# Sensor Network Navigation without Locations

Mo Li, Yunhao Liu, Jiliang Wang, and Zheng Yang
Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong
{limo, liu, aliang, yangzh}@cse.ust.hk

*Abstract*—**We propose a pervasive usage of the sensor network infrastructure as a cyber-physical system for navigating internal users in locations of potential danger. Our proposed application differs from previous work in that they typically treat the sensor network as a media of data acquisition while in our navigation application, in-situ interactions between users and sensors become ubiquitous. In addition, human safety and time factors are critical to the success of our objective. Without any pre-knowledge of user and sensor locations, the design of an effective and efficient navigation protocol faces non-trivial challenges. We propose to embed a road map system in the sensor network without location information so as to provide users navigating routes with guaranteed safety. We accordingly design efficient road map updating mechanisms to rebuild the road map in the event of changes in dangerous areas. In this navigation system, each user only issues local queries to obtain their navigation route. The system is highly scalable for supporting multiple users simultaneously. We implement a prototype system with 36 TelosB motes to validate the effectiveness of this design. We further conduct comprehensive and large-scale simulations to examine the efficiency and scalability of the proposed approach under various environmental dynamics.**

***Keywords—navigation; sensor networks; cyber-physical system***

## I. INTRODUCTION

Recent advances in wireless sensor network (WSN) technologies provide us with the ability to offer pervasive usage of sensor networks widely deployed over the space of interest. The increasing study of WSNs aims at enabling computers to better serve people by automatically monitoring and interacting with the physical world [5, 13, 16]. Existing works, however, largely focus on developing sensor network systems principally providing remote data collection. The possibility of in-situ interactions between the users and their physical environment is overlooked. Such interactions could significantly expand the capability of WSNs and thereby enhance their usability.

This work proposes to utilize the sensor network infrastructure as a cyber-physical system for navigating internal users during emergencies. The users are equipped with communicating devices like 802.15.4 compatible PDAs that communicate with sensors in the network. In the event of an emergency, the sensor network explores the emergent field and provides the necessary guidance information to navigate the user to safety.

The proposed application essentially differs from previous researches in several aspects. First, most of previous works view the sensor network as a mechanism for data acquisition, concentrating on organizing a data-centric network for efficiently collecting, routing, and processing in-network sensory data. In contrast, our application focuses more on in-situ user interaction with the sensor network infrastructure. There are not necessarily one or more sinks as data processing centers, yet there are no needs collecting back the sensory data distributed over the field. All operations are in-situ carried out by cyber-physical interactions among users and network nodes. Second, the navigation of human beings is inherently different from routing data packets. We have various methods to compensate for network issues like packet loss, e.g., packet reroute, multi-path routing, data redundancy and etc. The navigation of human beings, however, is the safety-critical selection of a single route, which prevents us from simply borrowing existing packet routing algorithms. For example, the recently suggested opportunistic routing paradigm, though it provides extra efficiency for the delivery of wireless data, obviously cannot be applied to design navigation protocols since it is impossible for human beings to be physically multicast or copied in their movement. Third, the time factor becomes critical in the context of human navigation other than the data delivery in the network. The limited human movement speed dictates that the navigation process is time consuming, while emergency situations might result in time variation. As emergency or dangerous situations change, it becomes necessary to frequently update the route plans for the guided users. Path dynamics in traditional packet routing process typically occurs between delivering different packets, while such dynamics in the navigation process might exist all along for guiding single individuals. People may even move backwards to seek broader opportunities under varied situations.

There have been attempts made at guiding navigation using wireless sensor networks. Most, if not all, existing approaches assume the availability of locations on each sensor node. Knowing the locations of dangerous areas, the sensor network can perform easy and efficient route calculations to navigate internal users out of the emergency area. The location information, however, may not always be available in many realistic situations where emergency guidance are needed, e.g., an underground tunnel or coal mine, a complex indoor area, and etc. The requirement of location information largely constrains the applicability of existing approaches to location-free environments. In addition, existing approaches usually do not specifically consider the impact of variations of dangerous areas, e.g., the expansion, shrinking, or disappearing of areas which is deemed dangerous. In reality, such variations often degrade the effectiveness of existing designs or even overwhelm them.

978-1-4244-3513-5/09/$25.00 ©2009 IEEE

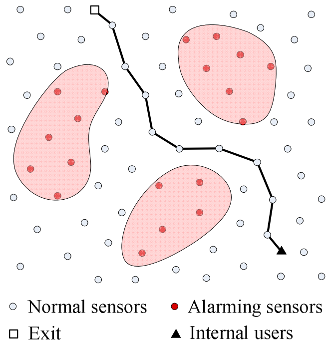


Figure 1. Sensor network navigation

The design objectives of this work are twofold. First, to release the necessity of utilizing location information. Specifically, neither the sensor nodes nor the users need to know their instant locations to achieve successful navigation. Second, to address emergency dynamics that can lead to variations of dangerous areas. Neither of the above is fully addressed in existing literatures. In this work, we propose to embed a distributed road map across the sensor network, which performs as a public infrastructure for providing guidance information for internal inquirers. Users would be able to issue queries from the road map and follow the recommended routes, avoiding dangerous areas. We also design efficient mechanisms to update the road map system according to the variations of dangerous areas thereby maintaining its accuracy and effectiveness. We implement a prototype system with 36 Telos Motes as well as conduct large scale simulations to examine the performance of our design. Experimental results show that this approach is effective and highly scalable when the network size is large and multiple users are simultaneously navigated.

The remainder of the paper is organized as follows. Section II presents problem specification. Section III describes our design principles. We present our implementation experience in Section IV. We further validate this design with experiments on our prototype system. We evaluate the scalability of our approach with large-scale trace driven simulations in Section V. We discuss the related work in Section VI and conclude the work in Section VII.

## II. PROBLEM SPECIFICATION

We consider the scenario of navigating human beings on the field under emergencies, where there might be several dangerous areas that threaten the safety of human beings, e.g., excessive heat, poisonous gas, passage obstacles, and etc. People need to be guided out of the field while keeping away from those dangerous areas. Figure 1 depicts such an example scenario. We characterize the navigation problem as a path planning problem and present its assumptions, objectives and requirements as follows.

### ***Assumptions***

We assume an emergent field containing several areas of dangers, as the red areas shown in Figure 1. The dangerous areas might emerge, disappear, expand or shrink as the time passes and the number of different dangerous areas at any time instance $t$ is a constant $L_t$.

A sensor network system is deployed on the field, where each sensor is able to detect the dangers distributed over the field. The sensor node triggers a "yes" alarm if it resides in the dangerous area (red nodes in Figure 1) and triggers "no" if outside (sky-blue nodes in Figure 1). Thus the boundary of a dangerous area can be outlined by the pairs of neighboring sensors with different outcomes. Each user carries a communicating device like 802.15.4 compatible PDA that can talk with sensors. By measuring the strength and direction of wireless signals, the user is able to track any targeted sensor node [12]. Thus the navigating route can be interpreted as a sequence of nodes.

### ***Objectives***

The objective of a successful navigation is to plan a path for each user to one or more pre-known exits on the field which lead to safe departure, bypassing all the dangerous areas. In Figure 1, we depict such an example route that leads the internal user to the exit. The navigation process is carried out in a fully distributed manner without any dedicated central agents like sinks. Each user is hand-off guided by sensors along the route.

### ***Requirements***

We mainly have the following three requirements on the navigation protocol.

- We require that the selected navigation route is *safe*, i.e., the route should be apart from the dangerous areas with guaranteed safety.
- We require that the selected navigation route is *efficient*, i.e., the route should not be excessively long. A shorter route results in a rapider departure from dangers.
- We require that the navigation protocol is *scalable*, i.e., the building and updating of the navigation routes should be local and lightweight.

## III. DESIGN PRINCIPLES

We elaborate the design of our navigation system in this section. The main idea is that we embed a distributed road map system in the sensor network. This road map system is built according to the distribution of dangerous areas and thus can characterize the features of the safety in the field. The navigation system maintains the road map as a public infrastructure across the network and guide different users across the field through the same road map, saving unnecessary overhead of individually planning routes for different users. The road map is updated in an event-driven manner when the dangerous areas vary.

In the following, we in detail describe the design principles in 4 components: *building the road map*, *guiding navigation on the road map*, *reacting to emergency dynamics* and *improving route efficiency*. In each component, we illustrate the design principles in continuous settings and describe their properties.

### *A. Building the Road Map*

We denote the entire emergent field as region $E$ and the combination of dangerous areas as region $D$. Thus the road map is built in the remainder region $R = E \backslash D$, since human beings can only move outside dangerous areas for ensuring their safety.

We build the basic framework of the road map by concatenating the medial axis of region *R*. The medial axis is a set of points, each of which is closest to at least two different points on the boundaries of dangerous areas. Figure 2 shows the basic framework of the road map (blue lines) on the continuous field of dangerous areas. We treat the area out of the sensor field as dangerous, since without any sensing information about such area we have to consider it possibly dangerous.

The consequent road map is then built as Figure 2 (a) depicts. We can also choose to consider the sensor field boundary safe if we have some preliminary information on the boundary, e.g., the sensor field is indoor environment, safely surrounded by walls or fences. In this case, the road map is built as Figure 2 (b) depicts. Since the two cases are essentially similar, without loss of generality, in the following, we mainly focus on the first case.

As proven in [3], the medial axis of region *R* is a finite set of continuous curves and it retains the topological features of this region. It is shown that any bounded open subset in the Euclidean space is homotopy equivalent to its medial axis [3]. For any continuous path in region R, we can find a homotopy transformation which maps the path into a segment of the medial axis. Thus our road map framework is expressive, which captures the topological features of the safe region *R*, representing the possible safe corridors among dangerous areas with curve segments on the medial axis. The road map framework is also compact, which represents the topological and geometric features of region *R* by a simple curve graph, the size of which is proportional to the complexity of large geometric and topological features on *R* [3].

### *B. Guiding Navigation on the Road Map*

We utilize the road map framework as a backbone for navigating different users inside the field. The road map divides region *R* into different cells. Each cell is separated by road segments from others and contains a dangerous area inside it.

#### *1) Connecting the exit to the road map backbone*

First, we find the exit in one of the cells and build a route connecting the exit and the road map backbone. The route is calculated based on the distance from the boundary of dangerous area inside that cell. We assign a virtual power field around the dangerous area in the cell, where the power $p$ of each point is inversely proportional to its distance $d$ from the dangerous area, e.g., $p = 1/d$. The route from the exit extends at each point along the most descending direction of the virtual field until it reaches the border of the cell, i.e., the road backbone.

***Lemma 3.1.*** The local minimum points of the virtual power field in each cell only reside on the medial axis.

**Proof.** Assume there is a local minimum point $s$ inside the cell with power $p = f(d)$, where $d$ is its distance from the dangerous area and function $f$ is monotonically decreasing. As $s$ is a local minimum point, it has the largest distance from the dangerous area within its local neighborhood. As figure 3(a) depicts, we find $r$, the closest point on the boundary of the dangerous area to $s$. Apparently, the distance $|sr| = d$. We prolong $rs$ which on the other side intersects with the $\varepsilon$-neighborhood of $s$ at $s'$. We can find another point $r'$ which is the closest point on the boundary of the dangerous area to $s'$. Since $s$ has the largest distance from the dangerous area within its local neighborhood, $|s'r'| < |sr| = d$, and in the triangle $\Delta ss'r'$, $|sr'| < |s'r'| + |ss'| = d + \varepsilon$. If we let $\varepsilon \to 0$, we have $|sr'| = |sr| = d$. $s$ is closest to the two different points on the dangerous boundary and thus on the medial axis. ■

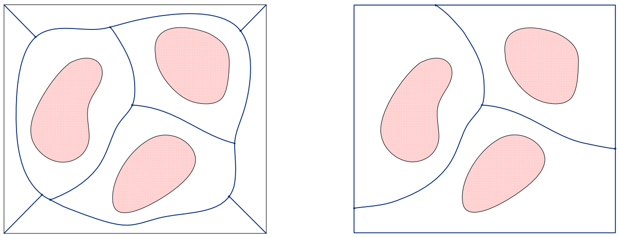
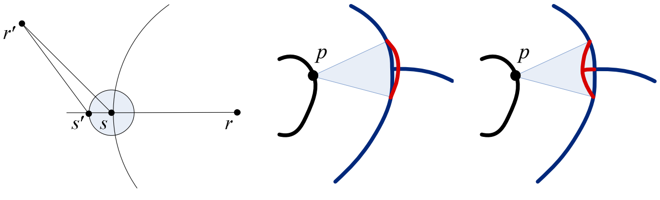

(a) (b)

Figure 2. The basic road map framework



(a) (b) (c)

Figure 3. (a) Local minimum point *s*; Local updating of emergency dynamics, (b) expanding and (c) shrinking

Lemma 3.1 guarantees that we can successfully build a route connecting the exit and the road backbone without halted at an intermediate point of local minimum. Such a route ensures that any point on the route is not any closer to the dangerous area than the destination exit. The route connecting the destination exit intersects the road backbone at a point we call *gateway*, which can be treated as exit on the road backbone.

#### *2) Assigning directions on the road map*

On the road backbone, we accordingly assign directions for each road segment, indicating a safe path towards the gateway for each point on the backbone.

This can be achieved by flooding from the gateway throughout the road backbone. The flooded information includes the closest distance to the dangerous areas, $d_c$, along the road to the gateway, the distance along the road to the gateway, $d_r$, and the direction $D$ along the road. Each point receives the flooded information from different directions. Usually the flooded information only comes from the two directions along the road but possibly multiple directions at the branch points where multiple roads segments intersect. Each point first compares $d_c$ and maintains the path with the largest $d_c$. Among the paths with the same value of $d_c$, the point keeps the shortest path with the smallest $d_r$. It records the direction $D$ of such a path. Finally, each point knows a path towards the gateway and maintains the direction of this path. Figure 4 depicts the finally maintained directions on the road map. If we break from the end of each path, the finally obtained directional road backbone is a continuous tree.

***Lemma 3.2.*** On the road backbone, from any point to the gateway, the path along the assigned directions maximizes the minimum distance to the dangerous areas.

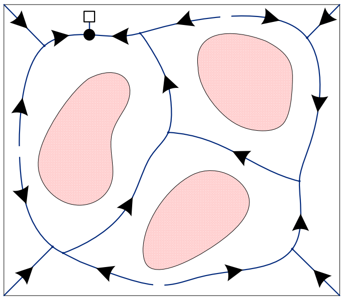


Figure 4. The finally obtained directional road backbone

**Proof.** According to the process of assigning directions on the road map, each point maintains the direction of the path to the gateway with the largest $d_c$. For the path that maximizes the minimum distance to the dangerous areas, the information flooded from this path includes the largest $d_c$ among all paths. If there are multiple such paths, the point keeps the shortest one. Thus the final assigned direction must be towards the path that maximizes the minimum distance to the dangerous areas. ■

#### 3) *Exploring the routes for users*

As we previously mentioned, the road backbone divides the region into different cells. Each user initially resides in one of those cells. Navigating each user to the destination exit includes three stages.

In the first stage, each user is guided from the inside of the cell to the road backbone. The route is calculated similarly as the route connecting the exit and the road map. By assuming a virtual field around the dangerous area in the cell, where the power of each point is inversely proportional to its distance from the dangerous area, the user at each step moves along the most descending direction of the virtual field until he reaches the road map backbone.

Along the road backbone, users from different cells will be navigated towards the gateway. The route is selected simply according to the directions assigned on the road map, i.e., each user moves along the directional roads. The last-mile navigation is guided along the route that connects the exit and the gateway on the road map.

#### 4) *The safety of the navigation route*

We show by following theorems that the selected navigation route provides *global safety* as well as *local safety*.

***Theorem 3.3.*** The selected navigation route maximizes the minimum distance of all possible routes to the dangerous areas.

**Proof.** Assume there is a Max-Min route connecting the user $u$ and the destination exit $v$. Such a route intersects with the road backbone at a series of points that divide the route into $k$ segments. Without loss of generality, we denote the points on the route from $u$ to $v$ as $s_0$, $s_1$, …, $s_k$, where $s_0 = u$ and $s_k = v$. For any intermediate segment $s_is_{i+1}$ we replace it with the road segment between $s_i$ and $s_{i+1}$ on the road backbone. Especially, for each of the two segments $s_0s_1$ and $s_{k-1}s_k$ on the end, we replace it with the route connecting $u(v)$ to the road backbone plus the road segment on the backbone. According to Lemma 3.1, the replaced new route is also a Max-Min route, and it is a route along the road map backbone. This Max-Min route differs with the selected navigation route only at the segment on the road map backbone. According to Lemma 3.2, the segment of the selected route on the road backbone is a Max-Min path among all possible paths on the road map backbone between $s_1$ and $s_{k-1}$. Thus the whole selected navigation route is a Max-Min route. ■

Theorem 3.3 shows that the selected route provides user guaranteed safety in global span. Each user along the selected route never moves unnecessarily close to the dangerous areas. And the safety is globally guaranteed in the sense that we cannot find another route which is more distant away from the dangerous areas.

Only providing the global safety, however, sometimes does not sufficiently utilize the safe space to guide safer behaviors for the user. We show by the following theorem that the selected navigation route also provides local safety for the user movement at each step.

***Theorem 3.4.*** For any given path segment on the selected navigation route, any substitute path will not be farther to the dangerous areas.

**Proof.** Consider a path segment $st$ on the selected navigation route. There are two cases. (1) $st$ is entirely on the road backbone, and (2) $st$ contains a segment on the route between $u(v)$ and the road map backbone. For the first case, we find on the path segment $st$ the point $p$ which is closest to the dangerous areas, and connect $p$ with its $k$ closest points $p_1$, $p_2$, …, $p_k$ on the boundaries of dangerous areas. Any substitute path segment will intersect with one of those lines, e.g., $pp_i$, or otherwise entirely enclose one of the dangerous areas. If it is the former case, we find the intersection point of $pp_i$ and the substitute path which is at least no farther to the dangerous area than $p$. If it is the latter case, we can find a path on the road backbone between $s$ and $t$ that is not closer to the dangerous areas than the substitute path similarly as we do in the proof of theorem 3.3. According to Lemma 3.2, such a path is not farther to the dangerous areas than $st$. Thus the theorem is true in the first case.For the second case, we find on the path segment $st$ the point $p$ which is closest to the dangerous areas. $p$ is on the road backbone or on the route between $u(v)$ and the road map backbone. If it is the former case, it is the same with the first case. If it is the latter case, $p$ must locate at point $t$ due to the construction of this route. Thus the substitute path has at least a point $t = p$ which is of the same distance to the dangerous area as $st$. ■

Theorem 3.4 guarantees that any intermediate local path segment on the selected route yields the largest distance to the dangerous areas. We call this property local safety. Local safety provides the user even stronger safety guarantee at each intermediate step such that at any local step, the selected route guides the user through the safest way.

### C. *Reacting to Emergency Dynamics*

Due to the emergency dynamics, the dangerous areas might vary during the navigation process. For example, as a fire spreads in the field, the borders of dangerous areas vary from time to time. There are several basic types of variation of dangerous areas, including emerging, expanding, shrinking, and diminishing. Obviously, the emergency dynamics introduce extra problems for the navigation. For example, the original medial axis might no longer provide a safe route under the expanding of dangerous areas. We need to rebuild the road backbone according to the variations of dangerous areas. A straightforward but highly inefficient mechanism is to entirely reconstruct the new road backbone whenever the variation of dangerous areas is detected. Such a mechanism introduces both expensive computation and communication costs to the resource limited sensor network. Furthermore, global collaboration will take relatively longer reaction time, not feasible under the emergent situations.

In this section, we describe an updating principle which additively rebuilds the road map according to the emergency dynamics and affects only a local district. We treat any targeted area or curve as a set of points and the variation of dangerous areas is considered as a continuous process of switching a series of points into or out of the dangerous areas. The expanding of a dangerous area corresponds to a point beside the dangerous area is switched into the dangerous area. The shrinking of a dangerous area corresponds to a point on the boundary of the dangerous area is switched out of the dangerous area. The emerging of a dangerous area corresponds to a point inside the safe district is switched into a dangerous area. The diminishing of a dangerous area corresponds to the last point of the dangerous area is switched to be safe.

In our approach, we let each point in the field maintain a status recording the set of the closest dangerous point to it and the distance between them. Each time a point is switched into a dangerous area, we only need to update those points which will take it as their closest dangerous point. Similarly, each time a point is switched out of a dangerous area, we only need to update those points which record it as their previously closest dangerous point.

***Lemma 3.5.*** When the dangerous area in a cell $c$ expands or shrinks continuously, only the points within $c$ are affected.

**Proof.** The border of a cell $c$ is part of the medial axis in the region. Thus all points out of $c$, is closer to at least one dangerous area other than the one within $c$. As the expanding or shrinking corresponds to the variation of the point on or beside the boundary of the dangerous area within $c$, only the points within cell $c$ and the $\varepsilon$-neighborhood of $c$ might be the closest to such a dangerous area and thus can be affected. The updating is within $c$ and its $\varepsilon$-neighborhood. If we let $\varepsilon \to 0$, we can conclude that any continuous variation on the dangerous area in cell $c$ affects only the points within $c$. ■

***Lemma 3.6.*** The emerging of a new dangerous point affects the points within the newly constructed cell and the diminishing of a dangerous point affects the points within the original cell.

**Proof.** The emerging of a new dangerous point creates a cell surrounding it. Such a newly emerged dangerous point is then the closest danger to all points within this cell. All points outside the cell do not have a change on their closest dangerous points and are thus not affected. Along with the diminishing of an existing dangerous point the cell around this point diminishes. All points within the cell are then the closest to other dangerous areas. All points outside the cell do not have a change on their closest dangerous points and are thus not affected. ■

***Theorem 3.7.*** The impact of the emergency dynamics in the field is local.

**Proof.** According to lemma 3.5 and 3.6, any emergency dynamic affects only a local district bounded within the cell of the dangerous area. All points outside this district maintain their original status. ■

Theorem 3.7 indicates that the updating operations are restricted within a local district. To be concrete, as Figure 3 (b) and (c) depict, when the dangerous area varies on point $p$ (expands or shrinks), it only floods and updates the statuses of the points within the sector-like region in the cell. The shape of the road map segment within this region is then accordingly updated by each node measuring the new distances to the closest boundaries of dangerous areas (the red part in Figure 3).

Assume the sizes of a dangerous area $A$ and the cell $c$ surrounding it are $s$ and $s'$ respectively. After expanding or shrinking one point on $A$, there is a sector-like region affected as above discussed. We by the following theorems show that the amortized size of such a region in all cases is $O(\sqrt{n})$, where the emerging or diminishing of the dangerous point is treated as a special case in the amortization when the size of $A$ is 0.

***Lemma 3.8.*** Assume the sizes of a dangerous area $A$ and its surrounding cell $c$ are $s$ and $s'$ respectively. After expanding or shrinking one point on $A$, the average sector-like region affected is of size $O(s'/\sqrt{s})$.

**Proof.** For any point $p'$ in $c$ but not in $A$, there exists one and only one point $p$ on the boundary of $A$ that is the closest dangerous point to $p'$. While the size of $c$ is $s'$ the perimeter length of $A$ is $\Omega(\sqrt{s})$. Thus the average size of the sector-like region affected by the point on the boundary of $A$ is $s'/\Omega(\sqrt{s}) = O(s'/\sqrt{s})$. ■

***Theorem 3.9.*** The amortized size of the sector-like region affected by the variation of dangerous areas is $O(\sqrt{n})$.

**Proof.** We let the size of the cell $c$ vary from 0 to the entire field $n$ and let the dangerous area $A$ inside $c$ vary from 0 to the size of the entire cell $s'$. Thus we have the amortized size of the affected regions in all cases is

$$D = \int_{s'=0}^{n} \Big( \int_{s=0}^{s'} O(s'/\sqrt{s})/s'ds \Big)/nds' = O(\sqrt{n}) \cdot \;\blacksquare$$

According to theorem 3.9, in each updating process when emergency varies, our method rebuilds the shape of the new road map with the cost of flooding and updating a local district of an amortized size $O(\sqrt{n})$. After rebuilding the shape of the new road map, we accordingly reassign the directions along the road backbone, which incurs local overhead on the compact backbone.

## IV. IMPLEMENTATION EXPERIENCE

In Section III, we elaborate the design principles of our navigation protocol in continuous settings. We show by theorems that our design provides safety, efficiency and scalability for the navigation process. Nevertheless, to implement such a protocol for real usage in practice, we need to carefully address some technical issues when applying our principles in the discrete sensor network.

### *A. Protocol Implementation*

As we do not have any location information in the network, we cannot obtain accurate distance measurement during system operations. Instead, we approximate the proximity of two nodes by the number of hops along their shortest communicational path in the connectivity graph.

Each sensor node maintains a list of variables, which record the status of this node (see Table I). *s.danger* marks whether the current node resides within or outside a dangerous area. *s.danger* is 0 if the current node is outside the dangerous area and *s.danger* is set to the ID of the dangerous area if the current node resides in such a dangerous area. *s.border* is a boolean variable that indicates whether the current node is on the boundary of the dangerous area. *s.mDist* records the distance from the current node to the nearest dangerous area, and *s.mSet* records the set of nodes on the boundaries of dangerous areas that are of *s.mDist* to the current node. *s.road* is a boolean variable that indicates whether the current node is on the road map backbone. *s.nextHop* and *s.rDist* are two variables that describe the status information for the node on the road backbone. *s.nextHop* stores the ID of the next hop node along the path direction on the road. *s.rDist* records the minimum distance to the dangerous areas on the path from the current node to the exit. *s.potential* records the potential value of ordinary nodes.

When emergency happens, each sensor node first senses the environment to detect dangerous areas. According to the sensing results, each sensor node determines whether it is inside or outside a dangerous area. Each node inside a dangerous area generates a random number as the ID of the dangerous area and floods it across the area. The ID of smaller value suppresses other IDs and the smallest ID dominates the entire area. Finally, different dangerous areas are set to different IDs. The nodes inside a dangerous area set *s.danger* to the ID value and those outside dangerous areas set *s.danger* to 0. The node on the boundary of the dangerous area easily detects its position from its neighborhood information and sets *s.border* to Y.

At the second stage, the nodes on the boundaries of dangerous areas flood the boundary information to the rest of the field. Each node accordingly records its distance to the nearest boundary nodes, *s.mDist*, and stores the corresponding boundary nodes in *s.mSet*. Each node examines *s.mSet*. If it contains boundary nodes on two or more dangerous areas, this node sets *s.road* to be Y, dedicated as a backbone node on the medial axis. In practical usage, however, when sensor nodes are sparsely deployed over the field, the nodes on the road backbone may not be connected into one component. To solve this problem, we compromise the accuracy of the medial axis, letting *s.mSet* of each node stores the boundary nodes of both *s.mDist* and *s.mDist* + 1 distance to it. By such means, the medial axis is indeed broadened and more nodes are dedicated on the road backbone, largely increasing the connectivity of the road backbone. The backbone nodes separate the sensor network into different components, corresponding to different cells of dangers in the field. As described in Section III, each ordinary node in each cell is assigned the potential value *s.potential* to be 1/*s.mDist*. The exit thus accordingly explores a route connecting to the backbone and the user can also explore their routes to the backbone based on the calculated potential fields in the cells.

Table I.

| variable | type | Size (bits) |
|---|---|---|
| *s.danger* | danger ID | 8 |
| *s.border* | Y/N | 8 |
| *s.mDist* | hops | 8 |
| *s.mSet* | node IDs | 80 |
| *s.road* | Y/N | 8 |
| *s.nextHop* | node ID | 8 |
| *s.rDist* | hops | 8 |
| *s.potential* | Value | 8 |

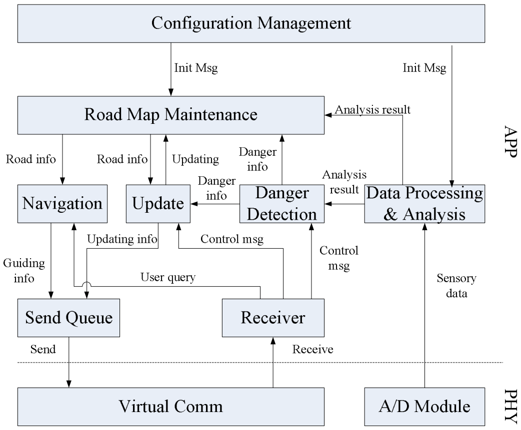


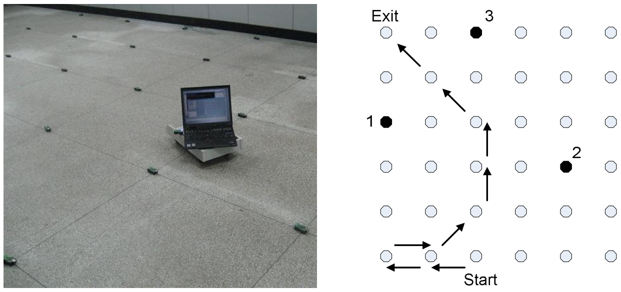
Figure 5. The block diagram of the system architecture

The gateway node then floods the exit information throughout the road backbone. The flooded message contains two items, $d_c$, which records the minimum number of hops to the dangerous areas along the road from the current node to the gateway, $d_r$, which records the number of hops along the road from the current node to the gateway. According to the principle described in Section III.B, each node initially sets its *s.nextHop* to be null, and *s.rDist* to be 0. On receiving the flooded message, each node first compares its *s.rDist* with $d_c$ in the message. If *s.rDist* $< d_c$, this node switches its *s.nextHop* to be the ID of the node that forwards the message and sets its *s.rDist* to be $d_c$. Then this node alters $d_c$ in the message to be min($d_c$, *s.mDist*) and resends this message. Otherwise this node simply discards the message.

When the dangerous area varies, nodes in its cell react to update the road backbone. Following the principle described in Section III.C, each time a sensor node is switched into or out of the dangerous area, it generates a report and floods it within those nodes that record it in their *s.mSet*. Those nodes accordingly update their *s.mDist* and *s.mSet*. The potential values *s.potential* of those nodes are also updated. The road map backbone is then updated, and the corresponding nodes update their *s.road*. As proven in theorem 3.7, such an operation only affects the nodes within a local sector-like district. The gateway node reinitiates a flood on the road backbone to reassign the path directions.

### *B.Prototype Experiment*

We implement the navigation system on the TelosB motes and a user interface on the laptop. Figure 5 depicts the block diagram of the architecture of our navigation system. The “Configuration Management” component manages the configuration information of the node, including node ID, sensor detection sensitivity, neighborhood status, and etc. The



(a) (b)

Figure 6. Experiment setup. (a) The miniature prototype deployment; (b) A showcase of the interactive experiment

"Road Map Maintenance" component accepts information composed of both local sensory results and network control messages. This component manages the status parameters of the node and generates guidance information for navigating users. It also provides updating information to other neighbor nodes for maintaining the road map backbone in the network. We implement a prototype system including 36 TelosB motes deployed into 6×6 grids in the atrium in the university campus, with 10 meter space in-between neighboring nodes. The system provides navigation for a person carrying a laptop computer or PDA that can talk with sensors. Figure 6 (a) exhibits a miniature deployment of our prototype system in the laboratory [9]. Our navigation protocol builds the road map infrastructure across the monitoring field and provides the route to guide the user out safely. The route is represented by a sequence of sensor nodes and the user is directed along those sensor nodes.

We design a black box challenging game to validate this system. At each step an internal user is provided the direction pointing from his current stop towards his next stop. At present, we do not equip the laptop with the antenna array that can detect the direction of received signals. Alternatively, we pre-load the positions of the sensor nodes in the laptop to facilitate the calculation of directions. Nevertheless, such information will not be revealed to the user so that the experiment truly demonstrates the effectiveness of our system when used in the location-free environment. The other participant uses a PC connecting to the sensor network. He behaves as a challenger to this navigation system, who manages the dangers within the field by setting certain areas from safe to dangerous and vice versa, simulating the emergency dynamics including danger emergence, disappearance, expanding and shrinking. The frequency and intensity of the update on the dangers represent the extent of the emergency dynamics. Neither of the two participants is aware of the other's operations. The person in the field moves according to the indications received from the navigation system. The challenging person freely sets the dangers without knowing the navigation progress. Such an interactive experiment achieves more than 95% success rate and validates the effectiveness of our navigation system under different emergent situations.

Figure 6 (b) showcases an instance of the interactive experiment, where the challenger sequentially sets the numbered sensor nodes to be dangerous areas when the internal user moves. Our navigation system accordingly guides the user along the route marked by arrows. Note that the user sometimes needs to go backwards to explore safer routes when emergency varies.

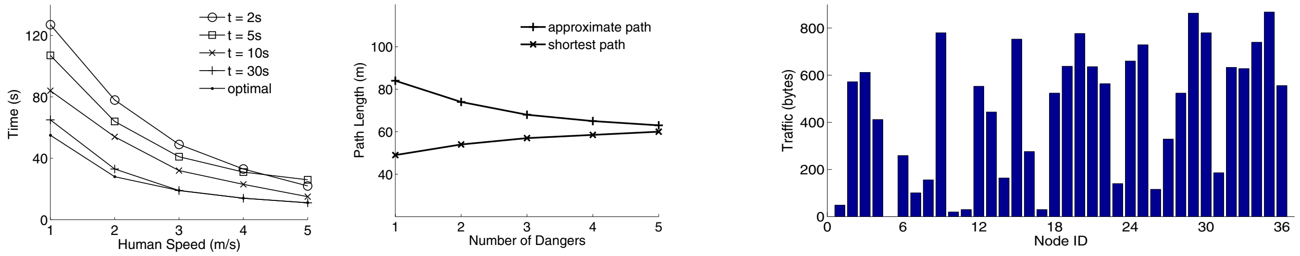
Figure 7 (a) depicts the time it takes to navigate the internal user out of the field. We vary the walking speed of the user as well as the frequency of updating the dangerous areas. The x axis represents different walking/running speeds of the user and the y axis represents the time of navigation. Different curves are recorded when we change the dangerous areas with different time intervals. We run 20 tests for each set of parameters. Apparently, a lager walking speed and a lower updating frequency leads to faster navigation. When the walking speed is 3m/s, our approach reaches nearly optimal navigation time if the environment is relatively static (30s updating interval). It reaches less than 3 times the optimal value even when the environment is highly instable (2s updating interval for the dangers).

Figure 7 (b) compares the length of the approximate path yielded in our navigation protocol with that of the theoretically shortest safe path. As the number of dangerous areas increases, the length of the shortest path increases from less than 50 meters to nearly 60 meters. Contradictive to the intuition, the length of the approximate path given by our protocol decreases as the number of dangerous areas increases. That is because more dangerous areas restrict the safest path from going a far way. The ratio of the two lengths decreases from around 2 to nearly 1.

Figure 7 (c) shows the cumulative traffic cost of each individual sensor node after 4 rounds of road map update, including the regular messages as well as retransmissions in case of packet loss. The y axis measures the bytes of received message. Most sensor nodes have similar traffic cost ranging from 600 to 800 bytes. Some of the nodes have apparently lower traffic cost, since they have been chosen as dangerous areas, thus not participating in navigating the users. The result in this figure shows that our protocol evenly distributes the work load over the entire network.

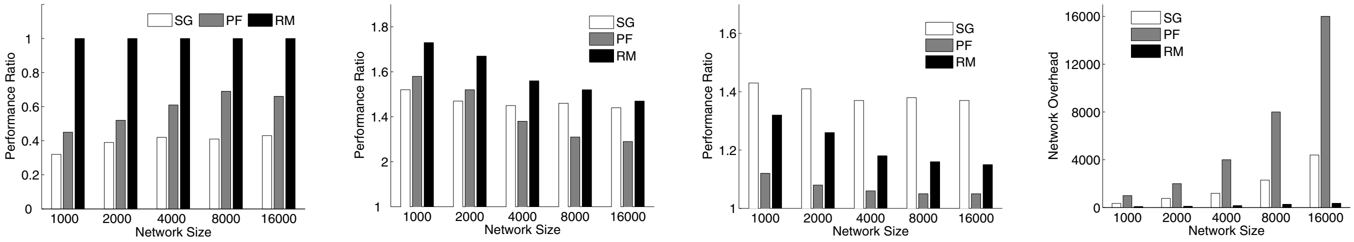
## V. PERFORMANCE EVALUATION

We conduct large-scale simulations to further evaluate the effectiveness and scalability of our approach. We compare the performance of this design with. the skeleton graph based approach proposed by Buragohain et al. [4] as well as the potential field based approach proposed by Li et al. [10]. We simulate randomly deploying sensor nodes in a rectangular area with an average node degree of 28, the same with that assumed in [4]. To examine the scalability, we vary the size of the field and the number of deployed nodes, while retaining the same network density. The network size ranges from 1000 to 16000. For each trial of the network size, we randomly generate 10 internal users in the field to navigate them outward and we take 20 runs, randomly inserting dangerous areas into the field. The number of inserted dangerous areas is uniformly randomly chosen from 3 to 6 and the size of each dangerous area is kept below 5% of the total field size. Indeed, our navigation protocol does not rely on any location information before it works. Nevertheless, since the two approaches we compare with all assume the availability of locations, in the simulations, we record the locations of all the nodes and reveal them to those two approaches to facilitate their operation.



(a) (b) (c)

Figure 7. The experimental results. (a) The time of navigation with different walking speeds and updating frequencies of dangerous areas; (b) The length of the navigation route compared with the shortest path; (c) The traffic cost of each node.



(a) (b) (c) (d)

Figure 8. Comparative results of the three approaches. SG represents the skeleton graph based approach proposed in [4]; PF represents the potential field based approach proposed in [10]; RM represents our road map based approach. (a) Performance ratio of the minimum distance to the danger; (b) Performance ratio to the shortest path; (c) Performance ratio to the minimum exposure path; (d) Average network overhead for updating the network in the event of changes in dangerous areas.

For the skeleton graph based protocol (SG for short), we choose the version based on adaptive skeleton graph, which has been shown superior to the uniform one in their original paper. For the potential field based protocol (PF for short), we choose the function of calculating the potential value to be $1/\text{dist}^2$, which has been used all through in their original paper. We compare the performance of our road map based approach (RM for short) with the two approaches from the following four aspects.

### *A. Minimum Distance to the Danger*

We first evaluate the absolute safety of the routes planned in the three approaches. We conduct our tests under static environment with fixed dangerous areas. Let $d$ denote the minimum distance from the planned route to the dangerous areas, and $d_{OPT}$ denote the minimum distance to the dangerous areas from the optimal path that maximizes $d$. The performance ratio is defined to be $d/d_{OPT}$. A larger ratio indicates a better safety of the planned route, as the minimum distance from the route to the dangerous areas is larger.

Figure 8 (a) shows the performance ratio of the three approaches. We can see that the proposed RM approach achieves the optimal result with the ratio = 1. Indeed, such a result is theoretically guaranteed by theorem 3.10. For the other two approaches, PF has a performance ratio around 0.6 while SG has a performance ratio below 0.4, due to the fact that SG is prone to guide the user close to the dangerous areas to achieve the shortest path on the skeleton graph.

### *B. Shortest Path*

We evaluate the path efficiency by comparing the length of the route planned in each approach $l$ with the length of the shortest path that does not cross the dangerous areas $l_{OPT}$. The performance ratio is defined to be $l/l_{OPT}$. A smaller ratio indicates a more efficient route, as the route is closer to the theoretically shortest safe path.

Figure 8 (b) shows the performance ratio of the three approaches under different network sizes. While SG keeps the ratio unchanged around 1.5, PF and our RM have decreased ratio as the network size is increased. That is probably because the hop count based distance measurement in the two approaches becomes more accurate when more sensor nodes are involved as the network scale increases. When the network size is increased to 16000, PF reaches the ratio of less than 1.3 and our RM reaches the ratio around 1.5.

### *C. Minimum Exposure Path*

By comparing the exposure value of the route planned in each approach with that of the minimum exposure path, we evaluate the cumulative safety along the route planned in each of the three approaches. The exposure value at each point of the route is calculated as $1/\text{dist}^2$. The performance ratio is defined as $S/S_{OPT}$, where S is the exposure along the route planned in each approach and $S_{OPT}$ is the exposure along the optimal path. A smaller ratio indicates a higher cumulative safety along the path.

Figure 8 (c) shows the performance ratio of the three approaches under different network size. SG has the highest ratio around 1.4 thus the lowest cumulative safety. PF has the lowest ratio around 1.1. Our RM performs in between with the performance ratio decreasing from 1.32 to 1.15 as the network size increases from 1000 to 16000.

Note that, such a set of tests indeed favors PF since the function for calculating the exposure value is chosen the same as the function for calculating the potential value of PF maximizing its performance.

### D. Update Overhead

Finally, we evaluate the network overhead incurred by the three approaches in the event of changes in the dangerous areas. We simulate a series of continuous changes in the dangerous areas and measure the average message transmissions in each round of network update process. Figure 8 (d) compares the network overhead of the three approaches. PF incurs the largest overhead as it relies on flooding the entire network to recalculate the potential value of each sensor node and accordingly rebuild the routes. Such a network cost is proportional to the network size. SG incurs relatively smaller overhead, yet proportional to the network size, as in SG the skeleton graph needs to be rebuilt once a dangerous area changes. The proposed RM incurs the least overhead as in RM only local communication is needed to update the road backbone when dangerous areas change. Hence, RM is scalable as the update overhead is merely related to the size of the network.

## VI. RELATED WORK

Path planning and navigation has been a crucial issue studied in the fields of Robotics [2, 8] and Computational Geometry [1] for a long time. Guiding navigation in wireless sensor networks, however, faces non-trivial challenges that have never been considered in previous works. The navigation process needs to be done in distributed manner over a self-organized network consisting of a huge number of sensor nodes. The environment conditions are no longer presumed as shared information globally available. Such information is instantly captured by the distributed sensor nodes. The navigation system needs to adapt and react to the environmental variations.

Li et al. first propose a distributed algorithm that explores the minimum exposure path for guiding navigation [10]. Their potential field based approach largely relies on exhaustive search over the entire network. Buragohain et al. propose to abstract the field by the skeleton graph and accordingly find navigation routes over the skeleton graph [4]. Some studies address the problem of finding the minimum or maximum exposure path in a network. Meguerdichian et al. [11] and Veltri et al. [15] propose heuristics to distributedly compute such paths. Although being similar with our navigation problem, finding the exposure path does not explicitly address the issue of navigating users among dangerous areas, treating individual sensor nodes as adversaries rather than utilizing them as infrastructures. Most of existing studies assume the availability of location information and consider a static field without changes in dangerous areas.

Data routing in sensor networks [3, 6, 7, 14] shares some similarity with the navigation problem we address in this work. Nevertheless, as we previously mentioned, navigating a human being essentially differs with routing a packet, as human navigation is a safety-critical and time consuming process, resulting in different design paradigms from data routing schemes.

## VII. CONCLUSIONS

We propose a road map based approach that provides human navigation in the distributed sensor networks. Primarily different from existing works, we validate our design without relying on location information, which surprisingly overcomes natural intuitions. We further discuss the situation in the event of emergency dynamics, which has not yet been explored by previous studies. We also introduce an updating scheme that locally updates the road map system in the network when the dangerous areas vary, which largely reduces the network overhead. We implement a prototype system consisting of 36 sensor nodes. Through a black box challenging game, we validate the effectiveness of our design. We further evaluate the performance of our approach through large scale simulations as well as compare it with two existing approaches. The simulation results show that although with much relaxed assumptions, our approach achieves comparable performance with significantly reduced communicational overhead.

## 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, NSFC grant No. 60803152 and NFSC Key Project grant No. 60533110.

## REFERENCES

[1] M. d. Berg, O. Schwarzkopf, M. V. Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications, 2nd Edition*: Springer-Verlag, 2000.

[2] S. Bhattacharya, N. Atay, G. Alankus, et al., "Roadmap Query for Sensor Network Assisted Navigation in Dynamic Environments," in Proceedings of DCOSS, 2006.

[3] J. Bruck, J. Gao, and A. A. Jiang, "MAP: Medial Axis Based Geometric Routing in Sensor Network," in Proceedings of ACM MobiCom, 2005.

[4] C. Buragohain, D. Agrawal, and S. Suri, "Distributed Navigation Algorithms for Sensor Networks," in Proceedings of IEEE INFOCOM, 2006.

[5] T. He, S. Krishnamurthy, J. A. Stankovic, et al., "Energy-Efficient Surveillance Systems Using Wireless Sensor Networks," in Proceedings of ACM MobiSys, 2004.

[6] Q. Huang, S. Bhattacharya, C. Lu, and G. C. Roman, "FAR: Face-Aware Routing for Mobicast in Large-Scale Sensor Networks," in Proceedings of IEEE INFOCOM, 2004.

[7] B. Karp and H. T. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks," in Proceedings of ACM MobiCom, 2000.

[8] J. C. Latombe, *Robot Motion Planning*: Kluwer, 1992.

[9] M. Li, J. Wang, Z. Yang, and J. Dai, "Demo Abstract: Sensor Network Navigation without Locations," in Proceedings of ACM SenSys, 2008.

[10] Q. Li, M. D. Rosa, and D. Rus, "Distributed Algorithms for Guiding Navigation across a Sensor Network," in Proceedings of ACM MobiCom, 2003.

[11] S. Meguerdichian, F. koushanfar, G. Qu, and M. Potkonjak, "Exposure in Wireless Ad-Hoc Sensor Networks," in Proceedings of ACM MobiCom, 2001.

[12] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AOA," in Proceedings of IEEE INFOCOM, 2003.

[13] R. Szewczyk, A. M. Mainwaring, J. Polastre, et al., "An Analysis of A Large Scale Habitat Monitoring Application," in Proceedings of ACM SenSys, 2004.

[14] P. Thulasiraman, S. Ramasubramanian, and M. Krunz, "Disjoint Multipath Routing to two Distinct Drains in a Multi-Drain Sensor Network," in Proceedings of IEEE INFOCOM, 2007.

[15] G. Veltri, Q. Huang, G. Qu, and M. Potkonjak, "Minimal and Maximal Exposure Path Algorithms for Wireless Embedded Sensor Networks," in Proceedings of ACM SenSys, 2003.

[16] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: Overload Traffic Management using Multi-Radio Virtual Sinks," in Proceedings of ACM SenSys, 2005.