TECHNICAL REPORT
APRIL 2009
RFID Trees: A Distributed RFID Tag Storage Infrastructure to Backtrack Hikers in a Forest*

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

Roy H. Campbell
Department of Computer Science
University of Illinois at Urbana-Champaign
Email: rhc@illinois.edu

Abstract

In this paper, we propose embedding RFID tags in trees in a forest to track hikers. Hikers are equipped with RFID readers, which read from and write to tags. Specifically, as a hiker moves through the forest, his/her reader leaves his/her ID and increasing sequence numbers (SNs) in tags. This creates a digital trail that allows the hiker to backtrack his/her route. That is, when the hiker decides to leave the forest, he/she scans for tags, following a path of tags with his/her ID and decreasing SNs. During backtracking, if a tag with a valid ID-SN pair is not nearby the current hiker's location, he/she has to wander around until he/she does find such a tag. Therefore, to minimize the backtracking time, we wish to avoid physical gaps in a hiker's digital trail, which are created if an ID is repeatedly deleted in tags near each other. These deletions occur because ID-SN pairs are overwritten in tags. That is, since tag memories are constrained, if there is no more space in a tag, a hiker leaving his/her ID-SN pair first deletes an existing ID-SN pair, according to one of three algorithms. In Random Selection (RN), the hiker randomly chooses an existing ID-SN pair to delete. In High Frequency Selection (HI), each hiker keeps a record of the frequencies of IDs it has seen thus far from previous tag encounters. The hiker chooses the ID-SN pair with the highest ID frequency to delete. High Frequency + Waiting Selection (HI+WT), is an extension of HI. After a hiker deletes an ID, that ID is immune from being deleted for a fixed number of consecutive tag encounters. Simulations indicate that HI+WT minimizes the distance travelled by a hiker during backtracking.

1. Introduction

A tracking system determines the locations of mobile objects, animals, or people in time. Passive RFID (radio frequency identification) technology is an attractive solution for tracking. Typically, passive RFID tags are attached to the mobile entities. Thereafter, a system of RFID readers scan the tags, and thus determine the entities' movements. The tags are small, inexpensive, and easily affixable to various objects. A tag draws power from the electromagnetic signal of a reader scanning it, thus eliminating the need for a battery [1]. In other situations, tags are not affixed to the objects, but facilitate the tracking system in another manner. In this paper, we consider this latter approach. We propose a mechanism of tracking hikers in a forest by dynamically storing information in tags embedded in trees.

Hikers typically carry navigation aids such as maps, compasses, and GPS (global positioning system) devices. Despite the prevalence of these tools, hikers often are lost. Maps and compasses sometimes are not useful to a layman hiker. In a highly wooded area, a GPS device may not function. We propose a solution motivated by the following physical ideas that help hikers navigate through a forest even without such tools.

- The authorities managing the forest (or national park for example) can post signage indicating hiking routes. These can be simple markers painted on trees, or full-fledged route maps installed at crossroads.
- Hikers can leave physical markers as well. This may be for the benefit of themselves or other hikers. For example, a hiker can carve an arrow on the bark of a tree.
- Hikers can leave physical tracks. For example, a hiker can leave a trail of rocks as he/she goes through the forest, similar to the German fairy tale, Hänsel and Gretel. Retracing the rocks backwards allows him/her to exit the forest.

1.1. Tagged Trees

We propose embedding RFID tags in trees. For example, we can store static map information in the tags. Hikers equipped with RFID readers (potentially integrated in their mobile phones) read information from the tags, in order

This work is supported by NSF grant CNS-0519817.

to assist them as they navigate through the forest. This is merely a digital version of the aforementioned idea of forest authorities posting signage. If tags are damaged or removed by people and/or animals, however, the system fails. Alternatively, we use the tags more dynamically in this paper. Hikers read and write information to tags as they move through the forest, allowing for a plethora of possibilities, including tracking. For example, a hiker leaves an identifier (ID) and increasing sequence numbers (SNs) in tags. This information forms a trail to allow him/her to backtrack a path. The digital trails can also be used to follow hikers. In the case of search and rescue, emergency responders can use the trail to locate lost hikers. In the unfortunate case of lost of life, the authorities can use the information in the afterward investigation. That is, we are considering the latter two aforementioned navigation physical ideas in digital form.

Implementing this system requires deploying tags and developing hardware and software. At one extreme, the forest authorities are responsible for embedding tags into trees and keeping an inventory of them. They also provide hardware and/or software to hikers, maintaining tight control over tags. At the other 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, we do not have to worry about battery consumption of tags. Furthermore, information does not have to be aggregated by access points. Instead, we use mobile readers to read from and write to the tags, allowing information to dynamically and physically 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 tracking hikers in a tagged forest.

1.2. Backtracking Hiker Routes

In this paper, we propose hikers storing their IDs and increasing SNs in tags as they move forward. ("Hiker" will sometimes refer to the person, sometimes refer to his/her associated reader device, and sometimes refer to both. The context will be clear.) Note that hikers generally do not know which particular trees have tags. That is, a reader can periodically scan for tags, automatically reading from and writing to them, even if the hiker is oblivious to the RFID communications as he/she progresses through the forest.

When a hiker decides to exit the forest, his/her reader starts to scan for tags containing his/her ID. The hiker can wander (randomly search for tags nearby), and try to follow a path of tags with his/her ID and decreasing SNs. That is, assuming an approximately circular RFID communications range, and using indicators such as received signal strength, the reader is able to give the hiker a general direction to follow. For example, at each step, the reader can increase its transmit power until a tag with a valid ID-SN pair is found. ("Valid" means the ID is associated with the reader and the SN is smaller than what the hiker has already seen thus far during backtracking.) The hiker can then try to move into the vicinity of that tag, before searching for a next tag with a next valid ID-SN pair. Note that the directions may be rather imprecise due to tags possibly being too far apart, causing the hiker to have to wander in finding a next tag containing a next valid ID-SN pair. That is, the maximum reader transmit power may not be sufficient for the reader to immediately find a next tag. The hiker thus approximately follows the digital trail that he/she had left behind to exit the forest.

If we use the time required for a hiker to exit the forest as a performance measure, our system improves or degrades gradually, according to marginal changes in the spatial density of tags in the forest. For example, if we start out with a sparse deployment of tags, a hiker has to often wander, since the reader may often not be able to scan for a next tag containing a next valid ID-SN pair. We call this lack of continuity in a digital trail as a physical gap. As more tags are slowly added to the forest, hikers 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 will happen sparsely 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 with his/her own if the tag is already full. As we add more hikers, more physical gaps in the digital trails are created due to ID-SN pair replacing, causing the same problem as that of sparse tag deployment.

In this paper, we consider three algorithms a hiker can use if he/she replaces an existing ID-SN pair with his/her own. In Random Selection (RN), an existing ID-SN pair is randomly chosen to be deleted. In High Frequency Selection (HI), hikers keep a record of the frequencies of IDs they have seen thus far in previous tags. If a hiker has to delete an ID-SN pair, it chooses the ID with the highest frequency in its record. High Frequency + Waiting Selection (HI+WT), is an extension of HI. After a hiker deletes an ID-SN pair, that ID is immune from being deleted for a fixed number of consecutive tag encounters.

The rest of the paper is organized as follows. In Section 2, we present relevant literature, including supporting technologies that indicate the feasibility of our proposed system. In Section 3, we present the three ID deleting algorithms in detail. In Section 4, we simulate our system using the three algorithms, and discuss the results. Finally, Section 5 concludes the paper.

2. Background Literature

We first present literature associated with tracking tagged objects. Typically, each object is affixed with a tag. That is, there is a one-to-one relationship between an object and a tag. We can thus model the problem as tracking tags with certain mobility patterns. Next, we present literature where tags are not affixed to the objects we are tracking. That is, tags assist the system in other (and often more novel) ways. Our work is more aligned to this latter paradigm. We also contrast our scheme to CenWits [2], a search and rescue system for hikers in a forest using a wireless sensor network. Finally, we present supporting technologies that indicate the feasibility of our proposed system.

2.1. Tracking Tagged Objects

Today, RFID is predominantly used by large companies such as Walmart and Walgreens to track their tagged inventory [3], [4]. This allows companies to accurately and cost-effectively track products as they move from the supplier to the consumer. The products are transported in containers and assisted by other expensive apparatus as they move through the supply chain. These objects can also be tracked (called mobile asset management) by affixing tags to them [5].

In RFID-enabled libraries, tags are affixed to books. Librarians can use handheld readers or the readers can be installed at checkout points. This tracking of books allows for accurate inventorying, automatic checkout, and theft prevention [6], [7].

2.2. Tracking Untagged Objects

The authors in [8] use a mobile robot to determine the locations of fixed tags in space. This creates an RFID map which the robot then uses to localize itself, as well as track the movements of other mobile objects.

The authors in [9] propose a "super-distributed" tag infrastructure. Citing the mu-Chip [10] as a small and inexpensive tag candidate, they envision deploying tags in space over large areas in a highly dense and redundant fashion. In applications built on top of such an infrastructure, the identity and behavior of a single tag becomes irrelevant. Instead, application performance depends on the system as a whole. The authors mention that vehicles may leave traces by writing IDs to tags. The path can be retraced by the vehicle or other vehicles, and overwritten slowly in time. These ideas are only briefly mentioned in the paper without further exposition. In our work, we further develop their ideas in the context of hikers in a tagged forest. Specifically, our work differs since we concentrate on leaving ID trails for the purpose of backtracking. 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.

2.3. Comparison with CenWits

Search and rescue for hikers in a forest is addressed in [2], in a system called CenWits (**C**onnectionless S**en**sor-Based Tracking System Using **Wit**ness**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, he/she can upload his/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 his/her witnesses. The authors in [11] provide optimizations to CenWits.

Although we do not specifically address search and rescue in this paper (though it is a natural future work, since the digital trails can be used for this purpose), we note the salient differences in our infrastructure compared to CenWits. Our system is based on hiker trails instead of specific location information. As a result, we do not require GPS (or any other explicit location determination mechanism). In CenWits, access points are used to collect location information. Since access points are expensive to deploy and maintain in a forest, 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. (We propose trees and argue below for the feasibility of this.) 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 him/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.

2.4. Supporting Technologies

Our proposed system relies on tags as storage devices. RFID technology was originally developed to have tags replace bar codes in tracking inventory. As a result, many tags adhering to the EPCglobal UHF Class 1 Gen 2 standard [12] have read-only memory for the EPC (Electronic Product Code), but have very little re-writeable memory. Nonetheless, many companies today are realizing the potential of RFID tag storage and manufacturing tags specifically for this purpose. GAO RFID and Atmel produce high frequency tags with 10 thousand and 64 thousand bits of re-writeable memory, respectively [13], [14]. For example, suppose we use a tag containing only 3200 bits of re-writeable memory in our proposed system. We allow each tag to hold 100 ID-SN pairs, each 32 bits in length. We assign 16 bits to the ID field, allowing for 65536 simultaneous hikers with unique IDs. The remaining 16 bits are for the SN field, allowing for 65536 SNs. Since a reader can continuously write ID-SN pairs to tags as a hiker moves through the forest, many bits may be required for the SN field. Alternatively, the reader may choose to increase the SN only after a certain number of ID-SN pair writes, especially if the hiker moves slowly.

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 him/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. They use a Texas Instruments glass capsule tag, 4 mm in diameter and 23 mm in length. 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.

3. Algorithms

In this paper, we consider three algorithms a hiker can use if he/she replaces an existing ID-SN pair with his/her own. In all three algorithms, if there is remaining space in the tag, the hiker writes to an empty memory location. If the hiker's ID is already in the tag, he/she replaces the existing SN with a larger one. Otherwise, the hiker chooses an existing ID-SN pair to delete. This is where the three algorithms differ. Intuitively, we would like hiker IDs to be distributed uniformly in space among the tags to reduce wandering time. Specifically, for each ID, we want to minimize large physical gaps between tags.

3.1. Random Selection (RN)

In Random Selection (RN), if necessary, the hiker randomly chooses an ID-SN pair to delete without any bias. That is, the random choice is uniform. RN aims to minimize the chance a hiker deletes the same ID in consecutive tag encounters. The reader uses minimal computational resources in RN.

3.2. High Frequency Selection (HI)

In High Frequency Selection (HI), each hiker keeps track of the ID frequencies he/she has seen thus far from previous tag encounters. If a hiker needs to delete an ID-SN pair, he/she picks the ID with the highest frequency. Intuitively, HI avoids deleting lower frequency IDs, since this would potentially create gaps. Each hiker has the small cost of maintaining a list of ID frequencies in HI

3.3. High Frequency + Waiting Selection (HI+WT)

High Frequency + Waiting Selection (HI+WT) slightly modifies HI. If an ID is deleted by a hiker, it is immune from being deleted for the next α tag encounters for that hiker. Intuitively, if there is an ID with a very high frequency, it is likely that the hiker deletes it in consecutive tag encounters according to HI, possibly creating gaps. Therefore, HI+WT alleviates this problem by protecting an ID for a short period of time after it is initially deleted from a tag. In HI+WT, each hiker has the additional cost of storing a counter for each ID.

4. Simulations

4.1. Simulation Model

The simulation model is a simplified description of our system described in Section 1.2. This provides a tractable

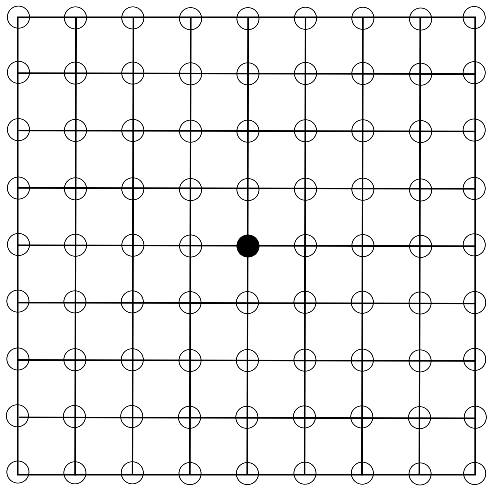


Figure 1. Forest grid with $d = 4$. Each point on the grid represents a tree in the forest. The circles represent tags embedded in trees. The disk at the grid center (starting point) represents many tags.

method to test our ideas. Nonetheless, our model is detailed enough to compare our algorithms in a practical setting. Specifically, we avoid the unrealistic random walk mobility pattern (and associated variations) [21] when describing hikers.

4.1.1. Forest Grid. A square grid with length $2d + 1$ represents the forest. At each grid point, there is a tree with one embedded tag, except for the tree at the grid center point, which has a large number of tags. Fig. 1 shows the grid. Each tag can store m ID-SN pairs. There are n hikers, each with a unique ID, initially located at the grid center (starting point). All hiker IDs are stored in the tags at the starting point. This models a dense deployment of tags in the forest. The starting point models a checkpoint or entry point in a forest, where the forest authorities can feasibly provide a large amount of memory storage.

4.1.2. Forward Hiker Movement. Hikers move along the grid in single steps. Initially, each hiker independently chooses four random numbers uniformly and independently between zero and one. These numbers are then normalized so that they sum to one. The numbers form the a priori direction distribution of the hiker moving {East, South, West, North}, respectively, in each step. This mobility pattern models hikers having predetermined directions and/or destinations when hiking.

Initially, all n hikers are located at the starting point. There are d rounds of movement and tag reading and writing. In each round, each hiker goes through the following four phases.

- The hiker chooses a direction to move according to his/her a priori direction distribution and moves a single step to the adjacent grid point.
- The hiker reads the ID-SN pairs from the tag at the new grid point.
- If there is no more space in the tag, the hiker deletes an ID-SN pair in the tag according to one of the three algorithms {RN, HI, HI+WT}.
- The hiker writes his/her own ID and an increased SN to the tag.

Note that hikers follow a predetermined order. That is, in each round, the l^{th} hiker completes all four phases before the $l + 1^{st}$ hiker begins his/her four phases, where $l \in \{1, \ldots, n - 1\}$. It is possible, therefore, that an ID-SN pair is written to a tag by a hiker, and then subsequently deleted by another hiker, within the same round. As well, since the grid has length $2d+1$, all hikers remain inside the grid at the end of the d rounds.

4.1.3. Hiker Backtracking. After d rounds of hikers moving forward, they attempt to backtrack their routes to the starting point. The hikers backtrack their paths by finding the IDs in the tags they left behind. They may not follow the exact path back, since IDs may have been deleted, and a tag may have been visited by a hiker more than once. Nonetheless, hikers can use the stored SNs to guarantee that they are always making progress towards the starting point.

A hiker uses his/her reader to scan for nearby tags with valid ID-SN pairs. The hiker increases his/her reader's transmit power up to a limit until he/she is successful. If no valid ID-SN pairs are found, the hiker is rendered lost. (In the simulation model, we assume such hikers stay at the lost point and do not reach the starting point. In practice, this may mean the hiker takes a very long time to exit the forest because he/she has to wander to reach the next tag. For simplicity, we consider such hikers as "lost".) We illustrate hiker backtracking through an example shown in Fig. 2. The hiker is located at the North East grid point indicated by the cross mark. The hiker first scans for tags within a Manhattan distance of one step. This is indicated by the inner concentric dotted circle. He/she does not find a tag with a valid ID-SN pair. He/she then scans for tags within two steps. This is indicated by the outer concentric dotted circle. The hiker continues this process up to a range of b steps. If it is still unsuccessful after b scans, it is rendered lost. Suppose that $b \geq 2$ and a valid ID-SN pair is in the tag located at the point indicated by the triangle. When the hiker makes the second scan (indicated by the outer concentric dotted circle), he/she has a positive result. He/she then proceeds to locate the tag by checking all the tags within a range of two steps (by moving to those tags). The hiker first makes a guess as to which tag to check first. In this case, suppose his/her a priori direction distribution is largest in the North direction. That means he/she should move South first when checking. (It is opposite snce we are backtracking.) Therefore, the hiker first

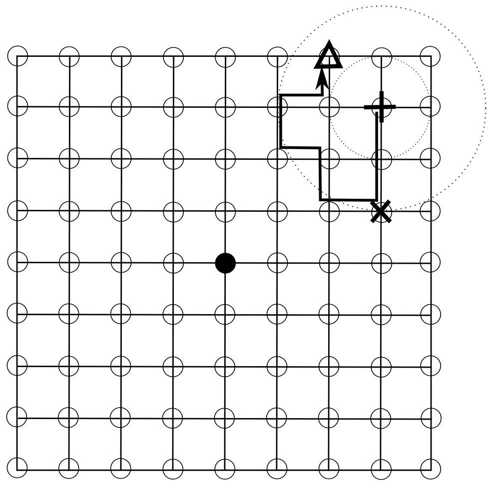


Figure 2. Hiker backtracking. The hiker is located initially at the cross mark. Using increasing transmit powers for the reader, the hiker detects the tag at the triangle. He/she then follows a path (arrow) to the triangle.

goes to the location marked with an x, and then proceeds to move in a somewhat circular motion, checking tags within a range of two steps, until he/she reaches the tag at the triangle. The arrow shows the path from the cross mark to the x, and finally to the triangle. This takes eight steps. Once the hiker reaches the triangle, the process repeats, and he/she searches for a next tag with a next valid ID-SN pair. Backtracking is complete when the hiker is lost or he/she reaches the starting point.

4.2. Simulations and Discussion

We simulate our system using a grid of length $2 \times 100 + 1 = 201$ ($d = 100$). Each tag can hold $m \in \{5, 10, \ldots, 50\}$ ID-SN pairs. $n = 1000$ hikers move forward $d = 100$ steps and attempt to backtrack their routes. The readers can scan up to a range of $b \in \{3, 4, 5\}$ steps. All three algorithms are simulated. $\alpha = b$ in HI+WT.

We consider three performance metrics. s_{avg} is the average Manhattan distance a hiker is from the starting point after it finishes backtracking. That is, if a hiker is able to backtrack to the starting point, his/her distance is zero. If the hiker becomes lost, his/her distance is measured from the final tag he/she reaches to the starting point. s_{avg} is averaged over all n hikers. f_{done} is the fraction of hikers that successfully reach the starting point. s_{avg} and f_{done} are shown in Figs. 3 and 4, respectively.

Figs. 3 and 4 indicate that increasing m improves performance, as expected. As m is made large, there are fewer chances that ID-SN pairs need to be deleted. As a result, the system is less affected by other parameters and algorithmic differences, causing the plots to tend to each other as m is increased. Nonetheless, there are still significant differences for the different cases at $m = 50$. Increasing b improves performance, also as expected. For example, if $b = 3$, hikers scan up to $4 + 8 + 12 = 24$ surrounding tags for a valid ID-SN pair. If $b = 4$, hikers scan up to $4 + 8 + 12 + 16 = 40$ surrounding tags. Therefore, the probability of finding the next tag during backtracking greatly increases with increases in b. We see that HI improves over RN, and HI+WT improves over HI, verifying our intuitions in Section 3. In particular, we see that the HI+WT plots perform very well. The plots asymptotically approach 0 and 1 for s_{avg} and f_{done}, respectively.

These results indicate that HI+WT is a viable candidate for implementing our proposed backtracking hiking system. The only cost of the algorithm is maintaining ID frequencies and counters. That is, the computational resources required to run the algorithm itself are minimal. As mobile devices continue to evolve and provide more storage, the HI+WT becomes a very attractive solution.

5. Conclusion

In this paper, we propose embedding RFID tags in trees, which is a feasible venture, as detailed in Sections 1.1 and 2.4. Hikers going through a forest read and write information to the tags, allowing for the tracking of hikers. We consider the specific problem of hikers leaving their IDs in tags and using that information to backtrack their routes. We provide three algorithms a hiker uses if he/she has to delete an existing ID-SN pair. Simulations indicate that the High Frequency + Waiting Selection (HI+WT) algorithm performs best.

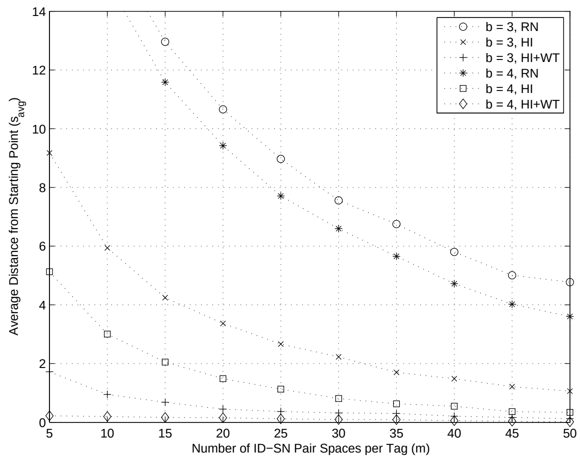
Future work includes investigating additional backtracking algorithms with more general system models. As well, we wish to further investigate how a dynamic and distributed storage infrastructure can be leveraged in different applications. RFID is an excellent technology to implement such an infrastructure and can provide many practical solutions to real-world problems.

References

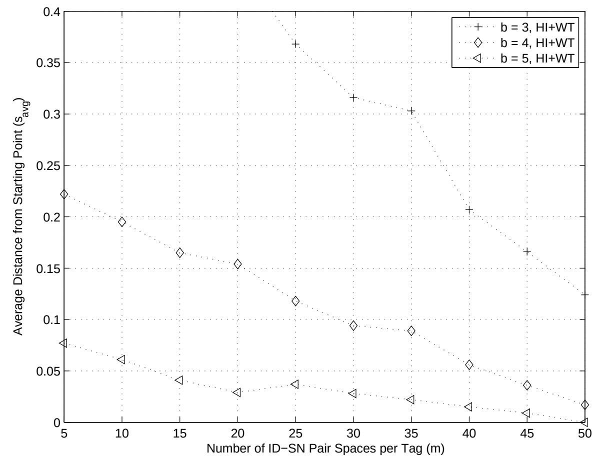
[1] D. M. Dobkin, *The RF in RFID*. Oxford, UK: Elsevier, 2008.

[2] J.-H. Huang, S. Amjab, and S. Mishra, "Cenwits: A sensor-based loosely coupled search and rescue system using witnesses," in *Proc. 2005 ACM Conference on Embedded Networked Sensor Systems*, San Diego, CA, Nov. 2005, pp. 180–191.

[3] (2009, Feb.) RFID news: Looking back at the wal-mart RFID time line. Supply Chain Digest. [Online]. Available: http://www.scdigest.com/assets/newsviews/09-02-23-1.pdf

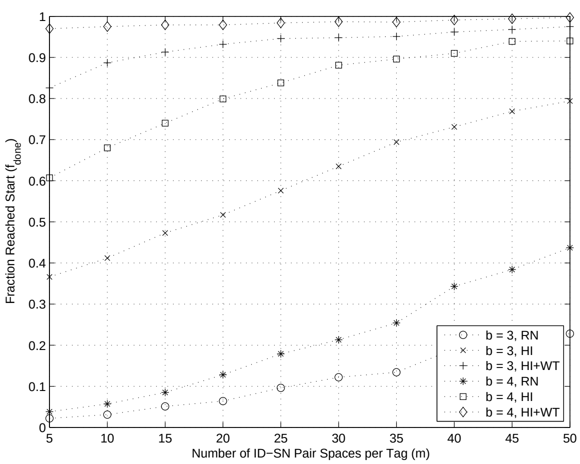


(a) $b \in \{3, 4\}$, {RN, HI, HI+WT}

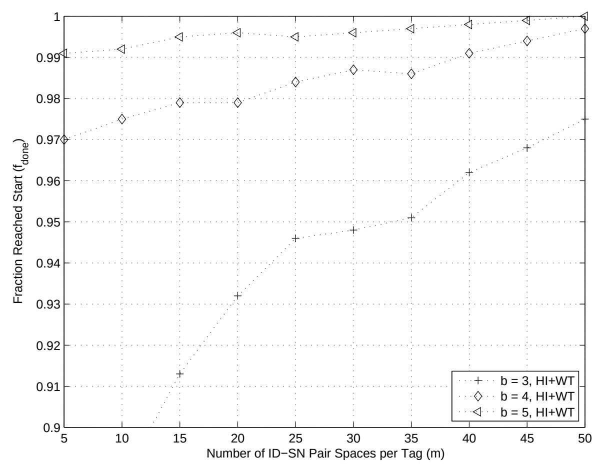


(b) $b \in \{3, 4, 5\}$, HI+WT

Figure 3. Average Distance from Start (s_{avg})



(a) $b \in \{3, 4\}$, {RN, HI, HI+WT}



(b) $b \in \{3, 4, 5\}$, HI+WT

Figure 4. Fraction Reached Start (f_{done})

[4] (2009, Mar.) Walgreens: RFID promotions tracking a game-changer. RFID Update. [Online]. Available: http://www.rfidupdate.com/news/03032009.html

[5] T. Inaba, "Value of sparse RFID traceability information in asset tracking during migration period," in *Proc. 2008 IEEE International Conference on RFID*, Las Vegas, NV, Apr. 2008, pp. 183–190.

[6] (2007, Oct.) Bibliotheca. Bibliotheca RFID Library Systems AG. [Online]. Available: http://www.bibliotheca-rfid.com/

[7] (2007) LibBest library information system. BookTec Information Co. [Online]. Available: http://www.rfid-library.com/

[8] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and localization with RFID technology," in *Proc. 2004 IEEE International Conference on Robotics and Automation*, vol. 1, New Orleans, LA, Apr. 2004, pp. 1015–1020.

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

[10] The world's smallest RFID IC mu-chip. Hitachi. [Online]. Available: http://www.hitachi.co.jp/Prod/mu-chip/

[11] P. Zhuang, Q. Wang, Y. Shang, H. Shi, and B. Hua, "Wireless sensor network aided search and rescue in trails," in *Proc. 2007 ACM Conference on Scalable Information Systems*, Suzhou, China, Jun. 2007.

[12] EPCglobal UHF class 1 gen 2. EPCglobal. [Online]. Available: http://www.epcglobalinc.org/standards/uhfc1g2

[13] 13.56 MHz high frequency (HF) rectangle paper RFID tag. GAO RFID. [Online]. Available: http://www.gaorfid.com/index.php?main_page=index&cPath=68

[14] 13.56 MHz cryptorf EEPROM memory 64 kbits. Atmel. [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/5006s.pdf

[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. 2009 IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, Jan. 2009.

[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. 2008 IEEE International Conference on 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. 2005 ACM International Conference on Smart Objects and Ambient Intelligence*, vol. 21, Grenoble, France, Oct. 2005, pp. 87–92.

[19] S. Hoyt, D. S. John, D. Wilson, and L. Bushnell, "A tree tour with radio frequency identification (RFID) and a personal digital assistant (PDA)," in *Proc. 2003 Annual Conference of the IEEE Industrial Electronics Society*, vol. 3, Roanoke, VA, Nov. 2003, 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] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, Aug. 2002.