RFID Trees: A Distributed RFID Tag Storage Infrastructure for Forest Search and Rescue

Victor K.Y. Wu* and Nitin H. Vaidya
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Email: {vwu3, nhv}@illinois.edu

***Abstract*—We create a distributed storage infrastructure by embedding passive RFID tags in trees, for forest search and rescue. As a hiker moves through the forest, her reader writes a unique identifier (ID) and increasing sequence numbers (SNs) to tags, called (ID,SN) pairs. This creates a digital path for searchers to follow if the hiker is lost. Since tag memory is limited, hikers must share this constrained resource to preserve their digital paths. At each tag, we consider a hiker overwriting an existing (ID,SN) pair if the tag is already full, according to one of four algorithms. In Oldest Selection (OS), the hiker deletes the oldest (ID,SN) pair. In Random Selection (RS), the hiker randomly deletes an (ID,SN) pair. In Highest Frequency Selection (HFS), the hiker deletes the (ID,SN) pair associated with the ID that she has seen the most in previous tag encounters. In Lowest Delete Frequency Selection (LDFS), the hiker deletes the (ID,SN) pair associated with the ID that she has deleted the least in previous tag encounters. HFS performs the best, but requires hikers to remember the number of ID encounters in the past, for each hiker ID.**

I. Introduction

RFID (radio frequency identification) technology is being used in search and rescue operations. In [1], the authors use a mobile robot to determine the locations of fixed RFID tags in space. This creates an RFID map that the robot uses to localize itself, as well as track the movements of other mobile objects. Although search and rescue is not specifically addressed in their work, it can be easily adapted for such a purpose. In [2], tags are distributed in the field by mobile robots. The robots communicate with each other through reading and writing to the tags, in order to create maps for search and rescue. In these systems, RFID technology only plays a supporting role. The robots are the main part of the system. However, robots may not necessarily be realistic in all situations, especially if they are cost-prohibitive to produce and maintain. North American safety authorities are opting instead to use active (battery-powered) RFID tags in conjunction with traditional human search and rescue [3]. Patients suffering from Alzheimer's (or similar diseases) wear wristbands, each with an active RFID chip emitting a unique identifier. If a patient wanders away, searchers use RFID readers tuned to the patient's identifier to quickly locate her. Similarly, in this paper, we consider search and rescue in the specific case of lost forest hikers.

A. Tagging trees for search and rescue

Despite navigational tools such as maps, compasses and GPS (global positioning system) devices, untrained hikers often become lost. Also, in a highly wooded area, a GPS device may not function. Even if a hiker can use a mobile phone, she may still not know her coordinates, or be able to communicate those coordinates to a search team. In this paper, we propose a search and rescue system for lost hikers in a forest. We deploy passive (battery-less) RFID tags throughout the forest by embedding them in trees. As a hiker moves through the forest, she uses an RFID reader (potentially integrated into her mobile phone) to write a unique identifier (ID) and increasing sequence numbers (SNs) to nearby tags, creating a digital path. We call these (ID,SN) pairs. (For ease of exposition, we define the less than relationship between two (ID,SN) pairs as follows. $(a, b) < (x, y)$ if $a = x$ and $b < y$.) Note that hikers generally do not know which particular trees have tags. (E.g. a tag may be embedded inside a tree bark, hidden from view.) That is, a hiker's reader can periodically scan for nearby tags, automatically interacting them, even if the hiker is oblivious to the RFID communications as she progresses through the forest. If the hiker becomes lost, searchers equipped with readers follow the digital path by scanning for increasing (ID,SN) pairs left by the lost hiker. (Searchers of course do not need to follow the digital path exactly. Some (ID,SN) pairs may be skipped depending on how searchers cover the area.) Our system removes the need for mobile robots, but instead uses human searchers. Nonetheless, it is conceptually similar to [1] and [2], since we use passive tags directly in the field to store information. Our system is similar to [3], since searchers use RFID readers in the rescue process. However, providing active RFID wristbands for every hiker is not feasible. Instead, we use passive tags deployed throughout the forest.

B. Replacement algorithms

If we use the search times for lost hikers as a performance measure, our system improves or degrades gradually, according to marginal changes in the spatial density of tags. For example, if we start out with a sparse deployment of tags, searchers may have difficulty following the digital path of a lost hiker, since tags containing (ID,SN) pairs may be few and far between. Searchers thus have to wander a lot when finding the next tag with a larger (ID,SN) pair. We call this

[*] This work is supported by NSF grant CNS-0519817.

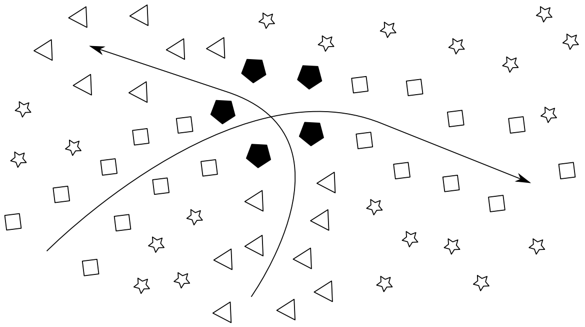


Fig. 1. Two hiker paths are shown. The polygons represent tags. The triangles and squares are storing the digital paths of the hikers moving toward the left and right, respectively. The filled pentagons are storing both paths. The stars are empty. If a third hiker moves through the center area, and the pentagons (tags) do not have any more memory, she may replace existing (ID,SN) pairs, potentially creating physical gaps for the two original digital paths.

lack of continuity in a digital path a physical gap. As tags are slowly added to the forest, searchers wander less, gradually improving performance. Conversely, tags may be damaged or removed by people or animals, degrading performance. Barring any significant natural disaster, however, this happens rarely in time and space. (In the event of a natural disaster, hikers would not visit those areas anyways.) The performance also depends on the number of hikers. Since tag memories are constrained, we assume a hiker replaces an existing (ID,SN) pair in a full tag with her own (ID,SN). As we add more hikers, more physical gaps in the digital paths are created due to these (ID,SN) pair deletions, causing the same problem as that of sparse tag deployment. This is illustrated in Fig. 1.

In this paper, we consider four algorithms a hiker can use if she replaces an existing (ID,SN) pair with her own. In Oldest Selection (OS), the hiker deletes the oldest ID-SN pair. In Random Selection (RS), the hiker randomly deletes an (ID,SN) pair without any bias. In Highest Frequency Selection (HFS), the hiker deletes the ID-SN pair associated with the ID that she has seen the most in previous tag encounters. In Lowest Delete Frequency Selection (LDFS), the hiker deletes the ID-SN pair associated with the ID that she has deleted the least in previous tag encounters. Our main contribution is evaluating the system performance of these algorithms.

C. System implementation

Implementing this system requires deploying tags and developing hardware and software. At one centralized extreme, the forest authorities are responsible for embedding tags into trees and keeping an inventory of them. Tag locations are strategically chosen and new tags are deployed as necessary. They also provide hardware and/or software to hikers, maintaining tight control over tags. At the other distributed extreme, the forest authorities exercise minimal control and restrictions on the system. Hikers buy special "tree-friendly" tags from manufacturers directly. They embed them into trees (through some safe and practical method provided by the forest authorities) as they move through the forest, and remove them if desired. Hardware and software and their supporting standards are developed through communities apart from the forest authorities. We envision a realistic implementation falls somewhere between these two extremes.

We note that our proposed system has the advantages of a sensor network. Many inexpensive tags are deployed in space. They sense their environment (albeit passively) for hikers, and store the hikers' information. Unlike a sensor network, however, battery consumption of tags is not a problem. Furthermore, information does not need to be aggregated by access points. Instead, we use mobile readers to read from and write to the tags, allowing information to flow through our distributed storage infrastructure. We envision that a variety of practical services can be developed on top of such a platform. In this paper, we narrow our scope to search and rescue.

The rest of the paper is organized as follows. In Section II, we present related work and supporting technologies that indicate the feasibility of our system. In Section III, we present the four replacement algorithms a hiker can use to delete an existing (ID,SN) pair at a tag. In Section IV, we evaluate our search and rescue system through analysis and simulations. Finally, Section V concludes the paper and provides future work.

II. Background literature

We present related work. We also present supporting technologies that indicate the feasibility of our system.

A. Related work

The authors in [4] propose a "super-distributed" tag infrastructure. They envision deploying tags in space over large areas in a highly dense and redundant fashion. Vehicles may leave traces by writing IDs to tags. The path can be retraced by other vehicles, and overwritten slowly in time. These ideas are only briefly mentioned in the paper without further exposition. Our work differs since we concentrate on leaving digital paths for the purpose of search and rescue. As well, we focus on using tags as constrained storage devices that are potentially overwritten very quickly. In other words, our work focuses on a more dynamic scenario where multiple readers are sharing storage among multiple tags.

In [5], the authors consider a wireless network composed of stationary RFID readers and mobile tags. Stationary readers read and write data to tags. The tags physically carry the data to other readers, thus achieving information flow. The authors focus on how such a network differs from traditional wireless networks, and the unique challenges these low data rate networks present. Our work is similar since hikers are also interacting with each other, using constrained tag storage as a medium. However, in our case, we consider the specific case of stationary tags and mobile hikers.

In [6], the authors propose RFID for indoor search and rescue. Tags are fixed in position and store location information, as well as other building-related information. As searchers move through a building during a disaster situation, this information is collected to aid in the search and rescue process, such as mapping and/or tracking. Our work differs since we focus on hiker interactions with tags, rather than searcher interactions. Specifically, we consider the dynamics of hiker digital paths being written and overwritten in tags. Conversely, [6] studies how searchers use information statically stored in tags a priori.

Search and rescue for forest hikers is addressed in [7], in a system called CenWits (**C**onnectionless S**en**sor-Based Tracking System Using **Wit**nes**s**es). In CenWits, a hiker in a forest wears a sensor containing a GPS receiver and a radio transmitter. When hikers come in contact with each other, they become location witnesses for each other by exchanging location information (retrieved from the GPS receivers). Dedicated access points are distributed throughout the forest, with connections to a processing center. When a hiker passes by an access point, she can upload her accumulated location information accordingly. If a hiker becomes lost at a later time, responders can be deployed to rescue the hiker using location information previously uploaded by the lost hiker and/or her witnesses. The authors in [8] provide optimizations to CenWits.

We note the salient differences in our infrastructure compared to CenWits. Our system is based on hiker paths instead of specific location information. As a result, we do not require GPS (or any other explicit location determination mechanism). CenWits uses access points, which are expensive to deploy and maintain in a forest. As well, they can only be installed at fixed and well-known positions. Each additional access point incurs an additional significant cost. Conversely, we use tags to collect information in our scheme. We can deploy tags densely throughout the forest. The maintenance cost of tags is merely the cost of replacing damaged tags. Tags can be placed at any location where feasible. Their locations do not even have to be known after deployment. Marginally adding tags only marginally increases costs. Since a dense deployment of access points in CenWits is not possible, information flowing to access points, and eventually reaching the processing center relies on witnesses trading information. If, however, there are few hikers, less information is garnered for a specific hiker, making search and rescue for her, if necessary, difficult. In contrast, having fewer hikers does not hurt our system. Finally, CenWits relies on a processing center external to the in-forest components (access points, hikers, and sensors) to aggregate information and compute search patterns. In our system, an external agent is not necessary for information aggregation. As well, any other computation is completely distributed.

B. *Supporting technologies*

Our proposed system relies on tags as storage devices. Companies are beginning to produce EPCglobal UHF Class 1 Gen 2 tags [9] with more re-writeable memory than the customary 512 bits in the past. ICsense and Tego have a Gen 2 tag with 32 kilobytes of re-writeable memory [10], [11]. We envision our system uses these Gen 2 tags.

In this paper, we do not focus on the physical layer RFID communications. However, previous work has demonstrated a plethora of tag singulation algorithms (based on query tree [12], and slotted Aloha [13]) are very effective at resolving tag collisions. As well, our experiments with the Alien ALR-9650 RFID reader [14] shows that up to 50 tags can be read within a second. If we embed each tree with two to three tags, we can scan over 15 trees within a second. Thus, our system can be easily supported with current passive UHF RFID technology.

Our proposed system requires users to carry readers. It is unreasonable, however, to expect users to have dedicated readers in many situations. Instead, we rely on the mobile phone. The mobile phone has matured into a ubiquitous communications and computing device. Most importantly, a person carries a mobile phone with her at all times, providing us with a realistic solution. In fact, manufacturers are already integrating readers into mobile phones. Nokia, Samsung, LG, and Motorola all offer NFC-enabled (near field communications) handsets [15]. Nokia has even integrated a UHF (ultra high frequency) reader into one of its handsets [16]. [17] and [18] offer reader software used in mobile phones.

The feasibility of tree tagging is demonstrated in [19], where trees are tagged as part of a tree tour. Tree-specific information stored in the tags are extracted by people on the tour, using PDAs (personal digital assistants) to scan the tags. [19] also investigates the physical constraints of embedding a tag in a tree. The tree is drilled and the tag is embedded below the bark. The forestry and logging industries also use RFID [20]. Embedded tags can be used to track the health of trees. Once a tree is chopped down, an embedded tag supports tracking of the log as it moves through the supply chain.

III. Replacement algorithms

Hikers move through the forest. Every time a hiker scans a tag, she writes her next (ID,SN) pair. If there is remaining space in the tag, the hiker writes to an empty memory slot. If an (ID,SN) pair belonging to the hiker is already in the tag (from a previous scan), she replaces it with the new (ID,SN) pair (thus, increasing the SN). Otherwise, the hiker chooses an existing (ID,SN) pair to delete, according to one of the following four replacement algorithms.

A. *Oldest Selection (OS)*

In Oldest Selection (OS), if necessary, the hiker deletes the oldest (ID,SN) pair. This may require hikers to include time stamps when writing (ID,SN) pairs to tags, thereby increasing the memory requirements of tags. Conversely, if the memory contents in a tag can be ordered, in a first-in first-out queue for example, then the additional memory is not necessary. OS assumes older hikers have left the forest, and thus gives priority to newer hikers.

B. *Random Selection (RS)*

In Random Selection (RS), if necessary, the hiker randomly deletes an (ID,SN) pair without any bias. That is, the random choice is uniform. RS aims to minimize the chance a hiker deletes the same ID in consecutive tag encounters.

C. Highest Frequency Selection (HFS)

In Highest Frequency Selection (HFS), if necessary, the hiker deletes the (ID,SN) pair associated with the ID that she has seen the most in previous tag encounters. Intuitively, HFS avoids deleting lower frequency IDs, since this potentially creates gaps. Each hiker has the cost of maintaining a list of ID frequencies in HFS.

D. Lowest Delete Frequency Selection (LDFS)

In Lowest Delete Frequency Selection (LDFS), if necessary, the hiker deletes the (ID,SN) pair associated with the ID that she has deleted the least in previous tag encounters. Similar to HFS, LDFS avoids potentially creating gaps. However, the approach is different. While HFS observes the ID frequencies (and thus hiker frequencies) in the field of tags, LDFS tries to spread out its effects of (ID,SN) deletion evenly among all hikers. Each hiker has the cost of maintaining a list of ID delete frequencies in LDFS.

E. Example

We illustrate the four algorithms through a simple example. Suppose a hiker scans a tag, and reads from it the five (ID,SN) pairs, $\{(2,13),(100,12),(15,99),(41,6),(75,92)\}$. The hiker plans to write her next ID-SN pair to the tag. Consider six cases. In the first case, the tag can store $T=6$ ID-SN pairs. The hiker has ID $=21$, and her next SN is 24. Since there is one remaining memory space, the hiker writes $(21,24)$ to the tag. In the second to sixth cases, the tag can store $T=5$ (ID,SN) pairs. In the second case, the hiker has (ID,SN) $=$ $(15,106)$. This means the hiker scanned this tag previously. The hiker thus updates $(15,99)$ to $(15,106)$ in the tag. In the third case, the hiker has (ID,SN) $=(21,24)$, and uses OS. Assuming the (ID,SN) pairs in the tag are ordered newest to oldest from top to bottom, $(75,92)$ is shifted out of the tag to make room for $(21,24)$. In the fourth case, the hiker has (ID,SN) $=(21,24)$, and uses RS. The existing (ID,SN) pair $(41,6)$ is randomly chosen and replaced with $(21,24)$. The first four cases are tabulated below. The first column shows the (ID,SN) pairs in the tag immediately before the hiker scans it. The other columns show the (ID,SN) pairs in the tag after the new (ID,SN) is written to it.

(ID,SN)				
Before	Case 1	Case 2	Case 3 (OS)	Case 4 (RS)
$(2,13)$	$(2,13)$	$(2,13)$	$(21,24)$	$(2,13)$
$(100,12)$	$(100,12)$	$(100,12)$	$(2,13)$	$(100,12)$
$(15,99)$	$(15,99)$	$(15,106)$	$(100,12)$	$(15,99)$
$(41,6)$	$(41,6)$	$(41,6)$	$(15,99)$	$(21,24)$
$(75,92)$	$(75,92)$	$(75,92)$	$(41,6)$	$(75,92)$
	$(21,24)$			

In the fifth case, the hiker has (ID,SN) $=(21,24)$, and uses HFS. Suppose the hiker has seen other hikers' IDs in previous tag encounters with the following frequencies.

ID	See Frequencies
2	8
15	65
41	50
79	43
86	5
92	72

Of the hiker IDs in the tag, 15 has the highest see frequency. Therefore, $(15,99)$ is replaced with $(21,24)$. In the sixth case, the hiker has (ID,SN) $=(21,24)$, and uses LDFS. Suppose the hiker has deleted other (ID,SN) pairs in previous tag encounters with the following frequencies.

ID	Delete Frequencies
2	13
12	2
41	10
75	11
83	5

Of the hiker IDs in the tag, 41 has the lowest delete frequency. Therefore, $(41,6)$ is replaced with $(21,24)$. The last two cases are tabulated below.

(ID,SN)		
Before	Case 5 (HFS)	Case 6 (LDFS)
$(2,13)$	$(2,13)$	$(2,13)$
$(100,12)$	$(100,12)$	$(100,12)$
$(15,99)$	$(21,24)$	$(15,99)$
$(41,6)$	$(41,6)$	$(21,24)$
$(75,92)$	$(75,92)$	$(75,92)$

IV. System evaluation

We evaluate our search and rescue system through analysis and simulations using the four replacement algorithms.

A. Analysis

We consider a simple geometry to allow for tractable analysis. Consider a line segment $\mathcal{L}$ in space of length L, with tags distributed along that line segment with linear density $\rho(u)$ tags/m that depends on the position u. Suppose a hiker moves through that line segment, writing a digital path to the $\int_{\mathcal{L}} \rho(u)\, du$ tags, and the tags are now all full. We wish to know how the digital path erodes as more hikers move pass (intersect) the line segment. We consider only the OS and RS replacement algorithms for ease of exposition.

First, we characterize how likely later hikers pass through that line segment using a probability distribution function (pdf). Consider a circular disk of radius $L/2$ centered at the origin, as shown in Fig. 2. The line segment is the y-axis portion of the disk (indicated by the arrow). We randomly select points $P_1:(x_1,y_1)$ and $P_2:(x_2,y_2)$ from the left and right semi-disks, respectively. $x_1 \sim unif\,[-L/2,0]\,, x_2 \sim unif\,[0,L/2]\,,$ and $y_1, y_2 \sim unif\,[-L/2,L/2]$. The line segment connecting P_1 and P_2 forms the path of a later hiker. In particular, the later hiker intersects the y-axis at $y = y_1 - \left(\frac{y_2-y_1}{x_2-x_1}\right) x_1$. The pdf of y, $f_y(u)$, is shown in Fig. 3. This models a situation where the middle of a forrest is a highly trafficked area. Thus, if the original hiker moves through the middle of a forrest, she should expect her digital path to erode quickly in the middle, as later hikers replace (ID,SN) pairs. In the following, we consider a generic pdf $f_y(u)$, so that the analysis applies for a general setting, depending on the statistics of hiker movements and the environment. $f_y(u)$ characterizes where later hikers are more or less likely to pass through $\mathcal{L}$.

Since the tags on $\mathcal{L}$ are all full, (ID,SN) pairs are deleted after one later hiker passes through $\mathcal{L}$, according to OS (We assume the ID-SN pairs of the original hiker are the oldest). In particular, if the scan range is small, and circular with radius Δ, then approximately $\rho(u)\, 2\Delta$ tags (ID,SN) pairs are deleted if the later hiker passes through at position u. Therefore, the

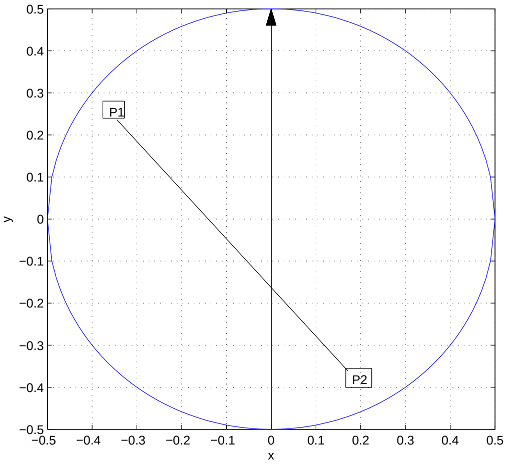


Fig. 2. The disk represents the field where hikers can move. In this case, $L = 1$. The original hiker leaves a digital path, shown by the arrow.

expected number of (ID,SN) pairs deleted is

$$\mathbb{E}\left[\text{number of (ID,SN) pairs deleted with OS}\right] \approx \int_{\mathcal{L}} f_y\left(u\right)\rho\left(u\right)du = \mathbb{E}\left[\rho\left(y\right)\right]. \quad (1)$$

If tags have T memory slots, then there is only a $1/T$ chance that the original hiker's ID-SN pair is deleted for a given tag. Therefore,

$$\mathbb{E}\left[\text{number of (ID,SN) pairs deleted with RS}\right] \approx \mathbb{E}\left[\rho\left(y\right)\right]/T. \quad (2)$$

Next, we characterize how the digital path erodes as multiple hikers pass through $\mathcal{L}$. Again, consider a small scan range and OS. For $\epsilon > 0$ small, the probability that an ϵ region of $\mathcal{L}$ at position v has the (ID,SN) pairs of the original hiker deleted is

$$\Pr\left(\text{pairs in } [v-\epsilon, v] \text{ deleted after } k \text{ hikers pass with OS}\right)$$

$$\begin{aligned} &\approx 1 - \left(1 - \int_{v-\epsilon}^{v} f_y\left(u\right)du\right)^k && (3)\\ &\approx 1 - \left(1 - f_y\left(v\right)\epsilon\right)^k && (4)\\ &\approx k f_y\left(v\right)\epsilon. && (5)\end{aligned}$$

For RS, we have

$$\Pr\left(\text{pairs in } [v-\epsilon, v] \text{ deleted after } k \text{ hikers pass with RS}\right)$$

$$\approx k f_y\left(v\right)\epsilon/T. \quad (6)$$

Thus, we see that the original hiker's digital path likely erodes as more hikers pass through.

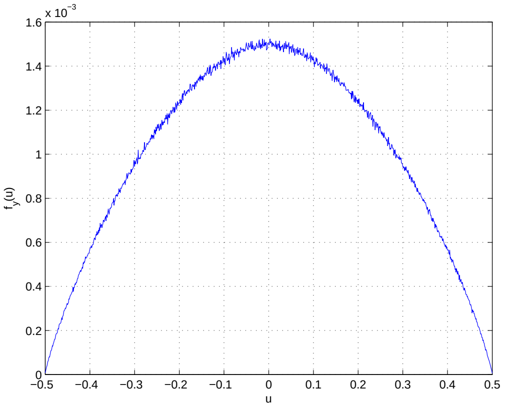


Fig. 3. Probability density function of where later hikers intersect $\mathcal{L}$

B. Simulations

1) Simulation model: To evaluate our system in a real-world environment, we consider a portion of the Yosemite Valley hiking trails [21] of Yosemite National Park in California, as shown in Fig. 4. The lengths of the trail sections $\{l_i\}$ are shown in Fig. 4. We focus on the trail sections shown in Fig. 5. We assume the widths of the trails are 1 m. The spatial tag density of the trails are ρ tags/m^2. We therefore generate $\rho l_i 1$ tags for the i^{th} trail section, randomly located in that section according to a uniform spatial distribution. Each tag has T memory slots for storing (ID,SN) pairs. Hikers are each equipped with a reader with a circular scan range of radius 1 m (typical of passive UHF RFID systems [22]). Hikers choose one of the following six paths and move through them, walking in the center of trail sections.

- Path 1: A, E, H, G, D
- Path 2: A, B, F, H, I
- Path 3: D, G, H, F, B, A
- Path 4: D, C, B, E, H, I
- Path 5: I, H, E, A
- Path 6: I H, F, C, D

For each simulation run, the tags are initially empty. The first hiker selects path 1 and moves through it, storing a digital path. Then, the next 24 hikers successively each pick a path at random, and moves through it, storing her digital path, using the replacement algorithms if necessary. (That is, after the first hiker, the next $T-1$ hikers do not need to use the replacement algorithms, since tags are guaranteed to still have space.) We want to know how the first hiker's digital paths erodes as later hikers move through the system.

In these simulations, we consider hikers moving along the specified park trails. If a hiker becomes lost, it is usually due to some unforeseen circumstance. For example, she may have mistakenly ventured off the trail. (In this case, her device could even notify her that it is no longer scanning tags. Such a system is beyond the scope of this work, but is definitely a future work). Or, a hiker may have accidently slipped off track and is now immobilized in a location near the trail, waiting for

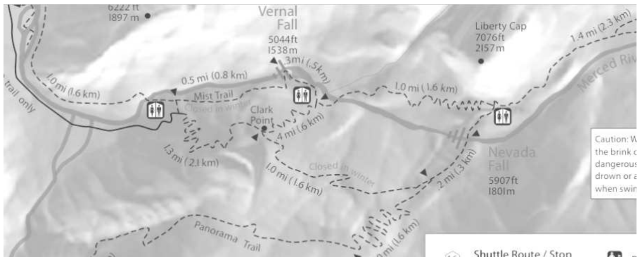


Fig. 4. Yosemite Valley hiking trail

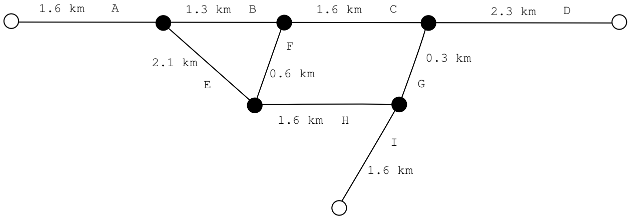


Fig. 5. Each trail section is labelled with a letter. Section lengths are shown.

help. In these cases, it is important that the digital path does not erode quickly. That is, searchers can follow the digital path and use the last known location of the lost hiker on the trail (indicated by the end of the digital path) as a check point during searching.

2) Simulation results: We consider the fraction of remaining ID-SNs in the first hiker's digital path as later hikers move through the system and delete the first hiker's (ID,SN) pairs with the replacement algorithms. Results are shown in Fig. 6 for $\rho \in \{1, 5, 10\}$ tags/m^2 and $T \in \{3, 7\}$. The x-axis is the cumulative distance travelled by later hikers. Therefore the spacing of points across the x-axis is non-uniform, since hikers randomly pick different paths composed of trail sections with different lengths. The results show that performance improves as we increase the tag density, and add more memory to tags, as expected. OS is a lower bound on the performance among the four replacement algorithms, since (ID,SN) pairs of the first hiker will obviously be deleted first if necessary. However, OS might be an attractive choice since it requires the very little computational resources in a tag. For $T = 3$, we see a fairly good separation among the four algorithms at medium cumulative distances. That is, in terms of increasing performance, the order is OS, LDFS, RS, and HFS. At higher distances, HFS still performs well, while the other three algorithms drop below 0.05 for fraction of remaining ID-SNs. For $T = 7$, OS performs very poorly, even worse than HFS with $T = 3$. It drops quickly at approximately 0.8×10^5 m cumulative distance travelled. LDFS and RS are similar and decrease with approximately constant slope. Finally, HFS performs the best, especially when $\rho = 10$ tags/m^2. In summary, HFS is the best choice, at almost all regions of operation. However, HFS requires a hiker counting IDs from previous encounters, which becomes nontrivial for a handheld device if hiker IDs are very long. OS does not require any difficult computations, but has poor performance. A good compromise is RS, which provides decent performance with minimal computational effort (only a random number generator is required). LDFS is the worse choice since it requires computational resources similar to HFS, but its performance is on par with RS, if not worse.

We also consider the sum of next (ID,SN) inter-tag squared distances, multiplied by π. That is, of the remaining ID-SNs of the first hiker, first order them according to increasing sequence numbers. Then, sum up the squared distances between their corresponding consecutive tags and multiple the result by π. For example, if the remaining (ID,SN) pairs are $(1, 244), (1, 9), (1, 259)$, and $(1, 43)$, corresponding to tag positions r_a, r_b, r_c, and r_d, respectively, then the metric we consider is

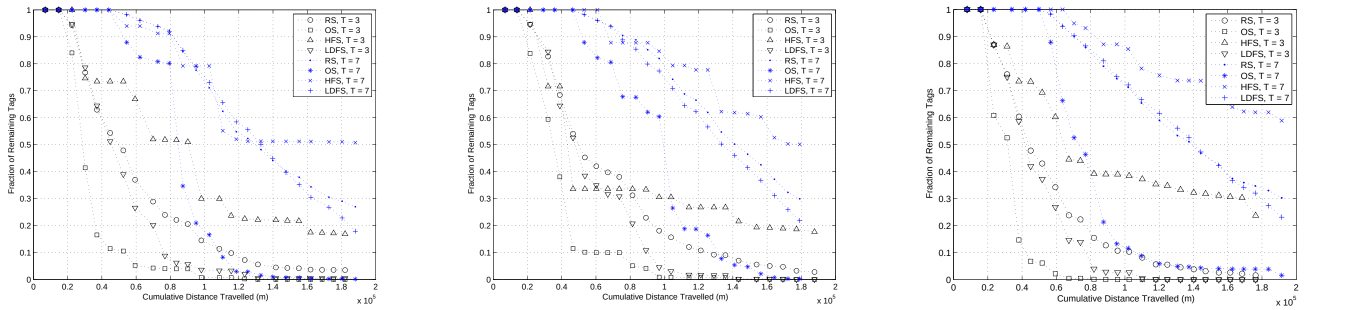
$$\pi \left(||r_d - r_b||^2 + ||r_a - r_d||^2 + ||r_c - r_a||^2 \right) \qquad (7)$$

This search metric reflects a cost that models a typical circular search pattern. For example, when searchers are following the digital path, they want to find the tag with the next higher (ID,SN) pair. They can use a circular search pattern to achieve this. The search metric then is a measure of the search time (which is naturally important in search and rescue). Alternatively, if a searcher is equipped with a strong RFID readers with customizable scan ranges, she can increase the scan range until the next tag is found. The search metric then reflects the power consumed by readers. Results are shown in Fig. 7 for $\rho \in \{1, 5, 10\}$ tags/m^2 and $T \in \{3, 7\}$. We see again that OS has poor performance for $T = 3$. (ID,SN) pairs are quickly deleted and thus the search metric becomes large. Indeed, if we compare Figs. 6 and 7, the search metric shoots up when the fraction of remaining (ID,SN) pairs drops quickly. HFS for both $T = 3$ and 7 perform fairly well. The search metric stays stable and low even as the cumulative distance travelled increases, unsurprisingly, since HFS performs the best in Fig. 6. Surprisingly, RS and HFS for $T = 3$ performs actually better than all four algorithms for $T = 7$ when $\rho = 10$ tags/m^2. In other words, too high a tag density actually hurts the search metric. A searcher can already sufficiently follow a path without having the tags too close to each other. Therefore, for $\rho = 10$ tags/m^2, $T = 3$ means that (ID,SN) pairs are deleted sooner, thus effectively reducing the tag density, allowing for a lower search metric.

V. Conclusion

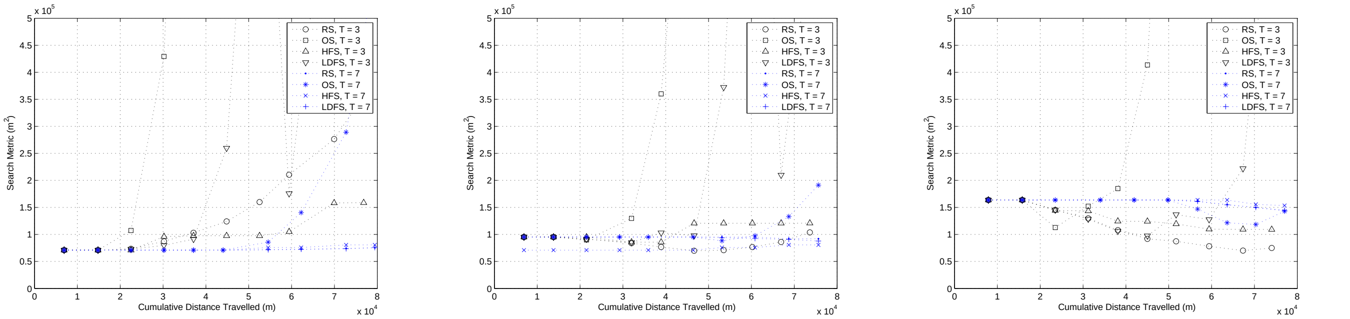
In this paper, we introduce a distributed RFID tag storage infrastructure for forest search and rescue. We provide four replacement algorithms that hikers can use to share the constrained tag memories. We evaluate the system through analysis and simulations in a realistic setting using the trails from a national park. Results indicate that HFS performs well, but requires hikers to remember the number of ID encounters in the past, for each hiker ID.

In future work, we will continue investigating additional ways to share the constrained tag memories. Specifically, we aim to enhance the indirect communication between hikers



(a) $\rho = 1$ tag/m^2 (b) $\rho = 5$ tags/m^2 (c) $\rho = 10$ tags/m^2

Fig. 6. Fraction of remaining tags with (ID,SN) pairs of the first hiker. $\rho \in \{1, 5, 10\}$ tags/m^2



(a) $\rho = 1$ tag/m^2 (b) $\rho = 5$ tags/m^2 (c) $\rho = 10$ tags/m^2

Fig. 7. Search metric. $\rho \in \{1, 5, 10\}$ tags/m^2

through reading and writing to tags. As well, we will continue to investigate more dynamic environments. This involves studying changes internal to the system itself (e.g. tags failing due to poor physical conditions), as well as changes external to the system (e.g. hikers adding and removing tags as they move through the forest).

References

[1] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and Localization with RFID Technology," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, Apr. 2004, vol. 1, pp. 1015-1020.

[2] A. Kleiner, J. Prediger, and B. Nebel, "RFID Technology-based Exploration and SLAM for Search and Rescue," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, Oct. 2006, pp. 4054-4059.

[3] "Search Teams Put RFID to the Rescue to Help Find the Missing," *RFID Journal*. [Online]. Available: http://www.rfidjournal.com/article/articleview/3362/1/1/

[4] J. Bohn and F. Mattern, "Super-distributed RFID Tag Infrastructures," *Lecture Notes in Computer Science*. Berlin, Germany. Springer, 2004. vol. 3295.

[5] Z. Yang and H. Wu, "Featherlight Information Network with Delay-Endurable RFID Support (FINDERS)," in *Proc. IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Rome, Italy, Jun. 2009, pp. 55-63.

[6] J.R. Guerrieri, M.H. Francis, P.F. Wilson, T. Kos, L.E. Miller, N.P. Bryner, D.W. Stroup, and L. Klein-Berndt, "RFID-assisted Indoor Localization and Communication for First Responders," in *Proc. European Conference on Antennas and Propagation (EuCAP)*, Nice, France, Nov. 2006.

[7] J.-H. Huang, S. Amjab, and S. Mishra, "CenWits: A Sensor-Based Loosely Coupled Search and Rescue System Using Witnesses," in *Proc. ACM Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, Nov. 2005, pp. 180-191.

[8] P. Zhuang, Q. Wang, Y. Shang, H. Shi, and B. Hua, "Wireless Sensor Network Aided Search and Rescue in Trails," in *Proc. ACM International Conference on Scalable Information Systems (INFOSCALE)*, Suzhou China, Jun. 2007.

[9] "EPCglobal UHF Class 1 Gen 2," *EPCglobal*. [Online]. Available: http://www.epcglobalinc.org/standards/uhfc1g2.

[10] "Gen-2 UHF RFID Front-end with 32 kB memory," *ICsense*. [Online]. Available: http://www.icsense.com/ic-design/business-cases/4.

[11] "High Memory & Functionality RFID Chips," *Tego*. [Online]. Available: http://www.tegoinc.com/chips.htm.

[12] C. Law, K. Lee, and K.-Y. Siu, "Efficient Memoryless Protocol for Tag Identification," in *Proc. ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Boston, MA, Aug. 2000, pp. 75-84.

[13] H. Vogt, "Multiple Object Identification with Passive RFID Tags," in *Proc. IEEE Conference on Systems, Man, and Cybernetics (SMC)*, Hammamet, Tunisia, Oct. 2002.

[14] "ALR-9650 Single-Antenna RFID Solution," *Alien Technology*. [Online]. Available: http://www.alientechnology.com/readers/alr9650.php.

[15] "NFC Research: Devices," *Near Field Communication Research Lab*. [Online]. Available: http://www.nfc-research.at/index.php?id=45.

[16] J.T. Savolainen, H. Hirvola, and S. Iraji, "EPC UHF RFID Reader: Mobile Phone Integration and Services," in *Proc. IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, Jan. 2009, pp. 521-525.

[17] A. Löffler, U. Wissendheit, H. Gerhäuser, and D. Kuznetsova, "GIDS - A System for Combining RFID-based Site Information and Web-based Data for Virtually Displaying the Location on Handheld Devices," in *Proc. IEEE International Conference on RFID (RFID)*, Las Vegas, NV, Apr. 2008, pp. 312-319.

[18] L. Pohjanheimo, H. Keränen, and H. Ailisto, "Implementing TouchMe Paradigm with a Mobile Phone," in *Proc. ACM International Conference*

on Smart Objects and Ambient Intelligence, Grenoble, France, Oct. 2005, vol. 21, pp. 87-92.

[19] S. Hoyt, D. St. John, D. Wilson, and L. Bushnell, "A Tree Tour with Radio Frequency Identification (RFID) and a Personal Digital Assistant (PDA)," in *Proc. IEEE Annual Conference of the Industrial Electronics Society (IECON)*, Roanoke, VA, Nov. 2003, vol. 3, pp. 2035-2040.

[20] "In the forest," *discover rfid*. [Online]. Available: http://www.discoverrfid.org/what-is-possible/work-better/in-the-forest.html.

[21] "Day Hikes in Yosemite Valley," *National Park Service*. [Online]. Available: http://www.nps.gov/yose/planyourvisit/valleyhikes.htm.

[22] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. Wiley, May 2003.