Automatic and Robust Breadcrumb System Deployment for Indoor Firefighter Applications

Hengchang Liu
Computer Science Dept.
University of Virginia
Charlottesville, VA, USA
hl4d@cs.virginia.edu

Jingyuan Li
Computer Science Dept.
University of Virginia
Charlottesville, VA, USA
jl3sz@cs.virginia.edu

Zhiheng Xie
Computer Science Dept.
University of Virginia
Charlottesville, VA, USA
zx3n@cs.virginia.edu

Shan Lin
Computer Science Dept.
University of Virginia
Charlottesville, VA, USA
shanlin@cs.virginia.edu

Kamin Whitehouse
Computer Science Dept.
University of Virginia
Charlottesville, VA, USA
whitehouse@cs.virginia.edu

John A. Stankovic
Computer Science Dept.
University of Virginia
Charlottesville, VA, USA
stankovic@cs.virginia.edu

David Siu
Science and Tech. Division
OCEANIT
Honolulu, HI, USA
dsiu@oceanit.com

ABSTRACT

Breadcrumb systems (BCS) have been proposed to aid firefighters inside buildings by communicating their physiological parameters to base stations outside the buildings. In this paper, we describe the design, implementation and evaluation of an automatic and robust breadcrumb system for firefighter applications. Our solution includes a breadcrumb dispenser with an optimized 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.

Experimental results from our study show that compared to the state of the art solution [14], our breadcrumb system achieves 200% link redundancy with only 23% additional deployed nodes. Our deployed crumb-chain can achieve 90% probability of end-to-end connectivity when one node fails in the crumb-chain and over 50% probability of end-to-end connectivity when up to 3 nodes fail in the crumb-chain. In addition, by applying adaptive transmission power control at each node after the crumb-chain deployment, we solve the link quality variation problem by avoiding significant variations in packet reception ratio (PRR) and maintain PRR of over 90% at the link level.

MobiSys'10, June 15–18, 2010, San Francisco, California, USA.

Categories and Subject Descriptors

C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based Systems; C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*wireless communication*

General Terms

Applications, Algorithms, Design, Experimentation

Keywords

Breadcrumb systems, firefighter, sensor networks, reliability, optimization

1. INTRODUCTION

Firefighter safety is a critical issue especially when fighting fires in large buildings. Monitoring physiological parameters such as heart rate and stress of these first responders in real-time can save many lives [5]. However, reliably transmitting this physiological data to a base station outside the building is a challenging problem. Existing solutions normally use one-hop communications and suffer from limited transmission range since it is sometimes difficult for wireless signals to travel through complex infrastructures. 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 [6] which allows a firefighter 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 firefighters are or where events occur.

In current breadcrumb systems, while the research focus is on the feasibility of an automated dispensing process, to date *ALL* prototypes built require manual deployment. This interferes with the firefighters' main tasks and also takes longer to deploy than a completely automated solution. Since firefighters 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 bread crumbs available, we address the problem of finding an optimized deployment scheme that minimizes the number of bread crumbs while maintaining high system reliability. The main contributions of this work are:

- We propose an automatic and robust breadcrumb system design and build a prototype system using 2.4 GHz based hardware (see Figure 1 for dispenser (1) and breadcrumbs (2 and 3)). To our best knowledge, this is the first prototype system that implements a *real automated deployment* process for breadcrumb systems.

- We investigate the *optimal redundancy degree* that should be applied in the system in terms of tradeoffs between reliability and efficiency. Results from theoretical analysis show that when the probability of physical node failure is less than 25%, maintaining two links at any time is the best choice.

- We compare the performance of various kinds of link quality estimators in order to make the most timely decision of when to deploy a new breadcrumb. Experimental results reveal that the exponentially weighted moving average approach outperforms other candidates in terms of avoiding dropping too late.

- We propose a novel *adaptive height-effect solver* and compare it to existing solutions. It is shown that our approach is more efficient in utilizing breadcrumbs while maintaining high system reliability.

- To maintain reliable communication after deployment we use an *adaptive power control* scheme where transmission power is dynamically adjusted to track dynamic environmental changes.

The remainder of this paper is organized as follows. We compare our work with state of the art in Section 2 and present the system requirements in Section 3. The detailed system description is presented in Section 4. The implementation and evaluation for our system are discussed in Sections 5 and 6, respectively. Finally, we conclude the paper in Section 8.

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

2. STATE OF THE ART

Although firefighter sensor systems are an active area of research, the challenges of designing such reliable, efficient, and automated platforms have only been explored partially. This type of system normally includes two parts [11]: Firefighters automatically deploy sensor nodes along their paths, effectively establishing an ad-hoc infrastructure for positioning, sensing and communication; and then firefighters 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 design mostly focused on the second part, by designing various kinds of wearable components that can be conveniently carried by firefighters. For example, the *FIRE* project [2] aims at designing new technologies such as small head-mounted displays (HMDs) for firefighting, and conducting experiments and exploratory research with firefighters. It basically includes three sub-projects: *SmokeNet* to design pre-deployed WSN for detecting fires; *FireEye* to equip firefighters with head mounted display units; and *eICS* to provide visual display showing resource allocation, personnel location and firefighter biometrics. Similarly, the *SIREN* project [10] provides reliable communication among firefighters using a WiFi-enabled PDA with a built in mote. The mote collects data from motes which are pre-deployed in the building to inform the firefighter of hazards and immediate danger. Pre-deployed motes also serve as location beacons that allow firefighters to navigate through the building. Other similar systems include *LIFENET* [11] and *MHMD* [18]. However, these first attempts of designing firefighter-assisting sensor systems rely heavily on pre-existing networks in the on-fire building, which is an invalid assumption at least in the near future. Thus, researchers have recently become more interested in the first part of system design we mentioned above, which is, how to deploy relaying nodes automatically and rapidly to maintain reliable communication between firefighters inside the building and base stations outside of the building.

There are now three categories of deployment approaches: no deployment, static deployment, and dynamic deployment. By no deployment, a firefighter usually carries a radio and communicates with the outside world within a single hop. One example system of this kind is the $P25$ system [4]. The main drawback of this approach is that due to hardware limitation, firefighters will inevitably lose their connections to the base station as they climb to tops floors in a tall building. Static deployment adopts a simple rule such as distance or time of the last deployment. However, simple static rules do not capture the wide variety of radio implementations, including different radio types, antenna types, and transmission power levels, all of which affect transmission range. More importantly, static deployment rules do not adapt to different channel propagation environments. For example, the range in an office corridor might be very different from that on a factory floor [14].

Compared to the first two methods, dynamic deployment is the most impressive as well as challenging. It monitors the run-time link quality and automatically deploys a new relay node whenever the communication metric (like PRR, RSSI, LQI, etc) satisfies some predefined rules. [14] 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 firefighter probes the channel periodically and measures link quality of measurement response. If 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 measuring link quality using a SNR based approach [17], interference avoidance [13], and UWB indoor localization techniques [9, 8].

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, which results 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 one breadcrumb leads to the failure of the whole system. Third, the link quality monitor they use is not appropriate and we will explain it in detail in later sections. Finally, they use 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 using a limited number of breadcrumbs.

The main differentiators of our proposed system over all the related work is in the sense of viewing the bread crumb based sensor system performance from an optimization point of view while maintaining high reliability.

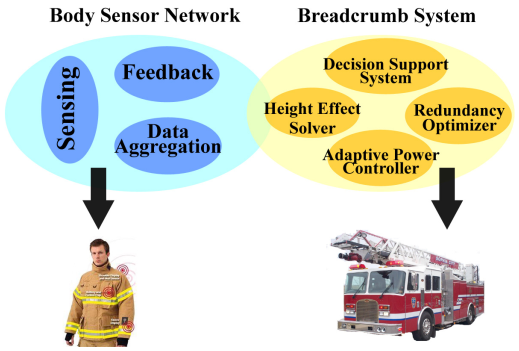


Figure 2: Breadcrumb system architecture.

3. REQUIREMENTS

To provide reliable and efficient assistance to firefighters, the breadcrumb sensor system must meet several requirements. First, it must provide a fast and automatic process to deploy new sensor nodes when necessary without disturbing the firefighters. This involves automatically monitoring the real time link quality to ensure reliable data transfer, designing a decision support system to identify when to drop a node, and automatically deploying new crumbs physically (e.g. using a dispenser with springs or rotation-enabled devices).

Second, the system must support reliable end-to-end communication in a consistent manner. Thus, each dropped crumb in the existing chain must be able to connect to the base station through more than one route to overcome the high possibility of physical failure of relay nodes due to harsh environments inside the building.

Finally, as firefighters only have a limited number of breadcrumbs carried at any given time, the system should be optimized to extend the overall transmission range as much as possible, to reduce the risk of disconnections as they move further away from the base station.

4. SYSTEM DESIGN

We propose a new breadcrumb system 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 address the specific requirements outlined in Section 3, and provide reliable and efficient means of automatic and robust breadcrumb deployment for in-door firefighter applications. Figure 2 illustrates the architecture of our proposed solution. We describe the overall system design and individual components in the following subsections.

4.1 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 trade-off 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% 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.

4.2 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 [13, 14], 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. Figure 3 describes the theoretical reliability of a crumb-chain when the RD is one. It shows the probability of end-to-end connection under various number of node failures in a breadcrumb chain of length $20, 30$, and 50 nodes. When the number of node failures varies from one to ten, we observe that even when four breadcrumbs are broken in a crumb chain of twenty nodes, there is still approximately 50% probability of end-to-end connection for the crumb chain. This demonstrates the necessity of some positive RD to make the whole system more robust and reliable.

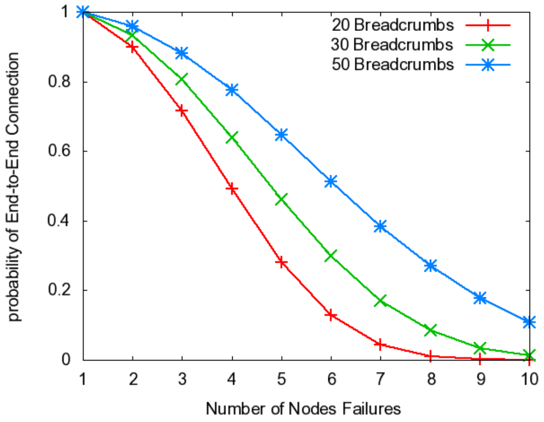


Figure 3: Probability of maintaining end-to-end connection when redundancy degree $= 1$, assuming independent and identically distributed node failures. *It can be observed that even when four breadcrumbs are broken in a crumb chain of twenty nodes, there is still approximately* 50% *probability of end-to-end connection for the crumb chain.*

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

$$\alpha = \frac{1}{n+1}\sum_{k=1}^{L} P(k) \cdot Survive(k) \qquad (1)$$

in which $P(k)$ indicates the probability that k breadcrumbs are dead in the breadcrumb chain, and $Survive(k)$ 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

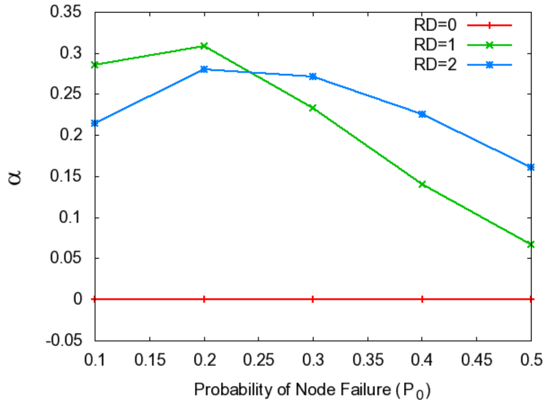


Figure 4: Comparison of metric α with different RDs when $L = 10$, as p_0 varies from 0.1 to 0.5. *It clearly shows that when p_0 is less than or equal to 25%, setting the redundancy degree to one is the optimal tradeoff.*

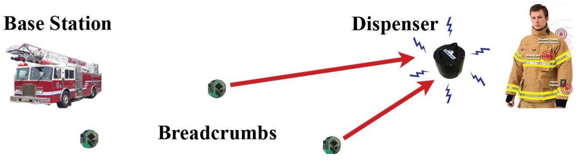


Figure 5: Double scout algorithm illustration.

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:

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

Figure 4 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% 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%, setting the redundancy degree to one is the optimal tradeoff. We argue that physical failure of breadcrumbs can be controlled to a low percentage (still positive though) by improving hardware design. Thus, in most cases we should set the redundancy degree to be one. We refer to this algorithm as the *d*ouble-scout algorithm.

Figure 5 illustrates how the double scout algorithm works. A firefighter always keeps “good” communication with two breadcrumbs. If there are more than two “good” communications available, two of the best link quality nodes are selected. Whenever there only exists at most one “good” breadcrumb connected to the firefighter, a new breadcrumb is deployed. Again, whether the communication between a breadcrumb and the firefighter is “good” or not is determined by the decision support system which we will investigate in detail in the next subsection.

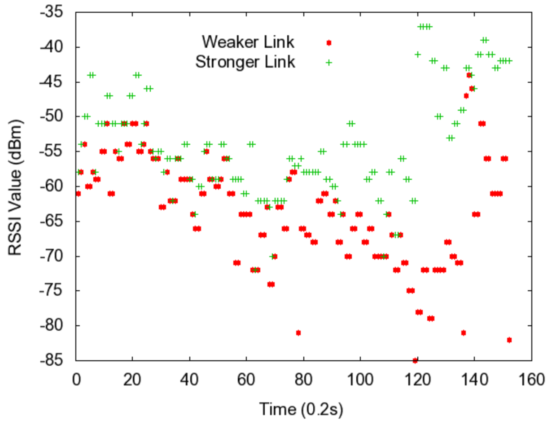


Figure 6: Double scout algorithm observation.

To show how the double scout algorithm operates at the dispenser side, we conduct some experiments on the double scout algorithm by placing two breadcrumbs on the floor and move the dispenser far away from them. Figure 6 shows the real time RSSI values of packets that the dispenser receives from the breadcrumbs. The threshold is set to -85 dBm and it can be observed that the weaker link reached the threshold after around 120. Then a new breadcrumb is deployed. We observe that there is always a strong link and another weaker link between a firefighter and the crumb chain.

The place of deployment for the second breadcrumb is crucial to the overall pattern of the final crumb chain. To further improve the link quality as well as packet reception ratio, we apply the “Max-min” optimization on the double scout algorithm by deploying the second breadcrumb in the crumb chain carefully so as to make the breadcrumb chain more even. Given the threshold, denoted by t_1, used by the decision support system, we move the third node in between two end nodes with threshold t_1 to find the point of threshold t_2 with the best fairness. Imagine a breadcrumb chain with deployed nodes numbered 1, 2, and 3, then t_1 means the RSSI between 1 and 3 and t_2 means the RSSI that is used as an indicator of where we should place breadcrumb 2 between 1 and 3.

4.3 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

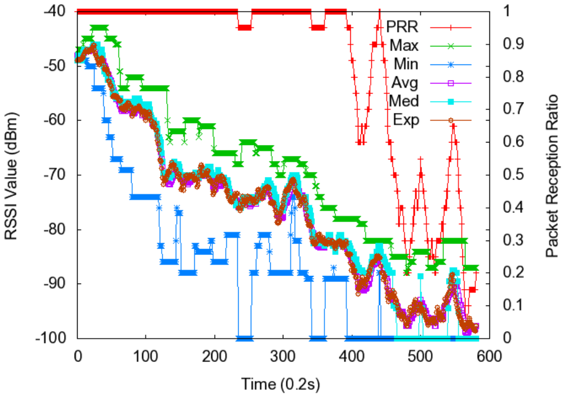


Figure 7: Relationship between metadata RSSI values and PRR until link disconnection.

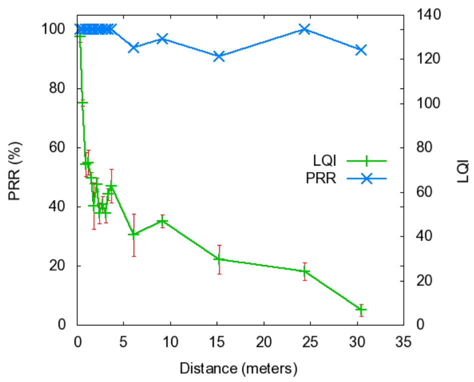


Figure 8: Relationship between LQI and PRR. ***The results indicate that the PRR is still around* 90% *while the LQI approaches very small numbers.***

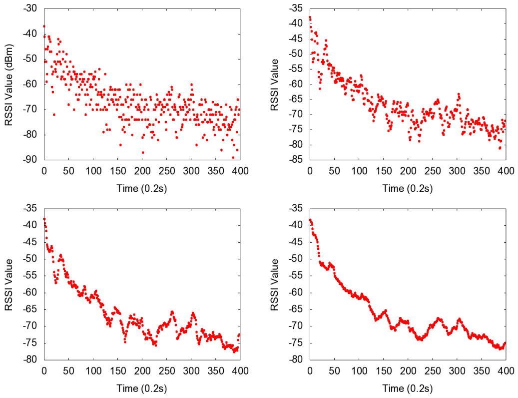


Figure 9: Mean estimator. Window size is 1 **(Topleft),** 5 **(Topright),** 10 **(Bottomleft), and** 20 **(Bottomright).**

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).

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 [14, 15]. 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. Figure 7 shows the experimental data of PRR and metadata in an example trace in which the communication device is moved far away from a breadcrumb on the floor until the connection is lost. The RSSI value for lost packets are set to be -100 dBm. It can be clearly seen that the metadata has some relations with the PRR and may help in making decisions.

LQI is a characterization of the strength and/or quality of a received packet [1]. The LQI value is limited to the range 0 through 255 and can be produced by the RSSI value [3]. In recent years researchers have been arguing about whether RSSI or LQI is a better representation for link quality [16]. We argue that LQI is not desirable in off-the-shelf hardware such as cc2420/2430 radio stack, since each RSSI value corresponds to a fixed LQI value and LQI equals to zero for each RSSI that is below -80. This makes the decision support system less accurate since it is possible that the PRR is still around 90% while the LQI approaches very small numbers, as shown in Figure 8.

SNR-based measurements takes advantage of physical layer signal-to-noise ratio for estimating run time link qualities. [17] shows that the SNR-based estimators are more efficient than PRR-based packet counting methods in terms of the number of measurements needed, and are more accurate when link quality variability increases. However, an important requirement of the SNR-based estimators is having *a priori knowledge* of the SNR-PDR relationship, which is unavailable for almost all buildings. This mapping is also both environment and hardware dependent. PRR-based approaches also suffer from problems like pattern recognition. Different patterns of “received/lost” strings with the same PRR value may reflect different real time situations and those pattern recognition algorithms are too heavy-weight to be applied in practice.

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 [14]. 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. Figure 9 shows the effect of window size on the smoothness of the $MEAN$ estimator.

- *EXP estimator* — Exponentially weighted moving average approach associates two parameters: current weighted value for RSSI Exp and weight coefficient β. Exp is

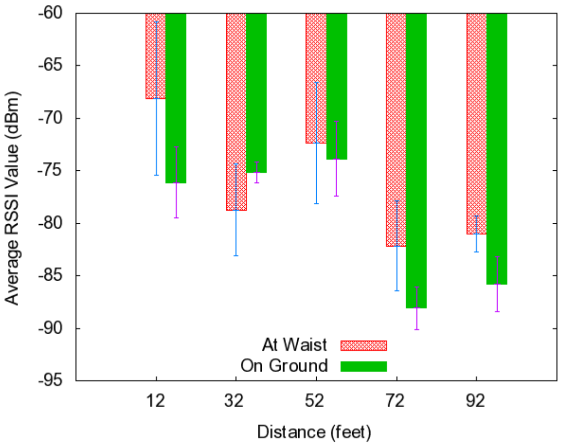


Figure 10: Height effect (stationary) on the floor. ***It can be observed that there is a*** 5 ***to*** 10 ***dBm degradation in most cases and the variance is around*** $10 - 20\%$**.**

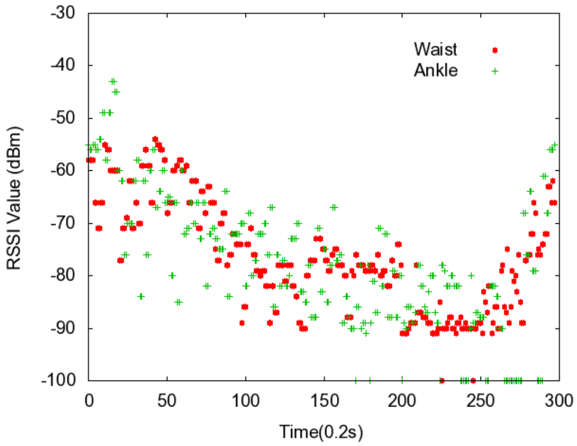


Figure 11: Comparison between locations of dispenser: Waist v.s. Ankle.

updated when new data packets with RSSI value R arrive using the formula below:

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

Compared to the MEAN estimator, the EXP estimator better reflects the dynamic characteristics of link quality by assigning heavier weight to more recently received packets. Additionally, it only requires recording two variables for each source instead of an array of RSSI values, so a lot of storage is saved.

- *RANGE estimator* — The last two estimators are based on the following observation: Link quality variation becomes very large in some complex environments such as where consecutive building corners are close to each other. In this case, the communication is likely to be lost even when the average performance is not that bad. Thus, we need to monitor the RSSI deviation as well. 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.

4.4 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 firefighter) 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 subsection, 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 firefighter. RSSI values between them are recorded while their distance varies from 10 feet 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.

Figure 10 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 $10 - 20\%$. 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. Finally, our experimental results in Figure 11 show that the RSSI values between the new node and an existing breadcrumb have extremely large variations and are not reliable when the link quality becomes weak.

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 6.

4.5 Adaptive Power Control

Adaptive power control (APC) 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 likely that if using the default transmission power, the new breadcrumb will be unable to maintain high quality link because of link dynamics.

Adaptive Power Control 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 6.

The adaptive power control scheme works as follows: When a breadcrumb is deployed, it begins to estimate pairwise link qualities between its neighbors by monitoring the RSSI value of 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 [12]. 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.

5. PROTOTYPE

We designed several custom hardware modules to accommodate the specific needs of the breadcrumb system. Figure 12 shows a family of hardware designed for evaluating our breadcrumb system. The hardware components of the system are as follows:

- *Dispenser* — The dispenser is designed to be lightweight and able to be hooked on the waist of firefighters. The current version can hold ten breadcrumbs adequate for testing and evaluating the system in a three-floor building. Figure 13 shows the dispenser with one breadcrumb placed in a slot in a turntable. When the decision support system determines to deploy a new breadcrumb, the turntable starts to rotate 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. Figure 13 shows the microprocessor in the dispenser.

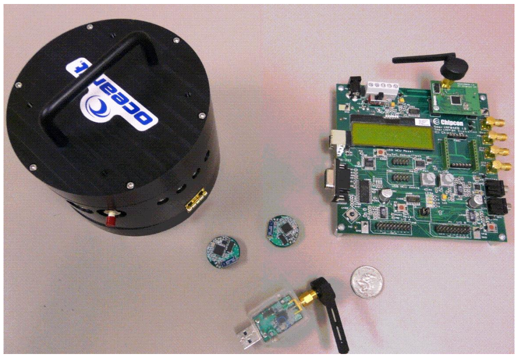


Figure 12: Breadcrumb system hardware prototype.

- *Breadcrumbs* — Each breadcrumb is a quarter-size mote powered by a lithium battery. It is placed in a covered box to be protected in the harsh environment. Current prototype of breadcrumbs are powered with a standard 3-volt, 560 mAh lithium battery cell from Panasonic. The battery is capable of running each router communication node at full power for 3 hours.
- *Base Station* — The base station is a USB-ported device connected to a laptop computer. This device can also be programmed using the evaluation board through a serial-port connector. The base station acts as the end device to receive data packets sent by the aggregation device on the firefighter's side through the breadcrumb chain, and stores the data to the database on the laptop.
- *Evaluation Board* — The $SmartRF04EB$ evaluation board from Texas Instruments Inc. is used to program the hardware (breadcrumb, dispenser and base station) using IAR programming environments.

Note that we do not use any PDA-like devices in our prototype systems as previous work did. According to the system requirements described in Section 3, firefighters should not be involved in dropping breadcrumbs since they are likely to be distracted from their rescue work. Instead, the processing unit embedded in the dispenser automatically drops breadcrumbs based on the decision support system.

5.1 Zigbee Protocol

We choose to base our implementation on Zigbee [7] because Zigbee networks offer advantages over other networking technologies such as Bluetooth and WiFi:

- Lowest power usage since the specification was designed for low power and low data rate communications for battery powered devices;
- Lowest cost due to wide industry acceptance and adoption;

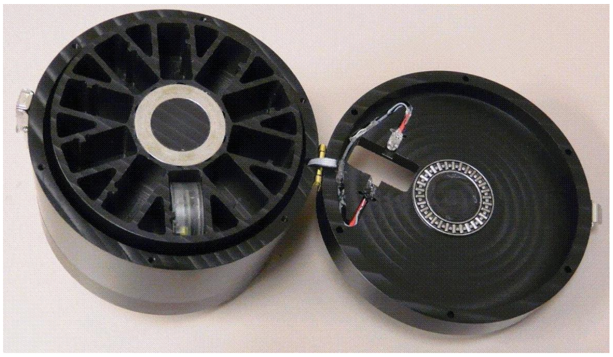


Figure 13: The dispenser with a breadcrumb inside.

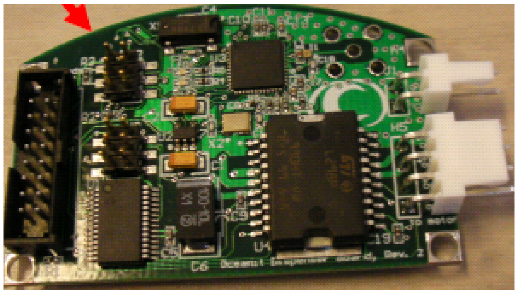


Figure 14: Microprocessor of the dispenser.

- Built using non-proprietary standardized hardware and software;
- Allows built in encryption for secure communications.

Table 1 compares the advantages and disadvantages of various networking technologies.

Table 1: Comparison of existing wireless technologies.

Name	Standard	Cost	Power	Bandwidth
Zigbee	802.15.4	Low	Low	Low
WiFi	802.11	High	High	High
Btooth	802.15.1	Medium	Medium	Medium

In real world applications, wireless communication systems must function in an environment containing interference from other wireless devices, appliances, and machinery as well as overcome losses due to construction materials and building topologies. The calculation of typical path loss is critical to determine the required transmitter power as well as a measure of the number of routers needed to blanket a building for reliable communications. Table 2 shows the various path losses for an indoor office environment with soft partitions at several distances across the same floor and through one floor.

As shown in the previous table, using the 2.4 GHz frequency would allow placing routers more than 50 meters apart on a single floor in a building with soft partitions.

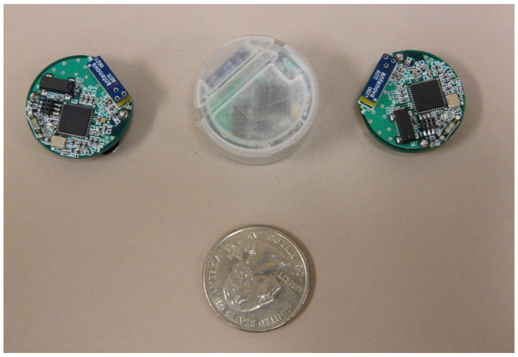


Figure 15: Breadcrumb prototype.

Table 2: Calculated path loss (dB) at various distances.

Distance (m)	on same floor	through one floor
10	65.6	91.8
20	73.4	99.6
30	78.0	104.2
40	81.3	107.5
50	83.8	109.9

6. EVALUATION

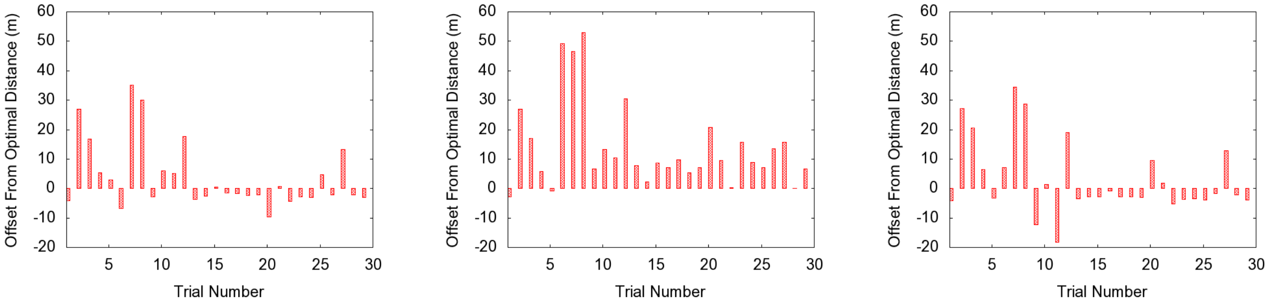
In this section we first describe the experimental evaluation of three key system components: the decision support system, the height effect solver, and the adaptive power control. Finally, we evaluate the overall system performance. Overall, our main performance results are:

1. The *Exponentially weighted moving average (EXP)* filter performs the best of the four candidate filters. Compared to the *MEAN* filter, *EXP* achieves less than a 20% possibility of false-negative errors in both the regression and prediction phases, which confirms our observations and analysis in Section 4.
2. The *Adaptive height adjustment* solution maintains an average RSSI above the threshold in all trials and has longer average distance than the conservative approach.
3. The *Adaptive Power Control* solution maintains a higher link quality over time than using a fixed transmission power and increases the likelihood of recovering unreliable links due to dynamic environmental changes.
4. Compared to the state of the art NIST work [14], our proposed system achieves 200% link redundancy at the expense of 23% additional deployed nodes. Our deployed breadcrumb chain also achieves 90% probability of end-to-end connectivity when one node fails in the crumb-chain and over 50% probability of end-to-end connectivity when up to 3 nodes fail.

6.1 Filter Evaluation

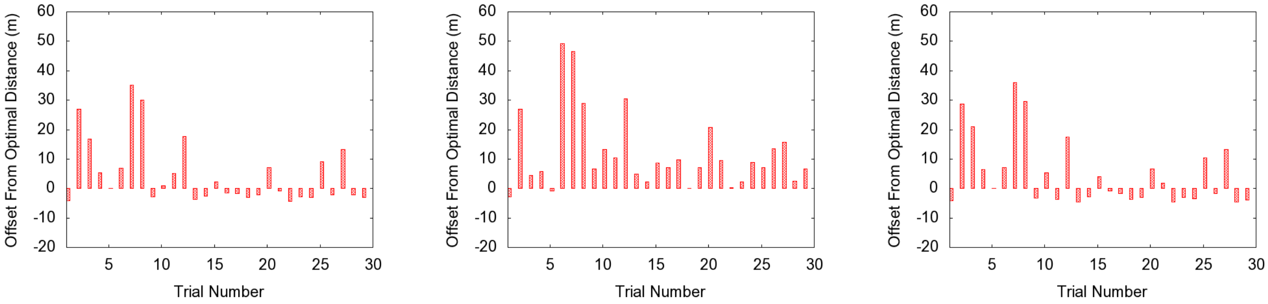
We first evaluate the performance of various filters we proposed in Section 4.3: *MEAN, EXP, RANGE,* and *MEDIAN.* The metric used for comparison is the packet reception ratio (PRR).

The evaluation approach is as follows. Totally 30 data sets are collected to evaluate these filters. The dispenser is



(a) *MEAN* filter. (b) *EXP* filter. (c) *MEDIAN* filter.

Figure 16: Difference distribution in regression step for filters: *MEAN*, *EXP*, and *MEDIAN*.



(a) *MEAN* filter. (b) *EXP* filter. (c) *MEDIAN* filter.

Figure 17: Difference distribution in prediction step for filters: *MEAN*, *EXP*, and *MEDIAN*.

hooked on the waist of one user, and the user begins to walk at an approximately constant speed after one breadcrumb is dropped. The runtime RSSI values between the dispenser and breadcrumb are recorded until their connection is lost. The sequence numbers of the packets are also recorded to calculate the PRR at runtime and the window size is set to 20. This experiment is conducted at six different starting points to cover all environments including hallways, corners, and stairways. The starting points are chosen to be the four corners of the same floor as well as the top and bottom of stairways. Each experiment is repeated 5 times. Note that the linkage between RSSI and distances is PRR. In section 4 we studied the RSSI-PRR mapping and here we convert it into optimizing the distances (determined by RSSI threshold) while maintaining high PRR.

The data is then analyzed off-line using Matlab. The analyzing process includes two parts: a *Regression* step and a *prediction* step. In the regression step, the dropping point (measured in distance) is calculated for each combination of the coefficients used in each filter, and is compared with the ground truth point (also in distance) which is defined as the point where the PRR drops to 90%. Finally, the Least Square Method is applied to find the best filter. The prediction step divides the trials into 5 groups (one experiment in each group), repeats the regression step for each 4 out of 5 groups, calculates the optimal coefficients for each filter, and applies it to the last group and obtains the difference between the estimated and ground truth dropping point. Then the Least Square Method is applied again to compare the performance of the filters.

6.1.1 Regression Step

Table 3 shows the results in the regression step. Results for three metrics are displayed: Least Square Distance, Standard Deviation, and number of false-negative cases. Optimal parameters in the results are as follows. In the *MEAN* filter, window size is 30 and threshold is -89.8 dBm; in the *EXP* filter, weight is 0.0313 and threshold is -81.8 dBm; and in the *MEDIAN* filter, the window size is 35, the threshold is -87.8 dBm, and the derivative is -22 dBm.

The first thing we observe is that the *RANGE* filter fails during its execution. The reason is that the filter did not decide to drop any new breadcrumb until the actual connection is lost in some traces. This fails to satisfy the reliability requirements of breadcrumb systems.

We also see that the *MEAN* and *MEDIAN* filters have similar performance in all three metrics. Both of them perform better than the *EXP* filter in terms of Least Square Distance. However, we note that they suffer from a huge problem of false negatives. In 30 trials, the *MEAN* filter results in 17 false-negative trials and the *MEDIAN* filter has 19. This implies that when applying one of these two filters, there is more than 50% chance that the resulting PRR of the new link in the breadcrumb link is below that predefined threshold. In contrast, the *EXP* filter results in only 3 false-negatives, which is only 19% of that in the *MEAN* filter case. The dropping point differences for all 30 trials are shown in Figure 16, in which the y axis means the distance between the ground truth dropping point (last 90% PRR point) and the dropping point calculated by the filters. For example, a -5 value means the calculated dropping points is 5 meters later than the ground truth dropping point.

Another observation worth mentioning is that on trials 6, 7, and 8, all estimators lead to a false-positive result as high as 50 meters. This is because these trials go through complex environments including consecutive corners and open-door offices, and thus the link may become extremely unstable. Therefore, although the final 90% PRR point of the ground truth for those trials is around 130 meters, the estimators decide to drop breadcrumb at around 80 meters because of the complex environment.

Table 3: Comparison of filters: Regression Step.

	MEAN	EXP	RANGE	MED
LSD (m^2)	4038	11455	Failed	4123
StdDev (m)	10.93	14.60	Failed	11.37
False-Neg	17	3	Failed	19

6.1.2 Prediction Step

Table 4 shows the results of the prediction step. Results for Least Square Distance and number of false-negatives are displayed. Similar trends as in the regression step can be observed: the *RANGE* filter fails again; the *MEAN* and *MEDIAN* filters have similar performance, and they perform poorly in terms of false-negatives with 50% for each of them. The *EXP* filter, on the other hand, has only 10% probability of false-negative predictions, which shows it is still the best candidate for the prediction step. The dropping point differences for all 30 trials are shown in Figure 17.

Table 4: Comparison of filters: Prediction Step.

	MEAN	EXP	RANGE	MEDIAN
LSD (m^2)	3940	10707	Failed	3113
False-Neg	15	3	Failed	15

6.2 Height Effect Solver Evaluation

We now compare the performance of the height effect solvers that we proposed in Section 4.4. According to the experimental results shown in Figure 10, we implement and compare the following three candidates: fixed 5 dBm offset, fixed 10 dBm offset, and adaptive offset.

The metric is the gap between the actual RSSI value of crumb-to-crumb link and the RSSI threshold set by the decision support system. We choose the *EXP* filter as the estimator of the decision support system based on the results and analysis in the last subsection. Experiments are conducted in several different trials and each trial is repeated five times. In each trial five new breadcrumbs are dropped.

Figure 18 shows the results of using different height effect solvers. Each bar value represents the average of five drops in one trial, and the maximum and minimum differences are also displayed. A negative difference means that the newly dropped crumb has worse link quality than the predefined threshold. We can observe that the conservative approach of having a fixed 10 dBm offset results in a highly reliable link, but has decreased average distance compared to the adaptive approach, as shown in Figure 19. Having a fixed 5 dBm offset results in a more unstable link and is likely to result in an unreliable breadcrumb chain, as evidenced by trial number 5 in Figure 18. The adaptive approach results in an average RSSI gap within 3 dBm for each trial as well as an average deployment distance of 9.3 meters. These results indicate that the adaptive approach leads to the best solution with average RSSI above threshold in all trials and has longer average distance than the conservative approach.

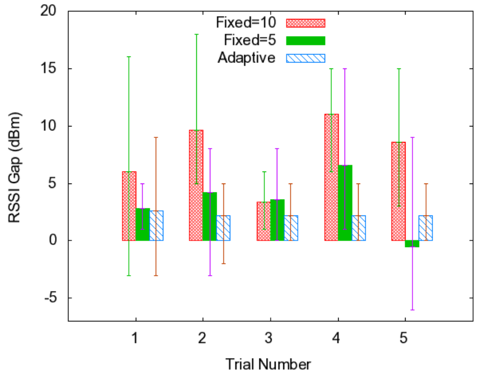


Figure 18: RSSI gap comparison of various height effect solvers.

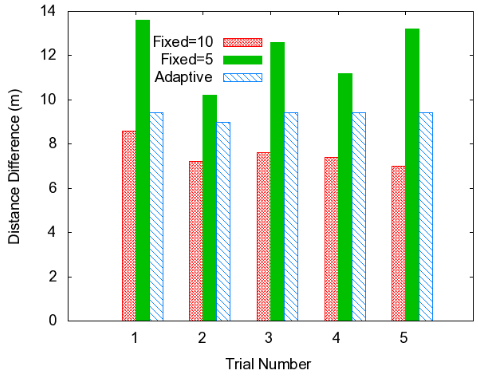


Figure 19: Average distance comparison of various height effect solvers.

6.3 Adaptive Power Control Evaluation

The evaluation approach for adaptive power control includes two parts. First, we conduct a series of experiments to validate the set point used in the approach. Second, the adaptive approach is compared to using fixed transmission power.

To validate the set point which is the threshold parameter (-81.8 dBm) in the *EXP* filter calculated in Section 6.1, we fix one transmitter and one receiver in many places in the building, then the transmitter sends 500 data packets to the receiver and both RSSI and PRR are recorded. The default power level is set to 27. Various environments are covered and five trials are done in each environment.

Figure 20 shows the average RSSI-PRR mapping in different environments. It is clear that the PRR is higher than 95% in all environments when the RSSI is set to the set point, -81.8 dBm, which implies that this set point is appropriate. We also observe that when RSSI is very low,

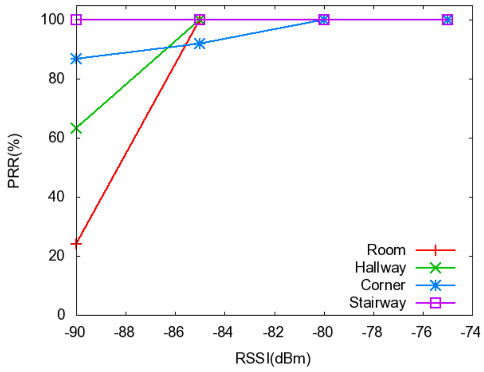


Figure 20: RSSI-PRR mapping in various environments.

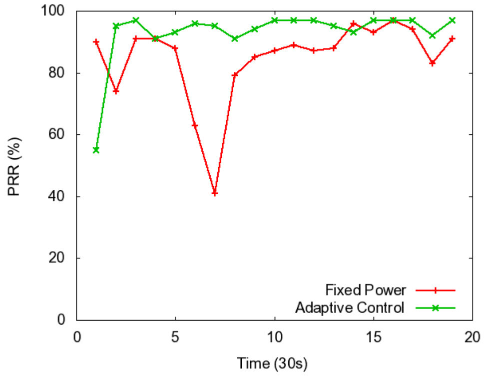


Figure 21: Performance comparison of fixed transmission power and adaptive control.

the packet reception ratio varies a lot in different environments. The main reasons that PRR are quite different under the same RSSI in different environments are obstacles and multi-path fading.

We then compare the adaptive power control approach against using fixed transmission power. A 10-minute continuous experiment is conducted to evaluate communication quality over time. A breadcrumb is placed on the ground and one person walks away until one breadcrumb is dropped out of the dispenser. Then this newly dropped breadcrumb begins to send packets to the previous one and the packet reception ratio is recorded every 30 seconds.

Figure 21 shows the result of link quality. First, we observe that the initial PRR in the adaptive control case is only 58% due to the multi-path fading effect. Then it increases to 92% by adjusting the power level and maintains a reliable link for the following nine minutes. On the other hand, the link quality varies a lot when the power level is fixed, for example, it drops to around 41% three minutes after the link is established. The results indicate that adaptive control approach helps maintain a reliable wireless link as well as recover from an unstable state.

6.4 Exploiting Reliability-Efficiency Tradeoff

After evaluating individual system components, the next issue is the tradeoff between reliability and efficiency in our system, i.e., how many more breadcrumbs are used in our system than in related work by NIST [14].

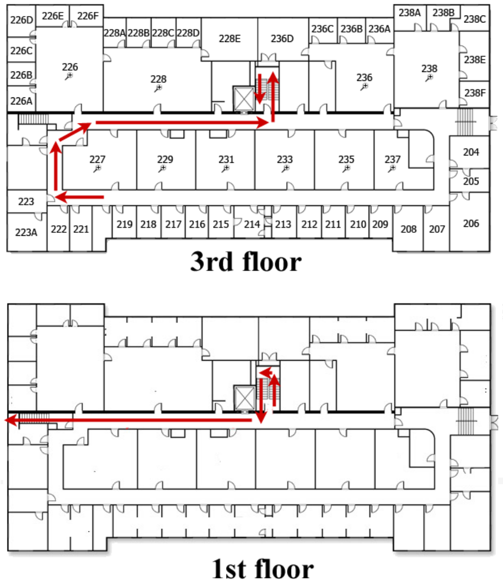


Figure 23: Trace for simulated rescue events.

A series of experiments are conducted in the building of Computer Science Department of University of Virginia. Figure 23 shows the path a firefighter walks. The base station is connected to a laptop and located outside one entrance of the first floor of the building and the firefighter 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.

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 5 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 [14]. The parameters used in [14] 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 from Section 6.1. We then apply each approach to the

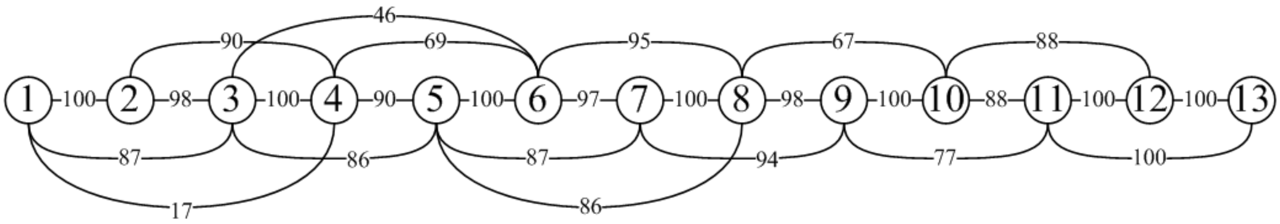


Figure 22: Logical topology for Trial 1 using our system.

trace described in Figure 23 and each case is repeated five times.

Table 5: Number of breadcrumbs dropped.

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

Table 5 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% link redundancy at the expense of 123% node redundancy. 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 Figure 22. It is clear that 11 out of 12 one-hop connections achieve more than 95% PRR and even three-hop wireless links exist ($1-4$, $3-6$, and $5-8$). We also observe that the PRR is only 69% between breadcrumbs 4 and 6 and 67% 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 (see Figure 23), which implies that complex environments may have a big impact on link quality.

Finally, we compare our work with [14] in terms of system robustness when breadcrumb failure occurs. Trial 1 in both cases are selected and the results are shown in Figure 24. We observe that our system has 90% probability to maintain end-to-end connection when one breadcrumb fails, while the NIST system has only less than 50%. Similar trends can be observed when more breadcrumbs fail, which implies that our system achieves better robustness than previous work.

7. DISCUSSIONS

Experimental results show that our automatic breadcrumb system achieves better reliability-efficiency tradeoff and can recover from unreliable wireless links. Our system is also more robust to breadcrumb failures. However, there are various opportunities to improve our breadcrumb system.

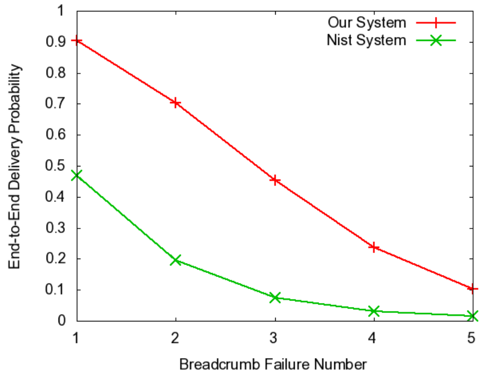


Figure 24: System robustness to breadcrumb failures.

For instance, it would be interesting to further investigate whether it is more desirable to use 2.4 GHz based or lower frequency hardware. Our current prototype uses 2.4 GHz based hardware instead of lower frequency like 900 MHz due to following reasons: First, by using a higher frequency, the size of the antenna is significantly reduced. This largely increases the possibility of having a lot of breadcrumbs placed in a single dispenser. In addition, Zigbee is a mature protocol stack for 2.4 GHz based hardware. However, there are some drawbacks for 2.4 GHz based hardware such as poor penetration through walls and poor operation if a sprinkler system is activated. It would be valuable to re-implement our system in lower frequency hardware and compare the performance with our current system.

Another interesting topic is to investigate the effect of different environments on the system performance. During the evaluation process, we observed that both RSSI and PRR perform differently under various environments, such as hallways, corners, and stairs, and link monitoring algorithm can be affected. This sheds some lights on further system optimization opportunities of setting different parameters in various environments. Looking into the data and trying to categorize is too complicated and beyond the scope of this paper. In addition, all experiments in the evaluation parts were repeated five times. During our experiments, we found that the results are very consistent and this is the main reason that we did not conduct more trials.

Moreover, 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 hard to address at this moment. In our opinion, one solution is to increase the redundancy degree, so as to improve the system robustness with best effort. Also adaptive power control can alleviate this problem by dynamically increasing power levels of isolated breadcrumbs to get a better chance to reconnect to the breadcrumb chain. In addition, the system becomes more robust in multiple firefighter scenarios because one group of firefighters can repair the destroyed link of another group when they happen to pass nearby.

Finally, considering multiple firefighter coordination results in a more realistic and efficient breadcrumb systems. All current work (including this paper) focused only on developing automatic and robust breadcrumb system assuming uncoordinated firefighters. In practice, firefighters are organized into small groups to execute different tasks and sometimes enter the building from several entrances simultaneously. Systems and algorithms designed for a single firefighter 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 firefighters are running along a hallway, in an uncoordinated scenario, the firefighter at the head of the group drops breadcrumbs all the time because his system always detects decreased link quality first. Later, when this firefighter takes another separate route by himself, he finds himself running out of breadcrumbs. Thus it is important to investigate efficient and automatic firefighter coordination algorithms to deal with multiple firefighters.

8. CONCLUSIONS

We have presented a new breadcrumb system deployment scheme that supports automatic and robust deployment of breadcrumbs for in-door firefighter applications. The system is composed of four components: redundancy degree optimization, decision support system, height effect solver, and adaptive power control. Experimental results show that compared to the state of the art work [14], our designed system achieves better reliability-efficiency tradeoff and can recover from unreliable wireless links. Our system is also more robust to breadcrumb failures. In the immediate future we have several items on our agenda, including leveraging multiple firefighter coordination and taking advantage of sensing capabilities on breadcrumbs. Developing reliability models for non-independent failures is another main task.

9. ACKNOWLEDGMENTS

This work was performed to support Oceanit on 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 supported, in part, by the NSF grants EECS-0901686 and CSR-0720640. Many thanks to Yongguang Zhang for shepherding this paper and helping us to improve its structure and readability.

10. REFERENCES

[1] Chipcon cc2430 datasheet, `http://www.alldatasheet.com/datasheet-pdf/pdf/241043/TAOS/CC2430.html`.

[2] Fire project at berkeley. `http://fire.me.berkeley.edu`.

[3] Ieee std. 802.15.4 - 2003, wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (lr-wpans). `http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf`.

[4] P25 projects, `http://en.wikipedia.org/wiki/Project-25`.

[5] Report of national fire protection association, 2005. `http://www.nfpa.org/`.

[6] Small business innovation research program, the department of homeland security, `https://www.sbir.dhs.gov/reference/FullSolicitation_FY07SBIR_7.%202_FINAL_Posted_06272007.pdf`.

[7] Zigbee alliance, `http://www.zigbee.org`.

[8] C. Gentile, A. Braga, and A. Kik. A comprehensive evaluation of joint range and angle estimation in indoor ultrawideband location systems. *In EURASIP Journal on Wireless Communications and Networking*, 2008.

[9] C. Gentile and A. Kik. A comprehensive evaluation of indoor ranging using ultra-wideband technology. *In EURASIP Journal on Wireless Communications and Networking*, 2007.

[10] X. Jiang, N. Y. Chen, J. I. Hong, K. Wang, L. A. Takayama, and J. A. Landay. Siren: Context-aware computing for firefighting. In *Proceedings of Second International Conference on Pervasive Computing, Pervasive 2004*.

[11] 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. In *Proceedings of Ubicomp 2007*.

[12] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic. Atpc: Adaptive transmission power control for wireless sensor networks. In *Proceedings of SenSys 2006*.

[13] M. T. Refaei, M. R. Souryal, and N. Moayeri. Interference avoidance in rapidly deployed wireless ad hoc incident area networks. In *Proceedings of IEEE Infocom Workshop on Mission-Critical Networking*, 2008.

[14] M. R. Souryal, J. Geissbuehler, L. E. Miller, and N. Moayeri. Real-time deployment of multihop relays for range extension. In *Proceedings of ACM MobiSys 2007*.

[15] M. R. Souryal, L. Klein-Berndt, L. E. Miller, and N. Moayeri. Link assessment in an indoor 802.11 network. In *Proceedings of IEEE WCNC*, 2006.

[16] K. Srinivasan and P. Levis. Rssi is under-appreciated. In *Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets)*, 2006.

[17] A. Wapf and M. R. Souryal. Measuring indoor mobile wireless link quality. In *Proceedings of IEEE ICC*, 2009.

[18] 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 increased indoor firefighting safety and efficiency. In *Proceedings of SPIE*, 2005.