An Automatic, Robust, and Efficient Multi-User Breadcrumb System for Emergency Response Applications

Hengchang Liu, Zhiheng Xie, Jingyuan Li, Shan Lin, David J. Siu,
Pan Hui, Kamin Whitehouse, and John A. Stankovic, *Fellow, IEEE*

Abstract—Breadcrumb systems (BCS) aid first responders by communicating their physiological parameters to remotely located base stations. In this paper, we describe the design, implementation, and evaluation of an automatic and robust multi-user breadcrumb system for indoor first response applications. Our solution includes a breadcrumb dispenser with a link estimator that is used to decide when to deploy breadcrumbs to maintain reliable wireless connectivity. The solution includes accounting for realities of buildings and dispensing such as the height difference between where the dispenser is worn and the floor where the dispensed nodes are found. We also include adaptive power management to maintain link quality over time. Moreover, we propose *UF*, a distributed cooperative deployment algorithm, to achieve longer breadcrumb chain lengths while maintaining fairness and high system reliability via selecting appropriate benefit and cost functions. We deployed and evaluated our system in real buildings with several different first responder mobility patterns. Experimental results from our study show that compared to the state of the art solution [27], our breadcrumb system achieves 200 percent link redundancy with only 23 percent additional deployed nodes. Our deployed breadcrumb chain can achieve 90 percent PRR when one node fails in the chain. In addition, by applying the *UF* coordination algorithm, the system can maintain connectivity for up to 87 percent longer distances than baseline greedy coordination approach while maintaining 96 percent packet delivery ratio.

Index Terms—Breadcrumb systems, emergency response applications, link monitoring, utility function, deployment and evaluation

1 Introduction

First responder (FR) safety is a critical issue especially when dealing with disasters in large buildings. Monitoring physiological parameters such as heart rate and stress of these first responders in real-time can save lives [2]. However, reliably transmitting this physiological data to a base station outside the building is a challenging problem. Existing solutions normally use one-hop communications [3] and suffer from limited transmission range since it is sometimes difficult for wireless signals to travel through complex infrastructures [24]. One promising approach to support reliable wireless communication is the so-called breadcrumb-based method spearheaded by the Science and Technology Directorate of the Department of Homeland Security [8] which allows a first responder to carry a small dispenser filled with sensor nodes and deploy them one-by-one in a manner that guarantees reliable communication. This paper describes the complete implementation and evaluation of a breadcrumb solution that automatically dispenses sensor nodes to achieve reliable communication and high packet reception ratio. While this paper focuses on reliable communication, it is important to note that breadcrumb based solutions, in general, have other potential major advantages over one hop radios, including: (i) by adding sensors to the dispensed nodes it is possible to map the fire, detect poison gases and smoke and help plan egress routes, and (ii) with additional algorithms it might also be possible to localize where first responders are or where events occur.

In current breadcrumb systems, while the research focus is on the feasibility of an automated dispensing process [17], [27], to date *ALL* functional and working prototypes built require manual deployment. This interferes with the first responders' main tasks and also takes longer to deploy than a completely automated solution. Since first responders will wear the dispensers on their hips, but once deployed nodes will be on the ground, there is a necessity to account for this height difference and its affect on resulting communication quality. Current solutions to account for the height effect adopt conservative approaches which lead to requiring a significantly large number of breadcrumbs. In this paper, we consider this problem from an optimization point of view. Given a limited number of breadcrumbs available, we address the problem of finding an optimized deployment scheme that minimizes the number of breadcrumbs while maintaining high system reliability. The main contributions of this work are:

- We develop an automatic and robust breadcrumb system using 2.4 GHz based hardware (see Fig. 1 for

- *H. Liu is with the Department of Computer Science, University of Science and Technology of China, Suzhou, Jiangsu 215123, China. E-mail: liu.heng.chang@gmail.com.*
- *Z. Xie, J. Li, K. Whitehouse, and J. Stankovic are with the Department of Computer Science, University of Virginia, Charlottesville, VA 22904.*
- *S. Lin is with the Department of Computer Science, Temple University, Philadelphia, PA 19122.*
- *D. Siu is with the OCEANIT Inc., Honolulu, HI.*
- *P. Hui is with the Deutsche Telekom Research Lab, Berlin, Germany.*

Manuscript received 26 Jun. 2012; revised 24 Nov. 2012; accepted 10 May 2013; date of publication 21 May 2013; date of current version 3 Mar. 2014.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2013.63

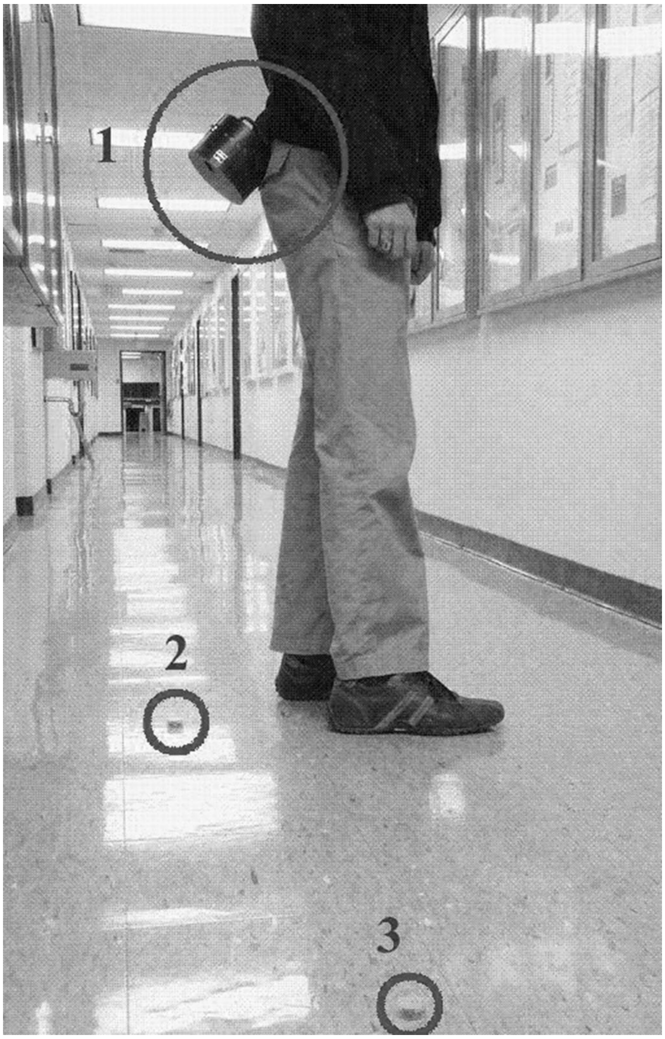


Fig. 1. Automatic breadcrumb system in action, with a dispenser (1) and breadcrumbs (2 and 3).

a dispenser (1) and breadcrumbs (2 and 3)). To the best of our knowledge, this is the first prototype system that implements a *real automated deployment* process for breadcrumb systems.

- We describe a novel reliability model for breadcrumb systems, which includes an optimal redundancy degree, a link monitoring algorithm, a height effect solver, and adaptive power control (APC). Evaluation results show that our breadcrumb system achieves 200 percent link redundancy with only 23 percent additional deployed nodes, compared to the state of the art solution [27].
- We present *UF*, a utility function based algorithm that provides an efficient and distributed coordination process via selecting appropriate benefit and cost functions. In addition, *UF* requires no *a priori* user mobility models, making the design practical. Experimental results indicate that this approach can maintain connectivity for up to 87.3 percent longer distances than baseline greedy approach while maintaining a 96 percent packet delivery ratio.
- We evaluated our system in three different buildings, two in University of Virginia and one in Washington DC, as well as a park in Seattle. The results demonstrate that our proposed work can achieve a reliable and efficient breadcrumb network in all indoor and outdoor environment.
- Finally, we successfully finished two demonstrations to the Computer Science Division at the Department of Homeland Security in August 2010 and April 2011. We showed three traces in the demos: one FR dropping breadcrumbs; two FRs walking together and fairly dropping breadcrumbs; and four FRs together and subsequently splitting off one by one. We were able to establish reliable communications between the FRs and the base station in all cases.

The remainder of this paper is organized as follows. We compare our work with state of the art in Section 2. The detailed description for the reliability model and multi-user coordination are presented in Section 3 and Section 4, respectively. The implementation and evaluation for our system are discussed in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

First response sensor systems are an active area of research, nevertheless, the challenges of designing such reliable, efficient, and automated platforms have only been explored partially. Previous work [12], [31], [32] on offline relay placement is not suitable for disaster response applications, because fault tolerance was not a primary design goal in their systems. On the other hand, building a dynamic infrastructure in real time has attracted more attention in recent years. This type of system normally includes two parts [17]: First responders automatically deploy sensor nodes along their paths, effectively establishing an ad-hoc infrastructure for positioning, sensing and communication; and then they interact with this sensor network by way of wearable computing equipment and receive navigational information on a head-mounted display or over a headset.

Previous system designs mostly focused on the second part, by designing various kinds of wearable components that could be conveniently carried by first responders. For example, the *FIRE* project [1] aimed at designing new technologies such as small head-mounted displays (HMDs) for firefighting, and conducting experiments and exploratory research with first responders. It basically included three sub-projects: *SmokeNet* to design pre-deployed WSN for detecting fires; *FireEye* to equip first responders with head mounted display units; and *eICS* to provide visual display showing resource allocation, personnel location and first responder biometrics. Similarly, the *SIREN* project [16] provided reliable communication among first responders using a WiFi-enabled PDA with a built in mote. The mote collected data from pre-deployed sensors in a building to inform the first responder of hazards and immediate danger. Pre-deployed motes also served as location beacons that allowed first responders to navigate through the building. Other similar systems included *LIFENET* [17], [18], [19] and *MHMD* [29]. However, these first attempts of designing first responder-assisting sensor systems relied heavily on pre-existing networks in the on-fire building. This is an invalid assumption at least in the near future. Thus, researchers have become more interested in the first part: how to deploy relaying nodes automatically and rapidly to maintain reliable communication between first responders inside the building and base stations outside the building.

There are currently three solution approaches: no deployment, static deployment, and dynamic deployment. In the no deployment approach, a first responder usually carries a radio and communicates with the outside world within a single hop. One example system of this kind of approach is the $P25$ system [3]. Static deployment adopts a simple rule such as dropping the next node based on distance or time of the last deployment. Dynamic deployment monitors the run-time link quality and automatically deploys a new relay node whenever necessary.

The $P25$ system is the standard for the design and manufacture of inter-operable digital two-way wireless communications products. Radio equipment that demonstrated compliance with P25 was able to meet a set of minimum requirements to fit the needs of public safety. However, due to hardware limitations, first responders would inevitably lose their connections to the base station as they climbed to tops floors in a tall building. Even worse, complicated indoor environments often contain substantial amounts of metal and other reflective materials that affect the propagation of radio frequency signals in nontrivial ways, causing severe multi-path effects, dead-spots, noise and interference [15]. In recent years, more and more reports have been published on the failure of public-safety technologies, especially the P25 system, on local news [4], webs [5], and Youtube [6], [7]. Please refer to [24] for more details on P25 system failures.

Due to the drawbacks of current P25 systems, breadcrumb sensor networks are emerging in recent years to provide more reliable communication links between first responders and the incident commander. The deployment process in breadcrumb sensor networks can be divided into static and dynamic deployment. Static deployment adopts a simple rule such as distance or time of the last deployment. However, simple static rules do not capture the wide varieties of radio implementations that affect transmission range, such as different radio types, antenna types, and transmission power levels. More important, static deployment rules do not adapt to different channel propagation environments. For example, the range in an office corridor might be different from that on a factory floor [27].

On the other hand, dynamic deployment monitors the run-time link quality and automatically deploys a new relay node whenever the communication metric (PRR, RSSI, LQI, etc.) satisfies some predefined rules. [27] was the first work to investigate the feasibility of dynamic breadcrumb deployment to extend the range of wireless communications, based on a stable PRR-RSSI mapping they observed in indoor environments. In this work, a mobile device on the first responder probes the channel periodically and measures link quality of measurement response. If the filtered measurements of link quality (based on a moving-average approach) are less than a threshold, deployment of a new node is triggered. The system is evaluated by experiments with Mica2 motes and a PDA. Several following works from NIST consider link quality measurements via a SNR based approach [30], interference avoidance [25], and UWB indoor localization techniques [13], [14].

However, there are several disadvantages of the NIST work. First, it needs human involvement such as deploying new breadcrumbs by hand and reading the PDA messages frequently. These activities are undesirable in real applications. Second, the NIST system only evaluates the case with no redundant breadcrumbs, resulting in a fragile crumb chain. Due to the harsh environment in an on-fire building, physical failure of breadcrumbs is likely to occur and the death of any breadcrumb leads to the failure of the whole system. Third, the link quality monitor used in the NIST system is not appropriate. We will explain it in detail in later sections. Finally, NIST used an uniform threshold for all environments and ignore the different characteristics in various locations like hallways, corners, and stairways. This lack of optimization makes it less efficient in use of limited breadcrumbs.

To summarize, the development of a breadcrumb sensor network is still in its infancy, and there is a lack of systematic system design and effort to make the deployment process automatic and efficient. In this paper, a sophisticated study of automatic, reliable, and efficient multi-user breadcrumb systems is presented.

3 Reliability Model

Breadcrumb sensor networks bring great challenges to system reliability problems, mainly due to the harsh environments in emergency response applications. Deployed breadcrumbs may be burned, or destroyed by collapsed walls, or moved out of the way by first responders or fire hoses. How to design highly reliable deployment algorithms is unknown.

We propose a new reliability model that consists of: (1) an optimized redundancy degree for breadcrumbs, (2) a decision support system for wireless link estimation that decides when to drop additional breadcrumbs, (3) a height effect solver to handle the gap in link quality after breadcrumbs drop from the dispenser, and (4) an adaptive transmission power control to handle link quality variation problems in harsh environments. These components together provide reliable and efficient means of automatic and robust breadcrumb deployment for in-door first responder applications. Note that these components and strategies may change if other system functions are introduced. We describe the overall system design and individual components in the following sections.

3.1 Solution Overview

We first describe the application scenario and how our proposed system is used for firefighter applications. Our goal is to establish a breadcrumb chain that can relay the physiological data from the body sensors on firefighter to base stations outside the building. Each firefighter carries m breadcrumbs in his crumb dispenser and our system automatically deploys a breadcrumb whenever connection to the deployed breadcrumb trail is getting weak. As firefighters run into the building, breadcrumbs are deployed automatically on the fly. Our deployment policy requires that each crumb keeps "good communication" with at least $n+1$ other crumbs at any time in order to have redundancies to tolerate crumb failures. Here, n represents the *redundancy degree* of each crumb. Note that the selection of redundancy degree requires a tradeoff between the number of breadcrumbs deployed and end-to-end reliability of the crumb-chain.

As the firefighter moves on for rescue work, the link quality between the dispenser on the firefighter and the breadcrumbs becomes weaker. The *decision support system* is used to monitor and estimate the link quality and make optimal decisions on when to deploy a new breadcrumb. Here the meaning of "optimal" is two-fold. First, the decision support system should be able to keep the packet reception ratio (PRR) of breadcrumbs above a predefined threshold. Second, it needs to avoid unnecessary breadcrumbs deployments, so as to efficiently use limited breadcrumbs to cover maximum distances.

Another key factor that needs to be taken into account while deciding when to deploy new breadcrumbs is the height effect. Since the dispenser (and the link estimator inside the dispenser) is normally placed at the waist of the firefighter, thus there is a gap between the estimated link quality and the actual link quality after the new breadcrumb is deployed on ground. For example, our experiments reveal that a new breadcrumb may fail to join the crumb chain even when PRR is 90 percent at the dispenser at that moment a breadcrumb is being dropped. Solutions must be proposed to eliminate this height effect and we call our solution the *height effect solver*.

After the new breadcrumb is deployed and joins the crumb chain, the link quality between this new crumb and its n neighbors may vary due to the dynamic impact from the environment. We propose an approach tailored to this situation: *adaptive power control*. More concretely, the newly deployed breadcrumb is able to adaptively increase its transmission power according to real-time link quality estimation so as to achieve more reliable link communication.

In summary, the combination of these four techniques provides a practical and optimized breadcrumb system to help firefighters communicate with base stations outside an on-fire building. Next, we introduce individual components of the system.

3.2 Redundancy Degree Optimization

Redundancy degree (RD) n refers to the number of redundant neighbors that each breadcrumb keeps in touch with at any moment. For example, if the dispenser always maintain "good" communications with at least three breadcrumbs, then the RD is set to be two. Previous works, such as [25], [27], only evaluated the situation in which the RD is zero, however, we argue that the RD must be some positive value to make the breadcrumb system practical in a harsh environment. On the other hand, over engineering the network by applying a very large RD is not desirable. Given a limited number of breadcrumbs in total, the system must efficiently use available resources to extend the transmission range as much as possible. Moreover, end-to-end delay time may suffer a lot due to continuous retransmission and received data packets are more likely to be corrupted. Frequent retransmission also leads to unnecessary energy consumption and shorter lifetime for breadcrumbs.

To represent the tradeoff between reliability and efficiency, we propose the following metric, α, to describe how

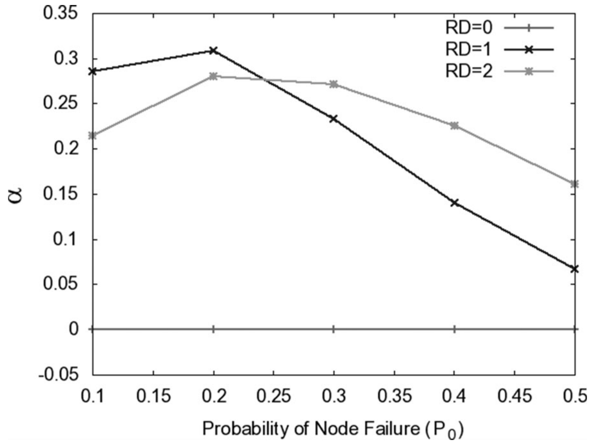


Fig. 2. Comparison of metric α with different RDs when $L = 10$, as p_0 varies from 0.1 to 0.5.

system reliability benefits/suffers as the RD various. Let n be the RD and L be the number of hops of the crumb chain, then α is defined as

$$\alpha = \frac{1}{n+1}\sum_{k=1}^{L} P(k) \cdot S(k), \tag{1}$$

in which $P(k)$ indicates the probability that k breadcrumbs are dead in the breadcrumb chain, and $S(k)$ is the probability that the breadcrumb chain can still maintain end-to-end connection when k breadcrumbs are dead. Thus the left side of the equation, α, represents the tradeoff between reliability and efficiency; it is defined by the ratio of system reliability gain to the efficiency degree, which is the right side. The system reliability gain is represented by the sum of probability to maintain communication links when node failure occurs, and the efficiency is measured by the number of breadcrumbs that a dispenser communicates with, which by definition is $n + 1$.

Let us assume that the event that each breadcrumb in the crumb chain fails are independent and identically distributed, and satisfies the regular binomial distribution with coefficient p_0. Then the function $P(k)$ becomes:

$$P(k) = \binom{L}{k} \cdot p_0^k \cdot (1 - p_0)^{L-k}. \tag{2}$$

Fig. 2 shows the comparison of different redundancy degrees in an example situation where the length of the crumb chain is ten. Metric α for different cases of p_0 are plotted. We observe that for a fixed redundancy degree, the metric α first goes up as p_0 increases to some extent, and then begins to drop asymptotically linear to p_0. In addition, the metric α for the $RD = 1$ case is around 30 percent better than that for the $RD = 2$ case when p_0 equals to 0.1. As p_0 increases, the differences become smaller and finally α for the $RD = 2$ case is better. This makes sense since when breadcrumbs are not vulnerable or fragile, it would be wasteful to use many redundant breadcrumbs. It clearly shows that when p_0 is less than or equal to 25 percent, setting the redundancy degree to one is the optimal tradeoff. We refer to this parameter selection as the *double-scout* *policy*. Finally, we assume independent failure model in our reliability analysis while in practice consecutive breadcrumbs may be destroyed due to harsh environmental realities like collapsed walls. This is an open problem and please refer to [21] for our discussions on this issue.

3.3 Decision Support System

As one of the most important components in the breadcrumb system, the decision support system monitors the link quality of all communications and determines when to deploy a new breadcrumb based on some predefined rules. The decisions it makes are extremely crucial to the system performance, since false-positive deployments (dropping too early) lead to decreased efficiency while false-negative deployments (dropping too late) result in poor end-to-end communication or even disconnections. Additionally, decisions must be made in time to represent dynamic change of link quality and to support a fast deployment process, so heavy-weight and time-consuming algorithms are not desirable in this situation. There are four categories of candidates based on the metric used to monitor link quality: Received signal strength indicator (*RSSI*), link quality indicator (*LQI*), signal-to-noise ratio (*SNR*), and packet reception ratio (*PRR*). Previous work [27] has shown that indicators such as LQI [9] and SNR are not suitable for breadcrumb systems.

A RSSI-based link quality monitor collects run-time RSSI values of received data packets and makes decisions using filtering techniques. The validity of this approach is proved by experimental results showing a stable RSSI-PRR mapping in indoor environments in stationary cases [26], [27]. However, we claim that the filtering approach using mean RSSI values in a sliding window used in previous works is inappropriate. The main reasons are that it considers all packets in the sampling window to have the same weight regardless of their temporal order, thus cannot represent dynamic link characteristics. Also, it is tricky to set an appropriate window size because of the accuracy-to-timeliness tradeoff. Finally, the accuracy is further decreased since lost packets are ignored. Instead, we observe that RSSI-related metadata may help construct a more accurate and efficient filter. The metadata includes max/min, median, deviation, and exponentially weighted moving average.

Based on the above analysis, we choose to exploit RSSI-based metadata and propose four candidate link quality estimators for the decision support system:

- *MEAN estimator*—This is used in previous work [27]. A new breadcrumb is deployed if the mean RSSI value of received packets in a sliding window with size N is below a threshold T_0.
- *EXP estimator*—Exponentially weighted moving average approach associates two parameters: current weighted value for RSSI Exp and weight coefficient β. Exp is updated when new data packets with RSSI value R arrive using the formula below:

$$Exp = (1 - \beta) \cdot Exp + \beta \cdot R. \tag{3}$$

- *RANGE estimator*—RANGE estimator makes use of the Max/Min value in the sliding window to detect false-positive or false-negative cases caused by noisy points.
- *Median estimator*—Another way to deal with noisy points is to use median value instead of mean, as used in many other scientific fields. The median estimator monitors the median RSSI value in a sliding window as well as RSSI deviation and drops new breadcrumbs in a similar way to other estimators.

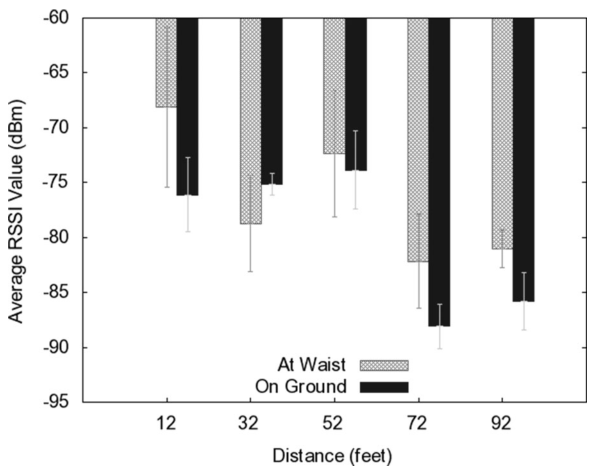


Fig. 3. Height effect (stationary) on the floor.

3.4 Height Effect Solver

Height effect refers to the gap between the estimated link quality at the dispenser's height (usually at the waist of the first responder) and the actual link quality at the crumb's height (on ground) after deployment. This is an important issue in practice in terms of reliability. For example, if the threshold of the decision support system is set to -85 dBm and there is a gap of 10 dBm due to the height effect, then the newly deployed breadcrumb is unable to join the crumb chain and the whole breadcrumb chain will be in trouble since the dispenser will then keep shooting out unhelpful breadcrumbs while the reliability becomes worse and worse. In this section, we investigate whether there is a consistent and constant degradation in link quality between an existing breadcrumb and the dispenser or the newly deployed breadcrumb.

To assess the height effect, we conducted a series of experiments using several breadcrumbs and a communication device we built. We first measure the degradation in link quality in stationary cases. One receiver is placed on the ground acting as the existing breadcrumb and one dispenser is hooked on the waist of a first responder. RSSI values between them are recorded while their distance varies from 10 to 90 feet. Experiments are repeated by placing the transmitter and receiver at different places on the floor to protect against the effects of fading. Then the transmitter is placed on the ground and the same experiments are conducted.

Fig. 3 shows the difference of RSSI with error bars for cases in which the transmitter and receiver are at different distances from one another. It can be observed that there is a 5 to 10 dBm degradation in most cases and the variance is around 10-20 percent. This indicates that applying a fixed offset on the original threshold may be a reasonable solution.

An alternative solution is to put an extra relay node at the ankle. This new node acts as link quality monitor and reports all results to the dispenser which is still at the waist. However, we argue that this approach is not desirable. First, this increases the overall complexity of the breadcrumb system and error propagation. Second, the communication between the new relay node and the dispenser becomes another tricky problem and suffers from problems like shadowing.

Based on above analysis, we propose a novel technique called adaptive threshold adjustment that solves the height effect problem. The principle behind this new approach is the temporal and spatial locality. The idea is to dynamically configure the offset that is applied to the threshold used in the decision support system, by recording the latest gap after a new breadcrumb has been deployed. For example, when the original threshold is set to -85 dBm and the current gap is 5 dBm at some moment, the actual threshold for deploying new breadcrumbs is then -80 dBm. A newly deployed breadcrumb then records the RSSI value as -88 dBm after it joins the crumb chain and sends this result to the dispenser. Finally, the dispenser updates the gap to be 8 dBm and the corresponding threshold for deploying the next breadcrumb as -77 dBm (calculated as $(-80)+((-85)-(-88))=(-77)$ dBm). We evaluate the performance of the adaptive threshold adjustment algorithm and compare it with other possible solutions such as applying fixed offsets in Section 5.

3.5 Adaptive Power Control

Adaptive power control is designed to handle link quality variation problems in harsh environments. APC further enhances the system reliability by enabling breadcrumbs to increase radio transmission power in the crumb chain when connection between two crumbs gets weak due to link quality variations. This is motivated by the fact that after a new breadcrumb is deployed and joins the crumb chain, the link quality between itself and the rest of the chain may satisfy the normal distribution centered with the threshold value determined by both the decision support system and the height effect solver. It is possible that if using the default transmission power, the new breadcrumb will be unable to maintain high quality link because of link dynamics or by being moved.

APC is a lightweight algorithm to: 1) make every node in a sensor network find the minimum transmission power levels that can provide good link qualities for its neighboring nodes, and 2) dynamically change the pairwise transmission power level over time as observed link quality varies. Through adaptive power control, we can maintain good link qualities between pairs of nodes with in-situ transmission power control. We evaluate how this approach helps optimize the crumb chain in Section 5.

Our implementation adaptive power control scheme works the same as in [20]: when a breadcrumb is deployed, it begins to estimate pairwise link qualities between its neighbors by monitoring the RSSI value of the received packets. If the RSSI value is higher than a "high set point", which is a predefined threshold to maintain reliable communication, a negative feedback message is sent to request its neighbor to decrease transmission power level by one. On the other hand, if the RSSI value is lower than a "low set point", then a positive feedback message is sent to request its neighbor to increase transmission power level by two. Note that currently the adaptive power control is only at the breadcrumb side and it includes the idea of topology control as breadcrumbs may increase their power level when their link quality with neighbors becomes weak.

4 Multi-User Coordination

In this section, we address the challenge associated with efficient and reliable automatic coordination among the dispenser systems carried by multiple first responders. This work is motivated by the fact that first responders are organized into small groups to execute different tasks and sometimes enter the building from several entrances simultaneously. Previous systems and algorithms do not fit into this situation and lead to suboptimal system resource (breadcrumbs) utilization as a result of inefficient breadcrumb deployments. One example is that a group of first responders are running along a hallway, in an uncoordinated scenario, the one at the head of the group drops breadcrumbs most of the time because his system usually detects decreased link quality first. Later, when this first responder takes another separate route by himself, he finds himself running out of breadcrumbs. On the other hand, simple coordination algorithms may not help increase deployment efficiency. For example, for fairness reasons, one simple algorithm requires the one with the most number of breadcrumbs in a group to deploy. In this case, breadcrumbs deployed by the one at the end of the group may not improve communication for the one at the very head of the group due to the distance between them, resulting in another request for breadcrumb deployment by the leading first responder very soon.

4.1 Baseline Algorithms

We first describe two baseline coordination algorithms. The first one aims to achieve complete fairness within groups during the deployment process. The key insight of this algorithm is the greedy approach, i.e., the user with most breadcrumbs always deploys a new node whenever there is a request within the group.

The greedy approach works as follows: when a group member decides to deploy a new breadcrumb according to the link monitoring algorithm, a request message is sent to the group leader, and then the leader accesses its list of group information and finds the user who has the most breadcrumbs. Finally, the leader sends a command to this user to deploy a new breadcrumb immediately.

However, the greedy algorithm may perform poorly in some scenarios. The reason is that the coordination process does not take into account the relative positions of first responders within the same group. If the one who finally deploys a new breadcrumb (the deployer) is far away from the requester, this deployment may not help too much since the requester will still suffer from weak wireless communication soon.

To overcome this inefficient coordination problem, one potential solution is to delay the dropping time. The deployer can choose to walk for a short time and then deploy the new breadcrumb at a place close to where the requester' system sent the requests, assuming they are walking in the same direction. More concretely, the deployer holds the breadcrumb for a while and deploys it when the link monitoring algorithm decides it is necessary for a new deployment. During this period, the requester uses other redundant links temporarily. We refer to this second baseline solution as a delay dropping algorithm.

However, this solution works under the assumption that moving information is a priori known within the group and group members stay close to each other during the coordination process. Moreover, it puts the requester at more risk during this delay period.

4.2 Utility Function Based (UF) Algorithm

Based on the insights from these baseline algorithms, we propose a third coordination algorithm which introduces the calculation of a utility function as the criteria for deploying new breadcrumbs.

The utility function based algorithm, *UF*, works as follows: The requester initiates the algorithm by broadcasting a help message. Then all of his neighbors send their information (number of breadcrumbs) to the requester; after a predefined timeout, the requester calculates the value of utility functions for each of its neighbors and sends a drop message to the one with the highest value to deploy a new breadcrumb.

The essential part of the algorithm is to define an appropriate utility function. A good utility function must precisely represent the tradeoffs between the gain of communication range extension to the requester/group and the cost of the breadcrumb counts to the deployer. The utility function for user i, denoted by $U(i)$, is defined as the weighted difference between a benefit function and a cost function, i.e.,

$$U(i) = \alpha \cdot B(i) - \beta \cdot C(i), \tag{4}$$

in which α and β and coefficients for the benefit function $B(i)$ and cost function $C(i)$, respectively. Without changing the final decisions, we use another variable $\gamma = \alpha/\beta$ and rewrite the formula as follows:

$$U'(i) = \gamma \cdot B(i) - C(i). \tag{5}$$

$B(i)$ represents the gain of communication link if first responder i deploys a new breadcrumb. This gain can be measured by either the $RSSI$ value for the requester himself or the average $RSSI$ value for both the requester and his neighbors. Thus it can be represented as:

$$B(i) = \frac{1}{n}\sum_{k=1}^{n} RSSI(i,k), \tag{6}$$

or

$$B'(i) = RSSI(i, req). \tag{7}$$

The tradeoff between these two functions $B(i)$ and $B'(i)$ is as follows. $B(i)$ considers all communication links in the group and takes the global optimal gain, but it requires each member to broadcast within the group, which increases both the communication overhead and the coordination delay. Moreover, the coordination delay may become even worse, since all group members are trying to broadcast simultaneously and thus schemes like random backoff timers have to be used. On the other hand, $B'(i)$ takes only a local optimal gain upon the requester, but it leads to much less communication overhead and shorter coordination delay. More concretely, assume there are n members in a group and perfect scheduling approaches are adopted such that packets can be sent one by one seamlessly. Also assume that broadcasting one packet takes t milliseconds. Then the communication overhead can be calculated as: $Overhead(B) =$ *one request message* $+\#$ *response message* $+\#$ *processed message* $= 1 + (n-1) + (n-1) = 2n-1$ and $Overhead(B') =$ *one request message* $+\#$ *response message* $= 1 + (n-1) = n$. Similarly, we have $Delay(B) = (2n-1)\cdot t +$ *calculation time* and $Delay(B') = n \cdot t$. We can see that the function $B(i)$ results in almost 100 percent more on both communication overhead and the coordination delay. Based on these reasons, we choose metric $B'(i)$ during the implementation and evaluation of our system.

To convert the $RSSI$ into a value for the calculation of $U(i)$, we propose the following approach: first, the time stamp when a predefined threshold of PDR is reached is set to be the minimum 0, and the time stamp when the experiment starts is set to be the maximum K. Then, we record the fraction of time μ when the exponentially weighted moving average ($EWMA$) value of $RSSI$ first reaches $RSSI(i, req)$ and calculate the corresponding value $k = \mu \cdot K$. The final value for the benefit function is then represented as the average of k for all 20 trials. We collected experimental data in traces in which the dispenser is moved far away from a breadcrumb on the floor until the connection is lost. To eliminate the effect of noise, we repeat the same experiment for 20 times, 5 in each different environments including hallway, corner, walking upstairs, and walking downstairs. Fig. 4 shows the results of PDR and $EWMA$ $RSSI$ in an example trace. To make our algorithm more general among different buildings, a linear approximation model is used to obtain the corresponding value $k = \frac{RSSI-(RSSI_{min})}{(RSSI_{max})-(RSSI_{min})} \cdot K$, here $RSSI_{min}$ stands for the minimum possible RSSI value, which for instance is -92 dBm for $CC2430$ radio [9].

$C(i)$ is the penalty for the number of remaining breadcrumbs for first responder i. A good cost function must take both relative ranking in the group and the absolute counts into consideration. The relative ranking is necessary since global information can provide important support for making decisions. The function of absolute counts, which grow exponentially as the counts decrease, is especially useful when most group members have only

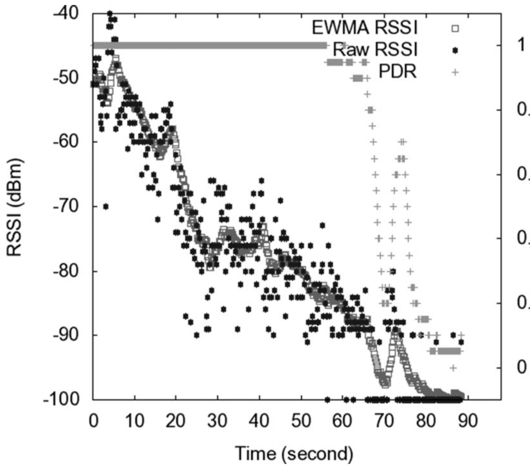


Fig. 4. PDR and $RSSI$ as time changes in an example trace.

a few breadcrumbs left after the system is running for a long time. Thus, taking both factors into account, we have

$$C(i) = R(i) + P(i), \tag{8}$$

in which $R(i) \in \{1, 2, 3, \ldots, n\}$ represents the ranking of group member i in the group in terms of breadcrumb counts and $P(i) = e^{n_0 - n}$. Here n_0 is a predefined threshold to indicate that this user has very few breadcrumbs and will be at high risk of running out of breadcrumbs soon. Fig. 5 shows how the value of $C(i)$ changes with the relative ranking and absolute counts. The number of users in the group is set to be 10, the initial number of breadcrumbs is 10 for each user, and the threshold for absolute counts is 4. We can observe from the graph that, relative ranking and absolute counts each affect the $C(i)$ value in a different way. $R(i)$ provides a constant increasing part to $C(i)$, $P(i)$ does not change much when there are plenty of remaining breadcrumbs, but as the counts decrease to the threshold, $P(i)$ starts to be the dominating factor for the total value of $C(i)$.

In summary, the utility function can be written as follows:

$$U(i) = \frac{RSSI - (RSSI_{min})}{(RSSI_{max}) - (RSSI_{min})} \cdot K - R(i) - e^{n_0 - n}. \tag{9}$$

5 Implementation and Evaluation

We use a 2.4 GHz based customized automatic deployment prototype (as shown in Fig. 6) for our experiments. An automated dispenser (displayed at the top) can contain five quarter-size breadcrumbs. The reason breadcrumbs are designed to be square-shaped is that round nodes are more prone to roll around when dropped, and leads to an unreliable breadcrumb chain. When the automatic decision support system determines that it is time to deploy a new breadcrumb, the turntable rotates until one breadcrumb is dropped out from a hole at the bottom of the dispenser, which allows for fast and automated deployment in real time. The dispenser is powered by two AA-size batteries and breadcrumbs are powered with a standard 3-Volt, 560 mAh lithium battery cell from Panasonic. The battery is capable of running each breadcrumb at full power for 3 hours. The dispenser and the breadcrumbs can communicate with each other via their $CC2430$ series radio chip. Finally, data packets are sent back to the incident commander (an IBM $T61$ laptop) via a USB-ported base station (displayed at the bottom left). Please refer to [21] for our discussions on using 2.4 GHz based hardware instead of lower frequency like 900 MHz.

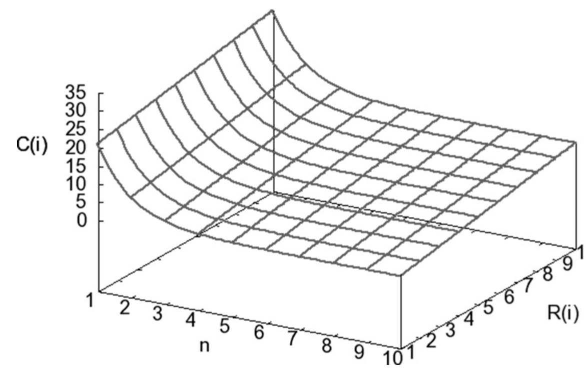


Fig. 5. $C(i)$ as $R(i)$ and $P(i)$ change.

5.1 Exploiting Reliability-Efficiency Tradeoff

We first evaluate our proposed reliability model. The results for individual components are skipped due to page limits. Please refer to [21] for more details.

To exploit the tradeoff between reliability and efficiency in our system, a series of experiments are conducted in the building of Computer Science Department of University of Virginia. The base station is connected to a laptop and located outside one entrance of the first floor of the building and the first responder walks along the path to the third floor. One user takes the dispenser with breadcrumbs inside and enters the building. The decision support system monitors the wireless link health and decides when to deploy a new breadcrumb. A new breadcrumb is automatically dropped out of the dispenser when necessary and begins to relay data packets to the base station.

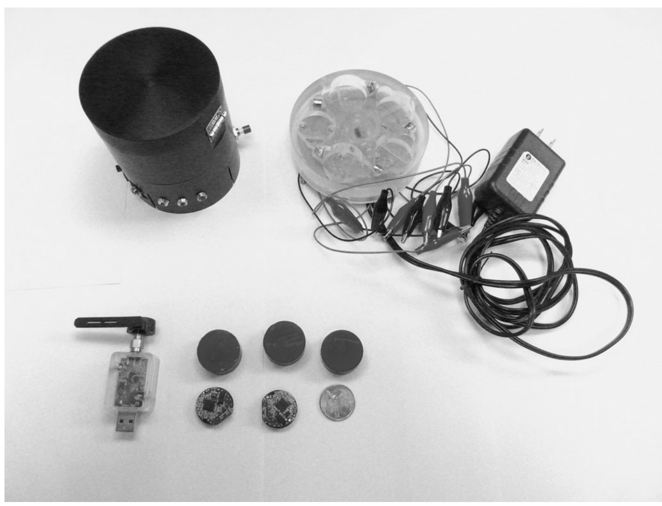


Fig. 6. Prototype of breadcrumb system.

TABLE 1
Number of Breadcrumbs Dropped

Trial number	1	2	3	4	5
Our work	13	14	13	15	15
NIST [27]	10	12	12	12	11

The double scout algorithm is used and two breadcrumbs are deployed at the start of the trace to initialize the Zigbee network. All breadcrumbs are placed in containers to protect against the simulated harsh environments. According to our experiments, these plastic boxes (non-conducting material) do not attenuate radio waves significantly.

Along the trace, the dispenser sends out request messages periodically at the rate of five packets per second in order to get responses from "active" breadcrumbs. Link quality information is then recorded according to the identity of breadcrumbs. Note that we did not try to find out the optimal rate for sending request messages, since this optimal value may be application-specific and thus does not have a general answer. Moreover, the battery life of the nodes exceeds the needed lifetime of the network in our experiments.

We integrate the double scout algorithm with the *EXP* filter based decision support system and adaptive height adjustment, and compare our system to the approach in [27]. The parameters used in [27] are the same as used in our paper, including mobile probe period 100 ms, averaging filter length 20, RSSI threshold -77 dBm (-92 dBm minimum value plus 15 dBm offset), and redundancy degree 0. Note that the RSSI threshold is adjusted for 2.4 GHz hardware. Parameters used in our system includes the results for *EXP* filter [21]. Each case is repeated five times.

Table 1 shows the results for number of breadcrumbs dropped in the trace. We observe that the average number of breadcrumbs used is 14 in our approach and 11.4 in the NIST work, which indicates that we achieve 200 percent link redundancy at the expense of 23 percent additional nodes. This is mainly because of the filter selection for the decision support system as well as the adaptive height control methods. In the NIST work, the *MEAN* filter is more likely to result in late dropping, so a more conservative threshold must be set in order to maintain high PRR, therefore the average number of breadcrumbs increases. Furthermore, they use a fixed offset to deal with height effect and the offset has to be set conservatively too.

The logical network topology along with average PRR for Trial 1 in our work is shown in Fig. 7. It is clear that 11 out of 12 one-hop connections achieve more than 95 percent PRR and even three-hop wireless links exist (1-4, 3-6, and 5-8). We also observe that the PRR is only 69 percent between breadcrumbs 4 and 6 and 67 percent between 8 and 10. The main reason lies in the consecutive corners on the third floor and the metal wall near the stairway of the first floor, which implies that complex environments may have a big impact on link quality.

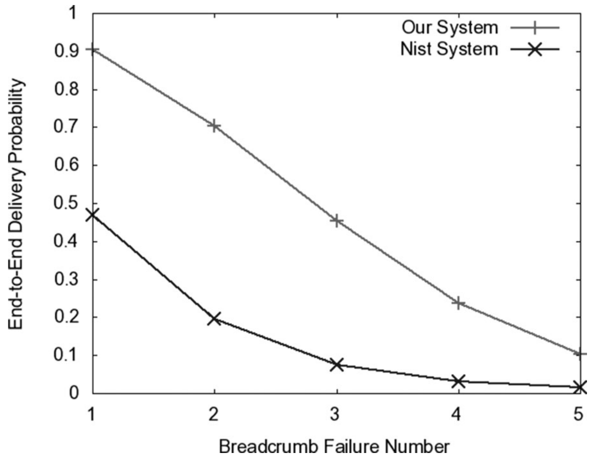


Fig. 8. System robustness to breadcrumb failures.

Finally, we compare our work with [27] in terms of system robustness when breadcrumb failure occurs. Trial 1 in both cases are selected and the results are shown in Fig. 8. We observe that our system can still achieve 90 percent PRR when one breadcrumb fails, while the NIST system has less than 50 percent. Similar trends can be observed when more breadcrumbs fail, which implies that our system achieves better robustness than previous work.

5.2 Multi-User Coordination

To evaluate our proposed *UF* algorithm, we conducted groups of experiments in the Computer Science Building at the University of Virginia. We compare UF to the two baseline algorithms described in Section 4.1: the greedy algorithm (Greedy) and the delay dropping approach (DD). Nine predefined branch points are used to record the experiment traces as shown in Fig. 9 (Points 1, 4, 7, and 9 are entrances/exits due to hilly environments), and users divide into subgroups or merge to a larger group at some of these branch points. The experiments involve four users in total, denoted by A, B, C, and D. Users walk in a zigzag style at normal walking speed. The three traces that they walked through are listed below.

1. *Branching and merging (BM).* This is to simulate the searching application. $ABCD$ all enter the building from branch point 1, and divide into two groups: $AB: 1 \rightarrow 2 \rightarrow 3$ and $CD: 1 \rightarrow 3$. $ABCD$ merge at 3 and walk through $3 \rightarrow 6$, and divide again: $AC: 6 \rightarrow 5 \rightarrow 8$ and $BD: 6 \rightarrow 9 \rightarrow 8$. Then $ABCD$ merge at 8, walk through $8 \rightarrow 7$, and leave the building.

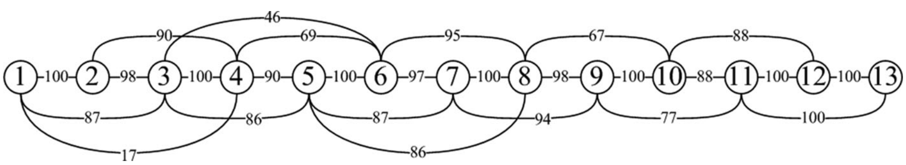


Fig. 7. Logical topology for Trial 1 using our system.

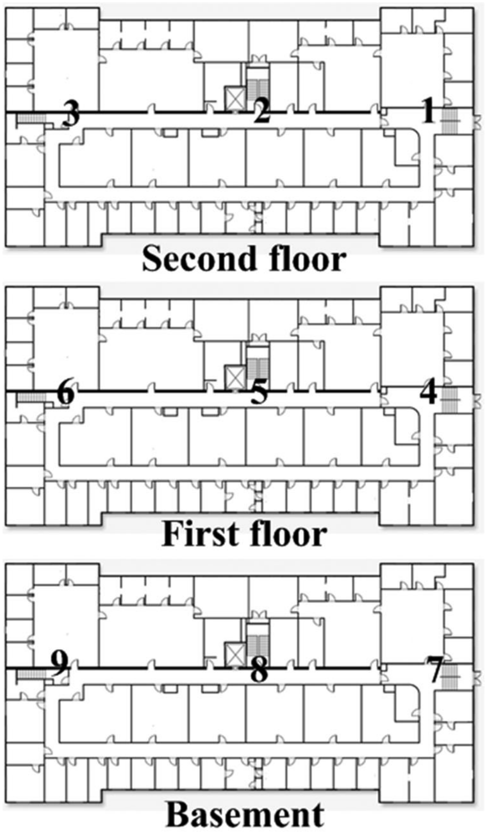


Fig. 9. Nine branching points in the computer science building.

2. *Single rescue point (SRP).* This is to simulate the rescuing application with a single rescue point. $ABCD$ enter the building from two different entrances 1 and 6, and the rescue point is 8. First, they walk through the trace: $AB: 1 \rightarrow 2 \rightarrow 5$ and $CD: 6 \rightarrow 5$. Then $ABCD$ merge at 5 and walk through $5 \rightarrow 8 \rightarrow 9$ and leave the building.
3. *Peeling off one by one (PEEL).* This is to simulate the rescuing application with multiple rescue points. $ABCD$ all enter the building from branch point 1, and walk through $1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 9$. At each branch point 2, 5, and 8, one user will leave the team and search via a different route.

During the experiments, users walked along the predefined traces and breadcrumb trails were automatically established. Multi-hop communication is then applied to transmit useful data packets to the base station. The redundancy degree is set to 1 and exponentially weighted moving average was adopted to guide when a new breadcrumb is needed, and the parameters are set to be the optimal value that results in the least probability of dropping late while maintain a low Least Square value: the weight β is 0.0313 and dropping threshold is -81.8 dBm. The timer used for waiting for responses from neighbors in *UF* is set to 1 second. For simplicity, the height effect is solved by a fixed offset 10 dBm. Along the trace, the dispenser sends out request messages periodically at the rate of five packets per second in order to get responses from "active" breadcrumbs. Link quality information is then recorded according to the identity of breadcrumbs.

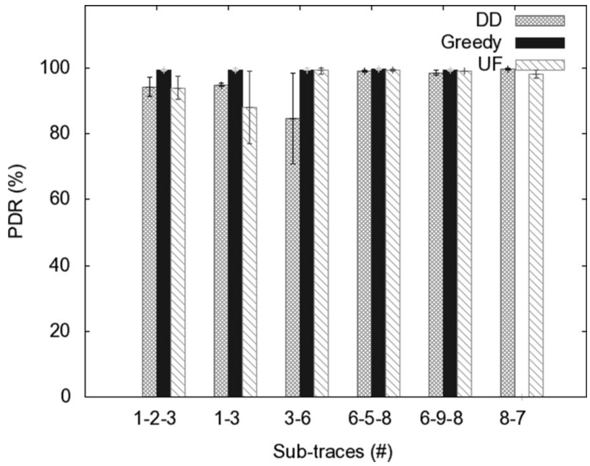


Fig. 10. Comparison of PDR between *DD*, *Greedy* and *UF* in the *BM* trace.

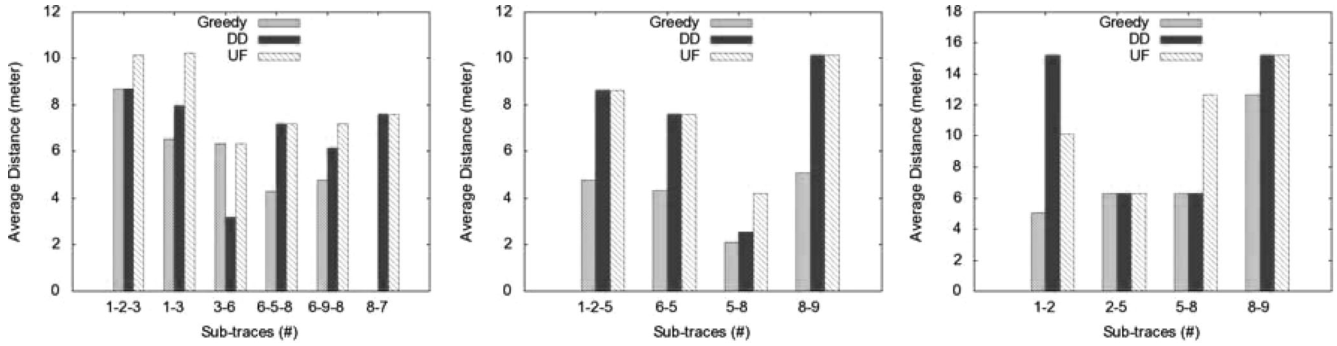
Physiological data are sent from the dispensers to active breadcrumbs at the rate of two packets per second. Due to the spatial locality, the synchronization message for group management is sent once per 2 seconds. For performance analysis purposes, in each data packet we included information such as times tamp and source node ID. Upon receiving the data packet, the intermediate breadcrumbs recorded this information in their own flash memory. Zigbee techniques [11] are used for the networking layer protocol during the experiments. To eliminate the effect of random noise, experiments were repeated five times when evaluating the reliability and the coordination delay of candidate algorithms and we found that the results have little variations. Unless stated otherwise, we used the above default values in all the experiments.

5.2.1 Reliability

To investigate whether the candidate coordination algorithm leads to a high PDR breadcrumb system, a group of indoor experiments were conducted. We attached sequence numbers to data packets for statistical purposes and recorded the PDR when running *Greedy*, *DD*, and *UF*. Due to page limitations, we selected the most complex BM trace as the experiment environment and recorded the PDR for each subtrace. The PDR for each user is recorded separately so as to see the variance.

Fig. 10 compares the average PDR with error bars of each subtrace when running *Greedy*, *DD*, and *UF*. We observe that 15 out of totally 18 bars achieve more than 95 percent PDR, which indicates that all three coordination algorithms lead to a high PDR. Particularly, *UF* achieves an average 96.3 percent PDR for all users in all subtraces, more concretely, 94, 88, 99.4, 99.3, 99, and 98 percent, respectively.

We also observed that sometimes people shadowing is a big factor in packet loss. This occurs when multiple users are walking in a narrow environment simultaneously and one stands in between the breadcrumbs and another user. As shown in Fig. 10, subtrace $3 \rightarrow 6$ (stairway from the third floor to the second floor, 4 users) when running *DD* only results in an average PDR 84.6 percent and the minimum PDR is only 70.7 percent.



(a) BM trace. (b) SRP trace. (c) PEEL trace.

Fig. 11. Average distances between breadcrumbs for sub-traces.

5.2.2 Efficiency

To compare the efficiency of candidate coordination algorithms, precise locations of deployed breadcrumbs in the previous experiments were recorded and analyzed. As shown in Fig. 11, due to the inefficient coordination mechanism, *Greedy* produces only 5.10, 4.08, and 9.82 meters in terms of average distances in three traces. *UF*, however, achieves 7.76, 7.64 and 11.09 meters, which is 52.2, 87.3, and 12.9 percent better than *Greedy*. This is because in *UF*, both breadcrumb costs and link quality gains are taken into account, situations in which the deployer is far from the requester rarely happen. *DD* has an average distance of 6.79, 7.22, and 10.77 meters, which is close to the performance of *UF*. This is reasonable since the delay dropping process allows the deployer to carry the breadcrumb for a while before dropping it, and thus extends the average distance between breadcrumbs. However, *DD* relies on the assumption that users are walking towards the same direction with the same speed all the time, and they cannot stop or go backwards during the process. *UF* does not have these impractical constraints and performs well via the well-designed utility function, i.e., considering both benefit function and cost function, and taking both relative ranking and absolute counts into account when calculating the cost function.

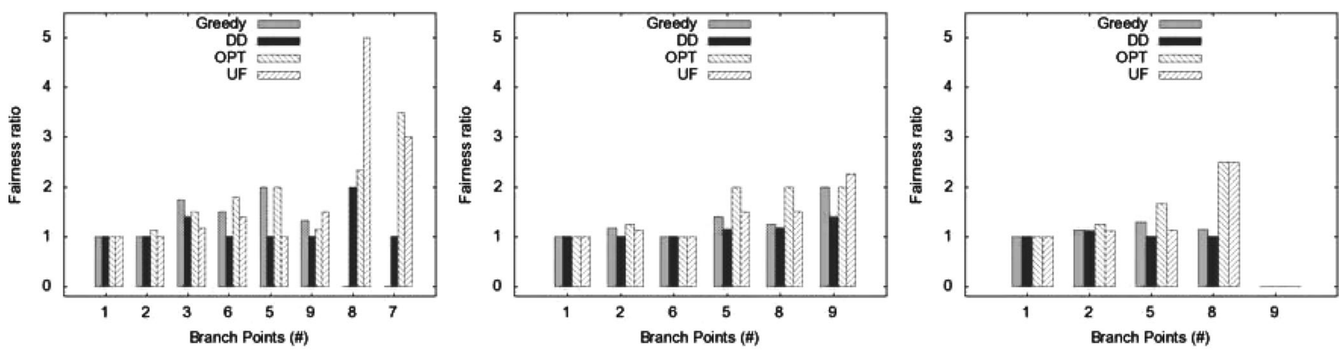
5.2.3 Fairness

We proceed to compare the performance of coordination algorithms in terms of fairness. The fairness metric is represented by the fairness ratio, defined as the maximum number of breadcrumbs within the group divided by the minimum number. Therefore, ideally this ratio should always be close to one to achieve complete system fairness. as the fairness ratio increases, it indicates that the group is not well balanced which may result in potential situations in which some group members have many breadcrumbs left while others only have few.

As shown in Fig. 12a, the fairness ratio is 1s for all candidates at the start since each user carries 10 breadcrumbs. As they go though the trace, *Greedy* and *DD* keep the ratio between 1 and 1.75 since they both adopt a round-robin style strategy. On the other hand, the fairness ratio of *UF* keeps increasing and reaches as high as 5. In *OPT*, since each user is assumed to have infinite number of breadcrumbs, the fairness ratio does not tell much. Also note that the value is set to 0 for branch points 8 and 7 since at least one user has no remaining breadcrumbs. Similar trends can be observed in the other two traces, as shown in Figs. 12b and 12c, respectively. The value is set to 0 at branch point 9 in Fig. 12c because only User A is walking through this subtrace. The results are mainly because that *UF* utilizes group resources efficiently from the global optimization point of view. Considering the group as a whole, *UF* always tries to select the right user that will provide the most contribution to the group by deploying a new breadcrumb instead of the one with most number of breadcrumbs.

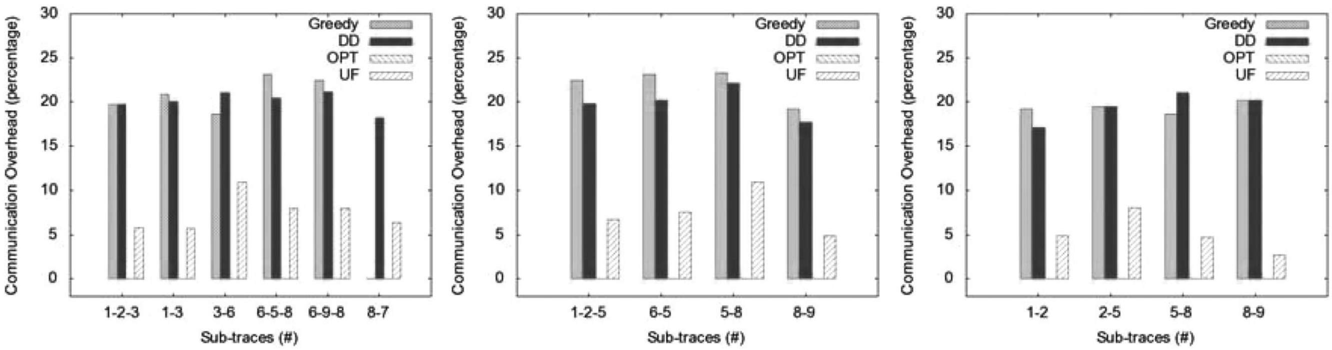
5.2.4 Communication Overhead

Last we compare the communication overhead between *UF* and other baseline algorithms. Note that *OPT* does not have this metric because no coordination is processed in this



(a) BM trace. (b) SRP trace. (c) PEEL trace.

Fig. 12. Fairness ratio for different coordination algorithms.



(a) BM trace. (b) SRP trace. (c) PEEL trace.

Fig. 13. Communication overhead for different coordination algorithms.

algorithm. *DD* has the same overhead with *Greedy* since they have exactly the same coordination algorithm except the deployment time, and this "delay" period does not introduce any extra packets.

The communication overhead of *Greedy* can be calculated as follows: Given a fixed time period of T seconds, and denote the number of users and average distance between consecutive breadcrumbs by n and D, respectively. Packets transmitted by dispensers inside the network in this period include: (a) physiological data are two (packets per second) $\cdot$ T (second) $\cdot$ $n = 2nT$; (b) link monitoring algorithm query messages are five (packets per second) $\cdot$ T (second) $\cdot$ n $\cdot$ (1 req + 1 ack) $= 10nT$; (c) coordination messages are $\frac{T}{D \cdot velocity} \cdot$ (1 req + 1 drop) $= 10T/D$; and (d) group management messages are $\frac{T(second)}{2(time/second)} \cdot (1 \text{ req} + (n-1) \text{ update}) = nT/2$. Thus, the percentage of communication overhead for *Greedy* can be calculated as: $\frac{(c)+(d)}{(a)+(b)+(c)+(d)} = \frac{1+0.2nD}{1+1.25nD}$. Similarly, the percentage of communication overhead for *UF* can be calculated as: $\frac{(c)}{(a)+(b)+(c)} = \frac{5(n+1)}{5(n+1)+12nD}$.

Fig. 13 displays the percentage of communication overhead in all subtraces for each trace when running different coordination algorithms. It is easy to see in all traces that, *Greedy* and *DD* consume around 20 percent of data packets for group management and coordination; while *UF* only has between 5 and 10 percent of overhead. More concretely, the average communication overhead for *Greedy* is 20.95, 22.00, and 19.38 percent in three traces, respectively. Since there is no need for group management, *UF* only consumes 7.49, 7.56, and 5.08 percent of overhead. These results imply that *UF* results in a 64.2-73.8 percent decrease of overhead compared with *Greedy*.

5.3 Outdoor Experimental Results

Up to now all experiments presented in this section are conducted in an indoor environment, but our breadcrumb system design is general enough to be applied to outdoor disaster and wild area exploring applications. To investigate the system performance in outdoor environments, we repeated some experiments in a park near the South Lake Union in Seattle, Washington. Two users, each carry a five-breadcrumb dispenser, walk together through a long route until both of them run out of breadcrumbs. The packet delivery ratio and the overall range extension were recorded. All parameters were the same as we used in Section 5.1. Table 2 shows the experimental results of the PDR for each user and overall range extension in each of the three trials. We can see that the packet delivery ratio for both users are above 90 percent on average. The average range extension is 87 meters in three trials, since the total number of deployed breadcrumbs is 10 in each trial, the average distance between breadcrumbs is 8.7 meters. These results provide initial evidence that the proposed breadcrumb system works in outdoor environments.

TABLE 2
Outdoor Experimental Results

Trial	FR1 PDR	FR2 PDR	Range Extension (meters)
1	95.59%	91.18%	84
2	100%	96.43%	92
3	79.07%	83.72%	84

6 Conclusion

We have presented a new multi-user breadcrumb system deployment scheme that supports automatic and robust deployment of breadcrumbs for in-door firefighter applications. The system is composed of five components: redundancy degree optimization, decision support system, height effect solver, adaptive power control, and multi-user coordination. Experimental results show that compared to the state of the art work [27], our designed system achieves better reliability-efficiency tradeoff, is more robust to breadcrumb failures, and can recover from unreliable wireless links. In addition, our proposed *UF* coordination algorithm achieves longer distances than baseline greedy coordination approach while maintaining high packet delivery ratio.

Acknowledgments

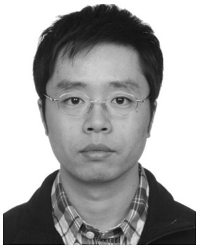
This work was performed to support a Phase II SBIR on Wireless Body Area Networks funded by the Science and Technology Directorate of the Department of Homeland Security. This work was also supported, in part, by the US National Science Foundation (NSF) Grants EECS-0901686 and CSR-0720640. Hengchang Liu is the corresponding author.

References

[1] Fire Project at Berkeley, http://fire.me.berkeley.edu, 2014.
[2] Report of National Fire Protection Association, http://www.nfpa.org/, 2005.

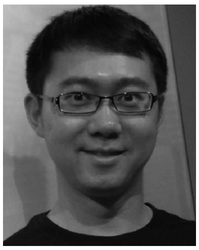
[3] P25 Projects, http://www.project25.org/, 2014.
[4] *Indianapolis Police and Fire Digital P25 Radio Problems*, http://blog.tcomeng.com/index.php/2011/what-we-have-is-a-failure-to-communicate-la-rics/, 2014.
[5] *Motorola P25 Problems in Indiana*, http://blog.tcomeng.com/index.php/2008/motorola-p25-problems-in-indiana/, 2014.
[6] "Indianapolis Police Worried about New Digital Radio System," http://www.youtube.com/watch?v=RrBJ8wmLCs4, 2014.
[7] "Indianapolis Police and Fire Digital P25 Radio Problems," http://www.youtube.com/watch?v=8yXFSsp1CMg, 2014.
[8] "Small Business Innovation Research, the DHS," https://sbir2.st.dhs.gov/, 2014.
[9] "ChipCon CC2430 Datasheet," http://www.alldatasheet.com/datasheet-pdf/pdf/241043/TAOS/CC2430.html, 2014.
[10] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Standard 802.15.4, http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf, 2003.
[11] ZigBee Alliance, http://www.zigbee.org, 2014.
[12] X. Cheng, D. Du, L. Wang, and B. Xu, "Relay Sensor Placement in Wireless Sensor Networks," *IEEE Trans. Computers*, vol. 14, no. 3, pp. 347-355, June 2001.
[13] C. Gentile and A. Kik, "A Comprehensive Evaluation of Indoor Ranging Using Ultra-Wideband Technology," *EURASIP J. Wireless Comm. and Networking*, vol. 2007, p. 12, 2007.
[14] C. Gentile, A. Braga, and A. Kik, "A Comprehensive Evaluation of Joint Range and Angle Estimation in Indoor Ultrawideband Location Systems," *EURASIP J. Wireless Comm. and Networking*, vol. 2008, article 36, 2008.
[15] S. Halder and W. Kim, "Adaptive Filtering for Indoor Localization Using Zigbee RSSI and LQI measurement," *IEEE Trans. Systems, Man, and Cybernetics—Part C: Applications and Rev.*, 2011.
[16] X. Jiang, N.Y. Chen, J.I. Hong, K. Wang, L.A. Takayama, and J.A. Landay, "Siren: Context-Aware Computing for Firefighting," *Proc. Second Int'l Conf. Pervasive Computing*, 2004.
[17] M. Klann, T. Riedel, H. Gellersen, C. Fischer, M. Oppenheim, P. Lukowicz, G. Pirkl, K. Kunze, M. Beuster, M. Beigl, O. Visser, and M. Gerling, "Lifenet: An Ad-Hoc Sensor Network and Wearable System to Provide Firefighters with Navigation Support," *Proc. Ninth Int'l Conf. Ubiquitous Computing (Ubicomp)*, 2007.
[18] M. Klann and M. Geissler, "Experience Prototyping: A New Approach to Designing Firefighter Navigation Support," *Proc. IEEE Pervasive Computing*, vol. 11, no. 4, pp. 68-77, Oct. 2012.
[19] M. Klann and M. Geissler, "How It May Feel: Making Firefighters Experience Future Support for Tactical Navigation,"*IEEE Pervasive Computing*, 2012.
[20] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J.A. Stankovic, "ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks," *Proc. Fourth Int'l Conf. Embedded Networked Sensor Systems (SenSys)*, 2006.
[21] H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, D. Siu, and J.A. Stankovic, "Automatic and Robust Breadcrumb System Deployment for Indoor Firefighter Applications," *Proc. ACM Eighth Ann. Int'l Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.
[22] H. Liu, Z. Xie, J. Li, S. Lin, K. Whitehouse, D. Siu, and J.A. Stankovic, "Efficient and Reliable Breadcrumb Systems via Coordination among Multiple First Responders," *Proc. IEEE 22nd Ann. Int'l Symp. Personal, Indoor and Mobile Radio Comm. (PIMRC)*, 2011.
[23] H. Liu, Z. Xie, J. Li, J. Stankovic, P. Hui, and D. Siu, "Intentional Forwarding: Providing Reliable and Real-Time Delivery in the Presence of Body Shadowing in Breadcrumb Systems," *Proc. IEEE 23rd Int'l Conf. Personal, Indoor and Mobile Radio Comm. (PIMRC)*, 2012.
[24] H. Liu, "Design and Implementation of an Automatic, Reliable, and Efficient Breadcrumb Sensor Network," PhD dissertation, Univ. of Virginia, 2011.
[25] M. Refaei, N. Moayeri, and M. Souryal, "Interference Avoidance in Rapidly Deployed Wireless Ad Hoc Incident Area Networks," *Proc. IEEE INFOCOM*, 2008.
[26] M.R. Souryal, J. Geissbuehler, L.E. Miller, and N. Moayeri, "Link Assessment in An Indoor 802.11 Network," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC)*, 2006.
[27] M.R. Souryal, J. Geissbuehler, L.E. Miller, and N. Moayeri, "Real-Time Deployment of Multihop Relays for Range Extension," *Proc. Fifth Int'l Conf. Mobile Systems, Applications and Services (MobiSys)*, 2007.
[28] K.Srinivasan, P. Levis, "RSSI Is Under-Appreciated," *Proc. Third Workshop Embedded Networked Sensors (EmNets)*, 2006.
[29] J. Wilson, D. Steingart, R. Romero, J. Reynolds, E. Mellers, A. Redfern, L. Lim, W. Watts, C. Patton, J. Baker, and P. Wright, "Design of Monocular Head-Mounted Displays for Indoor Firefighting Safety and Efficiency," *Proc. SPIE*, vol. 5800, pp. 103-114, 2005.
[30] A. Wapf and M.R. Souryal, "Measuring Indoor Mobile Wireless Link Quality," *Proc. IEEE Int'l Conf. Comm. (ICC)*, 2009.
[31] D. Yang, S. Misra, X. Fang, and J. Zhang, "Two-Tiered Constrained Relay Node Placement in Wireless Sensor Networks," *Proc. IEEE Seventh Comm. Soc. Conf. Sensor, Mesh, and Ad-Hoc Comm. and Networks*, 2010.
[32] W. Zhang, G. Xue, and S. Misra, "Fault-Tolerant Relay Node Placement in Wireless Sensor Networks," *Proc. IEEE INFOCOM*, 2007.

Hengchang Liu received the PhD degree from the University of Virginia in 2011 under the supervision of Professor John Stankovic. He is currently an assistant professor at USTC. His research interests mainly include cyber physical systems, mobile systems, named data networking, and wireless (sensor) networks.



Zhiheng Xie is currently working toward the PhD degree at the University of Virginia under the supervision of Professor John Stankovic. His research interests mainly include localization algorithms and cyber physical systems.

Jingyuan Li received the master's degree from the University of Virginia in 2011 under the supervision of Professor John Stankovic. He currently works at Microsoft in Seattle. His research interests mainly include cyber physical systems and self-healing systems.



Shan Lin received the PhD degree in computer science under the guidance of John Stankovic at the University of Virgina in 2010. He is an assistant professor in the CIS Department, Temple University. His primary funded research interests include the areas of cyber physical systems, networked information systems, and wireless sensor networks. In recent years, he has published papers in major conferences and journals in these areas. He is currently a PI of three US National Science Foundation (NSF) research grants.

David J. Siu received the master's degree from Stanford University in 1998. He is currently an electrical engineer in Oceanit Inc. Honolulu, Hawaii. His research interests mainly include energy metering and wireless body area networks.

Pan Hui received the PhD degree from the University of Cambridge under the supervision of Prof. Jon Crowcroft. He is currently with the Department of Computer Science and Engineering, Hong Kong University of Science and Engineering, and Telekom Innovation Laboratories, Berlin, Germany. His research interests mainly include mobile augmented reality, mobile computing, and other interesting topics related to social network.

Kamin Whitehouse received the BA and BS degrees from Rutgers University and the MS and PhD degrees from the University of California Berkeley. He currently works in the area of embedded systems, and is an associate professor in the Computer Science Department at the University of Virginia. He was awarded a CAREER Award by the US National Science Foundation (NSF).

John A. Stankovic received the PhD degree from Brown University. He is the BP America professor in the Computer Science Department at the University of Virginia. His research interests include real-time systems, distributed computing, wireless sensor networks, and cyber physical systems. He served as the chair of the department for eight years. He won the IEEE Real-Time Systems Technical Committee's Award for Outstanding Technical Contributions and Leadership. He also won the IEEE Technical Committee on Distributed Processing's Distinguished Achievement Award (inaugural winner). He is a fellow of both the IEEE and the ACM.