# Distributed Emergency Guiding with Evacuation Time Optimization Based on Wireless Sensor Networks

Lien-Wu Chen, *Member, IEEE*, Jen-Hsiang Cheng, and Yu-Chee Tseng, *Fellow, IEEE*

**Abstract**—This paper proposes a load-balancing framework for distributed emergency guiding based on wireless sensor networks. A load-balancing guiding scheme is designed and an analytical model is derived to reduce the total evacuation time of people indoors. The guiding scheme can provide the fastest path for people to reach an exit according to the evacuation time estimated using the analytical model. Based on thorough research, this is the first distributed solution in which corridor capacity and length, exit capacity, and the concurrent movement and distribution of people are considered in estimating the evacuation time and planning escape paths. Using the proposed framework, congestion in corridors and at exits can be eased to substantially reduce the total evacuation time. Analytical and simulation results show that this approach outperforms existing schemes and can prevent people from following local-optimal guiding directions that increase the evacuation time. A prototype called the *Load-balancing Emergency Guiding System (LEGS)* is implemented; this system can be used to compare the evacuation times and guiding directions provided by existing schemes and the proposed scheme for various distributions of people.

**Index Terms**—Emergency guiding, home security, load balancing, pervasive computing, wireless sensor network

## 1 Introduction

Recent progress in wireless communications and embedded microelectromechanical systems technologies has made research on wireless sensor networks (WSNs) increasingly attractive. Several WSN applications have been studied, such as vehicle security and tracking [1], cooperative collision avoidance [2], and emergency guiding and monitoring [3], [4].

In this study, load-balancing evacuation is examined to support emergency guiding for evacuating people using WSNs. Existing approaches do not facilitate complete load-balancing because multiple critical factors that affect the total evacuation time required for people indoors, such as corridor capacity and length, exit capacity, and concurrent movement and distribution of people, are not considered.

A load-balancing framework for quickly guiding people to safety during an emergency in a 2D indoor environment is proposed in this study. During normal times, the network monitors the environment in an idle mode; however, when an emergency is detected, all sensors switch to active mode to address this event. In addition, the network can adaptively modify its topology to ensure reliable transmission. This framework can rapidly identify hazardous regions (within a specified distance from emergency locations) that should be avoided and then determines the escape route that can lead people to exits in the shortest amount of time.

In this framework, each sensor is assigned an *altitude*, which can be considered as a degree of danger, as described in our previous studies [3], [4]. Sensors near exits are assigned smaller altitudes, whereas sensors near emergency locations are assigned higher altitudes, and the escape routes to exits are along sensors with higher altitudes to those with lower altitudes. Initially, each sensor is assigned an altitude according to its distance from the nearest exit. When an emergency occurs, sensors within a specified distance from the emergency locations form hazardous regions by raising their altitudes. After the above step, local-minimum sensors have to re-compute their altitudes to find ways out. The link reversal concept in the *temporally ordered routing algorithm (TORA)* [5] is used to solve this problem. Specifically, for quick convergence, variations of neighboring sensors' altitudes are used to increase a local-minimum sensor's altitude. After the algorithm converges as all sensors finds their ways out, each exit forms an emergency guiding tree as a root.

To further reduce the total evacuation time of people indoors in [3], [4], a load-balancing guiding scheme is designed and an analytical model is derived. The guiding scheme provides the fastest path for people to reach an exit based on the evacuation time estimated using the analytical model. Based on thorough research, this is the first distributed solution which corridor capacity and length, exit capacity, and concurrent movement and distribution of people are considered in estimating evacuation time and planning escape paths. By using the proposed framework, congestion in corridors and at exits can be eased to substantially reduce the total evacuation time. Furthermore, the framework can prevent people from following local-optimal guiding directions with long total evacuation times.

- L.-W. Chen is with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan. E-mail: lwuchen@fcu.edu.tw.
- J.-H. Cheng and Y.-C. Tseng are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan. E-mail: {jhcheng, yctseng}@cs.nctu.edu.tw.

*Manuscript received 28 June 2013; revised 16 Sept. 2013; accepted 22 Sept. 2013. Date of publication 12 Nov. 2015; date of current version 20 Jan. 2016. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPDS.2015.2500722*

TABLE 1
Comparison of Prior Works [3], [4], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] and our Load-Balancing Framework

| features | distributed guiding | distance awareness | capacity awareness | load balancing | concurrent move modeling | evacuation time prediction |
|---|---|---|---|---|---|---|
| reference [3], [4], [12], [13], [14], [15], [16] | √ | √ | | | | |
| reference [6], [7], [8] | | √ | | | | |
| reference [9] | | √ | √ | | | |
| reference [10], [11] | | √ | √ | √ | | |
| reference [17] | √ | √ | | √ | | |
| reference [18] | √ | √ | √ | √ | | |
| our framework | √ | √ | √ | √ | √ | √ |

Table 1 compares the functions provided by existing schemes (discussed in Section 1.1) and the proposed scheme. This framework offers the most efficient solution for load-balanced emergency guiding. The contributions of this framework are four-fold. First, corridor capacity and length, exit capacity, and people distribution are considered in selecting escape directions. Second, a method of modeling the concurrent movement of people during evacuation is proposed for determining the congestion in the escape path to an exit. Third, the framework predicts the evacuation time from each sensor to an exit during concurrent people movement, and thus can plan the fastest route to an exit. Finally, the proposed framework can reselect guiding directions based on the estimated time to all exits, which substantially reduces the total evacuation time for all people. Extensive performance studies are conducted and the simulation results show that our framework can achieve better escape efficiency and shorter evacuation times than existing evacuation schemes. In addition, a prototype called the *Load-balancing Emergency Guiding System (LEGS)* is implemented; this system can be used to compare evacuation times and guiding directions provided by existing schemes and the proposed approach for various distributions of people. This paper is an extended version of our previously published work [19], which (i) develops complete processing flow of the proposed solution, (ii) analyzes the time complexity of the proposed algorithm, (iii) presents more extensive simulation results to validate the proposed framework, and (iv) provides new analysis and discussion for performance comparisons under different network topologies.

The rest of this paper is organized as follows. Section 2 defines our load-balancing problem for emergency guiding. Section 3 describes our framework to solve this problem. Simulation results are presented in Section 4. Section 5 shows our prototyping results. Section 6 concludes the paper.

### 1.1 Related Works

Several centralized solutions have been proposed for emergency guiding [6], [7], [8], [9], [10], [11], which centrally determine guiding paths in a control host. Reference [6] deploys a large number of active RFID tags in a building for indoor localization. People have to use personal digital assistants (PDAs) connected by RFID readers through Compact Flash interfaces for emergency guiding. However, the method in [6] guides people to the nearest exit without accounting for the distribution of people or balancing evacuation loads of exits. Thus, uneven people distribution and unbalanced emergency guiding may cause congestion at the nearest exit. In addition to a large number of active RFID tags, reference [7] needs to deploy a small number of Bluetooth devices in a building for indoor positioning. However, people distribution and load balancing are also not considered in [7], and people are always guided to the nearest exits, which can result in longer evacuation times due to congested exits.

In references [8] and [9], emergency evacuation planning varies temporally because open spaces and corridors become unavailable as hazards spread. [8] introduces a dynamic model for determining the safest and shortest path for evacuation during fire emergencies, but it is assumed that the number of people to evacuate does not exceed corridor capacities and, for simplicity, that only one exit exists. [9] proposes a flow-based heuristic framework for the real-time evacuation guidance of pedestrians during emergencies, which provides high-safety escape paths to sites far from hazardous regions; however, like [8], only a single exit during evacuation is considered in this framework; therefore, the evacuation load of each exit is not considered in planning escape paths.

Reference [10] considers that, in situations of stress and panic, visual and auditory information channels might be unavailable or overwhelmed because of infrastructure breakdown or limited perception, and, therefore, designs navigation device called LifeBelt based on vibro-tactile guidance technology to guide people in individual directions. The predicted exit time (PET) proposed in [10] is calculated based on three parameters: (i) the time to reach an exit, (ii) the exit capacity, and (iii) the current number of people at an exit. However, because the possible congestion caused by the concurrent movement of people and uneven people distribution is not considered, the PET may become longer as people near exits.

Reference [11] addresses pedestrian congestion and rescue force flexibility and uses a directed graph to model emergency regions. By considering people movement as network flows and calculating the maximum flow and minimum cut on the graph, the algorithm in [11] directs firemen to eliminate key danger areas instead of finding the fastest path to an exit before the arrival of rescue force.

Conversely, distributed solutions for emergency guiding have been proposed in [3], [4], [12], [13], [14], [15], [16], [17], [18], which locally determine guiding directions in sensors. Reference [12] determined an escape path from each sensor to the exit without passing any emergency point. Although the algorithm in [12] can be used to determine a shorter and safer path from each sensor to the exit, balancing loads for multiple exits is not considered. With shortest-path routing,

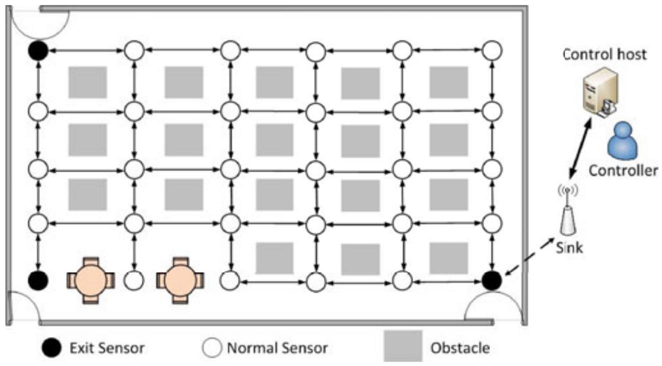



Fig. 1. System architecture of distributed emergency guiding.

this algorithm may guide people to a congested exit and thereby lengthen the total evacuation time, which also occurred in [3], [4], [13], [14], [15], [16].

Reference [17] aims to avoid congestion during emergency evacuation by guiding people to multiple exits along multiple corridors, but the load of each corridor is considered without taking the corridor capacity into account. In other words, people are distributed based on corridors rather than on exits, and some corridors can still become congested because of distinct corridor capacities.

To solve the congestion problem in [17], reference [18] proposes a distributed flow-based guiding protocol for indoor environments to evacuate peoples from danger areas to an exit. To construct paths that are less congested, each sensor in the network is assigned an artificial potential value determined by the moving speeds of people on a corridor, the distances to exits, and the capacity of exits, and people are directed to corridors with high potential values. However, local-optimal guiding adopted in [18], in which an escape path is selected based on the current speed of movement, could be fast initially and decrease later in the evacuation process.

## 2 System Model

Fig. 1 shows the system architecture of indoor emergency guiding. Sensor nodes (black and white circles) are deployed in a 2D indoor environment. These sensors form a multi-hop ad hoc network. One node serves as the *sink* of the network and is connected to the *control host*, which issues commands and configures the network. To support emergency-guiding services, sensors are classified as *normal sensors* (white circles), *exit sensors* (black circles), and *boundary sensors* (nodes with one or more neighboring sensors belonging to distinct emergency guiding trees). The method of finding the fastest escape path to an exit as detecting an emergency event is proposed in this paper. The details for constructing an initial guiding tree (i.e., shortest-path tree) rooted by an exit, preventing people from crossing hazardous region, and quickly reporting the status in the sensing field to the outside world can be found in our previous studies [3], [4].

In the proposed design, escape paths with the shortest possible evacuation times are determined and the guiding service is triggered by emergency events. Routing paths for packets are separated from escape paths for people. By overhearing wireless signals, a sensor can establish wireless links with its neighbors, and depending on how the plane is partitioned, a sensor can form navigational links (e.g., corridors) with its neighbors. However, because radio signals penetrate walls, a wireless link does not imply a navigational link, and, therefore, the navigational links are manually defined during deployment.

In the proposed system, the algorithm of emergency guiding is run based on detecting an emergency event, and the algorithm is rerun to prevent people from crossing a new dangerous area if the emergency event is detected outside the existing hazardous region. Assuming that all people follow guiding directions during emergency evacuation, and the load-balancing problem for emergency guiding is formulated as follows. The 2D indoor plane is represented by an undirected graph $G = (V, E)$, where $V$ is the set of all sensors and $E$ is the set of all corridors. Each exit has a specific capacity, and each corridor has a defined capacity and length. Each sensor knows the capacity and length of each corridor, the capacity of each exit, as well as its own location in the 2D plane. Moreover, the number of people near a sensor is detected by RFID or image recognition technology and reported to the sink periodically by each sensor. After collecting the number of people in each sensor, the sink broadcasts the people distribution information to all sensors. The goal is to minimize the total evacuation time for all people in $G$ by addressing the following issues:

1. *Escape Direction Selection*: How should the escape direction be selected during emergencies, to ensure that people indoors evacuate along the fastest path to an exit?
2. *Concurrent Move Modeling*: How should the concurrent movement of people during an evacuation be modeled, to ensure that the congestion in each escape path can be determined?
3. *Evacuation Time Prediction*: How should the evacuation time be predicted while escape paths are being determined, to ensure that the escape time from each sensor to an exit can be estimated accurately?
4. *Load-balanced Guiding*: How should a portion of people near a congested exit be guided to another exit when the predicted times to exits are unbalanced, to ensure that the total evacuation time can be minimized?

## 3 Load-Balancing Framework

The proposed framework is described in this section. A load-balancing guiding scheme is proposed in Section 3.1. An analytical model that aims to estimate the evacuation time is derived in Section 3.2.

### 3.1 Load-Balancing Guiding

To guide people to an exit with the shortest escape time, sensor nodes will reselect guiding directions based on the escape time to different exits estimated by the analytical model proposed in Section 3.2. Algorithm 1 gives the pseudocode of the load-balancing guiding algorithm, where $tree$ and $tree_{LB}$ are the sets of initial guiding trees and load-balancing guiding trees, respectively. As detecting an emergency event, all boundary sensors will execute the following steps:

**Step 1**: For each initial guiding tree $L_{\rm I}$ without hazardous region $R_{\rm H}$, boundary sensors will calculate the evacuation time $T_{exit}$ of $L_{\rm I}$ according to the proposed analytical model in Section 3.2. For each $L_{\rm I}$ with $R_{\rm H}$, the sensors and corridors inside $R_{\rm H}$ will be removed first. Then, the sensors (outside $R_{\rm H}$) losing their parent sensors (inside $R_{\rm H}$) will guide people to an exit through the shortest path so that a new guiding tree $L_{\rm N}$ is formed. In particular, if the emergency events become severe and no new guiding tree can be formed, the original one will be used for best-effort emergency guiding. For calculating $T_{exit}$ of $L_{\rm N}$, there are two possibilities as follows:

a) If there is an exit inside $R_{\rm H}$, all people inside $R_{\rm H}$ will be guided to the exit inside $R_{\rm H}$ through the shortest path. $T_{exit}$ of $L_{\rm N}$ can be estimated by the proposed analytical model in Section 3.2.
b) If there is no exit inside $R_{\rm H}$, the sensor detecting the emergency event and its corridors will be removed, and all other sensors inside $R_{\rm H}$ will guide people to the sensors outside $R_{\rm H}$ through the shortest path so that a new guiding tree $L_{\rm S}$ is formed. $T_{exit}$ of $L_{\rm S}$ can be estimated by the proposed analytical model in Section 3.2.

**Step 2**: The total evacuation time $T_{total}$ is decided by the emergency guiding tree $L_{\rm MAX}$ with the longest $T_{exit}$. After $T_{exit}$ of all emergency guiding trees are estimated by boundary sensors, boundary sensor $i$ in $L_{\rm MAX}$ will reset its guiding direction to the neighboring guiding tree $L_{\rm MIN}$ with the shortest $T_{exit}$ and recalculate $T_{exit}$ of $L_{\rm MAX}$ and $L_{\rm MIN}$. If $T_{total}$ decreases, $i$ will notify its neighboring sensors $b(i)$ in $L_{\rm MAX}$ that $b(i)$ become boundary sensors.

Assume that $L_{\rm MAX}$ has $m$ boundary sensors between $L_{\rm MAX}$ and $L_{\rm MIN}$, where boundary sensor $n_j$ is sorted in decreasing order by the number of people in $n_j$, for $j = 1, 2, \ldots, m$. The procedures of load-balancing guiding between $L_{\rm MAX}$ and $L_{\rm MIN}$ are as follows:

1) First, $n_1$ calculates $T_{exit}$ of $L_{\rm MAX}$ and $L_{\rm MIN}$ after removing and adding $n_1$, respectively. If $T_{total}$ decreases, $n_1$ will reset its guiding direction to $L_{\rm MIN}$ and notify its neighboring sensors $b(n_1)$ in $L_{\rm MAX}$ that $b(n_1)$ become boundary sensors.
2) Second, $n_2$ calculates $T_{exit}$ of $L_{\rm MAX}$ and $L_{\rm MIN}$ after removing and adding $n_1 + n_2$, respectively. If $T_{total}$ decreases, $n_2$ will reset its guiding direction to $L_{\rm MIN}$ and notify its neighboring sensors $b(n_2)$ in $L_{\rm MAX}$ that $b(n_2)$ become boundary sensors.
3) Similarly, $n_m$ calculates $T_{exit}$ of $L_{\rm MAX}$ and $L_{\rm MIN}$ after removing and adding $n_1 + n_2 + \cdots + n_m$, respectively. If $T_{total}$ decreases, $n_m$ will reset its guiding direction to $L_{\rm MIN}$ and notify its neighboring sensors $b(n_m)$ in $L_{\rm MAX}$ that $b(n_m)$ become boundary sensors.
4) Finally, new boundary sensors $b(n_1), b(n_2), \ldots,$ and $b(n_m)$ will repeat Step 2 to determine whether they should reset their guiding directions to $L_{\rm MIN}$ or not.

**Step 3**: Assume that there are $N$ emergency guiding trees in $G$. The load-balancing procedures under different numbers of emergency guiding trees are discussed in the following cases:

a) As $N = 2$, the total load-balancing guiding can be finished by Step 2.
b) As $N = 3$, the load-balancing guiding of Step 2 will be first done between the emergency guiding tree $L_1$ with the longest $T_{exit}$ and its neighboring guiding tree $L_2$ with the shortest $T_{exit}$. Then, the total load-balancing guiding can be finished by Step 2 between $L_1 + L_2$ and the third emergency guiding tree $L_3$.
c) As $N = 4$, the load-balancing guiding of Step 2 will be first done between the emergency guiding tree $L_1$ with the longest $T_{exit}$ and its neighboring guiding tree $L_2$ with the shortest $T_{exit}$. At the same time, the load-balancing guiding of Step 2 will be done between the remaining guiding trees $L_3$ and $L_4$. Then, the total load-balancing guiding can be finished by Step 2 between $L_1 + L_2$ and $L_3 + L_4$.
d) As $N \geq 5$, similarly, Steps 2 and 3 can be repeated to finish the total load-balancing guiding.

**Algorithm 1.** Load-Balancing Emergency Guiding

**Input:** $N, tree$
**Result:** $tree_{LB}$
**begin**
  **for** $i \leftarrow 1$ **to** $N$ **do**
    $T(i) =$ `Predict_Exit_Time` $(tree(i))$
  **while** `Number_of_Tree` $(tree) > 1$ **do**
    $L_{MAX} =$ `Longest_Exit_Time_Tree` $(T, tree)$
    $L_{MIN} =$ `Shortest_Exit_Time_Tree` $(L_{MAX}, T, tree)$
    $tree_{LB} =$ `Load_Balancing_Guiding` $(L_{MAX}, L_{MIN})$
    `Add_Tree_To` $(L_{MAX} + L_{MIN}, tree_{LB})$
    `Remove_Tree_From` $(L_{MAX}, tree)$
    `Remove_Tree_From` $(L_{MIN}, tree)$
    **if** `Number_of_Tree` $(tree) = 1$ **then**
      `Add_All_Trees_To` $(tree_{LB}, tree)$
      `Remove_All_Trees_From` $(tree_{LB})$
    **else if** `Number_of_Tree` $(tree) = 0$ **then**
      **if** `Number_of_Tree` $(tree_{LB}) > 1$ **then**
        `Add_All_Trees_To` $(tree_{LB}, tree)$
        `Remove_All_Trees_From` $(tree_{LB})$
  **return** $tree_{LB}$

### 3.2 Evacuation Time Analysis

Given an emergency guiding tree $L_{\rm G}$ rooted by an exit sensor, we derive its total evacuation time considering the corridor capacity and length, exit capacity, concurrent move, and people distribution. Assume that there are $n$ sensors in $L_{\rm G}$ and sensor 1 is the root. sensor 2, sensor 3, ..., and sensor $n$ are sorted in increasing order by their distances to sensor 1. Below, we first introduce some notations for sensor $i$, its parent sensor $p(i)$, and its child sensors $c(i)$ in $L_{\rm G}$, for $i = 1, 2, \ldots, n$:

- $T_i$: the time to evacuate from $i$ for the last person.
- $D_i$: the time to move from $i$ to $p(i)$.
- $N_i$: the number of people in $i$ as emergencies happen.
- $C_i$: the corridor capacity from $i$ to $p(i)$.
- $T^j_{c(i)}$: the time to evacuate from $i$'s child $j$ to $i$ for the last person.

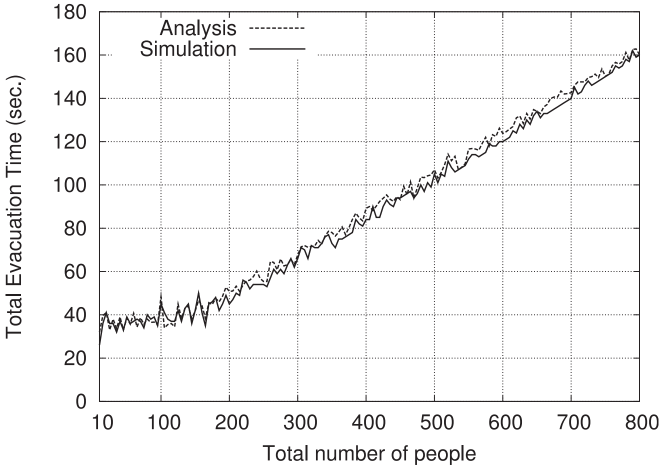



Fig. 2. Comparisons of analytical and simulation results.

- $D^j_{c(i)}$: the time to move from $i$'s child $j$ to $i$.
- $N^j_{c(i)}$: the number of people in $i$'s child $j$ as emergencies happen.
- $C^j_{c(i)}$: the corridor capacity from $i$'s child $j$ to $i$.

$T_i$ and $T^j_{c(i)}$ are the evacuation time of the subtrees rooted by $i$ and $i$'s child $j$ for the last person, respectively. Assume that there are $q$ child sensors in $c(i)$. We calculate the evacuation time $T_i$ of the subtree $L_{\mathrm{G}(i)}$ rooted by $i$ for the last person. The estimation of $T_i$ can be classified as follows according to whether there is congestion (i.e., the number of people arriving at the same time is larger than the corridor/exit capacities) in $i$ as the last person in $L_{\mathrm{G}(i)}$ evacuates from $i$:

**Case 1**: There is no congestion in $i$ as the last person in $L_{\mathrm{G}(i)}$ evacuates from $i$. The evacuation time $T_i$ is the sum of the time to evacuate from $c(i)$ for the last person and the time to move from $i$ to $p(i)$. So

$$T_i = \max_{j \in c(i)}(T^j_{c(i)}) + D_i.$$

**Case 2**: There is certain congestion in $i$ as the last person in $L_{\mathrm{G}(i)}$ evacuates from $i$. First, all sensors $j$ in $c(i)$ are sorted in increasing order by $D^j_{c(i)}$, for $j = 1, 2, \ldots, q$, such that $D^1_{c(i)} \leq D^2_{c(i)} \leq \cdots \leq D^q_{c(i)}$. Second, we find the smallest $k$ such that $C^1_{c(i)} + C^2_{c(i)} + \ldots + C^k_{c(i)} > C_i$. If no such $k$ exists, it represents that the corridor capacity between $i$ and $p(i)$ is large enough to be passed concurrently by the people from $c(i)$. In other words, there will be no congestion in $i$ and $T_i$ can be estimated by Case 1. Otherwise, there are two possibilities as follows:

a) If $D^k_{c(i)} \leq \frac{N_i}{C_i}$, it implies that there is a high probability of congestion in $i$ as the people evacuates from $c(i)$ to $i$. $T_i$ is modeled by summing the time to evacuate from $i$ for all people in $L_{\mathrm{G}(i)}$ and the time to move from $i$ to $p(i)$ as follows

$$T_i = \frac{\sum_{j=1}^{q} N^j_{c(i)} + N_i}{C_i} + D_i.$$

b) If $D^k_{c(i)} > \frac{N_i}{C_i}$, it implies that some people in sensor $1, 2, \ldots,$ and $k-1$ have evacuated from $i$ as the people in $k$ arrive at $i$. $T_i$ is modeled by summing the time to move from $k$ to $i$, the time to evacuate from $i$ for the remaining people in $L_{\mathrm{G}(i)}$, and the time to move from $i$ to $p(i)$ as follows

$$T_i = D^k_{c(i)} + T_{c(i)} + D_i,$$

where

$$T_{c(i)} = \frac{\sum_{j=1}^{q} N^j_{c(i)} - \sum_{j=1}^{k} C^j_{c(i)} \times (D^k_{c(i)} - D^j_{c(i)})}{C_i}.$$

In the analytical model, we adopt the maximum time estimated by Case 1 and Case 2 as $T_i$ since $T_i$ is the time to evacuate from $i$ for the last person in $L_{\mathrm{G}(i)}$, for $i = 1, 2, \ldots, n$. Thus, the total evacuation time of $L_{\mathrm{G}}$ is equal to $T_1$ that can be obtained by calculating $T_n, T_{n-1}, \ldots$ and $T_1$ in order, where $D_1 = 0$ and $C_1$ is the exit capacity.

Next, we analyze the time complexity of the load-balancing emergency guiding algorithm. For load balancing between guiding trees, the evacuation time analysis will be executed $N + \lceil \frac{N}{2} \rceil + \lceil \frac{N}{4} \rceil + \cdots + 1$ times, where $N$ is the number of exits in $G$. Since each sensor node has exactly one evacuation time, the time estimating step will be repeated at most $V$ times, where $V$ is the number of vertices in $G$. In Case 1, at most $M$ child nodes will be compared with their evacuation time to find the maximum one, where $M$ is the maximum number of children that a node has in all guiding trees. Similarly, in Case 2, at most $M$ numbers of people in child nodes will be summed up. The running time of load-balancing emergency guiding is $O(2N \times V(M + M))$. Thus, the time complexity is $O(NVM)$.

Fig. 2 shows evacuation time comparisons of analytical and simulation results under $10, 20, \ldots, 790$, and 800 people. We deploy $4 \times 4$ sensors in a 2D grid plane. The exit sensor is located on the top left corner, and the remaining sensors are normal sensors. The corridor and exit capacities, the moving time for each corridor, and the number of people in each sensor are randomly assigned as the same with Section 4. We compare the evacuation time estimated by the analytical model with the simulation ones. In Fig. 2, each simulation is repeated 100 times and we take the average value. As can be seen, the simulated and analytical results are quite close, which justifies the correctness of our derivation.

## 4 PERFORMANCE EVALUATION

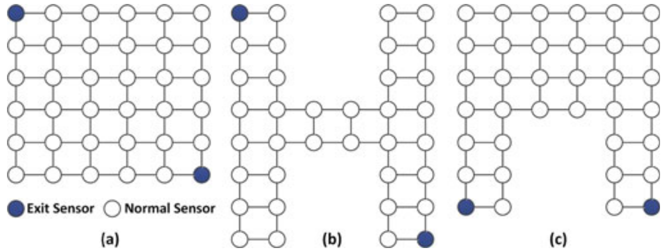



Fig. 3. Topologies (a)A, (b)B, and (c)C used in the simulation.

Fig. 3 shows three topologies including fully-square, non-square, and half-square layouts used in the simulation and basic parameters are summarized in Table 2. We deploy 36 sensors in each topology. There are two exit sensors located

TABLE 2
Simulation Parameters

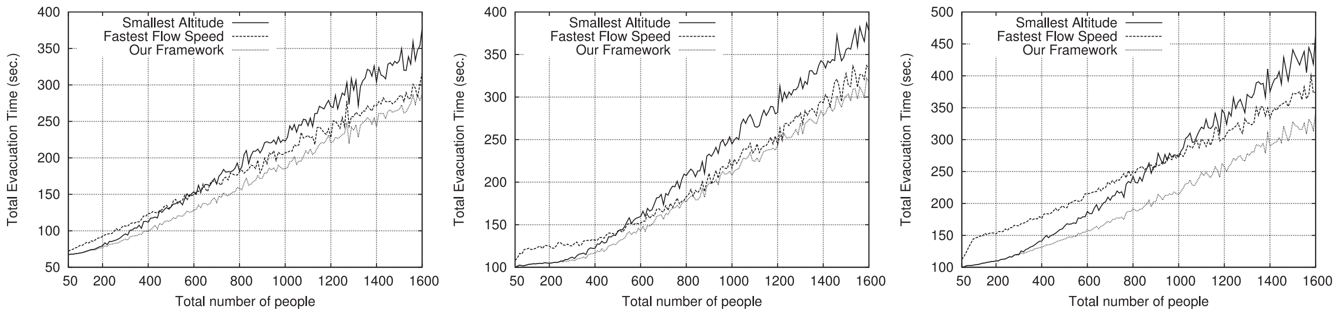
| Parameter | Value |
|---|---|
| Number of Sensor Nodes | 36 |
| Number of Exit Sensors | 2 |
| Number of Hot-spot Sensors | $2 \sim 16$ |
| Corridor Capacity | $2 \sim 6$ people/second |
| Exit Capacity | $2 \sim 6$ people/second |
| Moving Time of Corridors | $10 \sim 15$ seconds |

on the corners, and the remaining sensors are normal sensors. The corridor and exit capacities are randomly selected from 2 to 6 people/second, and the moving time for each corridor is randomly chosen from 10 to 15 seconds. We first randomly assign 50 percent people to all sensors and then randomly select $2 \sim 16$ hot-spot sensors for assigning the other 50 percent people to them such that hot-spot sensors have many more people to be guided than the other sensors. All people have the same speed to escape and they will be delayed at crossing points if the number of people arriving at the same time is larger than the corridor/exit capacities. Each simulation is repeated 100 times and we take the average value using a C++ simulator. We compare our scheme against the Smallest Altitude First (SAF) method [3], [4] that guides people to the neighboring sensor with the smallest hop count to an exit and the Fastest Flow Speed First (FFSF) method [18] that guides people to the neighboring sensor with the fastest moving speed.

Figs. 4a, 4b, and 4c show comparisons of total evacuation time under $50, 60, \ldots, 1,590,$ and 1,600 people in topology A, B, and C, respectively. From these figures, we can observe that our scheme has the shortest evacuation time with all numbers of people. This is because our scheme takes the corridor capacity and length, exit capacity, concurrent move, and people distribution into consideration. So the congestion of certain corridors and exits can be released to reduce the evacuation time of people. On the contrary, SAF will guide people to the sensor closer to the exit even if there is serious congestion in the nearest exit, and FFSF will guide people to the sensor with the faster moving speed even if the speed near the exit is slowdown due to congestion.

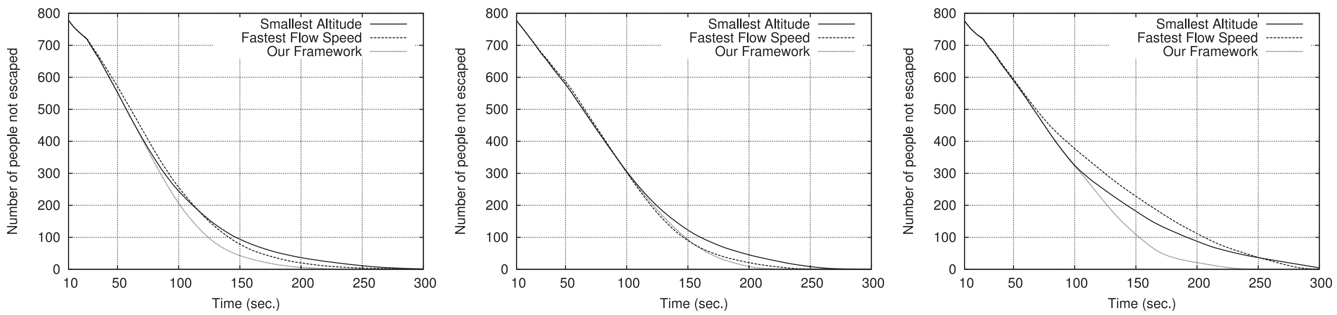
From Fig. 4c, we can find that both SAF and FFSF suffer from the local optimal selection problem. SAF may select an escape path with the shortest distance to an exit but longer evacuation time, especially during global congestion under a large number of people. FFSF may select an escape path with the faster moving speed currently but slower later, especially during local congestion under a small number of people. In particular, while SAF has the longer evacuation time than FFSF with more than 800 people, FFSF has the longer one than SAF with less than 800 people.

Fig. 5 shows comparisons of numbers of people not escaped every 5 seconds for nine hot spots under 800 people. We can observe that our scheme has the fastest escaping speed with the much shorter evacuation time than other schemes. More importantly, while SAF and FFSF have the low escaping speed due to serious congestions on corridors/exits, our scheme still can further reduce the total evacuation time significantly by balancing the evacuation load of each exit. In particular, using our scheme, there will be more people able to escape if the building collapses in a short time (i.e., each exit is blocked before all people escape). Note that in Fig. 5b, our scheme has similar performance to FFSF because most people are congested in the long-thin corridor connecting to two exits, which is the



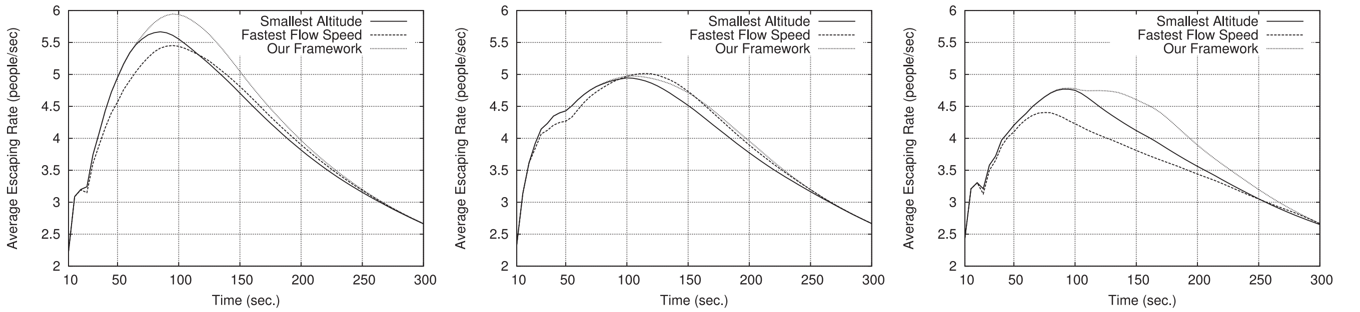

(a) Topology A (b) Topology B (c) Topology C

Fig. 4. Comparisons of total evacuation time for nine hot spots under different numbers of people.




(a) Topology A (b) Topology B (c) Topology C

Fig. 5. Comparisons of numbers of people not escaped every 5 seconds for nine hot spots under 800 people.




(a) Topology A (b) Topology B (c) Topology C

Fig. 6. Comparisons of average escaping rates every 5 seconds for nine hot spots under 800 people.

common bottleneck for evacuation. On the other hand, Fig. 6 shows comparisons of average escaping rates in Fig. 5.

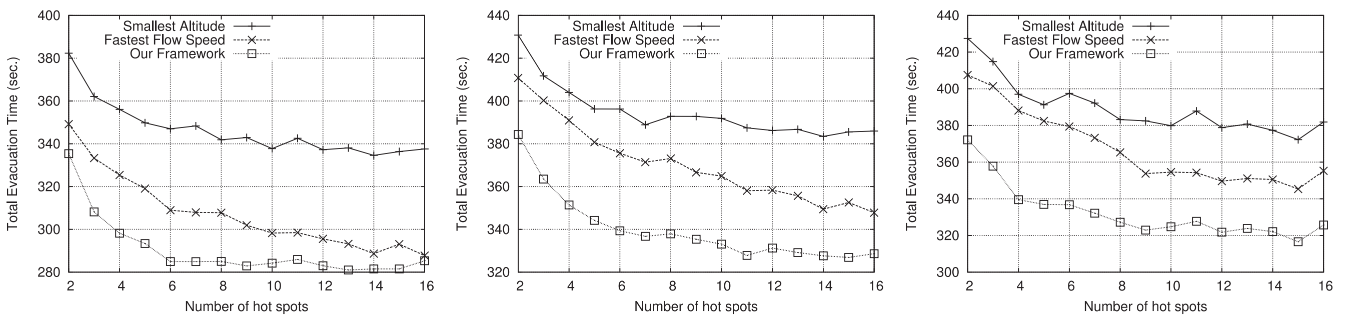
Figs. 7a, 7b, and 7c show comparisons of total evacuation time under $2, 3, \ldots, 15,$ and 16 hot-spot sensors in topology A, B, and C, respectively. The total number of people is set to 1,600 to show the guiding efficiency during heavy congestion under different numbers of hot-spot sensors. We first randomly select the specific number of hot-spot sensors to whom 50 percent of the people will be assigned and then randomly assign the other 50 percent people to all sensors. From these figures, we can observe that the evacuation time of our scheme is shorter than those of SAF and FFSF with all numbers of hot-spot sensors. Similarly, our scheme can release the congestion of certain corridors and exits to reduce the evacuation time of people. More specifically, the number of people congested in certain corridor or exit can be reduced by guiding them to different escape paths. On the contrary, both SAF and FFSF may guide people to the local optimal direction with the longer evacuation time in total. In particular, the guiding efficiency of our approach is increased and decreased during local and global congestion under small and large numbers of hot spots, respectively. We can find that the total evacuation time of our approach and FFSF are closer as the number of hot spots increases. This is because with more hot spots, more corridors are congested and more time is needed to escape from each exit. Although the load-balancing effect will be limited by the large number of hot spots, our approach still can evacuate people to exits in the shortest time.

## 5 Prototype Implementation

We implement a WSN-based prototype, called *Load-balancing Emergency Guiding System (LEGS)* [20]. Fig. 8a shows the front side and back side of the sensor equipped with a TFT LCD panel controlled by ATmega128 [21], a light sensor TSL2560 [22], and UI buttons as input devices. The SAF method [3], [4], the FFSF method [18], and our approach are implemented in Jennic JN5139 [23], which all have quick response time within 1 second after detecting an emergency event.

JN5139 [23] has a 16MIPs 32-bit RISC processor, a 2.4 GHz IEEE 802.15.4-compliant transceiver, 192kB of ROM, and 96 kB of RAM. The ROM/RAM architecture supports the storage of system software, including protocol stacks, routing tables, and application code/data. An external Flash memory is used to store application code that is boot-loaded into internal RAM and executed at runtime. JN5139 integrates hardware MAC and AES encryption accelerators, power-saving and timed sleep modes, and mechanisms for security key and program code encryption, which is a highly efficient, low-power, single-chip wireless microcontroller for battery-powered applications. In particular, JN5139 allows the flexibility of supporting mesh networking and packet routing inside a building.

TSL2560 [22] is a light sensor developed by LUMENOLOGY, which has a light-to-digital converter for transforming light intensity to a digital signal output. It combines a broadband photodiode (visible plus infrared) and a infrared-responding photodiode on a single CMOS integrated circuit capable of providing a near-photopic response over an effective 20-bit dynamic range (16-bit resolution). Two integrated ADCs convert the photodiode currents to a digital output that represents the irradiance measured on each channel. The digital output can be read by a microprocessor that ambient light level (in lux) is derived using an empirical formula to approximate the human eye response. For




(a) Topology A (b) Topology B (c) Topology C

Fig. 7. Comparisons of total evacuation time for 1,600 people under different numbers of hot spots.


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

Fig. 8. Prototype and hardware components of LEGS.

triggering emergency events, we integrates TSL2560 with JN5139 to sense the light intensity.

ATmega128 [21] is a 8-bit RISC-based microcontroller, which has 128 kB of Flash, 4 kB EEPROM, 4 kB SRAM, 53 general purpose I/O lines, 32 general purpose working registers, 4 flexible Timer/Counters, 2 UARTs, and a programmable watchdog timer with internal oscillator. IEEE 1149.1-compliant JTAG test interface is used for accessing the on-chip debugging system and software selectable power saving modes. The communication between ATmega128 and JN5139 is through the UART port for controlling the LCD panel.

For the demonstration of indoor people evacuation, we use a projector to simulate that people escape from an office as emergences happen, as shown in Figs. 8a, 8b and 8c. The evacuation simulator is developed by Processing [24] to create images, animations, and interactions. The people in the office can be dynamically added and removed on the screen by the cursor, and the guiding scheme can be selected by clicking one of the three green circles (the red one is the selected scheme and the yellow one is the reset button). The light sensor can be highlighted by a laser pen to trigger an emergency event and three emergency guiding methods can be selected for evacuating people to exits. The animation of people evacuation is shown in the projected screen and the evacuation time is counted until all people escape from the office. From the screen, we can compare the evacuation time and guiding directions of SAF, FFSF, and our approach under different people distribution. To show how indoor people evacuate based on different guiding schemes, the supplement provides a demo video of LEGS.

## 6 Conclusion

In this paper, we propose an emergency guiding framework consisting of a load-balancing guiding scheme and an evacuation-time predicting model. Through the proposed framework, the congestion of certain corridors and exits can be released to significantly reduce the total evacuation time, and people can avoid to follow the local optimal guiding direction with the longer evacuation time in total. Analytical and simulation results show that the proposed approach can accurately predicate evacuation time of people and evenly distribute evacuation load among exits to achieve the best performance, leading to more efficient distributed emergency guiding. In other words, adopting our scheme in emergency guiding can both avoid the congested corridors and exits wasting time due to incomplete information of spatial-temporal mobility and prevent people from following guiding directions with longer escape time caused by unbalanced evacuation load. In particular, the hazard spreading is not addressed in the current evacuation time analysis, which needs to be further investigated.

## Acknowledgments

This research is supported in part by NSC under Grant No. 101-2221-E-035-090 and 102-2221-E-035-031-MY3. Y.-C. Tseng's research is co-sponsored by MoE ATU Plan, NSC 101-2221-E-009-024-MY3, NSC 102-2218-E-009-002, IVF-NSC joint research grant 21280013 (102-2923-E-009-001-MY2), Academia Sinica AS-102-TP-A06, ITRI, hTC, and D-Link.

## References

[1] L.-W. Chen, K.-Z. Syue, and Y.-C. Tseng, "A vehicular surveillance and sensing system for car security and tracking applications," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sens. Netw.*, Apr. 2010, pp. 426–427.

[2] L.-W. Chen, Y.-H. Peng, and Y.-C. Tseng, "An infrastructure-less framework for preventing rear-end collisions by vehicular sensor networks," *IEEE Commun. Lett.*, vol. 15, no. 3, pp. 358–360, Mar. 2011.

[3] Y.-C. Tseng, M.-S. Pan, and Y.-Y. Tsai, "Wireless sensor networks for emergency navigation," *IEEE Comput.*, vol. 39, no. 7, pp. 55–62, Jul. 2006.

[4] M.-S. Pan, C.-H. Tsai, and Y.-C. Tseng, "Emergency guiding and monitoring applications in indoor 3D environments by wireless sensor networks," *Int. J. Sens. Netw.*, vol. 1, nos. 1/2, pp. 2–10, Jan. 2006.

[5] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 1997, p. 1405.

[6] L. Chittaro and D. Nadalutti, "Presenting evacuation instructions on mobile devices by means of location-aware 3D virtual environments," in *Proc. Int. Conf. Human Comput. Interaction Mobile Devices Services*, 2008, pp. 395–398.

[7] S. R. Gandhi, A. Ganz, and G. Mullett, "FIREGUIDE: Firefighter guide and tracker," in *Proc. IEEE Int. Conf. Eng. Med. Biol. Soc.*, 2010, pp. 2037–2040.

[8] T. Tabirca, K. N. Brown, and C. J. Sreenan,, "A dynamic model for fire emergency evacuation based on wireless sensor networks," in *Proc. Int. Symp. Parallel Distrib. Comput.*, Jun. 2009, pp. 29–36.

[9] T. Hadzic, K. N. Brown, and C. J. Sreenan, "Real-time pedestrian evacuation planning during emergency," in *Proc. IEEE Int. Conf. Tools Artif. Intell.*, 2011, pp. 597–604.

[10] A. Fersch and K. Zia, "On the efficiency of lifebelt based crowd evacuation," in *Proc. IEEE/ACM Int. Symp. Distrib. Simul. Real Time Appl.*, 2009, pp. 13–20.

[11] S. Li, A. Zhan, X. Wu, P. Yang, and G. Chen, "Efficient emergency rescue navigation with wireless sensor networks," *J. Inf. Sci. Eng.*, vol. 27, no. 1, pp. 51–64, Jan. 2011.

[12] Q. Li, M. DeRosa, and D. Rus, "Distributed algorithm for guiding navigation across a sensor network," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2003, pp. 313–325.

[13] C. K. Christakos, "Sensor networks applied to the problem of building evacuation: An evaluation in simulation," in *Proc. 15th IST Mobile Wireless Summit*, Jun. 2006, pp. 134–138.

[14] D. A. C. Buragohain and S. Suri, "Distributed navigation algorithms for sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, Apr. 2006, pp. 1–10.

[15] M. Barnes, H. Leather, and D. K. Arvind, "Emergency evacuation using wireless sensor networks," in *Proc. IEEE Int. Conf. Local Comput. Netw.*, Oct. 2007, pp. 851–857.

[16] S. Timotheou and G. Loukas, "Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios," in *Proc. ACM Symp. Appl. Comput.*, Mar. 2009, pp. 1171–1175.

[17] W.-T. Chen, P.-Y. Chen, C.-H. Wu, and C.-F. Huang, "A load-balanced guiding navigation protocol in wireless sensor networks," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2008, pp. 1–6.

[18] P.-Y. Chen, Z.-F. Kao, W.-T. Chen, and C.-H. Lin, "A distributed flow-based guiding protocol in wireless sensor networks," in *Proc. Int. Conf. Parallel Process.*, Sep. 2011, pp. 105–114.

[19] L.-W. Chen, J.-H. Cheng, and Y.-C. Tseng, "Evacuation time analysis and optimization for distributed emergency guiding based on wireless sensor networks," in *Proc. ACM/IEEE Int. Conf. Connected Vehicles Expo*, Dec. 2012, pp. 130–135.

[20] L.-W. Chen, J.-H. Cheng, Y.-C. Tseng, L.-C. Kuo, J.-C. Chiang, and W.-J. Lin, "LEGS: A load-balancing emergency guiding system based on wireless sensor networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2012, pp. 486–488.

[21] (2013, Jun. 28). TFT LCD microcontroller, ATmega128 [Online]. Available: http://www.atmel.com

[22] (2013, Jun. 28). Light sensor, TSL2560 [Online]. Available: http://www.taosinc.com/

[23] (2013, Jun. 28). Jennic, JN5139 [Online]. Available: http://www.jennic.com/

[24] (2013, Jun. 28). Processing [Online]. Available: http://www.processing.org/


**Lien-Wu Chen** received the BS and MS degrees both in computer science and information engineering from the Fu-Jen Catholic University and the National Central University, in 1998 and 2000, respectively. He received the PhD degree in computer science and information engineering from the National Chiao Tung University in December 2008. He was a research assistant at the Institute of Information Science, Academia Sinica (Jan. 2001-Feb. 2009). He joined the Department of Computer Science, National Chiao Tung University as postdoctoral research associate, in February 2009, and qualified as an assistant research fellow in February 2010. He is currently an assistant professor since February 2012 in the Department of Information Engineering and Computer Science, Feng Chia University. His research interests include wireless communication and mobile computing, especially in Mobile Ad Hoc and Sensor Networks. He has published 36 conference and journal papers and has 25 approved and filed patents. Among those patents, the technologies for file transfer, application execution, and goods data searching methods based on Augmented Reality and Cloud Computing have been transferred to Acer Incorporated in January, June, and November 2011, respectively. He was a guest researcher at the Industrial Technology Research Institute (February 2009-December 2009) and Chief Executive Officer at the RFID Resource Center, National Chiao Tung University (April 2009-March 2011). He received the Outstanding Demo Award (1st author) at the IEEE MASS 2009, the Best Paper Award (1st author) at the WASN 2010, the Audience Award (2nd prize) at the ESNC 2011, the Best Paper Award (1st author) at the MC-2012, the Prototyping Award (Stars prize) at the ESNC 2012, the Best Paper Award (1st author) at the WASN 2013, and the Best Demo Award (1st author) at the MC-2013. He is an honorary member of the Phi Tau Phi Scholastic Honor Society, and is a member of the ACM and IEEE.

**Jen-Hsiang Cheng** received the BS and MS degrees both in computer science and information engineering from the National Chiao Tung University in 2000 and 2012, respectively. He was an exchange student at the Department of Electrical & Computer Engineering, Carnegie Mellon University (Feburary 2000-July 2000). He is currently a research & development engineer at the Realtek Semiconductor Corp., Taiwan. He received the 2nd prize of Audience Award at the European Satellite Navigation Competition 2011, the Best Paper Award at the Mobile Computing Workshop 2012, and the Stars prize of Prototyping Award at the European Satellite Navigation Competition 2012. His research interests include wireless sensor networks and vehicular ad hoc networks.

**Yu-Chee Tseng** received the PhD degree in computer and information science from the Ohio State University in January 1994. He was/is chairman (2005-2009) and dean (2011-present), College of Computer Science, National Chiao-Tung University, Taiwan. He has been awarded as NCTU Chair Professor (2011-present) and Y. Z. Hsu Scientific Chair Professor (2012-2013). He received the Outstanding Research Award (National Science Council, 2001, 2003, and 2009), Best Paper Award (Int'l Conf. on Parallel Processing, 2003), Elite I. T. Award (2004), and Distinguished Alumnus Award (Ohio State University, 2005), and Y. Z. Hsu Scientific Paper Award (2009). His research interests include mobile computing, wireless communication, and sensor networks. He served on the editorial boards of *IEEE Trans. on Vehicular Technology, IEEE Trans. on Mobile Computing, and IEEE Trans. on Parallel and Distributed Systems*. His H-index is 50. He is fellow of the IEEE.