing our system is to position sensors so that every resolvable location can be identified unambiguously. Moreover, despite the projected decreasing cost of sensors, it is desirable to minimize the number of active sensors at a time (*i.e.,* not in sleep mode), thus extending the lifetime of the network. Thus, our goal is to perform location detection, at a given level of robustness, using a minimum number of sensors. For this purpose, we resort to the theory of identifying codes, which provide a general technique for uniquely identifying nodes in a graph.

At a high level, we model a location detection system as a graph by dividing a continuous coverage area into a finite set of regions. Each region is represented by a single point within its boundary. These points correspond to nodes in a graph and nodes are connected by links in the graph if the corresponding points in the physical system are able to communicate directly. The identifying code problem is then to determine the nodes on which to place the codewords such that each node of the graph is covered by a unique set of sensors; the location detection analog of this would be to designate special sensor nodes in such a way that every node in the graph is within communication range of a unique set of sensors.

The problem of finding an optimal identifying code for an arbitrary graph is NP-complete [5]. Instead, we propose a novel greedy algorithm, called ID-CODE, that produces *irreducible* identifying codes. An identifying code is irreducible if no codeword can be removed while still keeping every position uniquely identifiable. Our numerical results show that the solution produced by our algorithm is close to the optimal solution, for a wide range of parameters.

Furthermore, we introduce the concept of r-robust identifying codes. These codes are capable of correcting up to r errors. We propose a new algorithm called r-ID-CODE that generalizes the basic ID-CODE algorithm and produces irreducible r-robust codes. The degree of robustness, r, is a design parameter that can be traded off with the number of sensors required for the proper functioning of the system. We present numerical results illustrating this trade-off.

This paper is organized as follows. Section 2 briefly surveys related work in indoor location detection and identifying codes. In Section 3, we describe the outline of our proposed system, explaining the relationship between robust location detection in emergency sensor networks and the construction of identifying codes for arbitrary graphs. In Section 4, we describe our ID-CODE algorithm, prove some of its key properties, and describe how to apply it to an arbitrary topology. We introduce the concept of r-robust identifying codes in Section 5 and extend the ID-CODE algorithm to produce them. In Section 6, we evaluate the performance of our algorithms and illustrate the benefits of the proposed approach through simulation. The last section summarizes the main findings of the paper and provides some concluding remarks.

2 Related work

Location detection systems have been proposed and implemented in the literature for a variety of applications. For outdoor applications, the satellite based *Global Positioning System* (GPS) is commonly used [6]. GPS relies on trilateration of position and time among four satellites, and can determine location in many cases within a few meters. However, occlusions, reflections, and multipath effects limit the usefulness of GPS in indoor or dense environments.

Indoor location detection systems have been developed for cases when GPS usefulness is limited. These systems can be classified into three categories: *Infrared* (IR), *Ultrasound* (US), and *Radio* (RF). Each of these systems works well for its designed purposes, but lacks essential qualities needed for emergency networks.

2.1 Infrared

The *Active Badge* location system [7] developed at Olivetti Research Laboratory was one of the first indoor location detection systems and is representative of the IR-based approach to indoor location detection [8, 9]. This

system provides each person with a badge that periodically emits a unique ID using diffused IR that is received by one of several receivers scattered throughout a building. Badge location is then resolved by proximity to the nearest receiver.

IR systems require a path be present between the transmitter and the receiver through which light can travel. In an emergency setting, however, the environment can be very dynamic. This renders the system to be prone to failures since the path between the transmitter and the receiver may get blocked quite easily.

2.2 Ultrasound

Ultrasound-based systems also provide location detection based on proximity, but improve accuracy by measuring time-of-flight of ultrasound with respect to a reference RF signal. Systems such as the *Active Bat* [10] or MIT's *Cricket* [11] compare the arrival time of the two signals from various known beacons, allowing a listener to calculate his location. Active Bat claims an accuracy within 10 cm for this approach, and the Cricket system has been extended to determining target orientation [12].

Current ultrasound-based systems also are not designed for the type of robustness needed in an emergency setting. Line-of-sight paths may become obstructed or altered in the face of changing room environments, which results in loss of coverage. In emergency settings, it is also necessary to deal with the possibility of a beacon being destroyed, to which current systems are particularly sensitive.

2.3 Radio

Radio waves provide a powerful means of location detection because of their ability to penetrate various materials. Rather than using differences in arrival time, as done by ultrasound systems, RF-based location detection systems usually determine location based on received signal strength, predicated on a known *Signal-to-Noise Ratio* (SNR). RADAR [13], developed by Microsoft research, pre-computes an SNR-map for a building. The vector of signal strengths received at various base-stations is compared to the map to determine position. Other interesting RF-based systems include SpotON [14], which is designed for three dimensional location detection, and *Nibble* [15], a probabilistic location detection system developed at the University of California, Los Angeles. Nibble improves performance by incorporating a Bayesian model for predicting the likely origin of a signal

based on signal quality observed at access points.

As with the previously mentioned schemes, there are still inherent issues of robustness when applying RF to emergency networks. The failure of a sensor or the introduction of new signal path from shifting internal structures can severely impair existing systems. SNR-based systems have also the problem of being sensitive to environmental conditions. Recently, [16] suggested a simple localization scheme that declares a location to be the centroid of the reference points closest to it. However, this scheme applies to outdoor settings only.

2.4 Identifying Codes

The system proposed in this work overlays an identifying code on a proximity-based location detection system in order to improve resolution and robustness.

Identifying codes were introduced in [4] as a means of uniquely identifying faulty processors in a multiprocessor system. These codes, which are described in detail in Section 4, have enjoyed much attention in the coding theory literature. In general, finding an optimal identifying code is known to be an NP-complete problem [5]. The available constructions in the literature have so far been restricted to regular graphs such as hypercubes, meshes, and trees [17]. The works in [18, 19] suggest the use of these known identifying codes for surveillance purposes in an outdoor setting, but they require a regular, mesh topology. Though regular graphs are appropriate for multiprocessor networks, they are generally hard to realize for wireless networks, especially in indoor settings where there are many obstacles and reflectors. Moreover, emergency networks require robustness that is not available from standard identifying codes. Our system makes use of a robustness-oriented modification of identifying codes built over an arbitrary topology, as described in Sections 3 and 4. Our techniques for building these codes are practically realizable and provide codes whose sizes are close to known lower bounds and, hence, almost optimal.

3 System overview

The performance of a location detection system can be characterized by its *correctness* and *resolution*. Specifically, the correctness of the system is measured by the probability of correctly determining the region in which a target is located. The resolution of the system reflects the smallest distance

between the targets in a given area that can still be distinguished. In general, correctness can be traded off against resolution and vice versa.

In the context of emergency response systems, correctness is usually much more important than resolution. For example, to locate a trapped crew member, it is usually sufficient to know her floor and room; on the other hand, sending a rescue team to the wrong area in an emergency situation can be deadly. Thus, instead of continuous location detection, our system divides the coverage area into locatable regions, and reports a point in this region as the location for a given target.

The system can operate in either or both of two equivalent modes: *location service* or *location tracking*. In the location service mode, the system periodically broadcasts ID packets from designated sensors. An observer can determine his location from the packets that he receives. In the location tracking mode, an observer transmits his ID and the system determines his location from the sensors receiving the ID. Hereafter, we shall describe the system as it operates in the location service mode, though all results apply equally to the location tracking mode.

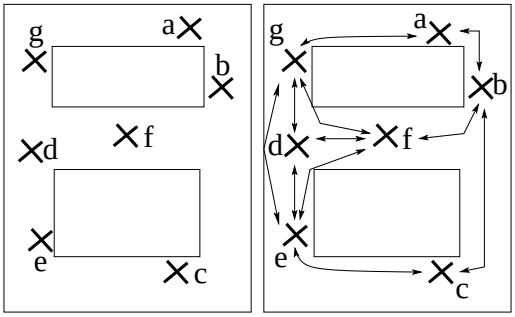
Our emergency sensor network is designed as follows: first, a set of points is selected for a given area. Then, based on the RF-connectivity between the points[1], transmitting sensors are placed on a subset of these points determined by a corresponding identifying code. This placement guarantees that each point is covered by a unique set of transmitters. Thus, an observer can determine his location from the unique collection of ID packets that he receives.

The transmitter placement induces an indistinguishable region around each locatable point (*i.e.,* an observer would receive the same set of ID packets anywhere in this region). This system alone does not guarantee coverage beyond the points incorporated into the graph model. To ensure more widespread coverage, additional techniques should be employed [20–22].

3.1 Example

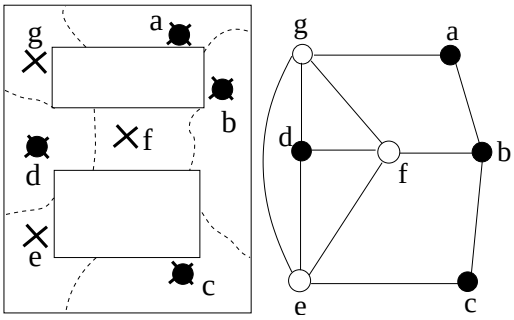
The following example illustrates our approach in more detail. Consider the points $P = \{a, b, c, d, e, f, g\}$ on a simple floor plan illustrated in Figure 1(a), and let the RF-connectivity among these points be represented by the arrows in Figure 1(b); in other words, there is an arrow between positions p_1 and p_2 if and only if there is an RF connection from p_1 to p_2. Given such

[1] Though we assume that RF-connectivity is used, the proposed framework supports any physical connection mechanism.



(a) Discrete Locations.

(b) Connectivity.



(c) Sensor placement (black dots). The coverage region of each point is delimited by the dashed lines.

(d) Position of the sensors on the graph.

Figure 1: Our proposed location detection system.

connectivity information between every pair of points, our objective is to build a system using a minimum number of transmitters that allows an observer to infer his location at any point in P.

For this purpose, we place four wireless transmitters at positions a, b, c and d, with each transmitter periodically broadcasting a unique ID. We assume that packet collisions are avoided by an appropriate medium access control (*e.g.,* simple randomization or a full-scale protocol [16]) and that the observer collects received packets over a (small) time T. For instance, in Figure 1(c), an observer in the region of point f would receive IDs from the transmitters at position b and d. The set of IDs received at a given position x is called the *identifying set* of x and denoted $ID(x)$.

If the identifying set of each point in P is unique, then targets can be correctly located at these points using a table-lookup of the packet IDs received. The reader can verify that, for this example, the identifying sets are unique and given as follows:

$v:$	a	b	c	d	e	f	g
$ID(v):$	$\{a,b\}$	$\{a,b,c\}$	$\{b,c\}$	$\{d\}$	$\{c,d\}$	$\{b,d\}$	$\{a,d\}$

In general, we model a physical environment with a graph $G = (V, E)$, whose vertices V model locatable regions and edges E connect regions with RF connectivity. Figure 1(d) shows the graph for the example in Figure 1(b). Note that the vertices of the graphs are mere points in space, and that physical transmitters need only be placed at those points designated by the chosen identifying code.

In the following section, we develop a theoretical framework, based on identifying codes, for determining the appropriate placement of transmitters in our location detection system.

4 Identifying codes for arbitrary graphs

As mentioned in Section 2, the problem of constructing optimal identifying codes for arbitrary graphs is known to be NP-complete. Therefore, rather than looking for an optimal solution, we propose a greedy algorithm to construct *irreducible* identifying codes. The irreducibility property means that the deletion of any codeword results in a code that is no longer an identifying code. Thus, the proposed algorithm always converges to a local minimum. In fact, our experimental results presented in Section 6 show that the solution achieved by this algorithm is close to the optimal solution for a wide range of parameters.

Moreover, we can show that the proposed algorithm is complete, meaning that all irreducible identifying codes that exist for a given graph, including optimal ones, can be produced with an appropriate selection of the input parameters for the algorithm. Therefore, as is typical with many of the algorithms for NP-complete problems, the algorithm can be used to produce an optimal solution, but at the cost of exponential time complexity.

4.1 Notations and Definitions

Let $G = (V, E)$ be a given graph with vertices V and edges E. Then, we define $\rho(u, v)$ to be the number of edges along the shortest path from vertex u to v. The *ball* $B(v)$ is defined to be

$$B(v) = \{w \in V : \rho(w, v) \leq 1\};$$

that is, $B(v)$ represents the set of vertices that are adjacent to v together with v.

Any non-empty subset $\mathbb{C} \subseteq V$ is called a *code* for the corresponding graph $G = (V, E)$, and its elements are called *codewords*. Given a code $\mathbb{C}$, the *identifying set* of a vertex $v \in V$ is defined to be

$$I_{\mathbb{C}}(v) = B(v) \cap \mathbb{C} \tag{1}$$

A code $\mathbb{C}$ is called an *identifying code* if for all $u, v \in V$

$$I_{\mathbb{C}}(u) \neq I_{\mathbb{C}}(v)$$

that is, the identifying set of every vertex in the graph is unique (so that every vertex is uniquely identified by its identifying set). An identifying code $\mathbb{C}$ is called *irreducible* if deletion of any codeword from $\mathbb{C}$ results in a code that is no longer an identifying code. A graph $G = (V, E)$ is said to be *distinguishable* if it permits an identifying code; otherwise, G is an indistinguishable graph.

4.2 Code Construction Algorithm

Formally, the problem of location detection is: *Given a distinguishable graph $G = (V, E)$, determine a subset $\mathbb{C} \subseteq V$ of minimum cardinality that is an identifying code.* Since this problem has been shown to be NP-complete [5], we instead consider a practical modification:

> *Given a distinguishable graph $G = (V, E)$, compute a subset $\mathbb{C}$ of V such that $\mathbb{C}$ is an irreducible identifying code for G.*

The first step in solving the above question is to determine whether a given graph is distinguishable. The following lemmas show that this determination is not hard to do in practice.

Lemma 1. *For a given graph $G = (V, E)$, if $\mathbb{C}$ is an identifying code, then every $\mathbb{D} \supseteq \mathbb{C}$ is also an identifying code.*

Proof. Assume that there exists $\mathbb{D} \supseteq \mathbb{C}$ that is not an identifying code. Then, by definition, there exist $u, v \in V$ such that

$$\begin{aligned} I_{\mathbb{D}}(u) &= I_{\mathbb{D}}(v) \\ \mathbb{D} \cap B(u) &= \mathbb{D} \cap B(v) \\ \mathbb{C} \cap \mathbb{D} \cap B(u) &= \mathbb{C} \cap \mathbb{D} \cap B(v) \\ \mathbb{C} \cap B(u) &= \mathbb{C} \cap B(v), \qquad \text{since } \mathbb{C} \subseteq \mathbb{D} \\ I_{\mathbb{C}}(u) &= I_{\mathbb{C}}(v) \end{aligned}$$

which is a contradiction of the assumption that $\mathbb{C}$ is an identifying code. ∎

Corollary 1. *V is an identifying code for any distinguishable graph $G = (V, E)$.*

An equivalent lemma was proven in [17]:

Lemma 2. *A graph $G = (V, E)$ is distinguishable if and only if*

$$\forall \;\; u, v \in V, \;\; B(u) \neq B(v).$$

Thus, to check if a graph is distinguishable, one must merely check that there are no two vertices with the same ball. In practice, we have seen empirically that almost all graphs are distinguishable unless their average degree is very low or very high. Graphs that are indistinguishable generally have a collection of vertices that are physically close to each other, and we describe in Appendix A a simple procedure for deleting a minimum number of vertices to make an indistinguishable graph into a distinguishable one.

Algorithm ID-CODE, presented in Figure 2, begins by designating every vertex in an input graph G as a codeword. Corollary 1 insures that this will be an identifying code for any distinguishable graph. At each step of the algorithm, one codeword is considered for deletion from the current code. If removing the codeword results in an identifying code, then the algorithm proceeds; otherwise, the codeword is reinserted into the code and

```
ID-CODE(G, a)
ℂ = V
if ℂ is not an identifying code
  do  EXIT
for each vertex x ∈ a, taken in order
  do  D = ℂ \ {x}
      if ∃ u, v ∈ V such that I_D(u) = I_D(v)
            ℂ = ℂ
      else  ℂ = D
return ℂ
```

Figure 2: The ID-CODE algorithm for generating an identifying code for an arbitrary graph.

the algorithm proceeds to consider other codewords, along a predetermined sequence of vertices, **a**, provided as a parameter.

By design, each iteration of the algorithm (including the last iteration) ends with an identifying code of the graph. Moreover, the algorithm performs one iteration for each vertex in the graph and at each iteration, checks for uniqueness of the identifying set of each node. Using an appropriate sorting to determine the uniqueness, the running time of the algorithm becomes $O(|V|^3 \log |V|)$. Also, using appropriate hash functions, the expected running time can be reduced to $O(|V|^2 \log |V|)$.

Theorem 1. *The code $\mathbb{C}$ returned by ID-CODE is irreducible.*

Proof. Assume, for sake of contradiction, that $\mathbb{C} \setminus X$ is an identifying code for some $X \neq \emptyset$. Choose any codeword $x \in X$ and let i be the iteration in which the ID-CODE considers codeword x and let $\mathbb{C}_i \supset \mathbb{C}$ be the resultant code at the beginning of this iteration. It must be that the set $D \triangleq \mathbb{C}_i \setminus \{x\}$ is not an identifying code, or else ID-CODE would have removed x from $\mathbb{C}$. Moreover, $D \supseteq \mathbb{C} \setminus \{x\} \supseteq \mathbb{C} \setminus X$. However, since $\mathbb{C} \setminus X$ is an identifying code, Lemma 1 implies that D is an identifying code as well, which completes the contradiction. ∎

Theorem 1 shows that a code returned by the ID-CODE is irreducible. The following theorem shows the converse, that is, that every irreducible

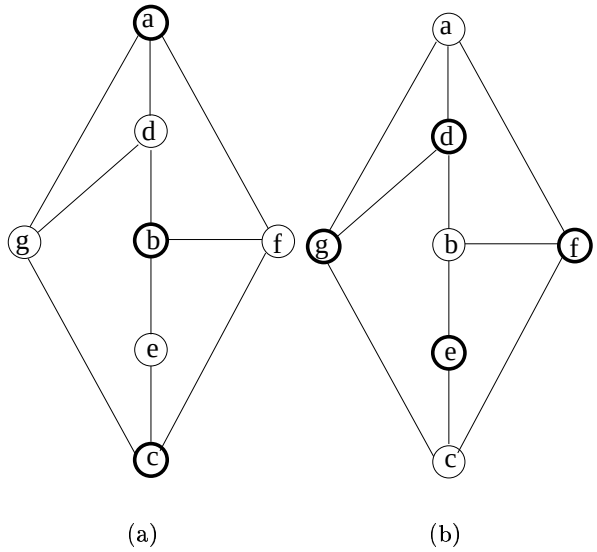


Figure 3: Different irreducible identifying codes for different sequences: (a) If the sequence of the vertices visited by ID-CODE is $\mathbf{a} = \{f, g, d, e, a, b, c\}$, then the resultant code is $\mathbb{C} = \{a, b, c\}$; (b) on the other hand, with the input sequence $\mathbf{a} = \{a, b, c, d, e, f, g\}$, the resultant code is $\mathbb{C} = \{d, e, f, g\}$.

identifying code, including optimal ones, can be generated by ID-CODE through an appropriate choice of the input parameters. The proof is given in Appendix B.

Theorem 2. *For every irreducible identifying code* $\mathbb{C}$ *of a given graph* $G = (V, E)$*, there exists an input sequence* $\mathbf{a}$ *such that ID-CODE(G,*$\mathbf{a}$*) returns* $\mathbb{C}$*.*

4.3 Example

The performance of ID-CODE depends on the sequence of vertices chosen. Figure 3 demonstrates concretely that different input sequences can result in different irreducible codes.

In section 6, we propose and evaluate some simple heuristic methods for effectively ordering the input sequence.

5 r-Robust code construction

Robustness is critical to emergency sensor networks; typical corruptions in such networks include:

- Destruction of ID-transmitting vertices by the emergency agent (*e.g.,* , fire, water, explosion).
- Variation of radio paths due to changes in building structure (*e.g.,* , walls collapsing, furniture shifting, people moving).
- Failure of ID reception due to medium access control scheme limitations.

In the previous section we described techniques for constructing identifying codes with as few codewords as possible. This framework inherently provides some amount of robustness since a point may be covered by sensors located far off, thereby creating spatial diversity. However, in practice, the identifying set received by an observer might fluctuate due to environmental conditions, and thus we seek to guarantee that the scheme works even if the received identifying set is different from the original one up to a certain limit. In this section we describe a novel generalization of identifying codes that achieves this goal by guaranteeing to be robust in the face of spurious fluctuations in observed identifying sets. First, we formalize our definition of robustness, making use $\oplus$ to denote symmetric difference (*i.e.,* , $A \oplus B = (A \setminus B) \cup (B \setminus A)$).

Definition 1. *An identifying code $\mathbb{C}$ over a given graph $G = (V, E)$ is said to be* r-robust *if*

$$I_{\mathbb{C}}(u) \oplus A \neq I_{\mathbb{C}}(v) \oplus B$$

for all $u, v \in V$ and $A, B \subseteq \mathbb{C}$ with $|A|, |B| \leq r$.

Simply stated, an identifying code is r-robust if the addition or deletion of up to r IDs at any vertex does not change its identifying capability. Alternatively, we may determine the robustness of a code $\mathbb{C}$ by measuring the minimum symmetric difference

$$d_{\min}(\mathbb{C}) \triangleq \min_{u,v \in V} |I_{\mathbb{C}}(u) \oplus I_{\mathbb{C}}(v)|$$

between the identifying sets of any two vertices. We thus have the following Theorem as a straightforward application of the definitions.

Theorem 3. *A code $\mathbb{C}$ is r-robust if and only if*

$$d_{min}(\mathbb{C}) \geq 2r + 1.$$

Adding codewords to an identifying code can only increase its minimum symmetric difference, as the following lemma proves.

Lemma 3. *For any two identifying codes $\mathbb{C} \subseteq D$ over the same graph G, $d_{min}(\mathbb{C}) \leq d_{min}(D)$.*

Proof. By definition, there exist $u, v \in D$ such that the symmetric difference $d(I_D(u), I_D(v)) = I_D(u) \oplus I_D(v) = d_{\min}(D)$.

Using (1) and the fact that $\mathbb{C} \subseteq D$, we get that:

$$\begin{aligned} I_D(u) \oplus I_D(v) &= (B(u) \cap D) \oplus (B(v) \cap D) \\ &= (B(u) \oplus B(v)) \cap D \\ &\supseteq (B(u) \oplus B(v)) \cap \mathbb{C} \\ &= (B(u) \cap \mathbb{C}) \oplus (B(v) \cap \mathbb{C}) \\ &= I_{\mathbb{C}}(u) \oplus I_{\mathbb{C}}(v). \end{aligned} \tag{2}$$

Therefore,

$$\begin{aligned} d_{\min}(D) &= |I_D(u) \oplus I_D(v)| \\ &\geq |I_{\mathbb{C}}(u) \oplus I_{\mathbb{C}}(v)| \\ &\geq d_{\min}(\mathbb{C}). \end{aligned}$$

∎

As it turns out, a simple modification of the greedy criterion of ID-CODE, depicted in Figure 4, produces an r-robust code if it exists. As with ID-CODE, examining vertices in the right order will necessarily produce the best possible r-robust codes for the given graph.

By construction, $\mathbb{C}$ is an r-robust identifying code at every iteration of the algorithm and the straightforward running time is $O(|V|^4)$. At the expense of storage complexity, the running time can be reduced to $O(|V|^3)$. Moreover, the following theorem, which is proven using Lemma 3 in a manner analogous to Theorem 1, shows that the resultant code is irreducible, meaning that the code is no longer r-robust if any codeword is removed.

Theorem 4. *The code $\mathbb{C}$ returned by r-ID-CODE is irreducible.*

```
r-ID-CODE(G, a, r)
ℂ = V
if d_min(ℂ) ≤ 2r
  do  EXIT
for each vertex x ∈ a
  do  D = ℂ \ {x}
      if d_min(D) ≤ 2r
            ℂ = ℂ
      else  ℂ = D
return ℂ
```

Figure 4: The r-ID-CODE algorithm for generating r-robust identifying codes for an arbitrary graph.

To decode location with an r-robust code, an observer must be provided a lookup table with the identifying set of every vertex in a given (uncorrupted) graph. Upon receiving an identifying set S, the observer finds the point p that minimizes $|I_{\mathbb{C}}(p) \oplus S|$. As long as no more than r IDs are corrupted, the observer is guaranteed to determine her location correctly.

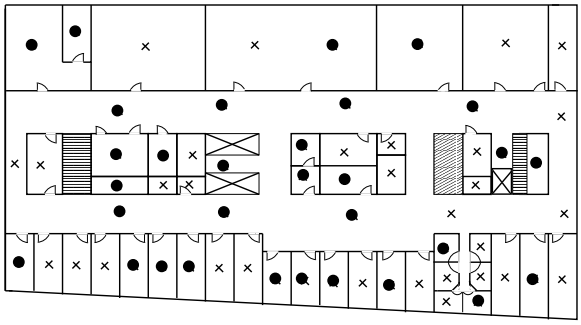
We conclude this section by giving an example of a 1-robust code, depicted in Figure 5. The figure shows the floor-plan of the fourth floor of the Photonics building at Boston University. The vertices of the graph are distributed in such a way that each room and corridor can be located, and two vertices are connected if there are fewer than six obstacles along the line-of-sight path between them.

6 Performance evaluation

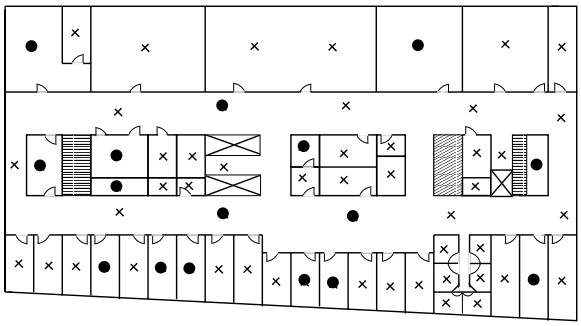
In this section we evaluate the performance of the proposed algorithms by simulation. Since both ID-CODE and r-ID-CODE leave as parameters the order in which vertices are to be visited, we first propose a few heuristic orderings.

6.1 Approaches for Ordering the Input Sequence

The simplest approach is to visit all vertices in random order. An alternative approach is based in the empirical observation that the proposed algorithms



(a) 1-robust code.



(b) 0-robust code.

Figure 5: Robust identifying codes. Solid circles represent transmitters. The 0-robust code, providing no robustness, requires 16 transmitters among the 64 points of resolution, whereas the 1-robust code requires 32.

are most likely to remove vertices that are visited first in sequence. Thus, performance should be improved by placing "good" codewords at the end of the sequence. Intuitively, a codeword is good if it is maximally distant from other codewords. To guess good codewords, we distinguish between the following two cases:

- If the average degree of vertices in the graph is low, then the good codewords are likely to have high degree since having larger degree minimizes the number of codewords required to cover all the vertices of the graph.
- On the other hand, if the average degree of the vertices in the graph is high, then a small number of vertices can cover the graph. However, if the codewords have high degree, then their balls will differ little, and, consequently, a larger number of codewords would be needed. So, good codewords are likely to have lower degree in this case.

Based on these observations we propose a hybrid heuristic for ordering: when the average degree of a graph is greater than half the number of vertices, we visit vertices in descending order of degree; otherwise we visit them in ascending degree order.

6.2 Simulation Results

To evaluate the performance of our algorithms, we applied the ID-CODE and r-ID-CODE algorithms on various graphs. The graphs used were random, connected, distinguishable graphs with average degree d_{ave}, where d_{ave} is a parameter. The graphs were generated by joining every two vertices with probability p, where $p = \frac{d_{\text{ave}}}{(|V|-1)}$, and discarding disconnected or indistinguishable graphs. For every d_{ave}, 100 different graphs were generated randomly and the results were averaged. The graphs used in this simulation are well suited to model an area comparable to the range of wireless transmitter with a large number of obstacles so that any two vertices might get connected. The simulation results are shown for varying values of d_{ave}.

Figure 6 shows the average size of the resultant code returned by the ID-CODE algorithm for $|V| = 128$ vertices graphs. The three upper curves corresponds to sorting of the vertex-sequence in ascending, descending and random orders, respectively. The bottom curve is a modified version of a lower bound provided in [4, Theorem 1(3)]. We observed similar behavior for graphs with 16, 32 and 64 vertices.

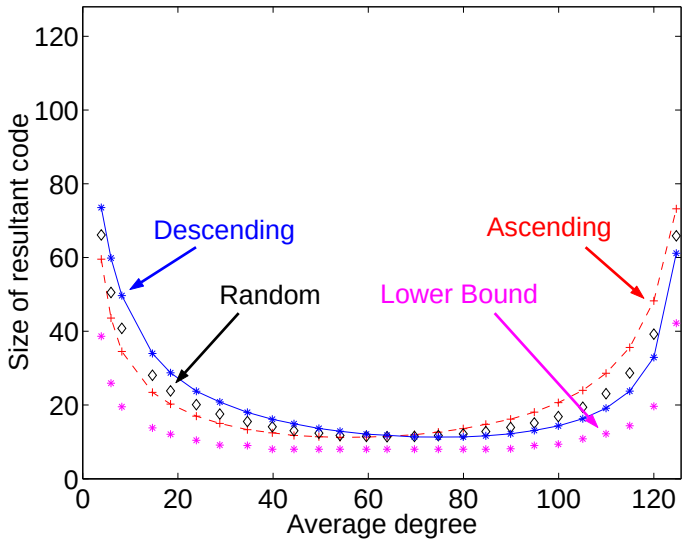


Figure 6: Performance of various heuristics for $|V| = 128$ vertices graphs.

As expected from the discussion of Section 6.1, ascending degree sorting of vertex-sequence gives the best results when $d_{\text{ave}} < |V|/2$ and descending degree sorting of vertex-sequence gives the best results when $d_{\text{ave}} > |V|/2$. The random sequences of vertices perform in between the other two for every choice of d_{ave}.

All three sorting approaches, including random sorting, are reasonably close to the lower bound. The performance of the algorithm does not seem to be much affected by the ordering of the input sequence.

We have also observed that the size of the resultant code is the smallest when the average degree is approximately $|V|/2$ and we conjecture that this is always the case. Nevertheless, the performance of the algorithm is not very sensitive to the average degree of the graph in a large plateau-like region. For instance, referring again to Fig. 6, we observe that the average code-size remains smaller than 15 for average degrees ranging from 40 to 90.

Also of interest is the scalability of ID-CODE to graphs with many vertices. Figure 7 shows our simulation results from graphs with 10 to 1000 vertices, each of average degree $|V|/2$. The figure clearly shows that the ratio of codewords to graph vertices decreases as the number of vertices increases. Thus, large graphs require relatively few transmitters for location detection. This shows that our approach scales well and is especially useful for graphs

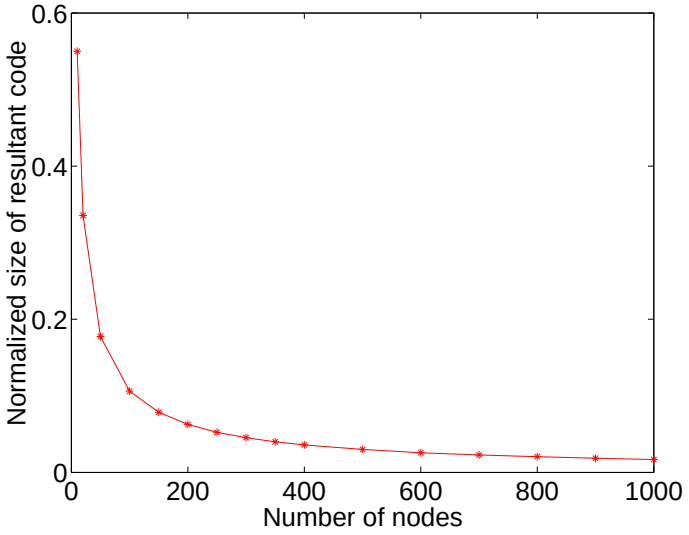


Figure 7: Scalability of the resultant identifying code $\mathbb{C}$. The normalized size $|\mathbb{C}|/|V|$ is plotted against the number of vertices $|V|$ in the graph.

with large number of vertices. Note that in simple proximity-based systems, where each position is covered by a single sensor, the ratio $|\mathbb{C}|/|V|$ remains always equal to 1.

Finally, we consider the behavior of r-robust codes. As in the previous set of experiments, we applied r-ID-CODE to random, connected graphs with the results shown in Figure 8. As expected, the code-size increases with increasing r so that there is a clear trade-off between the robustness of a code and the number of transmitting vertices that are required.

The general behavior of r-robust codes is similar to that of standard identifying codes. Minimum size is achieved if the average degree is about $|V|/2$, but the size of the resultant code is not too sensitive to the average degree. We see, again, that for a large range of degree values around $|V|/2$, the code-size is close to the one obtained for $d_{\text{ave}} = N/2$. However, the sensitivity increases as the robustness requirements are increased.

7 Conclusion

Indoor location detection is an integral part of advanced emergency response systems. In this paper, we have proposed a new framework for providing

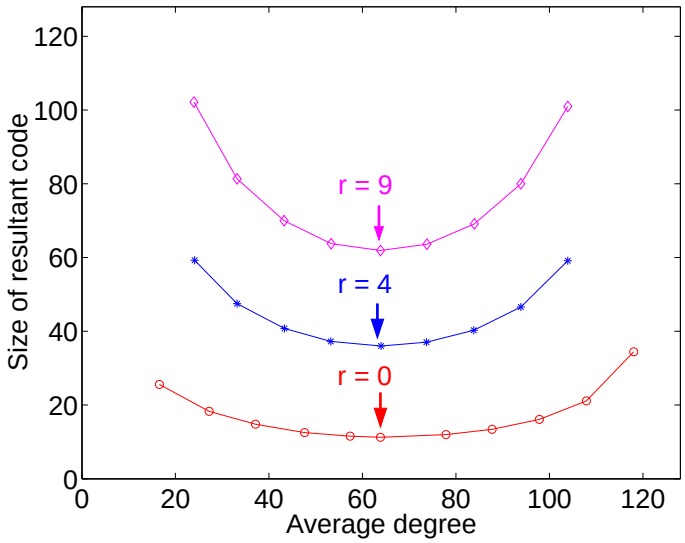


Figure 8: Behavior of r-robust codes with 128 vertices.

robust location detection for these systems, based on identifying codes. Our approach involved overlapping sensor coverage in such a way that each position on a floor is covered by a unique, and hence identifying, set of sensors.

We have proposed a polynomial-time algorithm, ID-CODE, for determining sensor placement by generating a corresponding irreducible identifying code. Since sensors may get destroyed and the connectivity between different positions may vary during emergency situations, we have also introduced the new concept of r-robust identifying codes. These codes can tolerate up to r errors during the collection of ID-packets, at any given position, while still providing accurate location information. We have also proposed a generalization of the ID-CODE algorithm, called r-ID-CODE, for generating irreducible r-robust identifying codes in polynomial time.

The performance of our algorithms has been evaluated through extensive simulations on random graphs of varying sizes and varying average degrees. Since the order in which vertices are processed by our algorithms could affect their output, we have proposed and tested several heuristic-based orderings. We have found that, for a wide range of parameters, the solution provided by our algorithms is close to a known theoretical lower bound and hence close to the optimal solution. Our results have also shown that the performance of the algorithms is rather insensitive to the ordering of the sequence,

and randomly ordered sequences generally perform well enough. Thus, the proposed algorithms do not require any particular parameter tuning.

The framework proposed in this paper does not impose any constraint on the physical communication layer implemented by the sensors. If the physical layer is inherently short-range, such as IR, then the system reduces to a proximity-based system, where each vertex is a codeword. On the other hand, our approach is more naturally suited to RF-communication, which allows sensors (codewords) to cover overlapping areas and is not limited to line-of-sight paths. We have shown that this approach leads to a drastic reduction in the total amount of sensors needed, or, alternatively, an increase in the robustness of the system.

Overall, this work has shown that the theory of identifying codes provides a promising means for enhancing the robustness and effectiveness of location detection systems in the context of emergency sensor networks. There are several open issues, however. Most importantly, wireless connections are not binary in nature, especially at the low SNR region. As a future work, it would be interesting to investigate whether it is possible to exploit signal attenuations due to structural properties of a typical indoor setting to create more stable connections (*e.g.,* by discarding packets that come with a low SNR).

Acknowledgment

The authors would like to thank Dr. M. Karpovsky and Dr. L. Levitin for introducing them to identifying codes. Insightful comments by the anonymous reviewers are gratefully acknowledged as well.

A. Constructing distinguishable graphs

Occasionally, the graph induced by available peer-to-peer connections might not be distinguishable, in which case any proximity-based location detection scheme will fail. Though the entire graph might not be distinguishable, there will always be an induced subgraph that is distinguishable (in the trivial case, this would be a subgraph consisting of a single vertex). We now describe a simple technique for generating such a subgraph with as many vertices as possible.

Consider a graph $G = (V, E)$ that is not distinguishable. We can partition vertices in the graph into equivalence classes based on their neighbors (denoted Neighborhood Equivalence Class or NEC). More precisely, v_1 and

v_2 will be in the same NEC iff $B(v_1) = B(v_2)$. The *contracted subgraph* of G is the subgraph $G' = (V', E')$ induced by picking one vertex from each equivalence class of V. The following lemmas and theorems prove that the contracted subgraph of G is its distinguishable subgraph with maximum number of vertices.

Lemma 4. *For any vertex $v \in V$ in a graph $G = (V, E)$, $B(v)$ is a union of NEC's of G.*

Lemma 4 follows from the fact that if $v' \in B(v)$ then $v \in B(v')$ and, therefore, $v \in B(v*)$ for all $v* \in NEC(v')$.

Theorem 5. *The contracted subgraph G' of G is distinguishable.*

Proof. By Lemma 4, ball around node v is $\cup NEC(v_i)$ for some v_i's. When inducing the subgraph G', we reduce the ball around v to $\cup v_i$. Thus, there is a one-to-one correspondence between balls in G and balls in G', implying that equivalent vertices in G' must also be equivalent in G. Thus, the only way for G' to have two equivalent vertices u and v (and, hence, be indistinguishable) is if u and v are in the same NEC of G, which contradicts the definition of a contracted subgraph. ∎

Theorem 6. *The contracted subgraph of G' of G has no fewer vertices than any other distinguishable subgraph of G.*

Proof. Any distinguishable subgraph G^* with more vertices than G' will necessarily have two vertices in the same NEC of G, by the pigeon-hole principle. These two vertices must thus also be in the same NEC of G^*, violating the assumption that G^* is distinguishable. ∎

Thus, for a given topology, the contracted subgraph is the largest subgraph that can be used for proximity-based location detection. The information needed to generate the contracted subgraph is local, since two vertices cannot be in the same NEC unless they are themselves neighbors. Hence, this subgraph can be determined efficiently and in a distributed fashion.

B. Proof of Theorem 2

We give the proof of Theorem 2 using the following lemma.

Lemma 5. *Given a graph $G = (V, E)$ with $|V| = N$, let $\hat{\mathbb{C}}$ be an irreducible identifying code with $|\hat{\mathbb{C}}| = q$. Let A be the set of sequences of vertices such that $\forall\ \mathbf{a} \in A$, ID-CODE($G$, $\mathbf{a}$) returns $\hat{\mathbb{C}}$. Then, $|A| \geq q!(N-q)!$.*

Proof. Let $\hat{\mathbb{C}} = \{c_1, c_2, \ldots, c_q\}$ and $V = \{v_1, v_2, \ldots, v_{N-q}, c_1, c_2, \ldots, c_q\}$. Consider the sequence
$\mathbf{a} = (v_1, v_2, \ldots, v_{N-q}, c_1, c_2, \ldots, c_q)$, i.e. a sequence where all the codewords of $\hat{\mathbb{C}}$ appear at the end. Then for $1 \leq i \leq N-q$, v_i will be removed from code by ID-CODE since at i-th step, $\mathbb{C}_{i-1} \setminus v_i = \{v_{i+1}, \ldots, v_{N-q}, c_1, c_2, \ldots, c_q\} \supseteq \hat{\mathbb{C}}$ and therefore an identifying code by Lemma 1. Thus, after $(N-q)$-th step, $\mathbb{C}_{N-q} = \hat{\mathbb{C}}$. But at no future step any more codeword can be removed since $\hat{\mathbb{C}}$ is irreducible. Therefore, the code returned by ID-CODE(G, $\mathbf{a}$) is $\hat{\mathbb{C}}$.

Now, $v_i,\ 1 \leq i \leq N-q$ can be arranged in $(N-q)!$ different ways and for each, rest of the vertices $c_i,\ 1 \leq i \leq q$ can be arranged in $q!$ different ways. Therefore, there are at least $q!(N-q)!$ sequences that result into $\hat{\mathbb{C}}$.

∎

The proof of Theorem 2 follows from Lemma 5 since $q!(N-q)! \geq 1$ as $0 \leq q \leq N$.

References

[1] Seth Edward-Austin Hollar, "COTS dust," M.S. thesis, University of California, Berkeley, 2000.

[2] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for 'smart dust'," in *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*, Seattle, WA, United States, 1999, ACM.

[3] Deborah Estrin, David Culler, Kris Pister, and Gaurav Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, January-March 2002.

[4] Mark G. Karpovsky, Krishnendu Chakrabarty, and Lev B. Levitin, "A new class of codes for identification of vertices in graphs," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 599–611, March 1998.

[5] Nageswara S. V. Rao, "Computational complexity issues in operative diagnosis of Graph-Based systems," *IEEE Transactions on Computers*, vol. 42, no. 4, pp. 447–457, April 1993.

[6] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, Springer-Verlag, 4 edition, 1997.

[7] Roy Want, Andy Hopper, Veronica Falcao, and Jon Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, January 1992.

[8] Sue Long, Rob Kooper, Gregory D. Abowd, and Christopher G. Atkeson, "Rapid prototyping of mobile context-aware applications: The Cyberguide case study," in *2th ACM International Conference on Mobile Computing and Networking (MOBICOM '96)*. ACM, July 1996.

[9] R. Azuma, "Tracking requirements for augmented reality," *Communication of the ACM*, vol. 36, no. 7, pp. 50–51, July 1993.

[10] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," in *Mobicom'99*. ACM, 8 1999.

[11] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan, "The cricket location-support system," in *6th ACM International Conference on Mobile Computing and Networking (ACM MOBICOM)*, Boston, MA, 2000, ACM.

[12] Nissanka B. Priyantha, Allen K. L. Miu, Hari Balakrishnan, and Seth Teller, "The cricket compass for contextaware mobile applications," in *7th ACM Conference on Mobile Computing and Networking (MOBICOM)*, Rome, Italy, July 2001, ACM.

[13] Paramvir Bahl and Venkata N. Padmanabhan, "RADAR: An inbuilding RF-based user location and tracking system," in *IEEE INFOCOM 2000*, Tel Aviv, Israel, 2000, IEEE.

[14] Jeffrey Hightower, Gaetano Borriello, and Roy Want, "SpotON: An indoor 3D location sensing technology based on RF signal strength," Tech. Rep. #2000-02-02, University of Washington, February 2000.

[15] Paul Castro, Patrick Chiu, Ted Kremenek, and Richard R. Muntz, "A probabilistic room location service for wireless networked environments," in *Ubicomp*, Atlanta, GA, 2001, ACM.

[16] Nirupama Bulusu, John Heidemann, and Deborah Estrin, "GPS-less low cost outdoor localization for very small devices," Tech. Rep. 00-729, University of Southern California/Information Sciences Institute, April 2000.

[17] Irène Charon, Oliver Hudry, and Antoine Lobstein, "Identifying codes with small radius in some infinite regular graphs," *The Electronic Journal of Combinatorics*, vol. 9, 2002.

[18] Krishnendu Chakrabarty, S. Sitharama Iyengar, Hairong Qi, and Eungchun Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *Accepted for publication in IEEE Transactions on Computers.*

[19] Krishnendu Chakrabarty, Hairong Qi, Sitharama S. Iyengar, and Eungchun Cho, "Coding theory framework for target location in distributed sensor networks," in *International Symposium on Information Technology: Coding and Computing*, 2001, pp. 130 –134.

[20] Nirupama Bulusu, John Heidemann, and Deborah Estrin, "Adaptive beacon placement," in *Twenty-first International Conference on Distributed Computing Systems (ICDCS-21).* University of California, Los Angeles, April 2001, IEEE Computer Society.

[21] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *IEEE INFOCOM 2001.* April 2001, IEEE.

[22] Seapahn Meguerdichian, Sasa Slijepcevic, Vahag Karayan, and Miodrag Potkonjak, "Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure," in *Proceedings of The 2001 ACM International Symposium on Mobile Ad Hoc Networking & Computing.* October 2001, pp. 106–116, ACM.