It Is Not Just A Matter of Time: Oscillation-Free Emergency Navigation with Sensor Networks

Lin Wang[*§], Yuan He[†], Yunhao Liu[†‡],Wenyuan Liu[*§], Jiliang Wang[†], Nan Jing[*§]

[*]School of Information Science and Engineering, Yanshan University
[†]School of Software, TNLIST, Tsinghua University
[‡]CSE, Hong Kong University of Science and Technology
[§]The Key Laboratory for Computer Virtual Technology and System Integration of HeBei Province
Email:{wlin, wyliu, jingnan}@ysu.edu.cn, {he, yunhao, wang}@greenorbs.com

Abstract—Emergency navigation is an emerging application of wireless sensor networks with significant research and social values. In order to ensure the safety and timeliness of navigation for the users, most of the existing works model navigation as a path-planning 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 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 test-bed experiments and extensive simulations. The results demonstrate that OPEN outperforms the current state-of-the-arts approaches with respect to user safety and navigation efficiency.

Keywords-Reachability; Oscillation; Emergency Navigation; Sensor Networks

I. INTRODUCTION

Wireless sensor networks (WSNs), born with the ability of automatic monitoring and interaction with the physical world under various environmental dynamics, are receiving increased attention in recent years [1]–[3]. Navigation is an emerging application of WSNs, in which sensor nodes collaboratively explore the dynamic environmental conditions and people's movements [4]–[6], and then prevent people in dangers from once again traversing into the dangerous area, such as geologic hazard, fire rescue, oil spill control, etc. A WSN system for forest monitoring, GreenOrbs [2], [3] was deployed in the TianMu Mountains of China. Regarding the potential disasters such as wildfires and landslides, navigation service is very important in ensuring visitor safety. Thus GreenOrbs was deployed in the mountain area, where sensor nodes monitor the environment and offer navigation service to the users when necessary. Following the instructions provided by the WSN, those in danger can move along a path to safely reach their desired destination. 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 on low-cost sensor nodes and the ad-hoc deployments 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 [7]–[10] 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, a 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. The users are therefore made to move back and forth in a local area, called oscillation.

– It is not just a matter of time!

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 in danger for a longer period of time and amplification of when the emergency is a 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 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.

Figure 1 shows an illustrative example of oscillation, which helps to understand the reachability of navigation. In our scenarios, three red 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, the 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 have spread and blocked all escape routes. We hope to predict the moving direction of the emergency sites and generate the 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 environments 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 the 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 abovementioned 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.

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. Using ENO, processing of the emergency dynamics in navigation is changed from passive reaction to proactive judgment. User oscillations can thus be avoided whenever possible.

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 II discusses the related work. The design of ENO and OPEN is introduced in Section III. Section IV presents theoretical analysis, proofs, and discussions on several important issues, followed by the performance evaluation in Section V. We conclude this work in Section VI.

II. RELATED WORK

A. Mobile Robot Navigation

Navigation using sensors is inspired by navigating autonomous robots with sensors in the field of Robotics [11]–[16]. The earliest relevant works in this direction formulate the motion safety to a collision avoidance problem, wherein dangerous objects traveling at a constant linear velocity must be avoided. For a robot operating in a planar environment with arbitrarily moving objects, collision-free motion is guaranteed if the maximum velocity of the robot is a multiple of the
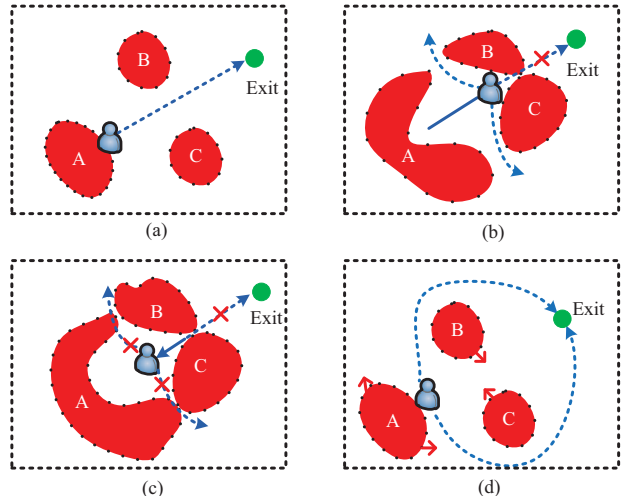

Fig. 1. Scenarios of emergency navigation using sensor networks. (a) indicates the generated path from a user to an exit. The user moves to the exit while the dangerous region "B","C" spread towards the path as shown as (b). Moreover, the user attempts to turn back to find an alternative path. (c) shows 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.

maximum velocity of the dangerous objects [11]. However, the proposal in [11] relies on some unrealistic assumptions, i.e., distributed algorithms are proposed for guaranteeing the collision avoidance [13], [14]. Nevertheless, this guarantee does not hold due to uncontrolled moving dangerous objects. To address general motion safety issues, the Inevitable Collision States (ICS) concept is proposed in [12]. ICS needs to reason the global information, which is usually unavailable in real-world applications. Nevertheless, ICS is a very complex model. The probabilistic versions of ICS [15] are studied so as to better capture the uncertainty that prevails in practical scenarios, but the probabilistic model does not meet the safety requirement of human navigation. Our work is inspired by a version of ICS corresponding to passive motion safety in [16]. Unfortunately, the 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.

B. Sensor Network Navigation

From the perspective of dealing with dynamic emergency, the existing navigation approaches of WSNs are classified into two categories: passive navigation [7]–[10], [17], [18] and proactive safety navigation [5], [19], [20].

Recent works in this area, e.g. [7], adopt the potential fields and the hop count as metrics to calculate the optimal navigation paths. The exit generates an attractive potential, pulling sensor nodes to the exit. At the same time, each obstacle (or emergency spot) generates a repulsive potential, pushing sensor nodes away from it. Each node calculates its potential value and tries to find a navigation path with the least total potential value. The authors propose an indoor navigation algorithm using temporally ordered routing with global flooding in [18]. 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 [8]. Similar to a roadmap, a skeleton graph is a sparse subset of

the original network. Sharing information across the skeleton graph of a network saves the communication overhead of navigation. To avoid the side-effect of inaccurate positioning, the authors in [9] propose to navigate people along the medial axis of the safe field. The approach in [10] ensures that every user maintains at least a usable path. The partial reversal method used in [10], 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.

Multi-user navigation with mobile sensor networks is developed in [20], which models the navigation problem as a Gaussian processes with truncated observations. To find the optimal locations that best predict the emergency dynamics and network conditions, one needs a spatial-temporal model of those two factors. 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.

III. 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.

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.

This section first introduces the model and definitions, followed by an overview of the workflow. Subsection C presents the details of the ENO metric. Subsection D introduces the process of path generation and navigation.

A. 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 seven-tuple: $< ID, NR, DS, h_d, h_e, h_u, Q_{ue} >$. ID is the unique node identifier, which is assigned when the sensor network is deployed. NR is the node role, which indicates the function of a node in navigation. There are two types of node roles: *dangerous node* and *general node*, which indicate a node inside or outside of the dangerous region. The emergency
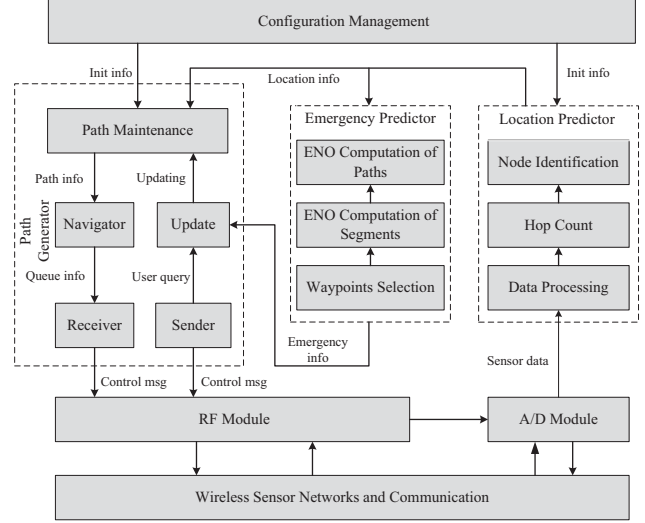

Fig. 2. Overview of the proposed navigation architecture

is regarded as the existence of a set of dangerous 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 dangerous nodes and their 1-hop neighbor nodes. DS denotes the set of neighbor dangerous nodes of a general node. h_d, h_e and h_u respectively denote the hop counts from a general node to its nearest dangerous node, exit, and user. The term "nearest" indicates the least hop-count distance between two nodes. Q_{ue} denotes the vertex sequence on the navigation path, on which the user node v_u is the header and the exit node v_d is the rear.

Our navigation protocol searches for a series of waypoints which correspond to sensor nodes deployed in the navigation scenario. Q_{ue} is constructed by joining multiple q_{mn}, while each q_{mn} 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 is mapped to the movement of a node towards the segment between two adjacent waypoints. For a segment, the speed of the dangerous 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 dangerous 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 Subsection C.

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.

B. Design Overview

Figure 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 to the WSNs which is deployed in advance. Firstly, each node uses the location predictor to evaluate the relative distance between them, the exit, and the danger. Secondly, 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 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.

1) Initialization: In the initialization phase, h_d, h_e and h_u are first set to 0 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 dangerous node and floods a probe MSG_d. Once MSG_d is received, its 1-hop neighbors accordingly change their roles to dangerous node and forward the message. After other nodes receive the massage, they change the hop count between them and the dangerous 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.

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_d is set to their shared dangerous node which is a convex point towards the segment

---

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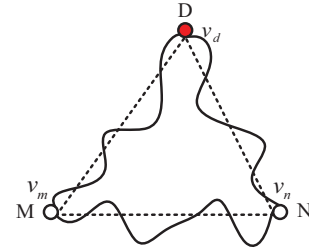19:   **end if**
20: **end while**

Fig. 3. Two adjacent waypoints with their shared dangerous node

$\overline{v_m v_n}$. We assume that h_{md} and h_{nd} 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 $\overline{v_m v_n}$.

For predicting the moving tendency of emergency, getting the reasonable k_{mn} 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. $\triangle DMN$ in Figure 3). The distance from v_n to v_m, h_{mn}, is bounded by

$$\sqrt{|h_{md}^2 - h_{nd}^2|} < h_{mn} \leq \sqrt{h_{md}^2 + h_{nd}^2} \tag{1}$$

Next, node v_n is pushed into the waypoint set V'. During the next iteration, v_n is viewed as 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.

Note that node density and waypoint location need to be considered when constructing the non-obtuse triangles. Here, we use the number of stationary nodes in the minimum non-obtuse triangle to define the node density approximately. In a
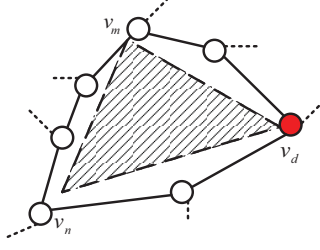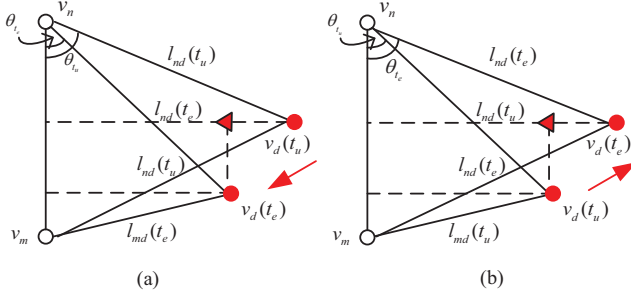
Fig. 4. Node density analysis

(a)   (b)

Fig. 5. Moving tendency of dangerous node v_d with a segment. (a) v_d moves close to a segment; (b) v_d shows moves away from a segment

minimum non-obtuse triangle (see Fig. 4), let $h_{md} = h_{nd} = 2$, we may have $k_{mn} = max(h_{mn}) \approx 3$ according to Formula 1. In this case, a user and an exit are located at the node v_n and v_m, respectively. Therefore, we need at least 7 stationary, infrastructure nodes, including 2 waypoints and 1 dangerous node to construct the non-obtuse triangle. Furthermore, our approach is independent from the number of user nodes (or mobile nodes). The user node only communicates with the neighbor stationary nodes, which transmit the messages to the exit and check their distances to the danger and the exit.

C. 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.

1) Calculate the movement velocity of dangerous region: 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 dangerous node's movement. The dynamics of a dangerous region is mapped to the movement of a dangerous 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 dangerous node. Suppose $l_{mn}(t) = k_{mn} \times \varpi$, ϖ denotes the average hop distance. In a real network, the triangle's edge may consist of many non-collinear vertexes. Using the hop distance instead of real geometric distance does not generate gross inaccuracies [8].

The principle of calculating the movement velocity is shown in Figure 5. θ_t denotes the angle between the segment $\overline{v_m...v_n}$ and $\overline{v_m...v_d}$. 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.

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

$l_{md}(t_{u/e})$ means the distance between v_m and v_d 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_d^{mn} of the mobile dangerous node v_d with the segment can be calculated as a projection of v_d on the vertical line $\overline{v_m...v_n}$ as follows:

$$l_d^{mn} = l_{md}(t_u) \times \cos\theta_{t_u} - l_{md}(t_e) \times \cos\theta_{t_e} \quad (4)$$

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

$$s_d^{mn}(t) = \frac{l_d^{mn}}{\triangle t} \quad (5)$$

When the dangerous node is approaching the segment (see Figure 5a), the velocity is a positive value. Otherwise, it is a negative value (see Figure 5b).

2) Expected Number of Oscillation: Now we formally introduce the metric Expected Number of Oscillation (ENO) which measures the number of possible reentrant 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

$$e_d^{mn}(t) = s_d^{mn}(t) \times \sum_{i=m\in q_{mn}(t)}^{n} \sum_{v_d\in DS_i} h_{id}^{-2}(t) \quad (6)$$

where $\sum_{i=m\in q_{mn}(t)}^{n}\sum_{v_d\in DS_i} h_{id}^{-2}(t)$ and $s_d^{mn}(t)$ are the spatial and temporal accumulated dangerous value during the time interval t. $q_{mn}(t)$ is generated by forwarding MSG_u and is stored in every general node between two waypoints.

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

$$E_{ue}(T) = \sum_{m,n\in Q_{ue}(T)}^{e} e_d^{mn}(t) \quad (7)$$

$Q_{ue}(T)$ is dynamically updated by forwarding MSG_e and stored in every waypoint between a user node and an exit node.

D. Path Generation and User Navigation

Upon receiving MSG_e sent by its nearest exit, a waypoint computes the moving velocity of the closest dangerous node value, i.e. $S_d^{mn}(t)$, 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.
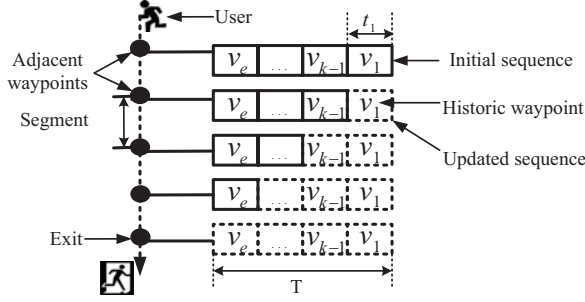
Fig. 6. The process of sequence mismatch and direction updating

Algorithm 2 Computing ENO of Path Algorithm

1: **while** Receiving a flooding message MSG_e from the exit node v_e do **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 $S_d^{mn}(t)$ of any two waypoints using Equations 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:       Forwording 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 path is significantly changed (see Figure 6). 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.

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.

IV. ANALYSIS AND DISCUSSIONS

For the trapped users, it is 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.

Algorithm 3 Navigation and Path Updating Algorithm

1: **while** Receiving $Q_{ue}(T)$ do **do**
2:    Selecting a $Q_{ue}(T)$ with the minimum $E_{ue}(T)$ to escape;
3:    **for** $n = n : 1$ /*$n = |Q_{ue}(T)|$ */ **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. Correctness

The waypoints selection is the essential link of the whole design. On one hand, a greater distance between the two adjacent waypoints lowers the computational accuracy of the velocity of the dangerous moving. On the other hand, two adjacent endpoint distances are smaller which means larger global computational overhead. Formula 1 is seen as a trade-off between efficiency and effectiveness.

Theorem 1. *Equation 1 holds in a non-obtuse triangle.*

Proof: A segment between two adjacent waypoints, together with the two lines connecting the waypoints and a dangerous node, form a non-obtuse triangle. If there is an obtuse angle in the triangle's interior angles, the dangerous node would be seen as a direct convex vertex towards the next segment in another triangle rather than in this triangle. In Figure 3, an interior angle of $\triangle DMN$ satisfies $\cos M = (h_{md}^2 + h_{mn}^2 - h_{nd}^2)/2h_{md}h_{mn}$ by the law of cosines. The interior angle M is a non-obtuse angle if and only if $h_{md}^2 + h_{mn}^2 - h_{nd}^2 \geq 0$, so $h_{mn}^2 \geq h_{nd}^2 - h_{md}^2$. Similarly, we can get $h_{mn}^2 \leq h_{nd}^2 + h_{md}^2$ for the interior angle D. And h_{md}, h_{nd} and h_{mn} are positive integers. After getting the square roots, Equation (1) is proved.

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 2. *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.

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.

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.

Fig. 7. Checking the propose approach on a testbed of 21 TelosB motes
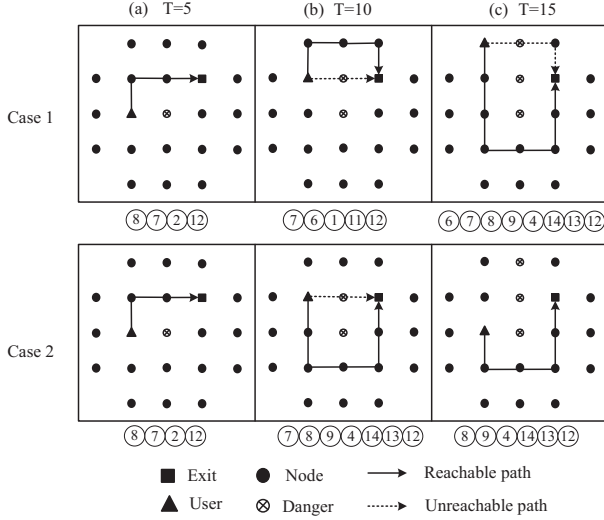

Fig. 8. Two cases with dynamic danger on the testbed

B. Discussions

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

1) Delay and cost: It is well known that periodic flooding may be a costly solution for data communication due to the high cost in network bandwidth and energy. We explain the reasons why we adopt periodic flooding to deal with node identification and path sequence updates.

First, the emergency navigation is a real-time service but it is an event-driven one. Therefore, emergency flooding is rare and transient in a long-term WSN system.

Second, the dynamic topology caused by the dynamic environments reduces the data delivery ratio and increases the latency to make a decision. Flooding is seen as such a scheme without topology information.

Last but not least, the fast reliable flooding protocol (e.g., Glossy [21]) is developed in the state of the art. A node receives the flooding packet with a probability higher than 99.99%, while having its radio turned on for only a few milliseconds during a flood. Taking advantage of the novel flooding scheme, the delay and cost may be no longer a problem in our proposed navigation process.

2) Distance measurement: We consider scenarios of large-scale deployments, where sensor capability is usually constrained to only communication and physical sensing, and might not include hardware supporting distance measurements.

In [22], the authors analyze the performance of Euclidean distance and hop distances proportionality approximation for uniform i.i.d deployment and the randomized grid deployment. Li and Liu [23] consider the ranger-free localization in isotropic and anisotropic networks. By constructing the virtual holes, the average distance estimation error is 3.2% and 3.7% in perturbed grid deployments and random deployments. Let the average distance error ratio be $\xi = \frac{k_{mn}\varpi}{g_{mn}}$ between hop distance l_{mn} and geometry distance g_{mn} for two nodes (e.g., v_m and v_n). We have an approximation to

$$g_{mn} = \frac{\varpi}{\xi}k_{mn} \qquad (8)$$

By combining Equation (8) and the Cosine Formula, the Cosine of the intersection angle q by means of geometry distance is calculated by

$$\cos\theta' = \frac{(\varpi^2/\xi^2)(k_{nd}^2 + k_{mn}^2 - k_{md}^2)}{2(\varpi k_{nd}/\xi)(\varpi k_{mn}/\xi)} = \frac{(k_{nd}^2 + k_{mn}^2 - k_{md}^2)}{2k_{nd}k_{mn}} \qquad (9)$$

From (2)-(3) and (9) we have $\theta' = \theta$. Therefore, using hop distance is acceptable for predicting the dynamics of emergency. Furthermore, we adopt the strategy based on neighborhood sensing to reduce the impact of inaccurate hop counting. The neighborhood sensing and hop-counting functions are piggybacked on the existing probe mechanism, which is transparent to the user and does not incur much additional overhead.

3) Network model: There are many different network models, depending on the specific assumptions and applications. We use the truncated observations graph, which includes the relative distance between any two nodes in a probabilistic connectivity graph. In this model, the idea of spatial-temporal community is embodied into it similarly with that proposed in [24]. The authors in [24] analyze and evaluate the performance using the Time-Varying Graphs (TVG) to measure the dynamic networks. Intuitively, TVG may be an alternative model to describe the dynamics of the emergency environments in the target scenario of this work, which is more suitable to measure the fine-grain dynamics of interactions.

V. EVALUATIONS

We evaluate the ENO-based oscillation-free navigation approach, named OPEN. This section presents the performance results in both experiments on real hardware and extensive simulations.

A. Experiments on Real Hardware

Through the experiments, we evaluate OPEN's efficiency in predicting the tendency of the emergency and the resiliency against increasing trapped users. We deploy 21 TelosB motes on our office floor to form a grid topology as shown in Figure 7. Their IDs are marked in the top-right red box of the motes. In Figure 7, mote-8 is configured to be a user node. Mote-12 is configured as an exit node. An external mote connected to a host server is configured as a sniffer to the network. After all motes receive the path queries, the navigation computation is triggered on each mote.
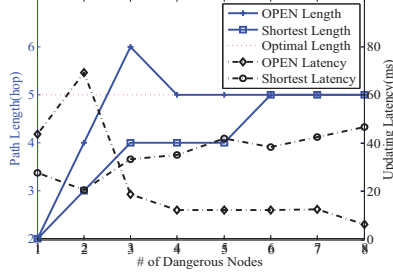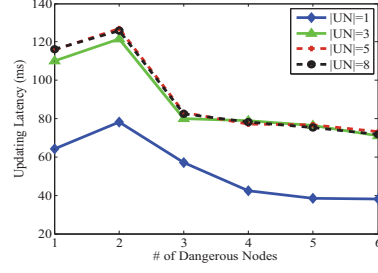
Fig. 9. The path length and updating latency

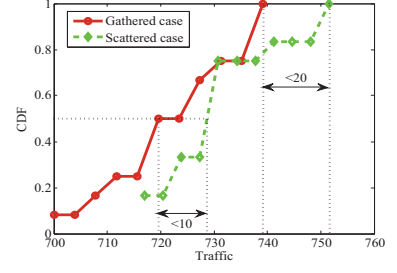Fig. 10. Average latency with multiple users

Fig. 11. CDF of communication traffic

1) Path Length and Latency: We implement an approach that finds the shortest path for navigation, shown by Case 1 in Figure 8. At the same time, we implement OPEN and the result is illustrated as Case 2. Periodically, some motes turn to sleep, which act as dangerous nodes. We assign two motes to be the user and the exit, respectively. The central node (a circle) is a dangerous node at the beginning, and the dangerous region is updated every 5 seconds. At the 10th second and the 15th second, the numbers of dangerous nodes increase to 2 and 3, respectively. We hope that our protocol can find the reachable path represented by the black solid line of Case 2c in Figure 8.

As shown in Figure 9, the left and right Y-axes denote the path length (hop count) and the response latency (total time waited for a navigation decision), respectively. When we set the number of dangerous node at 1, the path length of both the OPEN approach and the shortest path approach is 2 hops. However, the ENO-based approach incurs response latency of 48.2 milliseconds, while the shortest path needs only 26.3 milliseconds.

When the number of dangerous nodes increases, the latency of OPEN apparently decreases. When the number of dangerous nodes increases to 4, the response latency is around 12.3. At the same time, the latency of the shortest path approach keeps increasing. When the number of dangerous nodes exceeds 3, our protocol selects the reachable path (5 hops) and the path no longer needs to be recalculated, owing to the ability to predict the tendency of danger.

2) Efficiency with Multiple Users: Given the broadcast mechanism for information propagation, all the motes in the system may be aware of the waypoints and obtain the path ENO to make navigation decisions. Our approach thus has the potential to support applications of multiple concurrent users. The user number is set at 1, 3, 5 and 8 with the moving danger by controlling the number of sleeping motes.

As shown in Figure 10, the response latency of a navigation path for each user linearly increases with the proliferation of dangerous nodes, which varies from 1 to 2. Although the calculation overhead increases considerably in the navigation initialization stage, it declines rapidly when the danger spreads towards a certain direction. The result indicates that OPEN may predict the dynamics of danger in a timely fashion, which helps to reduce the response latency time and the overhead of path recalculation.

3) Communication Traffic: Communication traffic is a key indicator to evaluate the protocol performance. To evaluate the impact of user location on the traffic, we compare the communication cost caused by two types of user distributions: gathered and scattered distribution. We use the sniffer to capture the communication traffic, measured by the number of received packets on all 21 nodes. For the gathered distribution, we select 6 adjacent nodes located on a line to be the user nodes. For the scattered distribution, we select 6 disconnected nodes to be the user nodes. The cumulative distribution function (CDF) of the communication traffic in same monitoring period is shown in Figure 11. We can see that the scatter distribution generates more traffic. On one hand, the disconnected user nodes cause more retransmissions among the nodes. On the other hand, the gathered case incurs bursty queries from a local area where the users are located. Much communication traffic might be congested due to the bursty occurrences of queries. As shown as Figure 11, the maximum deference is less than 20 packets and the medium deference is less than 10 packets. Therefore, the traffic difference caused by the user distribution may be tolerated.

These experiments demonstrate that the communication cost of our approach is acceptable and can be implemented on real resource-constrained sensor motes.

B. Simulation

In order to evaluate the scalability and reachability of OPEN, we carry out extensive simulations to compare OPEN with two state-of-the-arts approaches, namely the potential field based approach (PF) [7] and the medial axis based approach (MA) [9]. For this purpose, we compare the efficiency and navigation safety of each approach. We are also interested in the approaches' robustness and behavior when the physical obstacles exist in the environment.

We tune the following parameters to evaluate the performance of OPEN: the network size, the node role, and the movement speed of dangerous nodes. We compare OPEN to PF and MA from five perspectives, namely average reachability, robustness, minimum average length, minimum dangerous distance, and minimum exposure path. The last two metrics are actually two indicators of user safety.

1) Path Reachability: We assume that the emergency exhibits dynamics in two patterns: shift and spread. The danger refreshing period is set to 10s, 30s and 60s. These experiments are repeated 20 times.

The purpose of this group of simulations is to compare OPEN with PF and MA with respect to path reachability, which is measured by the average path reachability of each approach. In order to compare the path reachability, we consider four different cases: (i) forty users reaching one exit against shifting emergency (named $[40, 1, p]$), (ii) forty
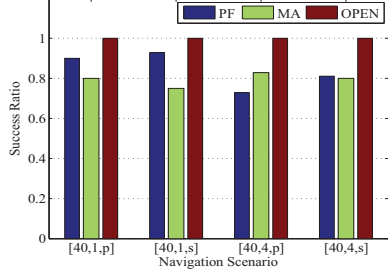
346

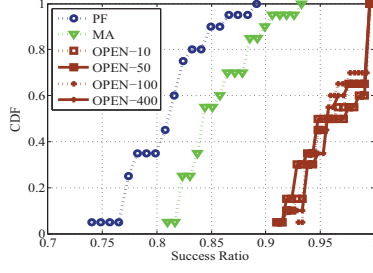Fig. 12. Average path reachability

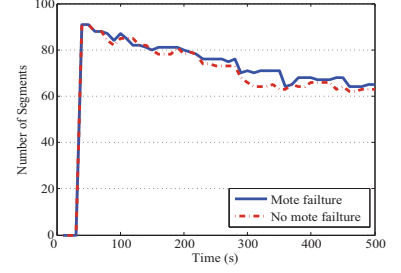Fig. 13. CDF of path reachability

Fig. 14. Robustness in the dynamic emergency
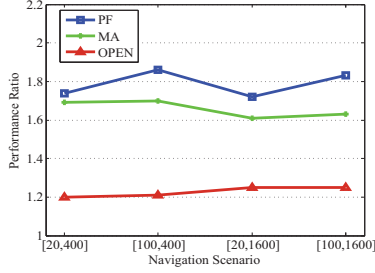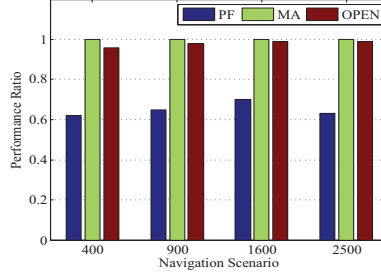

Fig. 15. Average minimum path length

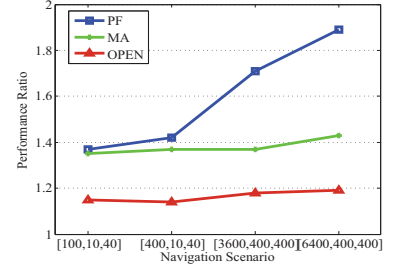Fig. 16. Minimum dangerous distance

Fig. 17. Minimum exposure path

users reaching one exit against spreading emergency (named $[40, 1, s]$), (iii) forty users reaching four exits against shifting emergency (named $[40, 4, p]$), (iv) four users reaching two exits against spreading emergency (named $[40, 4, s]$). The sensor field has 400 nodes deployed uniformly. Figure 12 shows that OPEN clearly outperforms MA and PF by achieving always 100% reachability. MA and PF fail to ensure the reachability in some scenarios, because they do not predict the tendency of emergency dynamics well. As a result, there are some dangerous nodes on the navigated path using MA or PF.

We further evaluate the scalability of OPEN at larger scales. Specifically, we carry out a group of simulations, where 1600 nodes are randomly deployed in an 40×40 area. The number of users is set to 10, 50, 100, 400. Figure 13 shows the CDF of path reachability of OPEN, MA, and PF. We can see that more than 93% users can be successfully navigated to the exit using OPEN, which is 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 when the scale of network increases from 10 to 400.

2) Robustness: We run the same emergency pattern with different preconfigured node failure ratio (10%, 20% and 30%). During the simulations, we randomly turn off some motes 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.

Figure 14 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 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.

3) Minimum Average Length of Navigation Paths: Using the minimum average length as the metric, we first evaluate the global reachability of the navigation path. The global reachability denotes the probability of generating reachable paths for all users with the least oscillations.

Let $\overline{l_{AVG}}$ be the minimum average length of all paths from the user nodes to the exit nodes, and $\overline{l_{OPT}}$ 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 $\frac{\overline{l_{AVG}}}{\overline{l_{OPT}}}$. $\overline{l_{AVG}}$ and $\overline{l_{OPT}}$ are computed by Equations (10-11).

$$\overline{l_{AVG}} = \min\left(\frac{\sum l_{ue}^{OPEN}}{Num_u}\right) \tag{10}$$

$$\overline{l_{OPT}} = \frac{\sum l_{ue}^{OPT}}{Num_u} \tag{11}$$

We inject 20 and 100 user nodes to the network of sizes 400 and 1600, respectively. Figure 15 shows the performance ratio of the three approaches under different network sizes and different numbers of users. PF keeps the ratio above 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.

4) Local Reachability: We evaluate the local reachability of navigation, 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 higher reachability of the path, namely a better chance for the guided user to safely bypass the dangerous regions.

Figure 16 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% 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.

5) Minimum Exposure Path: We vary the number of dangerous nodes from 40 to 200 to compare the exposure of the navigation path using different approaches. 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 Figure 17, $[100, 10, 40]$ indicates the network includes 100 nodes, amongst which there are 10 user nodes and 40 dangerous 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 users 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.

VI. 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. We also attempt to apply our algorithms to TDMA based networks. Potential research directions include study on the case with danger in non-convex polygon regions, the control mechanism of congested user flow, and mobile intergroup interaction and cooperation.

References

[1] M. Keally, G. Zhou, G. Xing, and J. Wu, "Exploiting sensing diversity for confident sensing in wireless sensor networks," in *Proceedings IEEE INFOCOM*, 2011.

[2] Y. Liu, Y. He *et al.*, "Does wireless sensor network scale? a measurement study on greenorbs," in *Proceedings of IEEE INFOCOM*, 2011.

[3] Z. Li, M. Li, and Y. Liu, "Towards energy-fairness in asynchronous duty-cycling sensor networks," in *Proceedings IEEE INFOCOM*, 2012.

[4] Z. Li, M. Li, J. Wang, and Z. Cao, "Ubiquitous data collection for mobile users in wireless sensor networks," in *Proceedings IEEE INFOCOM*, 2011.

[5] X. Mao, S. Tang, X.-Y. Li, and M. Gu, "Mens: Multi-user emergency navigation system using wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 12, no. 1-2, pp. 23–53, 2011.

[6] G. Xing, M. Li, T. Wang, W. Jia, and J. Huang, "Efficient rendezvous algorithms for mobility-enabled wireless sensor networks," *IEEE Transactions on Mobile Computing*, pp. 47–60, 2012.

[7] Q. Li, M. De Rosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," in *Proceedings of ACM MobiCom*, 2003.

[8] C. Buragohain, D. Agrawal, and S. Suri, "Distributed navigation algorithms for sensor networks," in *Proceedings of IEEE INFOCOM*, 2006.

[9] M. Li, Y. Liu, J. Wang, and Z. Yang, "Sensor network navigation without locations," in *Proceedings of IEEE INFOCOM*, 2009.

[10] D. Chen, C. Mohan, K. Mehrotra, and P. Varshney, "Distributed in-network path planning for sensor network navigation in dynamic hazardous environments," *appears in Wireless Communications and Mobile Computing*.

[11] V. Lumelsky and S. Tiwari, "Velocity bounds for motion planning in the presence of moving planar obstacles," in *Proceedings of the IEEE/RSJ/GI RIOS*, 1994.

[12] T. Fraichard and H. Asama, "Inevitable collision states. a step towards safer robots?" in *Proceedings of IEEE/RSJ IROS*, 2003.

[13] 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 *Springer Field and Service Robotics*, 2006, pp. 529–538.

[14] E. Lalish and K. Morgansen, "Decentralized reactive collision avoidance for multivehicle systems," in *Proceedings of IEEE CDC*, 2008.

[15] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic collision state checker for crowded environments," in *Proceedings of IEEE ICRA*, 2010.

[16] S. Bouraine, T. Fraichard, and H. Salhi, "Provably safe navigation for mobile robots with limited field-of-views in dynamic environments," *Springer Autonomous Robots*, pp. 1–17, 2011.

[17] S. Bhattacharya, N. Atay, G. Alankus, C. Lu, O. Bayazit, and G. Roman, "Roadmap query for sensor network assisted navigation in dynamic environments," *Springer Distributed Computing in Sensor Systems*, pp. 17–36, 2006.

[18] Y. Tseng, M. Pan, and Y. Tsai, "Wireless sensor networks for emergency navigation," *IEEE Computer*, pp. 55–62, 2006.

[19] G. Alankus, N. Atay, C. Lu, and O. Bayazit, "Adaptive embedded roadmaps for sensor networks," in *Proceedings of IEEE ICRA*, 2007.

[20] Y. Xu, J. Choi, and S. Oh, "Mobile sensor network navigation using gaussian processes with truncated observations," *IEEE Transactions on Robotics*, no. 99, pp. 1–14, 2011.

[21] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *Proceedings of IEEE IPSN*, 2011.

[22] S. Nath, P. V. Ekambaram, A. Kumar, and P. V. Kumar, "Theory and algorithms for hop-count-based localization with random geometric graph models of dense sensor networks," *ACM Transactions on Sensor Networks*, pp. 111–149, 2012.

[23] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," in *Proceedings of ACM MOBICOM*, 2007.

[24] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," *Ad-hoc, Mobile, and Wireless Networks*, pp. 346–359, 2011.