

On Oscillation-Free Emergency Navigation via Wireless Sensor Networks

Lin Wang, *Student Member, IEEE*, Yuan He, *Member, IEEE*, Wenyuan Liu, *Member, IEEE*,
Nan Jing, Jiliang Wang, *Member, IEEE*, and Yunhao Liu, *Fellow, IEEE*

Abstract—Emergency navigation is an emerging application of wireless sensor networks with significant research and social value. In order to ensure the safe and timely navigation of the evacuees, most of the existing works model navigation as a path-planning problem or movement decision support problem and adopt different metrics, such as the shortest route, the minimum exposure path, and the maximum safe distance. Without sufficient consideration of the dynamics of danger, the existing approaches are likely to cause users to move back and forth during navigation, known as oscillation. Frequent oscillations inevitably result in the user remaining in danger for a longer period of time, amplification of the user's panic, and eventual decrease in the chances of survival. In this paper we take users' oscillations in the dynamic environments into account and quantify the local success rate of navigation using a metric called ENO (Expected Number of Oscillations). We then propose OPEN, an oscillation-free navigation approach that minimizes the probability of oscillation and guarantees the success rate of emergency navigation. We implement OPEN and evaluate its performance through the trace from our system and extensive simulations. The results demonstrate that OPEN outperforms the current state-of-the-art approaches with respect to user safety and navigation efficiency.

Index Terms—Reachability, oscillation, emergency navigation, wireless sensor networks

1 INTRODUCTION

WIRELESS sensor networks (WSNs), born with the ability to automatically monitor and interact with the physical world under various environmental dynamics, have been receiving increased attention in recent years [1], [2], [3], [4], [5]. Navigation is an emerging application of WSNs, in which sensor nodes collaboratively explore the dynamic environmental conditions and people's movements [6], [7], [8], [9], and then prevent people in danger from once again traversing into the dangerous area, such as a geologic hazard, fire rescue, oil spill, etc. A WSN system for forest monitoring, GreenOrbs [1], [7] was deployed in the TianMu Mountains of China. Regarding potential disasters such as wildfires and landslides, navigation service is very important in ensuring visitor safety. Because the monitored area is very large and WSN deployment generally contains a number of sensor nodes, a directed user can only have a limited field-of-view and the local network information. Navigating the user safely to the destination becomes very challenging, especially considering the dynamically spreading danger area. The goal of guaranteeing the safety of directed users motivates us to study a highly efficient and reliable navigation approach.

Navigation with WSNs is attractive but challenging, due to the resource constraints of low-cost sensor nodes and the ad-hoc deployment of a WSN in large areas. A key issue in designing navigation approaches is the metric for evaluating a path's quality with respect to user safety and navigation efficiency. The existing works [10], [11], [12], [13] tackle the tradeoff between these two metrics. Those approaches, however, mostly consider the emergency as a static phenomenon and do not sufficiently address the dynamics (proliferation, shrink, and movement) of danger in the navigation solutions. As a result, navigation paths provided by those approaches are not necessarily passable in the end, due to the changes in emergency situations and environmental conditions. In order to keep the users safe, those approaches have to recalculate the navigation paths frequently when the dynamics of danger are present. *Oscillation* is defined as a phenomena that the navigated user is guided to move back and forth passively in local area, due to external impact (e.g., environmental dynamics or crowd congestion).

We note that the oscillation is an essential reactive to guarantee the path safety on the path the navigation system selected. However, the passive reentrant most likely consumes the escape opportunity for the evacuated. Thus, in the design of OPEN we minimize the passive oscillation as much as possible, while at the same time, we do not deny the necessary oscillation on the moving path in order to ensure safety.

It might be a general belief that oscillation of a user during navigation and the resulting prolonged period required to successfully navigate a user to safety is acceptable. Considering the practical cases with emergency navigation, however, oscillation is not just a matter of time. Oscillations inevitably result in users remaining

- L. Wang, W. Liu and N. Jing are with the School of Information Science and Engineering and the Key Laboratory for Computer Virtual Technology and System Integration of HeBei Province, Yanshan University, Qinhuangdao, China 066004. E-mail: {wulin, wylu, jingnan}@ysu.edu.cn.
- Y. He, J. Wang, and Y. Liu are with the School of Software and the TNLList, Tsinghua University. E-mail: {he, aliang, yunhao}@greenorbs.com.

Manuscript received 26 Apr. 2014; revised 29 Nov. 2014; accepted 4 Dec. 2014. Date of publication 17 Dec. 2014; date of current version 31 Aug. 2015. 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.2014.2381242

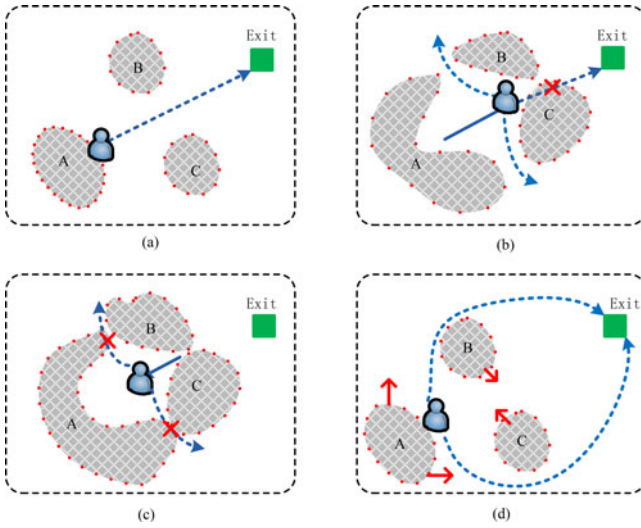


Fig. 1. Scenarios of emergency navigation using sensor networks. (a) indicates the generated path from a user to an exit. The user moves to the exit while the dangerous regions “B”, “C” spread toward the path as shown in (b). Moreover, the user attempts to turn back to find an alternative path. (c) shows that the alternative path is covered by dynamic danger “A”, which decreases the user’s chances of survival. (d) indicates the reachable path without oscillations.

in danger for a longer period of time and prolong the threat to the user’s safety. As the dynamic spreading danger (e.g., fire or gas leak) jeopardizes the user’s chances of survival, it is increasingly likely that frequent oscillations will cause the directed user to eventually miss the chance of survival.

Before further introducing the motivation of this work, we present a formal definition as follows: a navigation path is considered to be a reachable path, if and only if at any waypoint and the corresponding time point, the safety of the directed user is guaranteed. The reachability of a navigation path is then defined as the probability of a path to be reachable.

Fig. 1 shows an illustrative example of oscillation, which helps to understand the reachability of navigation. In our scenarios, three gray regions indicate dynamic emergency sites (designated *A*, *B* and *C*). The dotted arrow represents the selected direction of travel for a trapped user. The solid arrow indicates the path the user has previously traveled. According to temporal order, subgraphs (a)-(c) show the snapshots of navigating a user in three intervals. We can see that the user will be navigated to an exit. However, oscillation occurs when the danger zones encroach the selected path, forcing the user to turn back and find an alternate route. Unfortunately, the dynamic emergency has spread and blocked all escape routes. We hope to predict the moving direction of the emergency sites and generate a reachable path without oscillations as shown in subgraph (d).

The above example reveals that the dynamics of an emergency should be carefully taken into account in the design of navigation with WSNs. Nevertheless, it is unreliable to have only passive reactions to the dynamics of emergency, because frequent oscillations will decrease the user’s chances of survival. An efficient navigation approach should closely track the changes of an emergency in the environment and make proactive decisions for the

navigated users, so as to guarantee the eventual success of navigation.

In this paper, we propose OPEN, a navigation approach that provides oscillation-free paths in WSNs. OPEN smartly utilizes the sensing capacity of the sensor nodes to quantify the dynamics of emergency into ENO (Expected Number of Oscillations). The sensor nodes work collaboratively to distribute the ENO information across the network. Using ENO as a novel metric of path planning, OPEN finds navigation paths with the highest reachability and thus maximizes the success rate of navigation.

There are two main challenges in the above mentioned navigation process. One is how to accurately quantify the dynamics of emergency during a period of time, and the other is how to ensure the efficiency of distributed information exchange and state update, so as to support real-time navigation services. In our paper, we focus on utilizing the logical sequence to navigate the evacuated using WSNs. Our target is that the navigating path can predict the dynamics of the physical space in an emergency and reduce the invalid movement.

Our main contributions are summarized as follows:

- 1) We synthetically consider the spatial-temporal characteristics of emergency and propose the novel metric ENO to accurately quantify the emergency dynamics.
- 2) We design a light-weight distributed navigation approach, OPEN, that finds navigation paths with the minimum probability of oscillations and the best chance for the directed users to survive.
- 3) We theoretically analyze the reachability of navigation and prove the safety guarantee using OPEN. Moreover, we implement OPEN and demonstrate its performance advantages through extensive experiments and simulations.

The rest of this paper is organized as follows: Section 2 discusses the related work. The design of ENO and OPEN is introduced in Section 3. Section 4 presents theoretical analysis, proofs, and discussions on several important issues, followed by the performance evaluation based on real system and extensive simulations in Sections 5 and 6 respectively. The results show that the efficacy, efficiency, and scalability of the navigation service offered by OPEN outperform the existing methods. Finally, Section 7 concludes this work.

2 RELATED WORK

2.1 Mobile Robot Navigation

Navigation using sensors is inspired by navigating autonomous robots with sensors in the field of Robotics [14], [15], [16]. To address general motion safety issues, the inevitable collision states (ICS) concept is proposed in [14]. Our work is inspired by a version of ICS corresponding to passive motion safety in [17]. Unfortunately, passive motion safety does not hold in the scenario of emergency navigation with WSNs, which is a real-time service based on multi-hop and resource-constrained sensor networks.

Simultaneous localization and mapping (SLAM)[18] focuses on tracking an agent in an unknown environment

and constructing a real-time road map with a parallel mode. However, SLAM often relies on specific devices (e.g., stereo camera) and complex calculations (e.g., particle filter). Our target is presenting a fast, lightweight solution for emergency navigation.

2.2 Sensor Network Navigation

From the perspective of dealing with dynamic emergency, the existing navigation approaches of WSNs are classified into two categories: passive navigation [11], [19], [12], [13] and proactive safety navigation [9], [20], [21].

Recent works in this area, e.g., [10], adopt the potential fields and the hop count as metrics to calculate the optimal navigation paths. The authors propose an indoor navigation algorithm using temporally ordered routing with global flooding in [22]. To save the communication cost for initializing paths with global flooding, Buragphain et al. propose an algorithm based on the skeleton graph of a WSN [11]. Similar to a roadmap, a skeleton graph is a sparse subset of the original network. To avoid the side-effect of inaccurate positioning, the authors in [12] propose to navigate people along the medial axis of the safe field. The approach in [13] ensures that every user maintains at least a usable path. The partial reversal method used in [13], however, is likely to lead the directed users to emerging dangers.

Many passive navigation approaches embed a roadmap of the WSN that contains a collection of potential paths. It is worth noticing that the path reachability will be severely reduced when an oscillation takes place in any period along the navigation. In an emergency situation, the user flow and channel capacity are dynamic, which make those model-based methods lack robustness. By far, a lightweight solution is desired, which can efficiently predict the emergency dynamics and network conditions, and provides provable safety of navigation with WSNs.

2.3 Emergency Evacuation

Using wireless technologies and distributed decision, emergency response [23] or emergency management [24], [25], [26] (also called emergency evacuation [27]) enable intelligent and fast response to emergencies. We note that not only the dynamics of emergency but the crowd congestion can beget the oscillation path. Our work focuses on the former.

In [23], a real-time decision support system is proposed to provide reliable suggestions to the evacuees in the presence of a spreading hazard. It is supposed to work with a number of priori existing decision nodes. Such a scheme might not be scalable to large scale networks, due to the limit on the quantity of decision nodes. A distributed algorithm taking into account predictions of the relative movements of hazards in [23], [27]. The algorithm utilized a predetermined building model composed with two weighted graphs. The authors of [24], [25] focus on the distributed building evacuation simulator or evacuation decision system. The pedestrian congestion problem is addressed by means of the maximum flow and minimum cut theorem [28]. Cognitive packet network (CPN) is introduced to deal with routing and flow control in large scale and fast changing networks [26], [29]. Based on CPN, such a

approach in [26] can calculate the path traversal time and the maximum number of evacuees that can transit the concerned position over the period of time. From a practical point of view, the capacity of paths is bounded by the physical space, (e.g., the width of the exit or the corridors), which is susceptible to the dynamics of emergency instead of the path traversal time. Therefore, our work attempts to propose a new path metric that is sensitive to the dynamics of emergency environments. In addition, those works on congestion management attack on finding a navigation path with spatial-temporal constraints in the perspective of routing control, especially in the scene of multiple users, whereas the potential escape opportunity caused by environmental dynamics is ignored.

From the above description, you may notice that in our navigation problem, both the map and the paths are included, but modeled in a new way. The motivation behind this new model is that we try to leverage the wireless sensor network technique to offer more reliable and efficient navigation service.

3 DESIGN

In this section, we elaborate on the design of OPEN. We address the scenario that a trapped user is navigated towards a destination. In practice, the destination could be the exit of the dangerous area or the place where a rescuer is positioned. The goal is to ensure the user's safety during the navigation process.

This section introduces the model and definitions, followed by an overview of our basic idea. The third part presents the details of the ENO metric. The fourth part introduces the process of path generation and navigation.

3.1 Basic Idea

Our basic idea works as follows: A user node equipped with a radio module communicates with the WSN. Initially, each node uses a location predictor to estimate the relative distance from itself to the exit and the dangerous areas. Then the emergency predictor is triggered to select the waypoints and calculate the values of ENO regarding the optional segments and paths. According to the real-time collected information from the emergency predictor and the location predictor, the user node generates a navigation path and keeps updating it, minimizing the probability of oscillations during navigation until the user safely reaches the destination.

3.2 Model and Definitions

We use the truncated observations graph as the basic network model. The navigation scenario is mapped to a 2-D Euclidean plane, where a WSN is deployed. The WSN is modeled by an undirected graph $G(V, E)$, where V is the set of vertexes and E denotes the set of edges. Each vertex $v(v \in V)$ corresponds to a node and is presented as a six-tuple: $\langle ID, DS, h_p, h_e, h_u, Q \rangle$. ID is the unique node identifier, which is assigned when the sensor network is deployed. There are two types of node roles: *precarious node* and *general node*, which indicate a node inside or outside of the dangerous region. The emergency is regarded as the existence of a set of precarious nodes whose sensor readings (e.g., temperature) exceeds a predefined threshold. The

dangerous region is then modeled as a convex hull of the subset of precarious nodes and their 1-hop neighbor nodes. DS denotes the set of neighbor precarious nodes of a general node. h_p , h_e and h_u respectively denote the hop counts from a general node to its nearest precarious node v_p ($h_p = 0$), exit v_e ($h_e = 0$), and user v_u ($h_u = 0$). The term “nearest” indicates the least hop-count distance between two nodes. Q denotes the vertex sequence on the navigation path, on which the user node v_u is the header and the exit node v_e is the rear.

Our navigation protocol searches for a series of waypoints which correspond to sensor nodes deployed in the navigation scenario. Q is constructed by joining multiple q_{mm} , while each q_{mm} is a sequence of nodes between two adjacent waypoints (v_m and v_n). Note that the node's state corresponds to a time variable t as the environment changes. The connection between any two adjacent waypoints is viewed as a segment.

The node role may be transformed with the dynamics of the emergency. In our design, the dynamics of the dangerous region are mapped to the movement of a node towards the segment between two adjacent waypoints. For a segment, the speed of the precarious node is positive if the danger is approaching it. Otherwise, it means the dangerous region is moving away from the segment. The greater the velocity of the dangerous region, the higher the probability a directed user will encounter oscillations. Note that the node itself actually does not move. The movement velocity of a precarious node is calculated by quantifying the tendency of a node to be transformed from general to dangerous. This calculation procedure is introduced in detail in Section 3.3.2.

Joining the segments one by one will form an alternative path. In other words, a segment is the sequence of nodes between two waypoints which are on an alternative path. If a user is positioned at a node on a segment, we say that the user is sensed by the segment. The waypoint uses the node sequence, which is stored in the buffer as a queue, to guide a covered user to the next waypoint until reaching the exit. From the global view, the set of waypoints V' is a subset of V , and the set of segments E' is a subset of E .

3.3 Design Overview

Fig. 2 shows an architecture overview of the proposed navigation system. There are three main modules in our navigation system: location predictor, emergency predictor, and path generator. These modules are integrated on every sensor node, supporting a navigation workflow that consists of several phases, namely initialization, waypoints selection, path generation, and user navigation. There are three following challenges against our proposed scheme.

We give an imaginary scenario: a number of users equipped with portable devices are lost in an emergency environment. The portable device uses an RF module to access the WSNs which are deployed in advance. First, each node uses the location predictor to evaluate the relative distance between them, the exit, and the danger. Second, the emergency predictor is triggered to select the waypoints and calculate the proposed metric ENO of segments and paths. Finally, the user node generates a navigation path

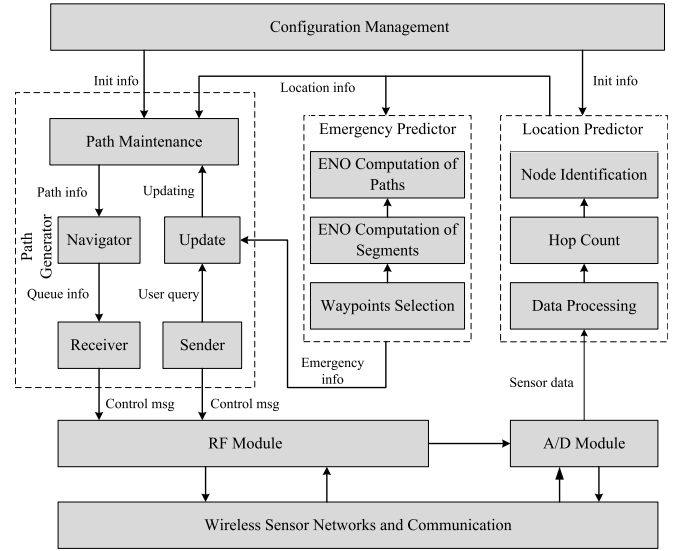


Fig. 2. Overview of our navigation architecture.

and updates it avoiding oscillations on the path according to the information from the emergency predictor and location predictor in a real-time pattern. The users avoid the dangerous regions and reach the exit without oscillations in accordance with the indicators on the device. There are three challenges in such circumstances:

- How to quantify the reachability of a specified navigation path in the distributed manner.
- How to predict efficiently the tendency of emergency using local in-network information.
- How to update the navigation path avoiding oscillations in the process of user navigation.

The details of our solution are described in the rest of this section.

3.3.1 Initialization

In the initialization phase, h_p , h_e and h_u are first set to zero on each node. The default node role is general node except for the nodes at an exit or a user. The real-time hop count may be captured using the periodic probes, presented as MSG . A node receiving a probe from another node determines whether to update the relative hop count, if the probe is the first valid MSG received during a probing period. The corresponding node role and hop count will be updated once there is a significant change on the node. For example, when the temperature reading on a node exceeds a given threshold, the node changes its role to precarious node and floods a probe MSG_p . Once MSG_p is received, its 1-hop neighbors accordingly change their roles to precarious node and forward the message.

Here, we are trying to express the dangerous status of each sensor in emergency scenarios, which is implemented via the periodic sensing and exchanging information with sensors. In our design, the hop count is used to indicate the difference of distance between the emergency site and the possible paths and therefore OPEN belongs to the location-free navigation scheme. To be precise, the severity of the danger can be represented by an increased hop count, which is closely related to the speed of emergency spread

and the update frequency of sensing. That is to say, the faster the emergency spread and the lower the frequency update, the greater the hop count should be set to, so that the sensor status can be satisfied with the realistic situation and vice versa. The size of the hop count in defining the precarious node is actually a quantitative reflection of the spreading speed of the emergency. In a situation where the emergency spreads faster, it is suggested to adopt a larger size of hop count to define the precarious node. In this way, the user's safety can be effectively ensured. Note that the calculation result of OPEN is essentially determined by the actual situation, instead of how a precarious node is defined.

Regarding the spreading emergency, we assume that the movement of emergency in a specified interval (i.e., the sensing interval of sensors in the network) does not exceed the average distance between any two sensor nodes. From this regard, we have a quick inference: if a node becomes a precarious node at some time point, the emergency in a sensing interval of the sensors can affect at most the 1-hop range centered at that node. In other words, if a node becomes a precarious node, its 2-hop neighbors will not be affected in a sensing interval of the sensors. We employ 1-hop as the hop count by means of our common knowledge. We note that the emergency might move at an extremely high speed in some special cases (e.g., a radiation leak). Then the assumption needs to be modified by changing 1-hop to k -hop, where k indicates the affected range of the emergency movement in a sensing interval of the sensors. It's worth noticing that the value of k does not affect the overall design effectiveness of our approach. After other nodes receive the message, they change the hop count between them and the precarious nodes. In the next period, the value is updated in the same manner. MSG_u and MSG_e is broadcast by a user node and an exit node, respectively. This process is periodically executed to obtain the relative position among the nodes.

3.3.2 Waypoints Selection

User nodes and exit nodes are pushed into the initial waypoint set. Our approach supports solving problems of multiple users and multiple exits. A periodic broadcast mechanism is maintained, so that all nodes receive the packets transmitted by the nodes which are potential users or exits. Each node can decide to discard or use the packet. Each user node is viewed as an origin; any exit node is viewed as a terminal. The distance between a node v_m and another node v_n is denoted by h_{mn} . v_p is set to their shared precarious node which is a convex point towards the segment $\overline{v_mv_n}$. We assume that h_{mp} and h_{np} are greater than or equal to two hops. We select the maximum of h_{mn} as k_{mn} , which denotes the length of the segment $\overline{v_mv_n}$.

For predicting the moving tendency of emergency, getting the reasonable k_{mn} is important with regard to an appropriate trade-off between the computation cost and the prediction accuracy. We can find the reasonable k_{mn} by searching v_n in a non-obtuse triangle (e.g., $\triangle MNP$ in Fig. 3). The distance from v_n to v_m , h_{mn} , is bounded by

$$\sqrt{|h_{mp}^2 - h_{np}^2|} < h_{mn} \leq \sqrt{h_{mp}^2 + h_{np}^2}. \quad (1)$$

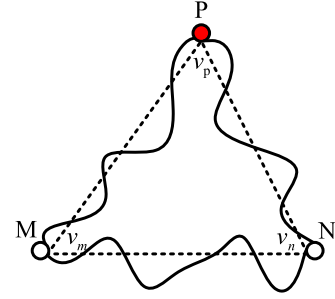


Fig. 3. The virtual triangle with two potential waypoints(M, N) and a precarious node(P).

Next, node v_n is pushed into the waypoint set V' . During the next iteration, v_n is viewed as the starting point. Similarly, we can find the next waypoint. After multiple iterations, the set of waypoints V' is generated for a user. The iteration process is not stopped until v_n is an exit node. The process of generating V' is shown in Algorithm 1.

Algorithm 1. Recognizing Waypoint Algorithm

```

1: while A node  $m$  receives a flooding message  $MSG_p$  from a
   precarious node  $p$  do
2:   if Hop count  $h_{mp} \geq 2$  & the node  $m$  is a user node then
3:     Pushing  $m$  into  $V'$  and transmitting  $MSG_m$  to the
       neighbor  $n$  of  $m$ ;
4:   else
5:     Forwarding  $MSG_p$  to next node;
6:     while Each  $n$  receives  $MSG_u$  do
7:        $h_{mn}++$ ;
8:       if  $h_{mn}$  satisfies by inequality (1) then
9:          $k_{mn} = \max(h_{mn})$ ;
10:        Pushing  $n$  into  $V'$ ;
11:       else
12:         Forwarding  $MSG_u$  to next node;
13:       end if
14:     end while
15:   else
16:     Deleting  $m$  from  $V'$ ;
17:   end if
18: end while

```

3.4 Proactive Navigation Metric

To quantify the reachability of a path, the node role and the relative position, the moving velocity of the dangerous region and the ENO of a path are studied in this section.

3.4.1 Calculate the Velocity of Emergency Spread

It is indeed non-trivial to avoid oscillation while navigating a trapped user to the exit in the dynamic environment. Our goal is to quantify the trend of the precarious node's movement. The dynamics of a dangerous region are mapped to the movement of a precarious node towards the selected segment. This conversion has two purposes: one is to ensure a reliable and stable symmetric link using handshake; the other is to obtain the time interval which is used to calculate the velocity and direction of a node's movement.

By constructing a virtual non-obtuse triangle ($\leq \pi/2$), the velocity of any node can be calculated. The triangle is composed of the segments among the two adjacent waypoints and a precarious node. Suppose $l_{mn}(t) = k_{mn} \times \varpi$, ϖ

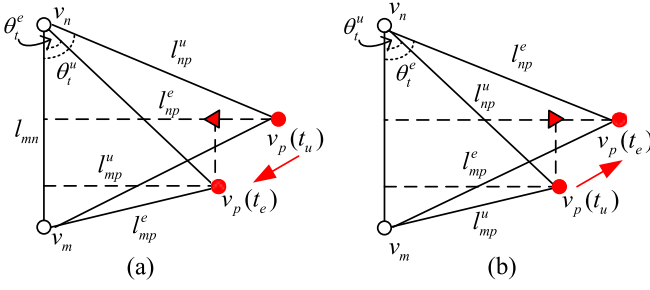


Fig. 4. Moving tendency of precarious node v_p with a segment. (a) v_p moves closer to a segment; (b) v_p shows moves away from a segment.

denotes the average hop distance. In a real network, the triangle's edge may consist of many non-collinear vertexes.

The principle of calculating the movement velocity is shown in Fig. 4. θ_t denotes the angle between the segment $\overline{v_m v_n}$ and $\overline{v_m v_p}$. The length of $\overline{v_m v_n}$, namely k_{mn} , is calculated by Algorithm 1. We can find the relationship among those segments with their angles by Cosines Formula as follows:

$$\cos \theta_{t_u} = (l_{np}^2(t_u) + l_{mn}^2 - l_{mp}^2(t_u)) / 2l_{np}(t_u) \times l_{mn} \quad (2)$$

$$\cos \theta_{t_e} = (l_{np}^2(t_e) + l_{mn}^2 - l_{mp}^2(t_e)) / 2l_{np}(t_e) \times l_{mn} \quad (3)$$

$l_{mp}(t_{u/e})$ means the distance between v_m and v_p at the time point $t_{u/e}$. The distance is approximately equal to the number of hops between them. Thus, the relative movement distance l_p^{mn} of the mobile precarious node v_p with the segment can be calculated as a projection of v_p on the vertical line $\overline{v_m v_n}$ as follows:

$$l_p^{mn} = l_{mp}(t_u) \times \cos \theta_{t_u} - l_{mp}(t_e) \times \cos \theta_{t_e}. \quad (4)$$

During the time interval $t(t = \Delta t = t_e - t_u)$, the velocity s_p^{mn} of the precarious node towards the segment is given by:

$$s_p^{mn}(t) = \frac{l_p^{mn}}{\Delta t}. \quad (5)$$

When the precarious node is approaching the segment (see Fig. 4a), the velocity is a positive value. Otherwise, it is a negative value (see Fig. 4b).

3.4.2 Expected Number of Oscillation

Now we formally introduce the metric Expected Number of Oscillation which measures the number of possible re-entrant oscillations of a user on a segment. The ENO of a navigation path is the sum of ENOs of all the segments on the path. Our goal is to find a navigation path with the minimum ENO, so that the roadmap provided to the navigated user is ensured reachable with the highest probability.

The ENO of a segment during a time interval t is given by

$$e_p^{mn}(t) = s_p^{mn}(t) \sum_{v_i \in q_{mn}(t)} \sum_{v_p \in DS_i} h_{ip}^{-2}(t), \quad (6)$$

where $\sum \sum h_{ip}^{-2}(t)$ and $s_p^{mn}(t)$ are the spatial and temporal accumulated emergency level on the segment from v_m to v_n during the time interval t . DS_i denotes the neighbor

precarious nodes set of v_i . $q_{mn}(t)$ is generated by forwarding MSG_u and is stored in every general node between two waypoints.

For example, by receiving the MSG_p , DS_m is recorded on v_m as $\{v_1, v_2, v_3, v_4\}$, and DS_n is recorded on v_n as $\{v_3, v_4, v_5\}$. At the same time, the hop count between the general nodes and the precarious nodes can be logged. In this case, h_{mp} is $\{5, 3, 3, 4\}$ and $\{5, 3, 3, 3\}$, h_{np} is $\{4, 3, 3\}$ and $\{4, 2, 3\}$ at time t_u and t_e , respectively. Note that there are two shared precarious nodes v_3 and v_4 for both waypoints. v_4 is selected to predict the emergency dynamics. Since $h_{mn} = 4$ is logged according to the received MSG_u and MSG_e at time t_u and t_e ($\Delta t = 10$), $s_p^{mn}(\Delta t) = (2.236 - 2.05)/10 = 0.12$, $\sum h_{mp}^{-2}(t_u) h_{np}^{-2}(t_u) \sum h_{mp}^{-2}(t_e) h_{np}^{-2}(t_e) = (1/5^2 + 1/3^2 + 1/3^2 + 1/4^2)(1/4^2 + 1/3^2 + 1/3^2)(1/5^2 + 1/3^2 + 1/3^2 + 1/4^2)(1/4^2 + 1/2^2 + 1/3^2) = 0.3225 + 0.2825 + 0.37 + 0.4225 = 1.3975$. Therefore, we have $e_p^{mn}(t) = 0.12 \times 1.3975 = 0.1677$ as the ENO of the segment $\overline{v_m v_n}$.

Note that both the dynamics of environment and the mobility of human beings are impacting factors to the navigation process. OPEN, as we propose, should take those two factors into consideration. What we want to clarify here is that the calculation of ENO in OPEN is a periodical process that continues throughout the navigation process. In other words, the ENO of the navigation path is kept updated as the user moves under the direction of OPEN.

The ENO of a path as $E_{ue}(T)$ across the network from the user node v_u to the exit node v_e , is given by

$$E_{ue}(T) = \sum_{v_m, v_n \in Q_{ue}(T)} e_p^{mn}(t). \quad (7)$$

$Q_{ue}(T)$ denotes the node sequence which is dynamically updated by forwarding MSG_e and stored in every waypoint between a user node and an exit node. T indicates the cumulative time which is the sum of Δt of the segments on the path $\overline{v_u v_e}$. We want to clarify a waypoint sequence as a logical path via WSN and the decision to move along a safety and moveable escape path. It is not emphasized that the logical path is mapped into the physical space perfectly in our paper. In this pattern, the evacuee is able to track any targeted sensor node by measuring the strength and direction. Under the circumstances, we consider the priorities of emergency navigation to be different from the physical path in emergency management.

3.5 Path Generation and User Navigation

Upon receiving MSG_e sent by its nearest exit, a waypoint computes the moving velocity of the closest precarious node value, i.e., $s_p^{mn}(t)$, using Equations (2)-(5) and the ENO $e_i^{mn}(t)$ of any segment using Equation (6) at the time slot t . Next, the waypoint adds the ENO of its corresponding segment to the ENO of the current path. Then the waypoint forwards the updated message to the next waypoint with the updated ENO and the current sequence of waypoints.

In a period T (T is a time interval in which a user node receives a waypoint sequence), $E_{ue}(T)$ is obtained for every user node. The pseudocode is shown in Algorithm 2.

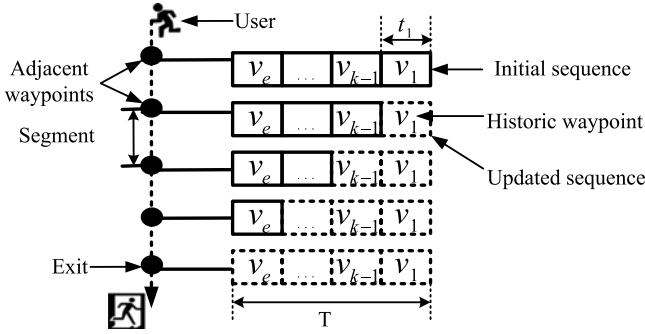


Fig. 5. The process of sequence mismatch and direction updating.

Algorithm 2. Computing ENO of Path Algorithm

```

1: while Receiving a flooding message  $MSG_e$  from the exit
   node  $v_e$  do
2:    $v_m \leftarrow v_e$ 
3:   if The node is neighbor waypoint of  $v_m$  then
4:     for  $v_n \leftarrow$  all neighbor node of  $v_m$  do
5:       Computing  $s_p^{mn}(t)$  of any two waypoints using Equa-
         tions (2)-(5);
6:       Pushing the waypoint  $v_n$  of the path into  $Q_{ue}(T)$ ;
7:       Computing  $e_i^{mn}(t)$  using Equation (6);
8:        $T = T + t$ ;
9:        $E_{ue}(T) = E_{ue}(T) + e_i^{mn}(t)$  and updating  $MSG_e$ ;
10:       $v_m \leftarrow v_n$  and forwarding it to next waypoint;
11:   end for
12: else
13:   Forwarding the  $MSG_e$  to its neighbor(s);
14: end if
15: Return  $E_{ue}(T)$ ,  $Q_{ue}(T)$ 
16: end while

```

Users need to select a reachable path using the metric. A user node selects the minimum ENO (corresponding to the shortest path) as its reachable path and records the waypoint set $Q_{ue}(T)$ and the set of segments E' . Movement direction of the user node will be updated if and only if the waypoint sequence of the path is significantly changed (see Fig. 5). That is, the original sequence $Q_{ue}(T_{n-1})$ on the user node significantly mismatches with the current sequence $Q_{node}(T_{new})$ on the waypoint, where T_{n-1} is the timestamp of departing from the previous waypoint, and T_{new} is the timestamp of arriving at the current waypoint. Actually, generating path and navigating user are designed as enqueue and dequeue operations, respectively. Updating movement direction is triggered once a mismatch operation happens at the header of the queue for both sequences. We show the pseudo code in Algorithm 3.

Assuming that there are N nodes in the network, including m user nodes and n precarious nodes. Algorithm 1 is executed once in the initialization phase in a distributed pattern, so the total amount of time to run steps 6 to 14 is n^2 . Due to the number of users gradually decreases as navigation proceeds, the total time required to run the algorithm can be expressed by $(1 + mn^2)$. Generally, the number of user gradually decreases with the change of time, which means that the upper bound of the algorithm's running time is $\mathcal{O}(mn^2)$. Intuitively, the running time of the algorithm is closely related to the growth of the number of

precarious nodes. Each node store and forward a constant number of data packets, so the space complexity is $\mathcal{O}(1)$. For Algorithms 2 and 3, the basic estimation of ENO has $\mathcal{O}(m(N - n))$ temporal complexity. As matter of fact, the number of precarious nodes is far less than that of general nodes. As $N \rightarrow \infty$, the temporal complexity can be expressed by $\mathcal{O}(mN)$. Algorithm 2 needs to calculate ENO between two adjacent waypoints, so the spatial complexity is bounded by $\mathcal{O}(mn)$. similarly, we can get $\mathcal{O}(m)$ spatial complexity for Algorithm 3. Totally, the protocol's spacial complexity is $\mathcal{O}(mN)$.

Algorithm 3. Navigation and Path Updating Algorithm

```

1: while Receiving  $Q_{ue}(T)$  do
2:   Selecting a  $Q_{ue}(T)$  with the minimum  $E_{ue}(T)$  to escape;
3:   for  $n = 1 : 1$  /*  $n = |Q_{ue}(T)|$  */ do
4:     Deleting its ID from  $V'$  when arriving a node;
5:      $n = n - 1$ ;
6:     if  $Q_{ue}(T_{n-1})$  mismatch with  $Q_{node}(T_{new})$  then
7:        $Q_{ue}(T) = Q_{node}(T_{new})$ ; /* updating the R*/
8:     else
9:        $Q_{ue}(T) = Q_{ue}(T_{n-1})$ ; /* continuing the R*/
10:    end if
11:  end for
12: end while

```

A user of the navigation system relies on the information computed using Algorithms 1-3 to get continuous feedback from the network on how to reach his/her destination.

4 ANALYSIS AND DISCUSSIONS

For the trapped users, it is a foremost requirement that the movement direction can guide them away from danger. Therefore, the algorithm must be correct and efficient. The theoretical analysis of correctness is presented as follows.

4.1 Correctness

The proposed protocol can dynamically update a potentially dangerous path and navigate correctly the user to escape along a reachable path. Now we prove the correctness of Algorithm 3. Moreover, it is proved that the aggregated ENO of the actual escape path is bound by the ENO of the proposed path.

Theorem 1. Algorithm 3 always generates a reachable path to an exit.

Proof. In Algorithm 2, the user node at which the user arrives keeps the latest $E_{ue}(T)$ and $Q_{ue}(T)$. Moreover, each general node keeps its prior segment's waypoint close to the exit node. Each node other than the user node has a smaller $E_{ue}(T)$. It's proved that there are no local minima in the network. \square

The user node can always find a node in its path sequence $Q_{ue}(T)$ that has a smaller ENO. If the process continues, the user node will successfully escape via the exit that has the smallest $Q_{ue}(T)$ ($= \emptyset$). Therefore, Algorithm 3 can correctly find a reachable path to an exit for a user.

Note that the network does not spread MSG_e in the worst case, i.e., all exits are not reachable. when there is not

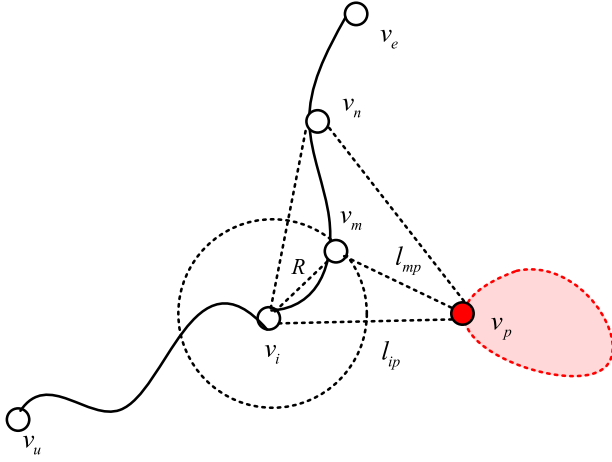


Fig. 6. The upper bound of the integrated ENO.

a reachable path to the exit, our approach cannot yield a path for the user. Specifically in our design, messages from an exit node are disseminated in an event-driven manner. As a result, the message can no longer be transmitted to the rest of the entire network, when no exit is really reachable. Then, the network does not spread a message and the algorithm will stop updating the navigation path. According to the most recent updated result, the user may choose to stay at a safe place and wait. At the same time, sensors keep sensing the environment, so MSG_e may be tracked periodically. Once MSG_e appears again in the network, the algorithm will be recovered and return to service.

We now compare the integrated ENO values on the path found by our protocol and the optimal path to show how reachable the found path is.

Theorem 2. *The aggregated ENO on the computed path is upper bound of the integrated ENO on the escape path.*

Proof. By running our protocol, we find a path from v_u to v_e , E_1 indicates the sum of the ENOs on the segments connected with waypoint nodes, and the path sequence of the whole path Q_{ue} is $\overline{v_u v_1} \dots \overline{v_{k-1} v_e}$. Let $\overline{v_0 v_1}, \overline{v_1 v_2}, \dots, \overline{v_{k-1} v_k}$ be the segment sequence connecting consecutive waypoints. E_2 denotes the aggregated ENO of all the segments on a navigation path. We first calculate the upper bound of E_2 and then prove that E_1 cannot be greater than that upper bound. Let the ENO value of $\overline{v_i v_m}$ with any precarious node v_p be e_p^{ij} , where $|\overline{v_i v_m}| \leq R$ (R is the transmission range of the sensor nodes). For a precarious node v_p moving towards the segment $\overline{v_i v_{i+k}}$, $k(\leq |\overline{v_m v_i}|)$ is the hop count of the eligible segment (see Fig. 6). \square

Let v_i be a 1-hop neighbor of v_m , and l_{mp} be the distance between v_m and v_p . We can get $l_{mp} \geq l_{id} - R$. For any point j on the segment $\overline{v_i v_m}$, let l_{ip} be the distance between j and v_p . Because $l_{jp} \geq l_{ip} - R$, the ENO of $\overline{v_i v_j}$ can be given as follows:

$$\begin{aligned} e_p^{ij}(t) &= s_p^{ij}(t) \times h_{jp}^{-2} = s_p^{ij}(t) \times R^2 \times l_{jp}^{-2} \\ &\leq s_p^{ij}(t) \times R^2 \times (l_{jp} - R)^{-2}. \end{aligned} \quad (8)$$

Thus we have

$$\frac{e_p^{ij}(t)}{e_p^{im}(t)} \leq \frac{s_p^{ij}(t) \times R^2 \times (l_{ip} - R)^{-2}}{s_p^{ij}(t) \times R^2 \times l_{ip}^{-2}} = \left(1 - \frac{R}{l_{ip}}\right)^{-2}.$$

Then,

$$e_p^{ij}(t) \leq \left(1 - \frac{R}{l_{ip}}\right)^{-2} \times e_p^{im}(t).$$

By summing up ENOs along the entire path, we have:

$$\begin{aligned} E_2 &= \int_{t_u}^{t_e} \sum_p e_p^{ij}(t) p(t) = \sum_{i=0}^{k-1} \int_i^m \sum_p e_p^{ij}(t) p(t) \\ &\leq \sum_{i=0}^{k-1} \int_i^m \sum_p \left(1 - \frac{R}{l_{ip}}\right)^{-2} e_p^{ij}(t) p(t) \\ &\leq R \left(1 - \frac{R}{l_{ip}}\right)^{-2} \sum_{i=0}^{k-1} e_p^{ij}(t), \end{aligned} \quad (9)$$

where $|\overline{v_i v_m}| \leq R$. If $(1 - \frac{R}{l_{ip}})^{-2} \leq q$ for all v_i and v_p , we get

$$E_2 \leq R \left(1 - \frac{R}{l_{ip}}\right)^{-2} \sum_{i=0}^{k-1} e_p^{im}(t) \leq RqE_1.$$

This tells us that the actual ENO value is not more than the computed ENO value of waypoints on the proposed path. Thus, we consider the path to be a reachable path.

4.2 Discussions

Here we have further discussions on several important issues regarding the efficacy and efficiency of our proposal.

4.2.1 Modeling Dynamic Danger

Modeling the dynamics of danger (i.e., the danger region and its dynamic change) is a necessary but challenging task in emergency navigation via WSNs. An appropriate model enables one to accurately characterize and predict the danger, which avoids the potential oscillations of the navigated user and ensures the safety and reliability of navigation. Note that the quality of a model actually depends on the domain information provided by the system designer and the resource constraints. Specifically, different amounts of information supports modeling at different granularity. The resource on the network devices (e.g., the memory space and the computing capacity on the sensor nodes), usually restricts the use of complex models.

Our work in this paper mainly focuses on the design and implementation of a navigation protocol, given a specific model of danger. With respect to the model parameters, we only assume the danger region has a computable propagation speed such that one can quantify the changing trend of danger. Our algorithms and protocol are not closely coupled with the model used in this paper, and thus can be generalized to many other models, depending on which kinds of dangers we address in the applications.

Nevertheless, model refinement may be an interesting issue to be studied in the context of emergency navigation. Even in the case with limited domain knowledge or even

zero prior knowledge, one can also exploit sensor readings to get feedback information from the environment and improve the quality of modeling in an online manner. We leave this issue to our future work.

4.2.2 Path Oscillation and Moving Oscillation

On one hand, we note that there may be oscillations of paths without any oscillations of people. Path oscillation is a classic issue in packet networks. In the context of ad-hoc and sensor networks, path oscillation might be due to dynamic factors like link quality and node mobility. What we want to clarify here is that in this paper we mainly focus on the oscillation of people, which is generally caused by the change in the navigation path provided by the navigation service. When a previously used navigation path becomes unreachable, a new path must be provided to the user. If the change in the path demands the user to change his/her moving direction, a user oscillation occurs. If the change in the path does not demand the user to change his/her moving direction, there will not be a user oscillation. Hence, it is different between path oscillation and user oscillation.

On the other hand, we note that wireless signals can pass through the space constraints (e.g., walls), but people can not. Based on topography and application context, carefully planned network deployment may help to alleviate this problem. However, we also recognize that the deployment can not completely guarantee that the problem does not occur. In some extreme cases, users may need to detour to reach the next hop the algorithms give. If there is unattended emergency right in the detour the user takes from one node to another, our approach still needs to rely on the user's subjective judgment to ensure basic safety.

5 REAL SYSTEM BASED EVALUATION

We evaluate the ENO-based oscillation-free navigation approach, named OPEN. This section presents the performance results evaluating data transmission in real system with 300 nodes [30].

Other than the outdoor experiments, we conduct some preliminary tests of our algorithms with a flexible indoor testbed with 21 nodes. Those results can be reviewed in our previous work [31].

Through the experiments, we evaluate OPEN's efficiency in terms of the path reachability.

5.1 Data

In the outdoor testbed, the sink node collects all the sensor data. For experimental purposes, we assume the sink node is the exit. The location information of all nodes is obtained prior to the experiments, namely during the deployment phase. Collection tree protocol (CTP) [32] is adopted to collect data from the nodes, which include the environmental sensor readings, RSSI among the nodes, and link quality of the wireless links. In our experiments, we use a 120 MB data trace from the network, which includes all data in three days. Based on the trace, we construct a dynamic emergency scenario and analyze the performance of our proposed algorithms. Ninety nodes whose illumination readings vary in the interval $[2 \times 10^7, 5 \times 10^7]$ are regarded as precarious nodes in the experiments.

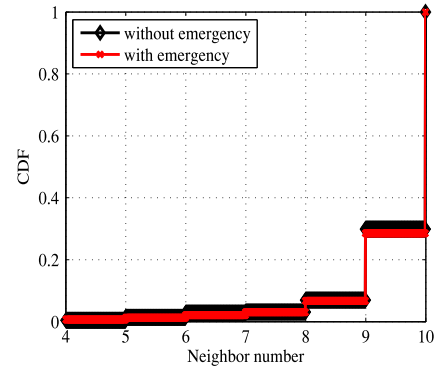


Fig. 7. Neighbor number varies with emergency.

5.2 Schemes for Comparison

We compare OPEN with other two schemes, namely *ETX + Hop Count* and *ETX + Dynamics Prediction*.

ETX + Hop Count. An ETX based link estimator is used to select the data transmission route. Hop count from any location to the exit or the precarious node to measure the path reachability.

ETX + Dynamics Prediction. This scheme includes OPEN's modules of virtual triangle based waypoint selection and movement speed prediction.

5.3 Scalability

In dynamic scenarios, the scalability of navigation service is an important metric for evaluating the applicability of our approach in real applications. We demonstrate the scalability of OPEN in terms of two different aspects. (1) Routing stability versus emergency dynamics; (2) Path generation versus emergency complexity.

5.3.1 Routing Robustness

Method. When an emergency happens, the network forms some precarious regions in which the packets may be transmitted but the nodes inside cannot be included in a navigation path. In order to evaluate the robustness of routing against emergency, we measure the size of the routing neighbor table on the nodes. We repeat the experiment for 20 different topologies, with transmissions generated from different packet sources.

Results. Fig. 7 plots the cumulative distribution function (CDF) of the neighbor table size for all the nodes on the navigation paths from the packet sources to the exit node. The mean length of navigation paths is around 8.5 hops in two scenarios (with emergency or without emergency). We can see that there is hardly any apparent change in the size of the neighbor table, which means routing is not affected by emergency factors. The routing mechanism used in our navigation approach ensures consistent and reliable routing across a variety of topologies and emergency patterns.

5.3.2 Efficiency on Emergency Scale

Method. In this experiment, we increase the number of precarious nodes in the emergency region and measure the impact of emergency scale on the navigation performance in terms of user safety. The user safety is quantified as the average hop distance from the nearest precarious mote to

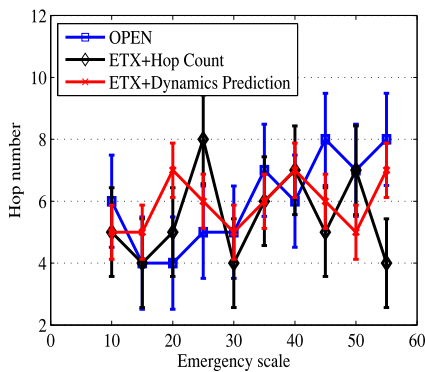


Fig. 8. Emergency scale versus danger distance.

the navigated user. We repeat the experiments for different topologies and emergency patterns.

Results. Fig. 8 plots the average hop distance from the nearest precarious mote to the navigated user, when using three different approaches (i.e., OPEN, ETX+Hop Count and ETX+Dynamics Prediction). While OPEN's performance scales gracefully, the performance of the two ETX-related approaches deteriorate when there are over 30 precarious nodes. Moreover, we can see the performance of ETX+Dynamics Prediction is close to that of OPEN and better than that of ETX+Hop Count. The main reason is the ETX+Hop Count approach cannot well react to the emergency dynamics. When a mote's state is changed from safe to dangerous, the ETX+Hop Count approach cannot effectively detect it, possibly leading to user oscillation. In comparison, the other two approaches can well react to the emergency dynamics. The ENO-based path planning solves this problem by extending the spacial dangerous value (precarious nodes scale) to the spatial temporal dangerous value (dynamics prediction).

Note that the performance of three approaches seems to oscillate regarding the hop count, especially ETX+Hop Count. The reason is that those nodes on the navigation path may be covered by an emergency in the abrupt change. ETX+Hop Count is sensitive in such cases, whereas our approach is much more resilient to such changes and shows stable increase and slight jitter with respect to the safe distance.

5.4 Oscillation Avoidance

In this experiment, we demonstrate OPEN's ability to offer navigation paths and avoid oscillations in an environment with dynamic emergency. We assume that path updating is implemented once a node of the path is converted to a precarious node.

5.4.1 Efficiency of Navigation

Method. This experiment is done in a selected scenario of navigation, with the terrain and the users' locations shown in Fig. 10a. User A needs to be navigated to Location Y as soon as possible, so a shortest path from his/her current location to Y is desired. However, the road heading for Y is blocked. So User A has to encounter an oscillation and make a U-turn to proceed. Using OPEN with timely ENO calculation based on the sensor information, User A can

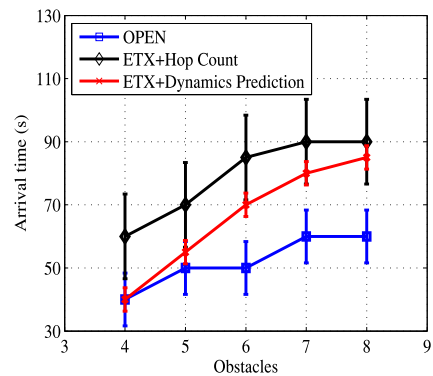


Fig. 9. Obstacle versus average time of arrival.

make a detour at the intersection, so that he/she reaches their destination via a slightly longer path. We assume that the average movement speed of a user equals to 3 meters per second (the average hop distance equals to 30 meters, while the user must be running in case of emergency). We repeat the experiment with different numbers of obstacles in the environment and measure the time taken for the user to reach the exit location Y.

Results. Fig. 9 plots the time taken for user A to arrive at Location Y versus the number of obstacles. All three schemes mentioned in Section 5.2 are tested. We observe that in a network of more than six obstacles, in average OPEN enhances the efficiency of navigation by 180 percent compared with the ETX+Hop Count scheme, and 140 percent compared with the ETX+Dynamics Prediction scheme. In this experiment, ETX+Hop Count predominantly selects the unreachable paths to the destination, while OPEN correctly finds the oscillation-free paths at the waypoints with high probability. Meanwhile, ETX+Dynamics Prediction performs marginally better than ETX+Hop Count because the dynamics prediction part adopts a more effective reachability metric, ENO. Nevertheless, the performance of ETX+Dynamics Prediction remains unsatisfactory when many precarious nodes exist, because path planning of that scheme is poor.

5.4.2 Oscillation Segments and Oscillation-Free Path

In this experiment, we present the efficiency of OPEN on avoiding oscillation in the scenes of Fig. 10.

Method. Consider two types of navigation paths: oscillation-free path and oscillation path. Here, if the number of

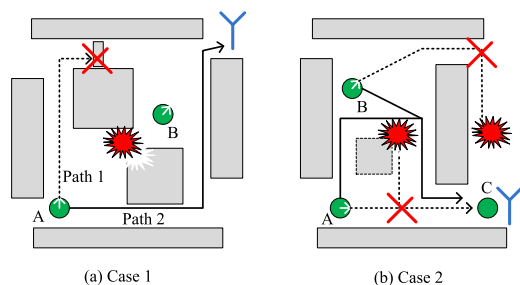


Fig. 10. Oscillation cases. For (a), the shortest path is blocked from A to destination by a NLOS obstacle. For (b), the path is unreachable as a result of the potential collision between A (or B) and the NLOS emergency. In both cases, either the path reachability or user safety is seriously threatened.

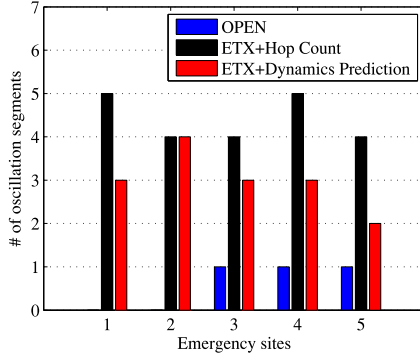


Fig. 11. Oscillation segments.

symmetrical sequence of adjacent segments on a navigation path is greater than 1, we define the segment corresponding to the symmetric sequence as an oscillation segment. For example, the segment of a navigation path q_t is $\{v_1, v_4, v_7, v_9, v_{11}\}$ at time t , while the adjacent segment of the same navigation path $q_{t+\Delta t}$ is $\{v_{11}, v_9, v_7, v_4, v_1\}$ at next time $t + \Delta t$. q_t and $q_{t+\Delta t}$ are called the symmetrical sequence of adjacent segments. Our goal is to provide a navigation path that contains the minimum oscillation segments. Such a path is called an oscillation-free path. Three methods are compared in terms of two performance metrics. One of them is the number of oscillation segments on a navigation path, and the other is the oscillation-free path ratio. The latter demonstrates the influence of the number of evacuee on the oscillation-free path.

Results. Fig. 11 plots the number of oscillation segments versus increased emergency sites. The number of oscillation segments in OPEN is well controlled within 1, and the number of oscillation segments with the other two methods are much higher. The reason is that OPEN can effectively sense emergency tendency and consider its cumulative changes in time and space, further avoiding the oscillation path. Fig. 12 plots the oscillation-free path ratio versus increased evacuee. The ratio with OPEN can be maintained at 100 percent, which means that any navigation path provided by OPEN contains oscillation segments less than 2. The reason is that the other methods need path updating once the evacuee meets the precarious node on the navigation path. OPEN always does path planning with ENO, which ensures the moving direction of the evacuee away from the emergency sites. Therefore, OPEN minimizes the number of oscillations shown in

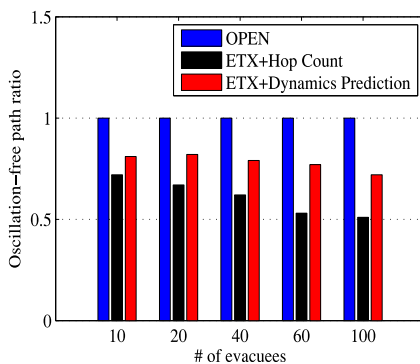


Fig. 12. Oscillation-free path ratio.

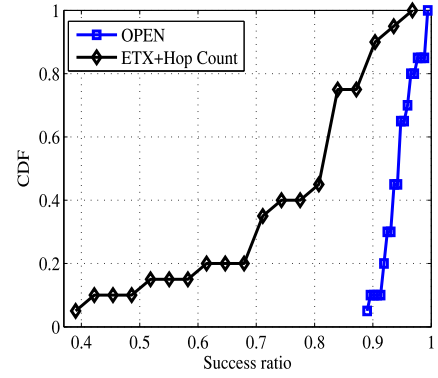


Fig. 13. CDF versus success ratio.

Fig. 10. If only considering accumulative effects of time or distance, as what the existing approaches do, leads to more oscillations.

5.4.3 User Safety

In this experiment, we evaluate the effectiveness of OPEN in terms of user safety during navigation.

Method. Consider a potential counter between the moving user and the dynamic emergency as shown in Fig. 10b emulating the common scenario in emergency navigation. User A is navigating towards a T-intersection and seeks to merge with a speeding danger on the path. That is a one of the most challenging problems in designing navigation systems, often requiring the human to run back and forth in the local area. In this topology, user C has passed the T-intersection before the danger arrives.

We note that all exits may be unavailable in the process of navigation. Fortunately, the potential escaping route will appear again with the dynamics of the emergency shown in Fig. 10b. The users A and B can be navigated towards the exit along the solid lines. Using sensor networks, the dynamics can be sensed. By informing users of the dynamics, new paths can be generated so as to adapt to the dynamic emergency. We evaluate the success ratio of navigation using a fraction of the reachable nodes in all nodes on the path. The higher the success ratio, the more reachable user A to detect the coming danger, while implementing OPEN against the ETX baseline implementation. We repeat the experiment with different numbers of emergency sites varying with different patterns (expanding or moving) in the environment.

Results. All three schemes are tested in the experiments. Fig. 13 plots the cumulative distribution of the success ratio of navigation paths. While the worst performance of ETX+Hop Count deteriorates to as low as 40 percent, OPEN successfully avoid danger with over 90 percent probability. That result demonstrates that OPEN is highly reliable in ensuring user safety during navigation. Moreover, OPEN performs relatively consistently under different experiment settings.

6 SIMULATIONS

In order to evaluate the scalability and the reachability of OPEN in large scale network configurations, we carry out extensive simulations to compare OPEN with three state-of-the-art approaches, namely the potential field

based approach (PF for short) [10], the medial axis based approach (MA for short) [12] and the CPN based approach (CPNST for short) [29]. CPNST takes the potential congestion into account and navigates the users with shortest evacuation time. In short, the congestions caused by the increased population density are regarded as emergencies in our simulation. For this purpose, we compare the efficiency and navigation safety of each approach. We are also interested in the approaches' robustness and behavior when physical obstacles exist in the environment.

6.1 Simulation Framework

We adopt an event-driven simulation framework written in C++ which is similar to Specksim simulator [33]. The simulation mainly contains two parts. The first part is a GUI which is implemented by C++. In this part, we allocate the movement of people, the current hazard situation and movement. We use a network topology of a grid partitioned with 900 to 16,000 nodes in a rectangular area. We partition the whole area into small cells with five nodes and the average distance of the both neighbor nodes is 20 meters. The maximum hop distance is set as 100 meters. The maximum moving speed of a person and the hazard are set as 3 and 10 meters per sensing interval (here the sensing interval is 1 second). The second part is a wireless sensor network that will monitor the environment by TOSSIM [34], collect the data, and send it to the sink node for further processing. A CSMA protocol and CTP-based routing protocol are implemented in our simulation. The radio model is based on the CC2420 radio, and the RF noise and interference a node hears that is manually configured based on meyer-heavy pattern in TOSSIM. For each round, the evacuated users and the emergency site are randomly generated in the field. The ratio of the size of emergency site to network size is kept below 15 percent. We tune the following parameters to evaluate the performance of OPEN: the network size, the node role, and the speed of emergency spread. We compare OPEN to the state-of-arts from five perspectives, namely average reachability, robustness, minimum average length, minimum dangerous distance, and minimum exposure path.. All simulation results are the output of average values taken after 20 runs.

6.2 Path Reachability

The purpose of this group of simulations is to compare OPEN with PF, MA and CPNST with respect to navigation success ratio, which is measured by the average path reachability. In our simulations, the average path reachability is expressed as $1 - \frac{N_e}{N_u \times J}$, where N_e indicates the number of nodes that become precarious nodes after f times of changes in the dangerous area for all of N_u users. In the controlled simulation, we consider that the only factor, which causes navigation failure, is the oscillation on the navigation path.

We assume that the emergency exhibits dynamics in two patterns: shift and spread. The danger refreshing period is set to 10, 30 and 60 seconds. In order to compare the path reachability, we consider four different cases: (i) 40 users reaching one exit against shifting emergency (named *SE40*),

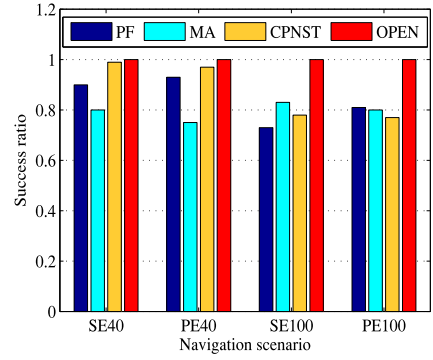


Fig. 14. Navigation success ratio.

(ii) 40 users reaching one exit against spreading emergency (named *PE40*), (iii) 400 users reaching four exits against shifting emergency (named *SE400*), (iv) 400 users reaching four exits against spreading emergency (named *PE400*). The sensor field has 400 nodes deployed uniformly. Fig. 14 shows that OPEN clearly outperforms PF and MA by always achieving 100 percent reachability in low occupancy rate (40 users). MA and PF fail to ensure the reachability in some scenarios, because they do not predict the tendency of emergency dynamics well. CPNST (with time metric) has similar success ratio with OPEN in same navigation scenario, because it takes the potential congestion into account. However, CPNST shows poor performance when the emergencies shift or spread. It is mainly because the CPN based approaches focus on the dynamics include by human mobility, PF, MA and our approach address the dynamics of environment. As a result, there are some precarious nodes on the navigation path using MA or PF or CPNST.

We further evaluate the scalability of OPEN at larger scales. Specifically, we carry out a group of simulations, where 1,600 nodes are randomly deployed in a 40×40 area. The number of users is set to 100 and 400. Fig. 15 shows the CDF of path reachability of OPEN, MA, PF, and CPNST. We can see that more than 99 and 96 percent users can be successfully navigated to the exit using OPEN and CPNST respectively. Both of them are much better than the cases with MA and PF. OPEN selects a reachable path starting from the current node to a reachable node using the metric ENO, which means the path reachability is not related to the scale of the network. This figure also shows that the path reachability using our approach presents notable stability as the scale of network increases.

6.3 Robustness

We run the same emergency pattern with different preconfigured node failure rate (10, 20 and 30 percent). During the simulations, we randomly select some nodes as having failed until the preconfigured node failure ratio is reached. The results indicate that OPEN is very robust and all users can reach the exit every time. Our maintenance module is able to keep the network connected and find alternative segments as the danger moves.

Fig. 16 shows the average number of segments (an indicator of the extent to which a navigation path is fragmented) in the roadmap as the emergency spreads. In the figure, experimental results for both mote-failing (motes are

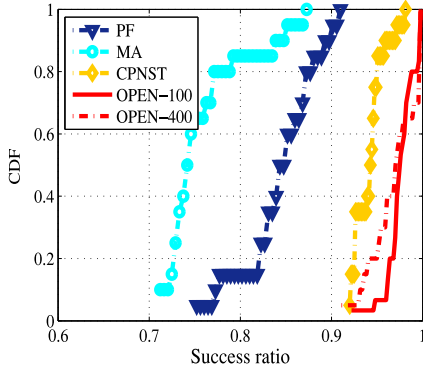


Fig. 15. CDF versus success ratio.

destroyed and no longer work) and non-mote-failing (motes are not destroyed and keep working) cases are shown. We can see that there is a slight decline in the number of segments, when motes in the dangerous region keep working. We also find that as the emergency spreads, only a few motes are used to maintain the roadmap. An interesting observation is that our network behaves similarly when motes fail or danger reaches them.

6.4 Minimum Average Length of Navigation Paths

Using the minimum average length as the metric, we first evaluate the global oscillation of the navigation path. The global oscillation denotes the probability of generating reachable paths for all users with the least oscillations.

Let \bar{l}_{AVG} be the minimum average length of all paths from the user nodes to the exit nodes, and \bar{l}_{OPT} be the minimum average length of the optimal path. l_{ue}^{OPEN} indicates a path length of OPEN. Num_u denotes the number of user nodes. The performance ratio is defined as $\frac{\bar{l}_{AVG}}{\bar{l}_{OPT}}$. \bar{l}_{AVG} and \bar{l}_{OPT} are computed by Equations (10)-(11)

$$\bar{l}_{AVG} = \min \left(\frac{\sum l_{ue}^{OPEN}}{Num_u} \right) \quad (10)$$

$$\bar{l}_{OPT} = \frac{\sum l_{ue}^{OPT}}{Num_u}. \quad (11)$$

We inject 20 and 100 user nodes into the network of sizes 400 and 1,600, respectively. Fig. 17 shows the performance ratio of the three approaches under different network sizes and different numbers of users. PF keeps the ratio above

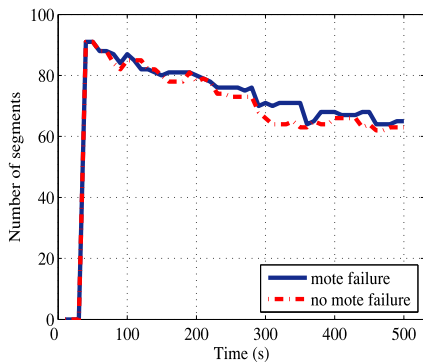


Fig. 16. Robustness.

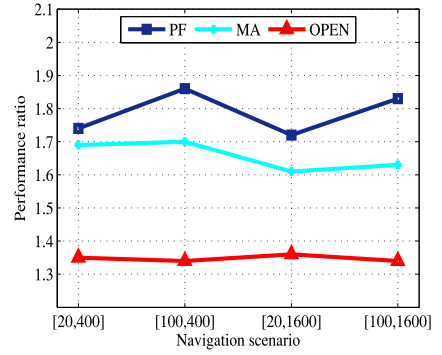


Fig. 17. On global oscillation.

1.7, MA keeps the ratio around 1.6, while OPEN achieves the ratio lower than 1.25. This result demonstrates the superior navigation efficiency using OPEN. When the dangerous areas change, OPEN can predict the motion tendency and estimate the reachability for the next node on the path. Moreover, the local oscillations are avoided.

6.5 Local Reachability

We evaluate the local reachability of the path, measured by the minimum distance to the danger. Let d be the minimum distance from the node on the path to the dangerous region, and d_{OPT} be the maximum minimum distance to the dangerous region from the optimal path. The performance ratio is defined as $\frac{d}{d_{OPT}}$. The larger ratio indicates less oscillation of the path, namely a better chance for the guided user to safely bypass the dangerous regions.

Fig. 18 shows that the performance ratio is not affected by the network size. We can see that the MA approach achieves the optimal result with the ratio of 1. Our approach and PF approach have performance ratios above 0.95 and 0.70, respectively. The performance of OPEN is 5 percent lower than that of MA with respect to the minimum safe distance, which does not indicate a survival threat. Indeed, the navigation decisions made by MA are often over-conservative, which tend to miss some potential survival opportunities. In comparison, OPEN greatly reduces the stay time of users in dangerous regions, enhancing the overall safety of the guided users as well.

6.6 Minimum Exposure Path

The minimum exposure is introduced to calculate the connectivity of wireless ad-hoc sensor networks in [35]. In this

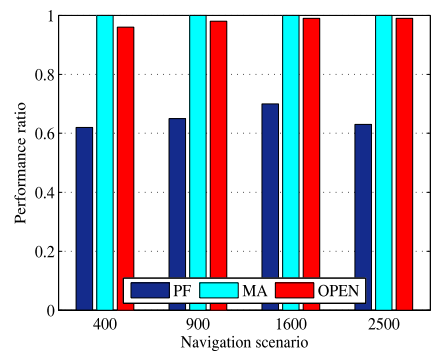


Fig. 18. On local oscillation.

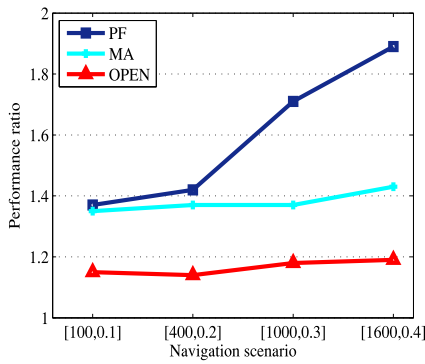


Fig. 19. On minimum exposure.

simulation, we use it to quantify the oscillation probability caused by unreliable network links. The exposure value of every point along the guiding path is calculated by $\frac{1}{h_d^2}$, which is also an indicator of the user safety.

S denotes the exposure value along the planned path, S_{OPT} denotes the optimal path of each approach. Let the performance ratio be $\frac{S}{S_{OPT}}$. The lower performance ratio means higher reachability of the path. The optimal exposure is calculated by BFS (Breadth-First Search). As shown in Fig. 19, [100, 0.1] indicates the network includes 100 nodes, amongst which there are 10 percent user nodes and 10 percent precarious nodes. We can see the performance ratio of OPEN is below 1.21, which is far less than the average values of PF and MA. PF uses the hop counts from the user node to the exit and to the danger as the metric directly, while MA uses the mid-perpendicular between two dangerous nodes as the metric based on Vironoi triangulation. The two approaches may increase the exposure value due to ignorance of the user's current location.

6.7 Summary

Here is a brief summary of the performance evaluation results. Through extensive experiments and simulations, we evaluate the efficacy, efficiency, and scalability of the navigation service offered by OPEN. Our experiments are based on the real-world data traces collected from a deployed WSN system. The simulations we conduct further evaluate the performance of OPEN under different settings of networks and parameters. The results are satisfactory, demonstrating the reliability and consistency of OPEN's navigation service.

7 CONCLUSION

Safety is always the first-place metric of emergency navigation with WSNs. When facing a dynamic environment with changing hazards, it becomes even more challenging to ensure the user's safety. This work for the first time studies the predictable reachability of navigation in the dynamic environment. We propose a reachability-based metric called ENO, upon which a practical navigation approach, OPEN, is designed. Our approach efficiently predicts the emergency dynamics in the navigation context and makes reliable and safe decisions to guide users to the exit. It minimizes the probability of oscillations of navigated users and thus enhances the reachability of navigation. The implementation and experimental results demonstrate the

advantages of our approach. In the future, we plan to take into account the sociological and psychological factors of moving crowds and the capacity constraints of roads into emergency navigation.

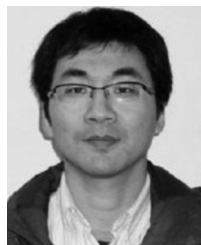
ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their insightful comments. They also thank their external members of the GreenOrbs Group for their suggestions and comments. This work is supported in part by the NSFC under grants 61303233, 61170213, 61272466, 61202359, the National Science Fund for Excellent Young Scientist under grants 61422207, and the Natural Science Foundation of Hebei Province under grant F2014203062, and the Natural Science Research Programs of Hebei Educational Committee for University Young Teachers under grant QN20131058. Lin Wang gratefully acknowledges financial support from China Scholarship Council. He and W. Liu are joint-corresponding authors.

REFERENCES

- [1] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, "Does wireless sensor network scale? A measurement study on greenorbs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 10, pp. 1983–1993, Oct. 2013.
- [2] Z. Li, W. Chen, C. Li, M. Li, X.-Y. Li, and Y. Liu, "Flight: Clock calibration using fluorescent lighting," in *Proc. ACM 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 329–340.
- [3] Z. Yang, L. Jian, C. Wu, and Y. Liu, "Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization," *ACM Trans. Sen. Netw.*, vol. 9, no. 2, p. 26, 2013.
- [4] Z. Yang and Y. Liu, "Understanding node localizability of wireless ad hoc and sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 8, pp. 1249–1260, Aug. 2012.
- [5] Z. Yang, Y. Liu, and M. Li, "Beyond trilateration: On the localizability of wireless ad-hoc networks," in *Proc. INFOCOM IEEE*, 2009, pp. 2392–2400.
- [6] Q. Ren, L. Guo, J. Zhu, M. Ren, and J. Zhu, "Distributed aggregation algorithms for mobile sensor networks with group mobility model," *Tsinghua Sci. Technol.*, vol. 17, no. 5, pp. 512–520, 2012.
- [7] Z. Li, J. Wang, and Z. Cao, "Ubiquitous data collection for mobile users in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2011, pp. 2246–2254.
- [8] G. Xing, M. Li, T. Wang, W. Jia, and J. Huang, "Efficient rendezvous algorithms for mobility-enabled wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 1, pp. 47–60, Jan. 2012.
- [9] X. Mao, S. Tang, X.-Y. Li, and M. Gu, "Mens: Multi-user emergency navigation system using wireless sensor networks," *Ad Hoc Sens. Wireless Netw.*, vol. 12, no. 1–2, pp. 23–53, 2011.
- [10] Q. Li, M. De Rosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," in *Proc. ACM 9th Annu. Int. Conf. Mobile Comput. Netw.*, 2003, pp. 313–325.
- [11] C. Buragohain, D. Agrawal, and S. Suri, "Distributed navigation algorithms for sensor networks," in *Proc. 25th IEEE Int. Conf. Commun.*, 2006, pp. 1–10.
- [12] 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, Jul. 2013.
- [13] D. Chen, C. K. Mohan, K. G. Mehrotra, and P. K. Varshney, "Distributed in-network path planning for sensor network navigation in dynamic hazardous environments," *Wireless Commun. Mobile Comput.*, vol. 12, no. 8, pp. 739–754, 2012.
- [14] T. Fraichard and H. Asama, "Inevitable collision states. a step towards safer robots?" in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 388–393.
- [15] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer, "Distributed search and rescue with robot and sensor teams," in *Proc. 4th Int. Conf. Field Serv. Robot.*, 2006, pp. 529–538.

- [16] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic collision state checker for crowded environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 1492–1498.
- [17] S. Bouraine, T. Fraichard, and H. Salhi, "Provably safe navigation for mobile robots with limited field-of-views in dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 267–283, 2011.
- [18] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Proc. IEEE/RSJ Int. Workshop Intell. Robots Syst. Intell. Mech. Syst.*, 1991, pp. 1442–1447.
- [19] S. Bhattacharya, N. Atay, G. Alankus, C. Lu, O. Bayazit, and G. Roman, "Roadmap query for sensor network assisted navigation in dynamic environments," in *Proc. IEEE Int. Conf. Distributed Comput. Sens. Syst.*, 2006, pp. 17–36.
- [20] G. Alankus, N. Atay, C. Lu, and O. Bayazit, "Adaptive embedded roadmaps for sensor networks," in *Proc. IEEE Int. Conf. Robotics Autom.*, 2007, pp. 3645–3652.
- [21] Y. Xu, J. Choi, and S. Oh, "Mobile sensor network navigation using gaussian processes with truncated observations," *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1118–1131, 2011.
- [22] Y. Tseng, M. Pan, and Y. Tsai, "Wireless sensor networks for emergency navigation," *IEEE Comput.*, vol. 39, no. 7, pp. 55–62, Jul. 2006.
- [23] A. Filippopolitis and E. Gelenbe, "A distributed decision support system for building evacuation," in *Proc. 2nd Conf. Human Syst. Inter.*, 2009, pp. 323–330.
- [24] N. Dimakis, A. Filippopolitis, and E. Gelenbe, "Distributed building evacuation simulator for smart emergency management," *Comput. J.*, vol. 53, no. 9, pp. 1384–1400, 2010.
- [25] A. Filippopolitis, G. Gorbil, and E. Gelenbe, "Spatial computers for emergency support," *Comput. J.*, vol. 56, no. 12, pp. 1399–1416, 2013.
- [26] A. Desmet and E. Gelenbe, "Reactive and proactive congestion management for emergency building evacuation," in *Proc. IEEE 38th Conf. Local Comput. Netw.*, 2013, pp. 727–730.
- [27] M. Barnes, H. Leather, and D. Arvind, "Emergency evacuation using wireless sensor networks," in *Proc. IEEE 32nd Conf. Local Comput. Netw.*, 2007, pp. 851–857.
- [28] S. Li, A. Zhan, X. Wu, and G. Chen, "Ern: Emergence rescue navigation with wireless sensor networks," in *Proc. IEEE 15th Int. Conf. Parallel Distrib. Syst.*, 2009, pp. 361–368.
- [29] H. Bi and E. Gelenbe, "Routing diverse evacuees with cognitive packets," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2014, pp. 291–296.
- [30] X. Mao, X. Miao, Y. He, X.-Y. Li, and Y. Liu, "Citysee: Urban co2 monitoring with sensors," *Proc. IEEE INFOCOM*, 2012, pp. 1611–1619.
- [31] L. Wang, Y. He, Y. Liu, W. Liu, J. Wang, and N. Jing, "It is not just a matter of time: Oscillation-free emergency navigation with sensor networks," in *Proc. IEEE 33rd Real-Time Syst. Symp.*, 2012, pp. 339–348.
- [32] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. 7th ACM Conf. Embedded Netw. Sens. Syst.*, 2009.
- [33] D. Davoudani, "Specknets: A case study for artificial immune systems," Ph.D. dissertation, School of Computing, Edinburgh Napier University, Edinburgh, 2012.
- [34] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *Proc. ACM 1st Int. Conf. Embedded Netw. Sens. Syst.*, 2003, pp. 126–137.
- [35] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *Proc. ACM 7th Annu. Int. Conf. Mobile Comput. Netw.*, 2001, pp. 139–150.



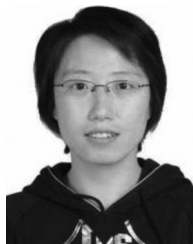
Lin Wang is an Associate Professor in the School of Information Science and Engineering at Yanshan University. He received the B.Ed. degree from Northwest Normal University and ME degree in Computer Engineering from Yanshan University. He is currently a PhD candidate in Computer Science at Yanshan University. He has worked as a visiting student at GreenOrbs Group since 2010, and as a research assistant. He is currently with Wireless Networking Lab of Illinois Institute of Technology. His research interests include sensor and wireless networks, mobile computing. He is a student member of the IEEE and ACM.



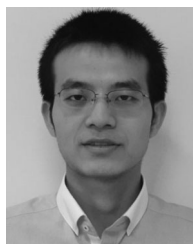
Yuan He received the BE degree from the University of Science and Technology of China, the ME degree from the Institute of Software, Chinese Academy of Sciences, and the PhD degree from the Hong Kong University of Science and Technology. He is an assistant professor in the School of Software and TNLIST at Tsinghua University. His research interests include sensor networks, peer-to-peer computing, and pervasive computing. He is a member of the IEEE and ACM.



Wen Yuan Liu received the BS and MS degrees in computer science at Northeast Heavy Machinery Institute, China, and the PhD degree in computer science at the Harbin Institute of Technology in 2000. Since 1996, he has been with the School of Information Science and Engineering at Yanshan University, Qinhuangdao City, China, where he is currently a professor. He is also a special invited researcher at IoT of TNLIST, Tsinghua University. His research interests include wireless sensor networks and mobile networks. He is a member of the IEEE and ACM.



Nan Jing received the BS and MS degrees in Circuit and System from Yanshan University, Qinhuangdao, in 2001 and 2004, respectively. Since 2013, she has been an Associate Professor with the the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. Her research interests include sparse channel estimation for MIMO OFDM system, sparse signal recovery and its applications in communications, wireless sensor networks and mobile networks.



Jiliang Wang is currently with School of Software and TNLIST, Tsinghua University, P.R. China. His research interests include sensor and wireless networks, network measurement, and pervasive computing. He received his BE degree in computer science and technology from the University of Science and Technology of China and his PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, in 2007 and 2011, respectively. He is a member of the IEEE.



Yunhao Liu received the BS degree from the Automation Department, Tsinghua University in 1995, and the MS and PhD degrees in computer science and engineering from Michigan State University in 2003 and 2004, respectively. He is now a Chang Jiang chair professor and a dean of the School of Software at Tsinghua University, China. He is the ACM distinguished speaker, and now serves as the chair of ACM China Council and as the associate editors-in-chief for the *IEEE Transactions on Parallel and Distributed Systems*, and the associate editor for the *IEEE/ACM Transactions on Networking* and *ACM Transactions on Sensor Network*. He is a fellow of the IEEE.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.