CANS: Towards Congestion-Adaptive and Small Stretch Emergency Navigation with Wireless Sensor Networks

Chen Wang, *Member, IEEE*, Hongzhi Lin, and Hongbo Jiang, *Senior Member, IEEE*

Abstract—One of the major applications of wireless sensor networks (WSNs) is the navigation service for emergency evacuation, the goal of which is to assist people in escaping from a hazardous region safely and quickly when an emergency occurs. Most existing solutions focus on finding the safest path for each person, while ignoring possible large detours and congestions caused by plenty of people rushing to the exit. In this paper, we present CANS, a Congestion-Adaptive and small stretch emergency Navigation algorithm with WSNs. Specifically, CANS leverages the idea of *level set method* to track the evolution of the exit and the boundary of the hazardous area, so that people nearby the hazardous area achieve a mild congestion at the cost of a slight detour, while people distant from the danger avoid unnecessary detours. CANS also considers the situation in the event of emergency dynamics by incorporating a local yet simple status updating scheme. To the best of our knowledge, CANS is the first WSN-assisted emergency navigation algorithm achieving both mild congestion and small stretch, where all operations are in-situ carried out by cyber-physical interactions among people and sensor nodes. CANS does not require location information, nor the reliance on any particular communication model. It is also distributed and scalable to the size of the network with limited storage on each node. Both experiments and simulations validate the effectiveness and efficiency of CANS.

Index Terms—Emergency navigation, WSN-assisted, congestion-adaptive, small stretch, level set method

1 Introduction

Recent advances in wireless sensor network (WSN) technologies provide us the ability of pervasive usage of sensors widely deployed over the fields of interest [1], [2]. While most earlier works are designed principally for intensive sensor data collection to monitor the physical environment, increasing interests are arousing in exploring the possibility of in-situ interactions between people and their physical environment [3], [4], [5], which could significantly expand the capability and usability of WSNs.

One important application of such in-situ interactions is WSN-assisted emergency navigation [6], [7], where the WSN infrastructure is utilized as a cyber-physical system. In this mobile environment, the internal users are equipped with PDAs or smart phones that can talk with the sensors [4], [8]. When emergency occurs, the WSN explores the emergent field and provides necessary guidance information to users, so that users can be guided to move out of a hazardous region through ubiquitous interactions with sensors.

At a first glance, one may have thought of a trivial usage of traditional packet routing protocols for guiding navigation. However, there are several inherent features of emergency navigation that significantly distinguish itself from packet routing schemes. Firstly, the navigation of human beings seeks for a safe-critical path, other than packet loss or energy efficiency which is the first priority as in packet routing. Note that here the safety of a path not only means to be far away from a hazardous area, but also refers to mild congestion, less detour as well as fast reaction to an emergency. Secondly, human navigation consumes much more time than traditional packet routing process, due to the limited movement speed of people. While during one packet delivery process the network is often considered static, human navigation in contrast deals with emergency dynamics almost all along the guiding process. As such, we cannot simply borrow existing packet routing schemes for emergency navigation with WSNs.

A diversity of specifically designed solutions for emergency navigation with WSNs has been proposed. Earlier approaches [6], [8], [9], [10] rely on either exhaustive network-wide flooding or the availability of location information on each sensor/user. The followup studies [4], [11], [12], [13] release the requirement of location information, and begin to consider the impact of variations of dangerous areas, which greatly enhance their applicability to more practical scenarios. Most if not all of these location-free methods seek for a global topological structure embedded in the network as the public infrastructure, through which different users can be safely guided to the exit and avoid unnecessary overhead of individually path planning. However, these methods neglect the underlying congestion and detour problems [3], which are critical for a fast evacuation, as they mainly focus on finding the shortest/safest path for each person, while other sub-optimal (yet safe) paths are left unused throughout most of the evacuation process.

• *The authors are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, P.R. China.*
E-mail: {cwangwhu, eihongzhilin2012, hongbojiang2004}@gmail.com.

Manuscript received 30 Mar. 2015; revised 24 May 2015; accepted 19 June 2015. Date of publication 1 July 2015; date of current version 31 Mar. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2015.2451639

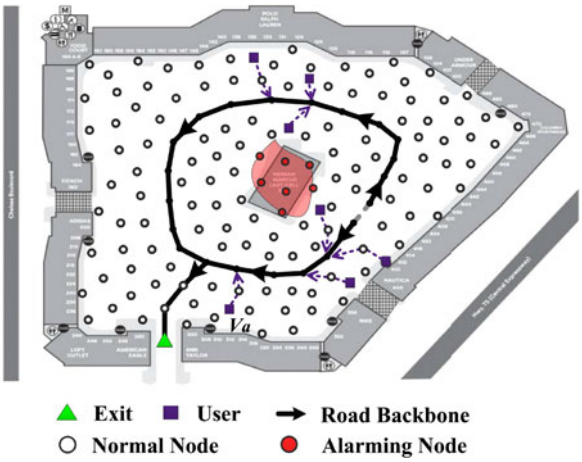


Fig. 1. Allen Premium Outlets in Texas. This network is homotopically equivalent to those shown in Figs. 2a and 7a. The medial axis based road map approach [4], [11]: all users are guided to the directional backbone of the road map, potentially resulting in heavy congestions and large detours.

Let us take one of the typical location-free emergency navigation approaches, the road map based method [4], [11], for instance. In this scheme, a distributed road map is embedded across the sensor network as a common facility for providing guidance information for internal inquirers. Specifically, the road map is built by connecting the medial axis[1] of the network, with a tail route concatenating the exit and the directions assigned for each road segment. To explore the safe path, each user is first led to the road backbone, then moves along the preset directional roads, and finally follows the directional tail route to evacuate from hazardous areas. Fig. 1 illustrates the road map and navigation instances of Allen Premium Outlets in Texas. Though the obtained navigation path for each user is safety guaranteed (in the sense that it is the most distant away from the dangerous areas), the number of people is more likely to exceed the safety capacity of the road map (especially the tail route), thus leading to congestions or even stampedes, as different users are guided through the same single-lane road map (e.g., the road backbone in Fig. 1) with the fixed direction. What is more, since users are required to first access to the identical road backbone wherever they are at the beginning, it would potentially result in large path detours and thus large path stretch (defined as the ratio between the length of the actual and optimal shortest paths). For instance, in Fig. 1, when emergency occurs, the user V_a near the exit has to mechanically turn backward toward the road backbone, involved in the crowd, and finally get to the exit in a great circle. Both heavy congestions and large detours inevitably give rise to a longer period of time for users remaining in danger, which would amplify users' panic and eventually decrease their chances of survival.

Our approach. To address the problems of above-mentioned heavy congestion and large detour, we present CANS, a Congestion-Adaptive and small stretch emergency Navigation algorithm with WSNs. We bring a novel *level set* perspective and do not seek for a topological structure that captures the global features of the network. Instead, we focus on local topology, specifically the regions around the hazardous areas. Our idea is based on the following observation: users far away from the emergency are often sufficiently safe so that they can tolerate more congestions, while users close to the dangerous areas are vulnerable to congestions. On this basis, we propose to construct a compound map of the network, including a so-called potential map and a so-called hazard level map, for emergency navigation. The potential map tracks the inverse evolution of level sets towards the exit, and thus develops a sense of navigation direction by which users can safely head for the exit with few detours, while the hazard level map tracks the evolution of level sets off the hazardous areas, and therefore can naturally tell where the hazardous areas are, and further assist users near the emergency to be branched into different paths to avoid heavy congestions. By doing so, users can be guided to the exit in a realistic safety way: users nearby the hazardous areas achieve a mild congestion at the cost of a slight detour, while people far away from the danger avoid unnecessary detours.

To the best of our knowledge, CANS is the first WSN-assisted emergency navigation algorithm achieving both mild congestion and small stretch, where all operations are in-situ carried out by cyber-physical interactions among users and sensor nodes. It is distributed and scalable to the size of the network without reliance on user/sensor's location information. It also considers the situation in the event of emergency dynamics, and designs an updating scheme that locally updates the hazard level map when the hazardous areas vary in time.

The remainder of this paper is organized as follows. In Section 2, we describe the problem specification. In Section 3 we expound upon the theoretical foundations, and in Section 4 we introduce CANS algorithm in detail, with further discussions in Section 5. Sections 6 and 7 conducts small scale experiments and extensive simulations, respectively, to evaluate the performance of CANS. Section 8 briefly overviews the related work. Finally Section 9 concludes the paper.

2 Problem Formulation

We represent the WSN as a connected (undirected) graph $G = (V, E)$, where $V = \{v_i\}$ is the set of sensor nodes, and $E = \{e_{ij} = (v_i, v_j)\}$ denotes the set of communication links between neighboring nodes. The neighbor set of a node $v \in V$ is denoted as $ngb(v)$, where $ngb(v) = \{v' | (v, v') \in E\}$.

We consider the scenario of navigating internal users in the network under emergencies, where there might be several hazardous areas, e.g., fire or poisonous gas, that threaten the safety of the users. The hazardous areas might emerge, disappear, expand, or shrink as the time passes, but the emergency is assumed to vary in only one direction. Users need to be guided out of the field as quickly as possible while keeping away from the hazardous areas.

Assumptions. We assume that each sensor in the WSN has the ability to sense information about its immediate environment, e.g., with temperature sensors or biochemical sensors. We consider a simple *sensor function* $s : V \to S$, where $S = \{0, 1\}$, representing whether it resides "out" or "in" a

1. In a WSN, the medial axis [14], or the skeleton [15], is a connected curve made up by a set of nodes with at least two closest neighbors on the (inner/outer) boundaries of the network.

hazardous area. We assume that users take along communicating devices like smart phones or 802.15.4 compatible PDAs that can communicate with the sensors. Meanwhile, users are able to track surrounding sensors by measuring the strength and direction of wireless signals [16]. In the event of an emergency, the sensor network explores the emergent field, and the users are guided to the preknown safe exit bypassing the hazardous areas through cyber-physical interactions with the sensors.

It is noted that, we assume that users are not surrounded by hazards so that we can always find one escaping path for emergency navigation. For cases in which people are fully surrounded by hazards, they may have to rely on the helicopter rescues or wait for rescuers to remove the obstacles.

Objectives. The objective of an emergency navigation algorithm is to plan for each user a path that is apart from the hazardous areas with guaranteed safety while avoiding excessive congestions and unnecessary detours. In particular, the following features of the navigation algorithm are expected:

- *Safety Guaranteed*. The navigation should be apart from the hazardous areas with guaranteed safety. Note that "safety" in previous studies has different definitions, but the bottom line is that any point on the path should not be within the hazardous area.
- *Congestion-Adaptive*. People are expected to be diverted onto different paths, and thus alleviating congestions.
- *Efficient and Scalable*. The path stretch cannot be too large; the building and updating of the paths are preferred to be local and lightweight.
- *Distributed and Location-free*. Centralized operations and the location devices/algorithms are undesired and not required.

Our proposed CANS algorithm fulfills above goals by constructing a compound map from a level set perspective. In the following, we first present the theoretical foundation of how CANS constructs the compound map in a continuous setting, and then describe how to adapt these ideas to a discrete WSN for emergency navigation.

3 Theoretical Foundation

3.1 Level Set Method

Consider a smooth, closed parametric curve $\gamma(p,0) : [0,1] \rightarrow \mathbb{R}^2$ in an Euclidean plane, and let $\gamma(p,t)$ the family of curves generated by the movement of the initial curve $\gamma(p,0)$ along its normal direction $\mathcal{N}$ with a moving speed $\mathcal{F}$, i.e.,

$$\frac{\partial}{\partial t}\gamma(p,t) = \mathcal{F}\mathcal{N}, \tag{1}$$

where p parameterizes the given curve and t parameterizes the family of curves.

In order to implement the above curve evolution, we can take a Lagrangian approach by discretizing the curve into a set of nodes and updating the node positions in time using a numerical approximation to Eq. (1) [17]. The main drawback of this approach is the difficulty of dealing with topological changes as the moving front splits and merges.

To tackle above issue, the level set method, initially developed by Osher and Sethian [18], represents the evolving curve as the zero level set ($\phi(\mathbf{x},t) = 0$) of a smooth (at least Lipschitz continuous) level set function $\phi(\mathbf{x},t)$, where $\mathbf{x} \in \mathbb{R}^2$. Using Eq. (1) and taking the derivative of $\phi(\mathbf{x},t) = 0$ with respect to time, we can obtain by the chain rule the flow guiding the propagation of $\phi(\mathbf{x},t)$:

$$\frac{\partial}{\partial t}\phi(\mathbf{x},t) = \mathcal{F}|\nabla\phi(\mathbf{x},t)|, \tag{2}$$

where $|\nabla\phi|$ denotes the gradient norm of ϕ.

Above level set formulation actually establishes a connection between the family of moving curves $\gamma(p,t)$ and the family of evolving surfaces $\phi(\mathbf{x},t)$, where the zero level sets of the function $\phi(\mathbf{x},t)$ always yield the moving front of the evolving curve. Note that in numerical computations the time-dependent level set formulation can be converted to a stationary one so that it can work with a fixed mesh [19], which has been proved to be accurate even for relatively large mesh sizes [20]. In view of the discrete nature of WSNs, we adopt the stationary level set method for our design.

There are several favorable features of the level set method, for which it is an ideal choice for supporting emergency navigation. First, it offers natural support on the estimation of the local geometric properties of the evolving curve, which can be used for tracking regions around the hazardous areas where people have to be dispersed. Second, every point on the evolving curve can reach the initial curve by tracing the gradient of ϕ, which provides a potential sense of direction for users to escape. Thirdly, it is able to account for topological changes, indicating possible simple procedures to react the emergency dynamics.

Recall that our idea of emergency navigation algorithm is to construct a topological structure, which on one hand schedules a safe path for each user with less detours, and on the other hand highlights the regions near the hazardous areas to disperse users to different paths. Therefore, we propose two level set variations for our specifical purpose. Combining both the variations brings about the compound map.

3.2 Level Set Variations for Emergency Navigation

To construct the level set variations, we have to define appropriate level set functions. Although all level set functions are equally good in theory, the signed distance function, as represented in the original paper, is preferred in practice for its stability in numerical computations[21]:

$$\phi(\mathbf{x}, t=0) = \pm\,\mathrm{dist}(\mathbf{x}), \tag{3}$$

where $\mathrm{dist}(\mathbf{x})$ is the distance from $\mathbf{x}$ to $\gamma(p,0)$ and the plus (respectively, minus) sign is used if $\mathbf{x}$ is interior (respectively, exterior) of $\gamma(p,0)$.

As the emergency is assumed to vary in only one direction, the signed distance function is no longer suitable for the usage as the level set function for our purpose. To tackle this problem, the notion of *distance transform* is introduced.

Distance transform. In computer vision community, distance transform [22] describes the shortest distance of any given point inside an object to the boundaries of the object. Specifically, let $\mathcal{D}$ denote an object, and $\partial\mathcal{D}$ the boundaries of $\mathcal{D}$. We refer to $d_E(x,y)$ as the distance between two points

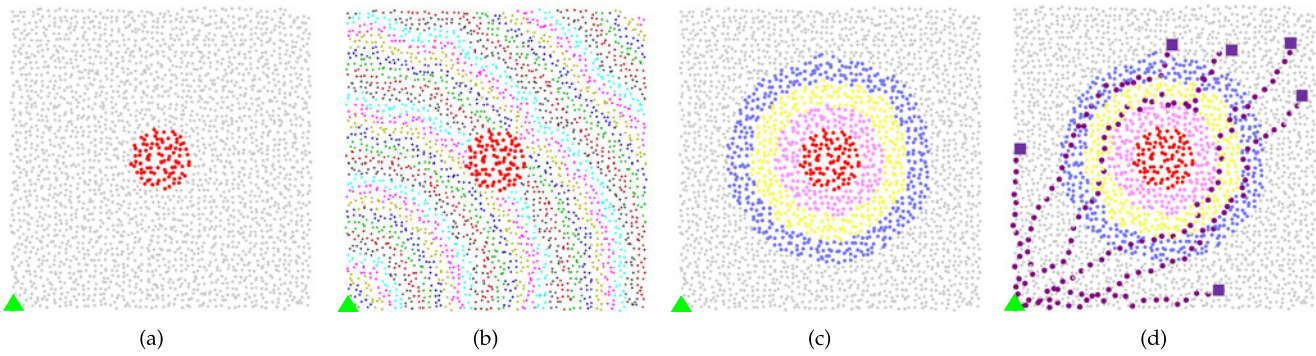


Fig. 2. The pipeline of CANS. (a) A sensor network with 2610 nodes, avg deg 13.9; with one exit (the green triangle) and one hazardous area (colored in red). (b) The potential map initiated by the flooding from the exit; level sets are distinguished in colors. (c) The hazard level map with $\kappa = 2, k = 2$; band of INTERIM, $1-$SAFE and $2-$SAFE is colored in pink, yellow, and blue, respectively. (d) Possible navigation paths for six internal users (the purple squares).

x and y. The distance transform of $\mathcal{D}$ is thus defined as

$$DT(\mathcal{D}) = \begin{cases} \min_{y \in \partial\mathcal{D}} d_E(x,y), & x \in \mathcal{D}, \\ 0, & x \notin \mathcal{D}. \end{cases} \quad (4)$$

For any point $x \in \mathcal{D}$, $d_{\partial\mathcal{D}}(x) = \min_{y \in \partial\mathcal{D}} d_E(x,y)$ is referred to as the distance of point x to the boundaries of $\mathcal{D}$. On this basis, we can redefine the distance transform as the level set function for each level set variation.

Single point level set. The single point level set tracks the (inverse) evolution of a single point, in particular, the exit of the field. Specifically, we treat the exit as the initial curve $\gamma(p,0)$, and utilize the distance of any point x to the exit r, denoted by $d_r(x) = d_E(x,r)$, as the level set function. After a certain time t when r has propagated the whole field, every point x must belong to one level set whose elements all have a distance value $d_r(x)$. As r propagates with time continuously, for any point x there is at least one neighboring point y satisfying that $d_r(y) < d_r(x)$. Thus we have

Lemma 1. *The function d_r has no local minimum.*

Lemma 1 indicates that there is at least one path from any point to the exit by tracing a neighboring point with a smaller d_r value. This property is appealing in that it is still valid in discrete WSNs, as will be discussed in Section 4.1, which allows each user to escape by following a local minimum free path to the exit.

Level set band. The level set band tracks the evolution of the boundary of a hazardous area, i.e., we treat the boundary of a hazardous area as the initial curve $\gamma(p,0)$. The following distance function is used as the level set function:

Definition 1. *Let $\mathcal{H}$ and $\partial\mathcal{H}$ denote a hazardous area and its boundary, then the distance of any point x exterior of $\mathcal{H}$ to $\partial\mathcal{H}$ is $d_{\partial\mathcal{H}}(x) = \min_{y \in \partial\mathcal{H}} d_E(x,y)$.*

Instead of tracing every level set, the proposed level set band traces level sets in successive bands. To be more concrete, after the evolution of $\partial\mathcal{H}$, any point x with $d_{\partial\mathcal{H}}(x)$ satisfying $\Upsilon < d_{\partial\mathcal{H}}(x)/d_\epsilon \leq (\Upsilon + 1)$ forms a band consisting of levels from level-Υd_ϵ to level-$(\Upsilon+1)d_\epsilon$, where $d_\epsilon > 0$ is a width parameter and $\Upsilon \in \mathbb{N}_+$ is the band ordinal. Having Υ and d_ϵ, a series of bands around the hazardous area can be established. In a discrete setting, such bands can be used to reflect how close users to the dangerous area, as well as to disperse users into different bands to achieve mild congestions.

4 CANS ALGORITHM

In this section, we describe the implementation of our proposed CANS algorithm in a discrete setting. Notice that the level set method cannot be applied directly in discrete WSNs, as sensors are randomly deployed in the field and often have no location or distance information. With mere connectivity information, it is infeasible for each node to calculate its gradient of the distance transform function, and discrete form of the distance transform function also needs careful design. Therefore, we leverage the hop count distance in discrete WSNs to approximate the distance in the continuous setting.

We first present the outline of CANS algorithm, followed by the details of each step:

1) *Establishing the potential map.* To provide a local ordering that helps to guide users safely to the exit with less detours. To this end, a hop count based potential field is initiated by the exit, such that every node in the network is assigned a potential value and nodes with the same potential value form a level set (see Fig. 2b). On this basis, every user has a sense of direction to the exit by following the gradient of the potential function from high to low values.
2) *Building the hazard level map.* To facilitate users around the hazardous areas to find different paths so as to avoid heavy congestions. This is done by enforcing nodes around the emergency to form a number of bands with different hazard level weights, based on which users can be dispersed to go across paths in different bands (see Fig. 2c).
3) *Planning a safe path for each user.* To schedule for each user a safe path to the exit with mild congestion and small stretch (see Fig. 2d), on the basis of the constructed compound map (i.e., the combination of the potential map and the hazard level map).

The whole process, as an example, is shown in Fig. 2.

4.1 Establishing the Potential Map

The first step of CANS is to establish the potential map, as shown in Fig. 2b, to track the inverse evolution of level sets towards the exit. With mere connectivity information in a discrete network, we propose to establish a hop count based potential function $f : V \to D$ for the exit r, where D is an

integer set of the hop count distance from each node to the exit. Specifically, the exit r initiates a flooding across the whole network (here we regard the hazardous areas as obstacles and thus nodes in hazardous areas are not involved in the flooding; hereafter nodes are only referred to those outside the hazardous areas). After receiving the flooded message from r, every node knows its hop count distance d to r, and then records a hop count distance pair $\langle d, r\rangle$. Therefore, for any node v_i with the hop count distance d, we have $f(v_i) = d$. Accordingly, the *level set* of the potential function is given by $f^{-1}(d) = \{e_{ij} = (v_i, v_j) \in E | f(v_i) = f(v_j) = d\}$. All the level sets $f^{-1}(\cdot)$ constitute the potential map of the network.

One salient property of the established potential map is that, as shown in Lemma 1, there is no *local minima* (whose potential value is less than that of its neighbors) in the network, in that every node must have at least one neighbor with a smaller potential value (e.g. the predecessor from which the node receives the flooded message from the exit). Therefore, we have:

Lemma 2. *Any node is on some strictly monotonically decreasing path (with respect to the potential function) connecting to the exit.*

Given the potential map, the simplest way for a user v to progress to the exit r is by following the gradient of the potential function f from high to low values, which is sketched by pseudo-codes in Algorithm 1.

Algorithm 1. PLAIN NAVIGATION

Input Exit r; Current node $v \neq r$
Output Next hop u
1: **for each** $v' \in ngb(v)$ **do**
2: **if** $f(v') < f(v)$ **then**
3: $u \leftarrow v'$; **break**
4: **return** u

Recall that the potential map is local-minima-free, we thus have:

Theorem 3. *The plain navigation path from any node to the exit is guaranteed by following its downstream (with respect to the potential function) nodes.*

Whereas the potential map locally provides a sense of direction for users to the exit in a consistent way, it is of much randomness and blindness in the sense that users might be guided, without location information, to regions quite close to the hazardous areas. To tackle this issue, the hazard level map is thereby put forward.

4.2 Building the Hazard Level Map

The hazard level map (Fig. 2c shows an example) tracks the evolution of level sets off the hazardous areas, and therefore can naturally tell where the hazardous areas are. Motivated by information-guided routing schemes in WSNs [23], [24], [25], which use gradient descent by exploiting the natural continuity of the signal field, we propose to build the hazard level map by enforcing nodes around the emergency to form different bands with different hazard level weights. Based on the hazard level map, users are dispersed to go across paths in different bands, so that it is not so crowd as in a single-lane road map.

Building a hazard level map begins when the sensors detect an emergency event, and it is initiated by the emergency boundary nodes defined as [26], [27]:

Definition 2. *An* emergency boundary node *is a node $v \in V$ such that there exists a neighbor $v' \in ngb(v)$, where $s(v) \neq s(v')$.*

When an emergency occurs, a trivial way to build the hazard level map is to let the emergency boundary nodes simply initiate outward flooding to form bands with different hop count distances. Normally, bands with larger hop counts are safer than those with smaller hop counts, and therefore users are more likely to be guided to pass through those safer bands. However, from the perspective of passing around an emergency as fast as possible, following the bands with smaller hop counts is preferred for potentially shorter path lengths. So there is a trade-off between safety and path stretch. What is more, as boundary nodes may change in time due to emergency dynamics, the emergency can spread readily to bands with small hop counts, e.g. the 1-hop band, yielding such bands unsafe any longer. To tackle these issues, we propose to only keep track of the bands with "sufficient safety", which is formulated in terms of the *hazard level*:

Definition 3. *Let $ngb_k(v)$ denote the set of nodes at most k hops from v, and $S_k(v)$ denote the set of sensed values within the k-hop neighborhood of v (including v itself), then the* hazard level*, $h(v)$, of node v is defined as:*

- *HAZARD: $0 \notin S_k(v)$, i.e., v and all of its k-hop neighbors have a sensed value 1.*
- *SAFE: $1 \notin S_k(v)$, i.e., v and all of its k-hop neighbors have a sensed value 0.*
- *INTERIM: $0 \in S_k(v)$ and $1 \in S_k(v)$, i.e., v is neither HAZARD nor SAFE.*

By introducing the hazard level, we can see that a HAZARD node cannot be adjacent to a SAFE node: there is an INTERIM band separating the HAZARD nodes from the SAFE nodes. By properly choosing the value of k for different scenarios (e.g., a larger value of k for rapidly spread poisonous gas), we can only consider those steady-state SAFE nodes during a period of time, while neglect those highly dynamic sensors with "salt-and-pepper" type of the sensed values. By doing so, it is more robust to emergency dynamics, and is especially beneficial to the local updating of the navigation paths when reacting to emergency dynamics, as will be discussed in Section 5.1.

Based on the hazard level, we can build the SAFE bands with different weights. Specifically.

Definition 4. *A* 1-Safe node *has an INTERIM k-hop neighbor. A* κ-Safe node *has an INTERIM node in its κk-hop neighborhood but not in its $(\kappa - 1)k$-hop neighborhood; κ is called the* hazard level weight *of the node. All κ-SAFE nodes constitute the* κ-Safe band.

By checking its k-hop neighborhood using a similar way like the expanding ring algorithm [28], every SAFE

node can locally identify its weight and to which SAFE band it belongs. Thus, during the navigation, if users are close to the emergency, i.e. they encounter the SAFE bands, they will be branched to bands with different weights, reaching safe paths with few congestions (detailed in Section 4.3).

However, there exist some special cases, when there is only the INTERIM band, or the user flow exceeds the capacities of the SAFE bands. Under these circumstances, to ensure choosing safer INTERIM nodes, we define the *criticality* to reflect the hazard level among INTERIM nodes.

Definition 5. *Let $N_{k0}(v)$ denote the number of nodes in $ngb_k(v)$ whose $s(v) = 0$, and $N_{k1}(v)$ denote the number of nodes in $ngb_k(v)$ whose $s(v) = 1$. Then the* criticality *of an INTERIM node v, $C_k(v)$, is defined as*

$$C_k(v) = \frac{N_{k1}(v)}{N_{k1}(v) + N_{k0}(v)} = \frac{N_{k1}(v)}{|ngb_k(v)|}. \quad (5)$$

Combining the SAFE bands and the INTERIM nodes with different criticality gives rise to the hazard level map of the network, as shown in Fig. 2c, that facilitates the discovery of multiple navigation paths as we will discuss in the next section.

4.3 Planning a Safe Path for Each User

Given the constructed compound map (i.e., the combination of the potential map and the hazard level map), each node in the network has the knowledge of its hop count distance to the exit; each SAFE node knows its hazard level; each INTERIM node is aware of its criticality. On this basis, we design a navigation scheme to schedule a safe path, along which a user can reach the exit apart from the hazardous areas while avoiding heavy congestions.

Denote the user as v_0. As we assume that any user is equipped with communicating devices to interact with the sensors in the network, the navigation path can be interpreted as a sequence of sensor nodes $P = \{v_1, v_2, \ldots, v_n\}$. Then we have the following definitions:

Definition 6. *A path $P = \{v_1, v_2, \ldots, v_n\}$ is a* safe path *if $\forall$ $v_i \in P$ is not HAZARD.*

The obtained plain navigation path by Algorithm 1 is obviously a safe path, as the potential function is built on the non-obstacle nodes (i.e., nodes that are not HAZARD). However, as aforementioned, the plain navigation has no awareness of where the hazardous areas are, and it is more like to guide users too close to the hazardous areas. In addition, it also neglects the congestion problem.

Taking the congestion into consideration, our idea of navigation is as followed. For users far away from the emergency who are relatively safe and can tolerate more congestions, they will be guided by the plain navigation as in Algorithm 1 to achieve small stretch. For users close to the hazardous areas who are vulnerable to congestions, they are dispersed into different bands to avoid heavy congestions. Concretely, the navigation begins with the user, then Algorithm 2 is sequentially invoked to guide the user apart from the hazardous areas. The following theorem guarantees the safety of the scheduled paths:

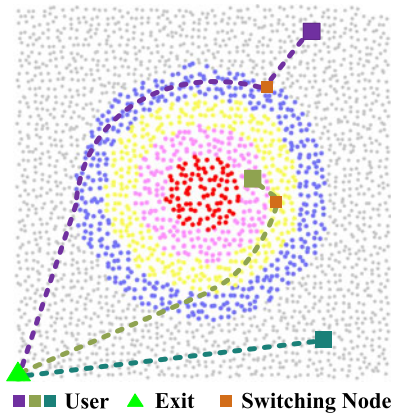


Fig. 3. Illustration of different situations in Algorithm 2.

Algorithm 2. CONGESTION-ADAPTIVE NAVIGATION

Input Exit r; Current node $v \neq r$
Output Next hop u

if v is a switching node **then**
  $\kappa_0 \leftarrow$ UniformRandVar randv(1, κ)
  v follows the downstream (with respect to the hazard level weight) until u' whose weight is κ_0; $u \leftarrow u'$
else if v is INTERIM **then**
  if $\exists\, v' \in ngb(v)$ is SAFE **then**
    $u \leftarrow v'$;
  else if $\exists\, v' \in ngb(v)$ with $C_k(v') < C_k(v)$ **then**
    $u \leftarrow v'$;
else if v is SAFE **then**
  if $\exists\, v' \in ngb(v)$ with $v'.weight = v.weight$ and $f(v') < f(v)$ **then**
    $u \leftarrow v'$;
  else if $\exists\, v' \in ngb(v)$ with $v'.weight > v.weight$ and $f(v') < f(v)$ **then**
    $u \leftarrow v'$;
  else if $\exists\, v' \in ngb(v)$ neither SAFE nor INTERIM and $f(v') < f(v)$ **then**
    $u \leftarrow v'$;
else
  for each $v' \in ngb(v)$ **do**
    if $f(v') < f(v)$ **then**
      $u \leftarrow v'$; **break**
return u

Theorem 4. *All possible navigation paths on the compound map scheduled by CANS are safe paths.*

Proof. Algorithm 2 deals with the following situations according to the spatial location of the user (see Fig. 3).

Situation 1: The user is entering the SAFE bands, i.e., current node is SAFE while its previous hop is not (we refer to the current node in this case as a *switching node*, see the brown spots in Fig. 3 for instance). In this situation, the user will randomly choose one SAFE band to follow. To be more concrete, the user first randomly chooses an integer $\kappa_0 \in [1, \kappa]$ and follows the downstream (with respect to the hazard level weight) until encounters a node whose weight is κ_0. Then Situation 3 comes.

Situation 2: The user is in the INTERIM bands. In this situation, the user first checks if there is a SAFE node with

a smaller potential value in its neighborhood. If so, the user will turn to the SAFE node, and there follows Situation 3. Otherwise, the user will choose a neighbor with a smaller criticality as its next hop.

Situation 3: The user is in the SAFE bands. This situation happens when the user is originally in the SAFE bands or after the guidance in Situations 1 or 2. In this case, the user will try to follow the corresponding SAFE band towards the exit until get out of the SAFE bands. Particularly, the user first chooses a neighbor within the same SAFE band with a potential value no greater than its own. If not, the neighbor with a larger SAFE weight and a smaller potential value is to be chosen. If still unavailable, the neighbor neither in the SAFE bands nor in the INTERIM band with a smaller potential value is the choice.

Situation 4: The user is far from the hazardous areas, i.e., none of the neighbors with a smaller potential value of current node is SAFE. Thus the user is neither in the SAFE bands nor in the INTERIM band. Then the plain navigation as in Algorithm 1 is performed to guide the user to the exit.

Situation 5: Exceptions for above situations. In Algorithm 2, we rule out specific techniques to deal with some special cases. For example, in Situation 2 there may be the case when the INTERIM node has neither a SAFE neighbor nor a neighbor with a smaller criticality. We also set aside scenarios with complicated topologies such as a narrow strip network or a dumbbell-shaped network. In these cases, we simply evoke the plain navigation to find the next hop.

Based on above five situations, we can see that any user will find a next hop to the exit by evoking Algorithm 2. As every next hop is within the compound map and thus is not HAZARD, Theorem 4 is valid. □

5 DISCUSSIONS

5.1 Reacting to Emergency Dynamics

In reality, hazardous areas may vary in time, e.g. they may emerge, expand, shrink or disappear. The impacts of such dynamics on the compound map and further on our navigation algorithm mainly reflect in two aspects. Firstly, Lemma 2 may no longer hold, as there may be some local minima in the potential map that hinders the user to progress to the exit. Secondly, the hazard level map may need to be reconstructed, as the hazard levels of the nodes may change due to the dynamics. We now discuss in the following how to compensate for each of the impacts.

5.1.1 Dealing with the Local Minima

The construction of the potential map in a static network, according to Lemma 2, guarantees that every non-exit node in the network has a neighbor whose potential value is smaller than its own, such that the plain navigation (Algorithm 1) proceeds smoothly.

However, several nodes are likely to become local minima when their neighbors with smaller potential values are not valid (e.g., dead) due to the emergency dynamics. In this case, if a node v find itself a local minima, it adjusts its potential value according to the discrete Laplace's equation [29]:

$$f(v) = \frac{1}{|ngb(v)|} \sum_{v' \in ngb(v)} f(v'), \tag{6}$$

which is actually the average of the potential values of its neighbors, and thus is guaranteed to be free of local minima. As such, the node v can avoid becoming a local minima and its next hop can be found.

It is noted that the local minima may theoretically fall into a plateau region [25], [30], where all its neighbors have the same potential value. If this situation happens, local discovery by searching the local neighborhood through either a random walk or a local flooding can be performed to reach a nearby non-stationary node, so that the navigation can proceed without being suspended.

5.1.2 Rebuilding the Hazard Level Map

Due to emergency dynamics, the hazard level map may no longer valid, and thus the reconstruction of the hazard level map has to be considered. A trivial but highly time/message costly and inefficient method is to entirely reconstruct the new hazard level map whenever the emergency varies. Instead, the emergency dynamics only induce a local impact on our established hazard level map, as will be proven below, and thus only a local operation on the reconstruction of the hazard level map is required.

Theorem 5. *The emergency dynamics affect the hazard level map reconstruction locally.*

Proof. Recall that the process of building the hazard level map in Section 4.2 mainly includes three aspects: identifying node's hazard level, constructing the SAFE bands, and recognizing INTERIM node's criticality, each aspect of which is conducted by the corresponding node within its k-hop, κ-hop, and 1-hop neighborhood, respectively. Accordingly, a local status updating scheme is straightforward: when the hazard area varies, each node only needs to check its local neighborhood to update its status, and thus the hazard level map can be locally renewed in accordance with the emergency dynamics. That is, Theorem 5 holds. □

Note that the presented status updating scheme can also be utilized for local failure recovery in the presence of temporary or permanent node/link failures.

5.1.3 Reacting Speed and Relevant Overhead

As discussed before, CANS only needs local information (i.e., each node requires only information from its local neighbors) to react to the emergency dynamics. Therefore, the reacting speed to emergency dynamics and the resulting communication overhead largely depend on the updating frequency of CANS: more frequent updating obviously results faster reacting speed but higher communication overhead. In addition, the updating frequency is closely related to the emergencies of different spreading speeds. For example, compared to a fire hazard, a chemical gas leakage incident often requires a faster reacting speed. Thus, it is essential to take into consideration such aforementioned

factors, so that an appropriate updating frequency can be obtained for the implementation of the algorithm in reality.

5.2 Trade-off Among Path Safety, Congestion, and Stretch

Emergency navigation design with WSNs requires a proper tradeoff among three conflicting factors: path safety, congestion and stretch. Early proposals typically consider the level of danger based on the distance of the node to hazardous areas: the farther, the safer. Accordingly, the media axis based methods such as [4], [8], [11], [13] are proposed to obtain navigation path that maximizes the minimum distance of all possible paths to the hazardous areas. Whereas selecting the path farthest from the hazard area assures path safety to the greatest extent, it is more likely to result in path congestions and unnecessary detours, as all users are guided to the specific path no matter where the users are.

In contrast, CANS provides multiple navigation paths for users to escape, so that the tradeoff among path safety, congestion and stretch can be achieved. To be more concrete, in CANS we regard the areas outside the SAFE bands are sufficiently safe, such that users therein can tolerate the congestions as they are far away from the hazardous areas. For users within the SAFE bands, they are required to be branched into SAFE bands with different weights to avoid heavy congestions. Particularly, for users within the INTERIM band without the chance turning to the SAFE bands, they will locally follow the direction of a lower criticality. With such design, users outside the SAFE bands reduce needless detours, while users within the SAFE bands avoid heavy congestions. Meanwhile, the safety remains guaranteed as proved in Section 4.3. It is worth noting that, another neat part of this design is that, the width and the number of the SAFE bands can be adapt to different applications by tuning the values of parameters k and κ, so that the path safety can be enhanced and guaranteed in various scenarios.

5.3 Complexity and Storage Cost

Message complexity, time complexity and storage cost are of great importance for a distributed algorithm relying on mere connectivity information. The message complexity is the traffic cost an algorithm incurs, the time complexity is determined by the number of iterations during the running time of an algorithm, and the storage cost is measured by the number of nodes (their coordinates or IDs) stored. All three factors significantly affect the scalability of a distributed algorithm. For the complexity of CANS we have:

Theorem 6. *Both the message complexity and the time complexity of the compound map construction of CANS are $O(N)$, where N is the network size.*

Proof. The compound map construction of CANS mainly contains two steps: establishing the potential map and building the hazard level map. During the first step, a network-wide flooding to construct the potential map incurs an $O(N)$ message complexity and a best/worst case time complexity of $O(1)/O(\sqrt{N})$ [15], [31]. For the second step, this process usually has a message complexity of $O(\sqrt{N})$ due to the local flooding and a time complexity of $O(1)$ as is performed in a distributed fashion. In total, the compound map construction of CANS has a linear message and time complexity of the network size. □

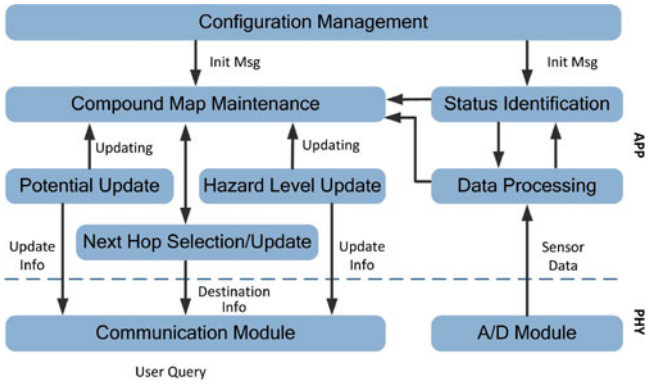


Fig. 4. Architecture of the experiment testbed.

Theorem 6 implies that while flooding is frequently used to collect necessary information in each step of CANS, the message and time complexities will not increase too much as the flooding is in most cases limited in a local area. For instance, in the step of building the hazard level map, the flooding is only executed within the k-hop neighborhood of each node, containing much fewer nodes compared with the whole network.

For the storage cost, as each node only needs to record the number of its neighbors and the exit, we have:

Theorem 7. *In CANS, the storage cost of each node is $O(\overline{|ngb(v)|})$, where $\overline{|ngb(v)|}$ is the average node degree.*

As the storage cost is dominated by the number of its neighbors, according to Theorem 7, the storage cost of each node is thus close to a constant and scales well with the network size.

6 PROTOTYPE EXPERIMENT

We implement a prototype testbed using *TI CC2530* chips; the sensor operating system is *Tiny OS*. Fig. 4 shows the architecture of the testbed. The configuration management takes charge of sensor's information, e.g., sensor ID, sensor status, exit information, etc. The Compound Map Maintenance provides updating information of node's potential and hazard level for maintaining the compound map; it also generates guidance information for query users. The communication module receives queries from internal users and sends the computed next hop information back to them.

Our testbed consists of 36 sensor nodes deployed into 6×6 grids with 10 meter space between a pair of nodes in the campus, the same deployment as that in [4], [11], (see Fig. 5a for a miniature prototype in the laboratory). In our experiments, we mainly focus on node's congestion (i.e., the number of paths the node involved in) and the generated path length. During the experiments, the top-right corner sensor (coordinate (6, 6)) is set as the safe exit, and we randomly choose 10 initial positions of the users, and 10 tests are performed for each position. Besides, we set $k = \kappa = 1$, and set aside the INTERIM band due to the small scale nature.

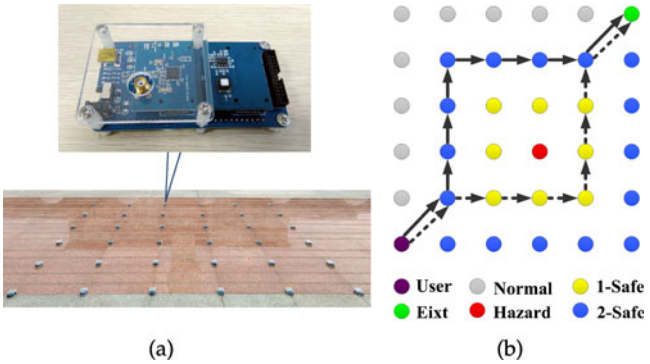


Fig. 5. Experiment setup. (a) The miniature prototype deployment. (b) A showcase of the navigation.

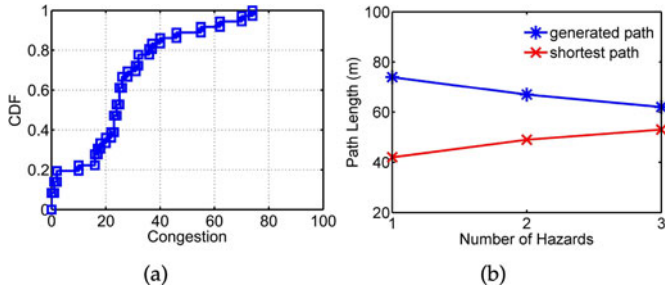


Fig. 6. (a) Congestion distribution. (b) Path length versus number of hazards.

Fig. 6a depicts the cumulative distribution function (CDF) of node's congestion, when there is only one hazard (coordinate (4, 3)), as showcased in Fig. 5b. We can see that the majority of nodes (over 80 percent) are involved in less than 40 paths, and there are only a small number of nodes over crowed, say involved in more than 60, which are nodes near the exit. We will see in the large scale simulations, the alleviation of node's congestion is even more significant. Fig. 6b compares the average length of the generated path with the shortest safe path (the ground truth). It can be seen that, the ratio of the two path lengths (i.e., the path stretch) decreases from around 1.7 to 1.2 as the number of hazards increases. Due to the limitation of the experiment scale, it is not that obvious of the advantage of our algorithm in terms of path stretch. However, when the size of the network increases, the superiority would be particularly evident, as will be shown in the next section.

7 Performance Evaluation

The prototype experiments testify the effectiveness of our algorithm for small scale networks. In this section, we evaluate the feasibility and scalability of our approach through a series of simulations in large scale networks, and compare the proposed CANS algorithm with the existing algorithms [4], [8], [11].

7.1 Simulation Setup

In our simulation settings, nodes are uniformly yet randomly distributed within the sensing field, and have the same communication range R. The communication model used by default is the Unit Disk Graph model (UDG) [32], i.e. two nodes are connected iff their separation is no greater than R. The default parameters for hazard level map construction are $\kappa = k = 2$. For each trial of the network we randomly generate 10 internal users in the field, and then run our algorithm 100 times to report the aggregate statistics.

We also compare the performance of our algorithm with the skeleton graph based protocol (SG for short) [8], as well as the road map based method (RM for short) [4], [11]. To obtain a better performance, we follow the simulation settings used in their original papers: for SG approach, the adaptive SG version is utilized, for its superiority to the uniform one as discussed in [8]; for both SG and RM schemes, we enlarge nodes' communication range so that the average node degree of the network is around 28. It is noted that our algorithm requires no location/distance information and is less dependent on a high network density; an average node degree above around 9 would meet the general requirement, as will be shown below.

7.2 Simulation Results

We examine the performance of our algorithm as well as SG and RM in three networks, with the average node degree between 9.9 and 12.9 and the number of hazardous areas from 1 to 3, as shown in Figs. 7a, 7e, and 7i. We observe from the simulations that despite the variation in scale and complexity, our algorithm guarantees to generate a safe path for every internal user. Figs. 7b, 7f, and 7j show the constructed hazard level map for each network, as well as the possible navigation paths by CANS for selected internal users in the network. We also notice that SG and RM are able to assure a safe path for each user as well, so we further focus on two factors for navigation performance evaluation: congestion distribution and path stretch.

Congestion distribution. We measure a node's congestion by the number of scheduled paths the node involved in, where the scheduled paths refer to the paths generated by a specific navigation algorithm (i.e., CANS, SG or RM in our simulations). The results of CANS and RM are illustrated in the last two rows in Fig. 7, and only those nodes with a congestion more than 200 are darkened. SG has a similar results to RM so here it is not shown due to space limits. Obviously, CANS has a better performance with respect to congestion distribution, and there are far less over involved nodes. In comparisons, most parts of the road backbones of RM and SG are over congested, especially the parts near the exit. The reason is explicit: all scheduled paths are on the road backbone, yielding the path endings near the exit excessively crowded, where the backbone colored in black indicates a congestion more than 800.

We also simulate the CDF of the congestions of the nodes by randomly selecting 1,000 paths from all the 3,000 paths (10 users for 100 times tests in each network) generated in the three tested networks, which is shown in Fig. 8a. It is observed that, nodes in CANS are involved in less than 400 paths, and the CDF rapidly increases to 1, which means CANS generates no overcrowded nodes and the nodes are involved in relatively fewer paths than SG and RM. In contrast, though most nodes have a congestion less than 200, about 10 percent of the nodes are extremely over involved in more than 600 paths. These small parts of the nodes are on the road backbone, and they have to burden almost all the users. Meanwhile, it is quite a threaten, from a practical

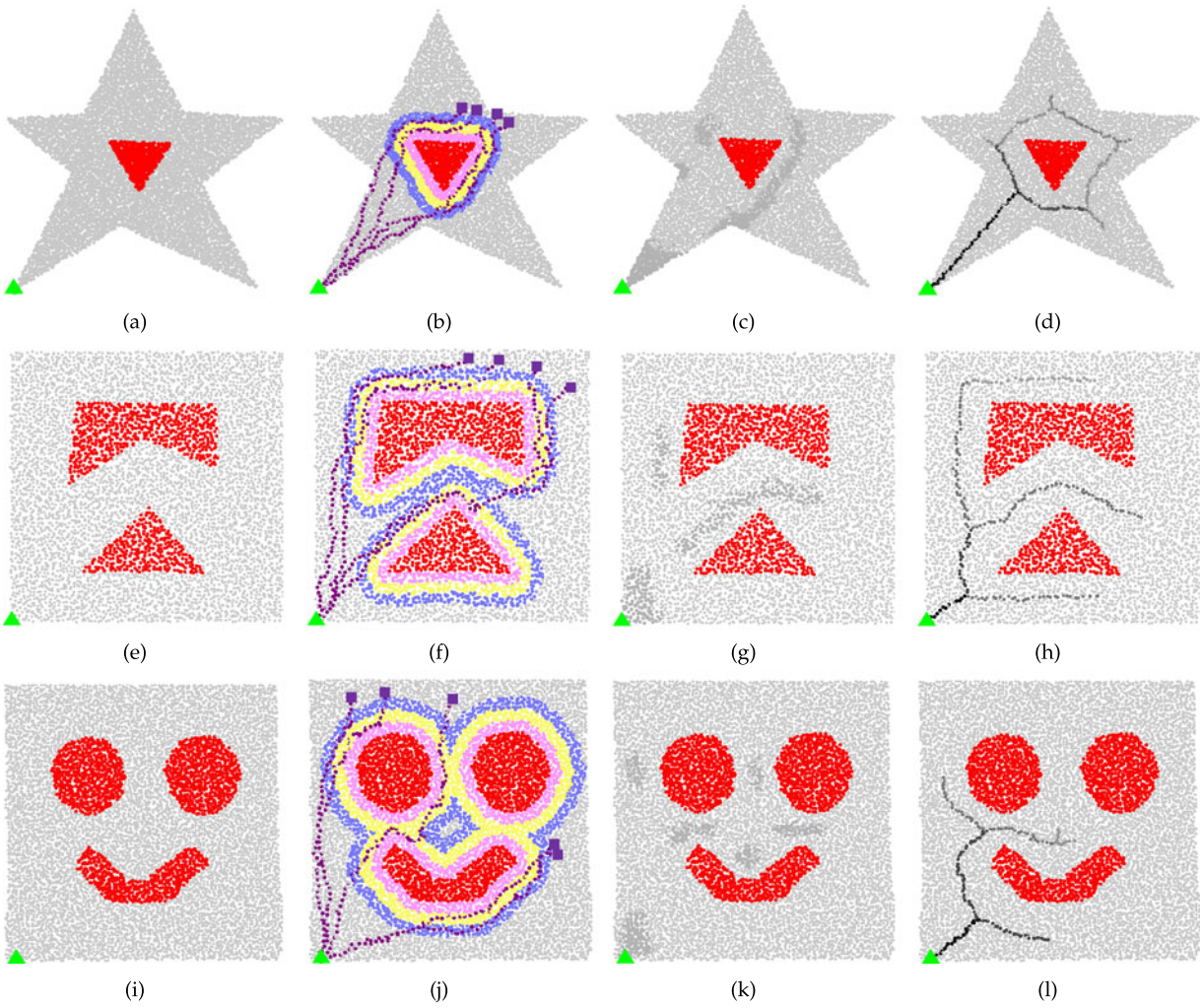


Fig. 7. Rows: (1) A star-shaped network with 5,303 nodes; avg deg is 12.9. (2) A wedge-shaped network with 4,766 nodes; avg deg is 11.2. (3) A smiley-shaped network with 3,909 nodes; avg deg is 9.9. Columns from left to right: (1) The original network, with one exit (the green triangle) and one or more hazardous areas (colored in red). (2) Possible navigation paths by CANS for selected internal users. (3) Congestion distribution of CANS. (4) Congestion distribution of RM approach [4], [11]. Darker colors indicate heavier congestions. Band of INTERIM, $1-$SAFE and $2-$SAFE is colored in pink, yellow, and blue, respectively.

perspective, that paths composed of nodes with such heavy congestions may not be that safe any more.

Path stretch. Fig. 8b shows the average/maximum stretch results of CANS, SG and RM methods for each network. It can be seen that CANS has an average stretch very close to 1 in all cases, the lowest among the three schemes. This means CANS has the highest navigation path planning efficiency, as the stretch is an indicator of navigation overhead. For example, an average stretch factor of 1.68 in RM indicates its overhead of 68 percent on the basis of the shortest path. Therefore, CANS reduces navigation overhead by 24 and 46 percent, in comparison with SG and RM, respectively. This is not much surprising as both SG and RM require that users start off on the road backbone first wherever they are, while CANS only needs those users near the hazardous areas to pay for slight detours, so that they can avoid heavy congestions. In terms of the worst-case stretch, the difference of the performance is even greater. Apart from invariably guiding users to the road backbone at first, the direction assigned on the backbone further aggravates the performance of RM: a user led to the backbone may have to go in an opposite direction with a large detour even there is a short path directly to the exit. We also notice that, the stretch of SG/RM, contradictive to the intuition, decreases as the number of emergency increases. That is because the network will be divided into more smaller regions by more hazardous areas, and thus the road backbone, restricted to smaller regions, gets closer to the users.

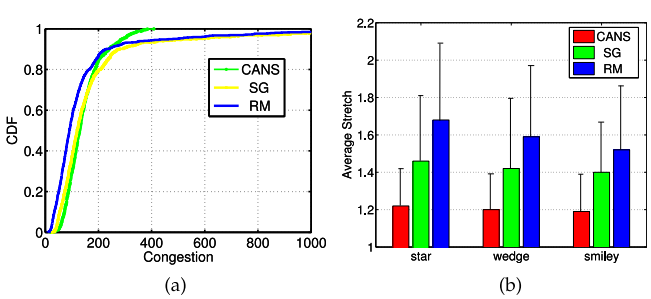


Fig. 8. (a) Congestion distribution. (b) Average/maximum path stretch.

7.3 Reacting to Emergency Dynamics

To test the robustness of CANS under environment dynamics, we expand the hazardous areas in Fig. 7f with different

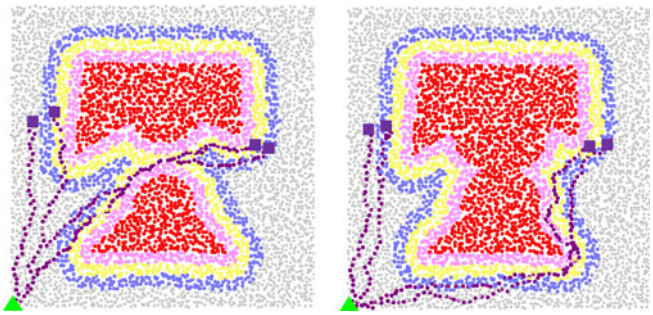


Fig. 9. Reacting to emergency dynamics: (a) Slow spread. (b) Fast spread.

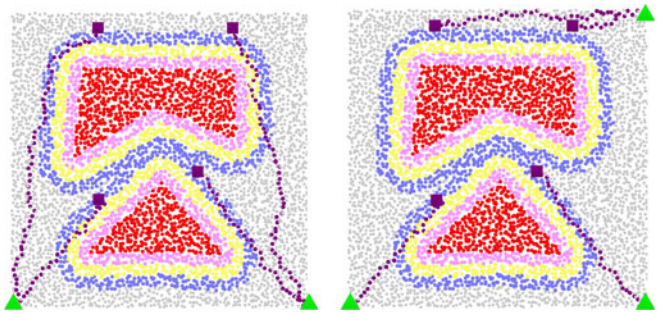


Fig. 10. Adaption to situations of (a) two exits and (b) three exits.

hazard spread speed. We choose the same four users at first, and the dangerous areas expands as the users escape in the guidance of CANS. Figs. 9a and 9b show how CANS can react to emergency dynamics with a slow and fast spread speed, respectively. It can be seen that both cases the four users can be scheduled a safe path apart from the dangers. When two hazardous areas emerge into one, as shown in Fig. 9b, the two users on the right are able to adjust their ways in time to get around the dangerous area. Obviously this requires a local and fast responds to the environment changes, which can be done by CANS as we discussed in Section 5.1.

7.4 Adaption to Multiple Exits

In previous evaluations, the number of exit is set to 1. However, our algorithm can be adapted to multi-exit situations with moderate efforts: as every node records the hop count distance pair $\langle d, r\rangle$ to every exit r, during the navigation users can simply choose the nearest exit as its destination, and then conduct the navigation process for evacuation. Note that though such a treatment may not be the most effective solution, it is efficient in respect to its simplicity. Figs. 10a and 10b demonstrate the navigation paths in multi-exit situations in the wedge-shaped network as in Fig. 7e. We can observe that by dealing with multiple exits in such a straightforward way, CANS can be well adapted to more than one exit, while still preserving mild congestion and small stretch.

7.5 Effect of Communication Models

Though we use the UDG model in our simulations, our algorithm needs mere connectivity information and assumes a practically constant maximum transmission range. Thus a uniform communication range is not required, as in a number of connectivity based algorithms in WSNs, e.g., [33], [34], [35]. To testify this, we also examine the performance of our algorithm by using the Quasi-Unit Disk Graph (QUDG) model [36], and obtain similar results as in UDG. Here we skip the illustrations due to space limitations.

8 Related Work

8.1 Path Planning and Navigation of Robots

In the fields of robotics, navigation is regarded as one of the most important and fundamental issue, and it is closely related to robot's path planning [37], [38]. Traditional robot navigation relies on on-board sensors of the robots to sense the environment, and often requires the location or geometric information to minimize the navigation path length while avoiding obstacles, e.g., the moving obstacles planner (MOP) [39] and the rapidly-exploring random trees (RRT) [40], [41]. One major concern is that these methods need a central controller to collect information from the entire network, and further conduct intensive computations, e.g., to calculate the shortest path.

To avoid building a centralized and global representation of the environment, distributed navigation approaches of robots are considered in the context of sensor networks [42], [43]. Whereas these methods bring about better scalability and efficiency, they cannot directly applied to WSN-assist emergency navigation as these approaches only design one path for one specified robot, which will result in a heavy congestion path in WSN scenarios.

Another thread of research, which is more related to our work, studies multi-robot navigation problems. This kind of problems often requires that each robot follow different paths to cover or sense the whole workspace as in search-and-rescue problems [44]. It is usually formulated as finding navigation paths under homotopy class constraints [45], [46]. Unfortunately, existing attempts at classifying navigation paths of different homotopy classes require centralized operations and computation, yielding them infeasible for distributed sensor networks as discussed in this paper.

8.2 Emergency Navigation with WSNs

Emergency navigation is one of the most important reactive tasks in WSNs that serve as adaptive distributed repositories of information [3], [10]. Li et al. [6] proposed the first distributed algorithm for guiding navigation across a WSN. To seek a solution naturally following the discrete nature of WSNs, they established an *artificial potential field* by flooding from dangerous nodes, so that the exit generates an attractive potential while each dangerous node generates a repulsive potential, so as to obtain a navigation path with the least total potential value. The main drawback of this method is its exhaustive network-wide flooding.

In order to lower the communication cost, Buragohain et al. [8] proposed to abstract the network by establishing a *skeleton graph*, over which approximate optimal safe paths can be found. Tseng et al. [9] proposed to adopt the idea of the *partial link reversal* by adding the weight of safety in path selection to guide users farther away from hazardous regions. Note these two methods assume the availability of location information.

Recently, Li et al. [4], [11] proposed a distributed road map based navigation algorithm, providing users safety guaranteed navigation without any preknowledge of user and sensor locations. Followups also release the location assumption. Xiao et al. [12] presented a reliable navigation algorithm that can handle the unreliability in wireless communications. Zhan et al. [13] designed a safe, ordered, and speedy emergency navigation algorithm in WSNs to minimize users' evacuation time.

However, the related works above only consider how to find a shortest/safest path for each person while ignore the number of people on the path and the path's stretch. In some emergency, excessive people running in a narrow path may cause heavy congestion (or even stampede) and large stretch.

8.3 Homotopic Routing and Information-Guided Routing

To exploit the existence of multiple navigation paths in a WSN, our work is inspired by previous works in homotopic routing [47], [48], [49] and information-guided routing [23], [24], [25].

Homotopic routing aims to find routing paths of a specific homotopy type, to improve load balancing and routing resilience. Current methods to distinguish paths of different homotopy types fall into two categories. One is based on the demarcation of *cut edges* of the obstacles [47], and the other is based on *conformal mapping* that embeds the original network into a virtual coordinate space [48], [49]. Both methods are designed for static networks and their performances are more likely to deteriorate in highly dynamic emergency settings.

Information-guided routing has been explored as a scalable approach for scenarios with high query frequency [23]. It is based on the fact that the spatial distribution of many physical quantities follows a natural diffusion law, and therefore the natural gradients of physical phenomena are utilized to guide the routing process. One major concern of the gradients imposed by natural laws is that the signal field may have multiple peaks and valleys, forcing the routing to deteriorate to a random walk [24], [26].

Apart from the intrinsic distinction between routing and our in-situ interaction based navigation, our approach is designed with substantial differences to above two groups of routing algorithms by devising new structures. The potential map and the hazard level map maintain the high-level topological features of the network and provide a local-minima-free structure to guarantee users to find safe and efficient escape paths with mild congestion and small stretch. What is more, our approach is able to cope with the dynamics of the hazardous areas.

9 Conclusion

In this paper, we have presented CANS, a novel distributed algorithm towards congestion-adaptive and small stretch emergency navigation with WSNs. CANS does not require in advance knowledge of location or distance information, nor the reliance on any particular communication model. It is also scalable since the time and message complexities of our algorithm are linear to the network size. Both small scale experiments and extensive simulations demonstrate the efficiency and effectiveness of the proposed algorithm.

Acknowledgments

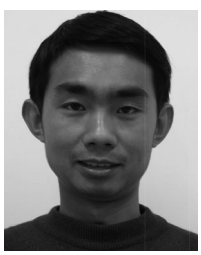
This work was supported in part by the National Natural Science Foundation of China under Grants 61271226, 61272410, 61202460, and 61471408; by the National High Technology Research and Development Program ("863" Program) of China under Grants 2014AA01A701 and 2015AA011303; by the National Natural Science Foundation of Hubei Province under Grant 2014CFA040; by the China Postdoctoral Science Foundation under Grants 2014M560608; by the Fundamental Research Funds for the Central Universities under Grant 2015QN073; and by the Science and Technology Plan Projects of Wuhan City under Grant 2015010101010022. Hongzhi Lin is the corresponding author.

References

[1] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. New York, NY, USA: Wiley, 2010.
[2] Y. Song, B. Wang, Z. Shi, K. Pattipati, and S. Gupta, "Distributed algorithms for energy-efficient even self-deployment in mobile sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 5, pp. 1035–1047, May 2014.
[3] C. Fischer and H. Gellersen, "Location and navigation support for emergency responders: A survey," *IEEE Pervasive Comput.*, vol. 9, no. 1, pp. 38–47, Jan.–Mar. 2010.
[4] J. Wang, Z. Li, M. Li, Y. Liu, and Z. Yang, "Sensor network navigation without locations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1436–1446, July 2013.
[5] M. Bocca, O. Kaltiokallio, N. Patwari, and S. Venkatasubramanian, "Multiple target tracking with RF sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 8, pp. 1787–1800, Aug. 2014.
[6] Q. Li, M. De Rosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw.*, 2003, pp. 313–325.
[7] E. Xu, Z. Ding, and S. Dasgupta, "Target tracking and mobile sensor navigation in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 177–186, Jan. 2013.
[8] C. Buragohain, D. Agrawal, and S. Suri, "Distributed navigation algorithms for sensor networks," in *Proc. 25th IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–10.
[9] Y.-C. Tseng, M.-S. Pan, and Y.-Y. Tsai, "Wireless sensor networks for emergency navigation," *Computer*, vol. 39, no. 7, pp. 55–62, July 2006.
[10] Q. Li and D. Rus, "Navigation protocols in sensor networks," *ACM Trans. Sensor Netw.*, vol. 1, no. 1, pp. 3–35, 2005.
[11] M. Li, Y. Liu, J. Wang, and Z. Yang, "Sensor network navigation without locations," in *Proc. 28th IEEE Int. Conf. Comput. Commun.*, 2009, pp. 2419–2427.
[12] Q. Xiao, B. Xiao, J. Luo, and G. Liu, "Reliable navigation of mobile sensors in wireless sensor networks without localization service," in *Proc. 17th Int. Conf. Quality Service*, 2009, pp. 1–9.
[13] A. Zhan, F. Wu, and G. Chen, "SOS: A safe, ordered, and speedy emergency navigation algorithm in wireless sensor networks," in *Proc. 20th Int. Conf. Comput. Commun. Netw.*, 2011, pp. 1–6.
[14] J. Bruck, J. Gao, and A. Jiang, "MAP: Medial axis based geometric routing in sensor networks," in *Proc. 11th ACM 11th Annu. Int. Conf. Mobile Comput. Netw.*, 2005, pp. 88–102.
[15] H. Jiang, W. Liu, D. Wang, C. Tian, X. Bai, X. Liu, Y. Wu, and W. Liu, "Connectivity-based skeleton extraction in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 710–721, May 2010.
[16] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proc. 22nd IEEE Int. Conf. Comput. Commun.*, 2003, pp. 1734–1743.
[17] S. Osher and R. P. Fedkiw, "Level set methods: An overview and some recent results," *J. Comput. Phys.*, vol. 169, no. 2, pp. 463–502, 2001.

[18] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.
[19] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci.*, vol. 93, no. 4, pp. 1591–1595, 1996.
[20] M. Falcone, T. Giorgi, and P. Loreti, "Level sets of viscosity solutions: Some applications to fronts and Rendez-vous problems," *SIAM J. Appl. Math.*, vol. 54, no. 5, pp. 1335–1354, 1994.
[21] D. Adalsteinsson and J. A. Sethian, "A fast level set method for propagating interfaces," *J. Comput. Phys.*, vol. 118, no. 2, pp. 269–277, 1995.
[22] G. Borgefors, "Distance transformations in digital images," *Comput. Vision, Graph., Image Process.*, vol. 34, no. 3, pp. 344–371, 1986.
[23] R. Sarkar, X. Zhu, J. Gao, L. J. Guibas, and J. S. Mitchell, "Iso-contour queries and gradient descent with guaranteed delivery in sensor networks," in *Proc. 27th IEEE Int. Conf. Comput. Commun.*, 2008, pp. 960–967.
[24] X. Zhu, R. Sarkar, J. Gao, and J. S. Mitchell, "Light-weight contour tracking in wireless sensor networks," in *Proc. 27th IEEE Int. Conf. Comput. Commun.*, 2008, pp. 1175–1183.
[25] H. Lin, M. Lu, N. Milosavljevic, J. Gao, and L. J. Guibas, "Composable information gradients in wireless sensor networks," in *Proc. 7th Int. Conf. Inf. Process. Sens. Netw.*, 2008, pp. 121–132.
[26] J. Lian, L. Chen, K. Naik, Y. Liu, and G. B. Agnew, "Gradient boundary detection for time series snapshot construction in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 10, pp. 1462–1475, Oct. 2007.
[27] M. Duckham, D. Nussbaum, J. Rudiger Sack, and N. Santoro, "Efficient, decentralized computation of the topology of spatial regions," *IEEE Trans. Comput.*, vol. 60, no. 8, pp. 1100–1113, Aug. 2011.
[28] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks* (ser. The Kluwer International Series in Engineering and Computer Science), vol. 353. Norwell, MA, USA: Kluwer, 1996, ch. 5, pp. 153–181.
[29] S. G. Krantz, *Handbook of Complex Variables*. Boston, MA, USA: Birkhäuser, 1999.
[30] P. Skraba, Q. Fang, A. Nguyen, and L. Guibas, "Sweeps over wireless sensor networks," in *Proc. 5th ACM/IEEE Int. Conf. Inf. Process. Sens. Netw.*, 2006, pp. 143–151.
[31] H. Jiang, T. Yu, C. Tian, G. Tan, and C. Wang, "CONSEL: Connectivity-based segmentation in large-scale 2D/3D sensor networks," in *Proc. 31st IEEE Int. Conf. Comput. Commun.*, 2012, pp. 2086–2094.
[32] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Math.*, vol. 86, no. 1, pp. 165–177, 1990.
[33] S. MacLean and S. Datta, "Reducing the positional error of connectivity-based positioning algorithms through cooperation between neighbors," *IEEE Trans. Mobile Comput.*, vol. 13, no. 8, pp. 1868–1882, Aug. 2014.
[34] H. Jiang, T. Yu, C. Tian, G. Tan, and C. Wang, "Connectivity-based segmentation in large-scale 2D/3D sensor networks: Algorithm and applications," *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 15–27, Feb. 2015.
[35] C. Wang and H. Jiang, "SURF: A connectivity-based space filling curve construction algorithm in high genus 3D surface WSNs," in *Proc. 34th IEEE Int. Conf. Comput. Commun.*, 2015, pp. 981–989.
[36] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad-hoc networks beyond unit disk graphs," in *Proc. Joint Workshop Found. Mobile Comput.*, 2003, pp. 69–78.
[37] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. New York, NY, USA: Wiley, 2006.
[38] G. E. Jan, K. Y. Chang, and I. Parberry, "Optimal path planning for mobile robot navigation," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 4, pp. 451–460, Aug. 2008.
[39] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *J. Guidance, Control, Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.
[40] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. 17th IEEE Int. Conf. Robot. Autom.*, 2000, pp. 995–1001.
[41] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *Proc. 24th IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1617–1624.
[42] Z. Yao and K. Gupta, "Distributed roadmaps for robot navigation in sensor networks," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 997–1004, Oct. 2011.
[43] Z. Zhang, Z. Li, D. Zhang, and J. Chen, "Path planning and navigation for mobile robots in a hybrid sensor network without prior location information," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 172, pp. 1–12, 2013.
[44] S. Bhattacharya, D. Lipsky, R. Ghrist, and V. Kumar, "Invariants for homology classes with application to optimal search and planning problem in robotics," *Ann. Math. Artif. Intell.*, vol. 67, nos. 3/4, pp. 251–281, 2013.
[45] S. Kim, K. Sreenath, S. Bhattacharya, and V. Kumar, "Optimal trajectory generation under homology class constraints," in *Proc. IEEE 51st Annu. Conf. Decision Control*, 2012, pp. 3157–3164.
[46] M. Kuderer, C. Sprunk, H. Kretzschmar, and W. Burgard, "Online generation of homotopically distinct navigation paths," in *Proc. 31st IEEE Int. Conf. Robot. Autom.*, 2014, pp. 6462–6467.
[47] W. Zeng, R. Sarkar, F. Luo, X. Gu, and J. Gao, "Resilient routing for sensor networks using hyperbolic embedding of universal covering space," in *Proc. 29th IEEE Int. Conf. Comput. Commun.*, 2010, pp. 1–9.
[48] R. Jiang, X. Ban, M. Goswami, W. Zeng, J. Gao, and X. Gu, "Exploration of path space using sensor network geometry," in *Proc. 10th Int. Conf. Inf. Process. Sens. Netw.*, 2011, pp. 49–60.
[49] K. Huang, C.-C. Ni, R. Sarkar, J. Gao, and J. S. B. Mitchell, "Bounded stretch geographic homotopic routing in sensor networks," in *Proc. 33rd IEEE Int. Conf. Comput. Commun.*, 2014, pp. 979–987.

Chen Wang received the BS and PhD degrees from the Department of Automation, Wuhan University, China, in 2008 and 2013, respectively. He is currently a postdoctoral fellow at the Huazhong University of Science and Technology, China. His recent research interests include wireless networking and communication protocols, especially geometric algorithms for wireless sensor networks. He is a member of the IEEE.



Hongzhi Lin received the BS, MS, and PhD degrees from the Huazhong University of Science and Technology, China, in 2000, 2003, and 2008, respectively. He is currently an assistant professor with the Huazhong University of Science and Technology. His current research interests include the areas of wireless networking and digital signal processing.

Hongbo Jiang received the BS and MS degrees from the Huazhong University of Science and Technology, China, and the PhD degree from Case Western Reserve University in 2008. He then joined the faculty of the Huazhong University of Science and Technology, where he is currently a full professor and the dean of the Department of Communication Engineering. His research interests include computer networking, especially algorithms and protocols for wireless and mobile networks. He is a senior member of the IEEE.