Probabilistic Optimal Tree Hopping for RFID Identification

Muhammad Shahzad and Alex X. Liu
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan, USA
{shahzadm, alexliu}@cse.msu.edu

ABSTRACT

Radio Frequency Identification (RFID) systems are widely used in various applications such as supply chain management, inventory control, and object tracking. Identifying RFID tags in a given tag population is the most fundamental operation in RFID systems. While the Tree Walking (TW) protocol has become the industrial standard for identifying RFID tags, little is known about the mathematical nature of this protocol and only some ad-hoc heuristics exist for optimizing it. In this paper, first, we analytically model the TW protocol, and then using that model, propose the Tree Hopping (TH) protocol that optimizes TW both theoretically and practically. The key novelty of TH is to formulate tag identification as an optimization problem and find the optimal solution that ensures the minimal average number of queries. With this solid theoretical underpinning, for different tag population sizes ranging from 100 to 100K tags, TH significantly outperforms the *best* prior tag identification protocols on the metrics of the total number of queries per tag, the total identification time per tag, and the average number of responses per tag by an average of 50%, 10%, and 30%, respectively, when tag IDs are uniformly distributed in the ID space, and of 26%, 37%, and 26%, respectively, when tag IDs are non-uniformly distributed.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *Wireless communication*; C.2.8 [**Mobile Computing**]: Algorithm Design and Analysis

General Terms

Algorithms, Design, Performance, Experimentation

Keywords

RFID; Tags; Identification

SIGMETRICS'13, June 17-21, 2013, Pittsburgh, PA, USA.
Copyright 2013 ACM 978-1-4503-1900-3/13/06 ...$15.00.

1. INTRODUCTION

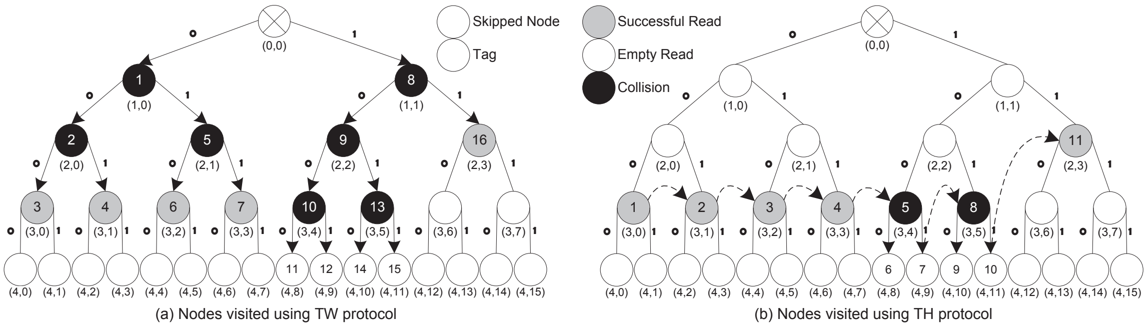
1.1 Background and Problem Statement

As the cost of commercial RFID tags, which is as low as 5 cents per tag [21], has become negligible compared to the prices of the products to which they are attached, RFID systems are being increasingly used in various applications such as supply chain management [13], indoor localization [18], 3D positioning [27], object tracking [17], inventory control, electronic toll collection, and access control [5, 16]. For example, Walmart has started to use RFID tags to track jeans and underwear for better inventory control. An RFID system consists of tags and readers. A tag is a microchip combined with an antenna in a compact package that has limited computing power and communication range. There are two types of tags: (1) passive tags, which do not have their own power source, are powered up by harvesting the radio frequency energy from readers, and have communication ranges often less than 20 feet; (2) active tags, which come with their own power sources and have relatively longer communication ranges. A reader has a dedicated power source with significant computing power. RFID systems mostly work in a query-response fashion where a reader transmits queries to a set of tags and the tags respond with their IDs over a shared wireless medium.

This paper addresses the fundamental RFID *tag identification* problem, namely reading all IDs of a given set of tags, which is needed in almost all RFID systems. Because tags respond over a shared wireless medium, tag identification protocols are also called *collision arbitration*, *tag singulation*, or *tag anti-collision* protocols. Tag identification protocols need to be scalable as the number of tags that need to be identified could be as large as tens of thousands with the increasing adoption of RFID tags. An RFID system with a large number of tags may require multiple readers with overlapping regions. In this paper, we first focus on the *single reader* version of the tag identification problem and then extend our solution to the *multiple reader* problem.

1.2 Summary and Limitations of Prior Art

The industrial standard, EPCGlobal Class 1 Generation 2 (C1G2) RFID [9], adopted two tag identification protocols, namely framed slotted Aloha and Tree Walking (TW). In framed slotted Aloha, a reader first broadcasts a value f to the tags in its vicinity where f represents the number of time slots present in a forthcoming frame. Then each tag whose inventory bit is 0 randomly picks a time slot in the frame and replies during that slot. Each C1G2 compliant tag has



(a) Nodes visited using TW protocol

(b) Nodes visited using TH protocol

Figure 1: Identifying a population of 9 tags over ID space of 2^4 using Tree Walking and Tree Hopping

an inventory bit, which is initialized to be 0. In any slot, if exactly one tag responds, the reader successfully gets the ID of that tag and issues a command to the tag to change its inventory bit to 1. The key limitation of framed slotted Aloha is that it can not identify large tag populations due to the finite possible size of f, which is typically no more than 512 [23]. Qian *et al.* have shown that framed slotted Aloha is most efficient when f is equal to the number of tags [20]. Therefore, although theoretically any arbitrarily large tag population can be identified by indefinitely increasing the frame size, practically this is infeasible because during the entire identification process, Aloha based protocols require all tags, including those that have been identified, to stay powered up and listen to all the messages from the reader in order to maintain the value of the inventory bit. This results in high instability because any intermittent loss of power at a tag will set its inventory bit back to 0, leading the tag to contend in the subsequent frame. The instability of Aloha based protocols has formally been proven by Rosenkrantz and Towsley in [22].

TW is a fundamental multiple access protocol, which was first invented by U.S. Army for testing soldiers for syphilis during World War II [4]. TW was proposed as an RFID tag identification protocol by Law *et al.* in [12]. In TW, a reader first queries 0 and all the tags whose IDs start with 0 respond. If result of the query is a successful read (*i.e.*, exactly one tag responds) or an empty read (*i.e.*, no tag responds), the reader queries 1 and all the tags whose IDs start with 1 respond. If the result of the query is a collision, the reader generates two new query strings by appending a 0 and a 1 to the previous query string and queries the tags with these new query strings. All the tags whose IDs start with the new query string respond. This process continues until all the tags have been identified. This identification process is essentially a partial Depth First Traversal (DFT) on the complete binary tree over the tag ID space, and the actual traversal forms a binary tree where the leaf nodes represent successful or empty reads and the internal nodes represent collisions. Figure 1(a) shows the tree walking process for identifying 9 tags over a tag ID space of size 2^4. Here a `successful read` node is one that an identification protocol visits and there is exactly one tag in the subtree rooted at this node, an `empty read` node is one that an identification protocol visits and there is no tag in the subtree rooted at this node, and a `collision` node is one that an identification protocol visits and there are more than one tags in the subtree rooted at this node. The key limitation of TW based protocols is that they visit a large number of collision nodes in the binary tree. Although several heuristics have been proposed to reduce the number of visits to collision nodes [15, 19], all these heuristics based methods are not guaranteed to minimize such futile visits. Prior Aloha-TW hybrid protocols also have this limitation.

1.3 System Model

As most commercially available tags and readers already comply with the C1G2 standard, we do not assume changes to either tags or the physical protocol that they use to communicate with readers. We assume that readers can be reprogrammed to adopt new tag identification software. For reliable tag identification, we are given the probability of successful query-response communication between the reader and a tag.

1.4 Proposed Approach

To address the fundamental limitations that lie in the heuristic nature of prior TW based protocols, we propose a new approach to tag identification called Tree Hopping (TH). The key novel idea of TH is to formulate the tag identification problem as an optimization problem and find the optimal solution that ensures the minimal expected number of queries (*i.e.*, nodes visited on the binary tree). In TH, we first quickly estimate the tag population size. Second, based on the estimated tag population size, we calculate the optimal level to start tree traversal so that the expected number of queries is minimal, *hop* directly to the left most node on that level, and then perform DFT on the subtree rooted at that node. Third, after that subtree is traversed, we re-estimate the size of remaining unidentified tag population, re-calculate the new optimal level, hop directly to the new optimal node, and perform DFT on the subtree rooted at that node. Hopping to optimal nodes in this manner skips a large number of collision nodes. This process continues until all the tags have been identified. Figure 1(b) shows the nodes traversed by TH for the same population of 9 tags as in Figure 1(a). Here a `skipped node` is one that TW visits but TH does not. We can see that TH traverses 11 nodes to identify these 9 tags. In comparison, TW traverses 16 nodes as shown in Figure 1(a). This difference scales significantly as tag population size increases.

1.4.1 Population Size Estimation

TH first uses a framed slotted Aloha based method to quickly estimate the tag population size. For this, TH requires each tag to respond to the reader with a probability q. As C1G2 compliant tags do not support this probabilistic responding, we implement this by "virtually" extending the frame size $\frac{1}{q}$ times. To estimate the tag population size, the reader announces a frame size of $\frac{1}{q}$ but terminates it after the first slot. The reader issues several single-slot frames while reducing q with a geometric distribution (*i.e.*, $q = \frac{1}{2^{i-1}}$ in the i-th frame) until the reader gets an empty slot. Suppose the empty slot occurred in the i-th frame, TH estimates the tag population size to be $1.2897 \times 2^{i-2}$ based on Flajolet and Martin's algorithm used in databases [6].

1.4.2 Finding Optimal Level

To determine the optimal level γ_{op} that TH directly hops to, we first calculate the expected number of nodes that TH will visit if it starts DFTs from nodes on any given level γ. Let b be the number of bits in each tag ID (which is 64 for C1G2 compliant tags), then, we have $1 \le \gamma \le b$. If γ is small, more collision nodes will be visited while if it is large, more empty read nodes will be visited. Our objective is to calculate an optimal level γ_{op} that will result in the smallest number of nodes visited. To find γ_{op}, we first derive the expression for calculating the expected number of nodes visited by TH if TH directly hops to level γ. Then we calculate the value of γ which minimizes this expression. This value of γ is the value of optimal level γ_{op}. We present the technical details of finding γ_{op} in Section 3.

1.4.3 Population Size Re-estimation

If the tags that we want to identify are uniformly distributed in the ID space $[0, 2^b - 1]$, then performing DFTs from each node on level γ_{op} will result in minimum number of nodes visited. However, in reality, the tags may not be uniformly distributed. In such cases, each time when the DFT of a subtree is finished, TH needs to re-estimate the total tag population size to find the next optimal level and the hoping destination node. TH performs the re-estimation as follows. Let z be the first tag population size estimated using the Aloha based method, x be the number of tags that have been identified, and s be the size of the tag ID space covered by the nodes visited. Naturally, $z - x$ is an estimate of the remaining tag population size; however, we cannot use this estimate to calculate the next optimal level because the remaining leftover ID space may not form a complete binary tree. Instead, based on the node density in the remaining ID space, TH extrapolates the total tag population size to be $\frac{z-x}{2^b-s} \times 2^b$ and uses it to find the next hopping destination node. Note that if tags are uniformly distributed, we have $\frac{z-x}{2^b-s} \times 2^b = z$.

1.4.4 Finding Hopping Destination

Each time after a DFT is done and the new optimal level is recalculated, TH needs to find the next node to hop to, which may not be the leftmost node on the optimal level. Consider the example shown in Figure 1(b). Assuming a uniform distribution, the optimal level to start the DFT is 3. In this paper, we use (l, p) to denote the p^{th} node on level l. TH performs DFTs on the subtrees of nodes $(3, 0)$ to $(3, 5)$ and identifies 8 out of 9 tags. Based on the number of remaining tags after the last DFT, which is 1, the optimal level for the next hop is changed from 3 to 1. However, if TH starts the DFT from the leftmost node on level 1, which is $(1, 0)$, it will result in identifying all tags in its subtree again which is wasteful. Similarly, if TH starts the DFT from the second leftmost node on level 1, which is $(1, 1)$, it will visit the subtree of $(2, 2)$, which is wasteful as all the tags in the subtree of $(2, 2)$ have already been identified. Similarly, if there had been a third leftmost node on the new optimal level and if TH starts the DFT from that third left most node, it will not visit the subtree of $(2, 3)$, resulting in tag $(4, 13)$ not being identified. To avoid both scenarios, *i.e.*, some subtrees being traversed multiple times and some subtrees with tags not being traversed, after the optimal level is recalculated, TH hops to the root of the largest subtree that can contain the next tag to be identified but does not contain any previously identified tag. The level at which this root is located can not be smaller than the new optimal level. For the example in Figure 1(b), after the subtree rooted at node $(2, 2)$ has been traversed, the recalculated optimal level is 1 and the next node that TH hops to is $(2, 3)$.

Our experimental results in Figure 2 show that when the tags are not uniformly distributed in the ID space, our technique of dynamically adjusting γ_{op} according to the leftover population size significantly reduces the total number of queries and the average number of responses per tag. The two curves "TH w re-estimation-Seq" and "TH w/o re-estimation-Seq" show the total number of queries needed, respectively, with and without the dynamic adjustment of γ_{op} for non-uniformly distributed tag IDs. For example, for 10K tags, this dynamic level adjustment reduces the total number of queries by 31.5%. Our experimental results in Figure 2 also show that when the tags are uniformly distributed in the ID space, there is no need to dynamically adjust γ_{op}. The two curves "TH w re-estimation-Uni" and "TH w/o re-estimation-Uni" show the total number of queries needed, respectively, with and without the dynamic adjustment for uniformly distributed tag IDs. These two curves are quite close because for uniformly distributed tag IDs, γ_{op} does not usually change after each DFT and thus the benefit of dynamically adjusting γ_{op} is relatively small.

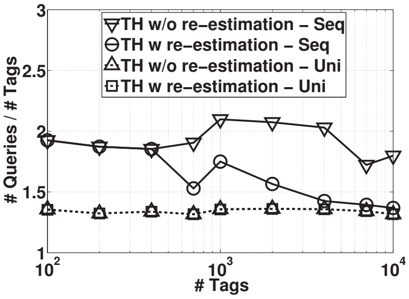


Figure 2: Impact of re-estimation

2. RELATED WORK

We review existing tag identification protocols, which can be classified as nondeterministic, deterministic, or hybrid protocols.

2.1 Nondeterministic Identification Protocols

Existing such protocols are either based on framed slotted Aloha [28] or Binary Splitting (BS) [2]. As we discussed above, Aloha based protocols only work for small tag populations. In BS [2], the identification process starts with the reader asking the tags to respond. If more than one tag responds, BS divides and subdivides the population into smaller groups until each group has only one or no tag. This process of random subdivision incurs a lot of collisions. Furthermore, BS requires the tags to perform operations that are not supported by the C1G2 standard. ABS is a BS based protocol that is designed for continuous identification of tags [14].

2.2 Deterministic Identification Protocols

There are 3 such protocols: (1) the basic TW protocol [12], (2) the Adaptive Tree Walking (ATW) protocol [24], and (3) the TW-based Smart Trend Traversal (STT) protocol [19]. ATW is an optimized version of TW that always starts DFTs from the level of $\log z$, where z is the size of tag population. This is the traditional wisdom for optimizing TW. The key limitation of ATW is that it is optimal only when all tag IDs are evenly spaced in the ID space; however, this is often not true in real-world applications. In contrast, during the identification process, our TH protocol adaptively chooses the optimal level to hop to based on distribution of IDs. STT improves TW using some ad-hoc heuristics to select prefixes for next queries based upon the type of response to previous queries. It assumes that the number of tags identified in the past k queries is the same as the number of tags that will be identified in the next k queries. This may not be true in reality.

2.3 Hybrid Identification Protocols

Hybrid protocols combine features from nondeterministic and deterministic protocols. There are two major such protocols: Multi slotted scheme with Assigned Slots (MAS) [15] and Adaptively Splitting-based Arbitration Protocol (ASAP) [20]. MAS is a TW-based protocol in which each tag that matches the reader's query picks up one of the f time slots to respond. For large populations, due to the finite practical size of $f \leq 512$, for queries corresponding to higher levels in the binary tree, the response in each of the f slots is most likely a collision, which increases the identification time. ASAP divides and subdivides the tag population until the size of each subset is below a certain threshold and then applies Aloha on each subset. For this, ASAP requires tags to be able to pick slots using a geometric distribution, which makes it incompliant with the C1G2 standard. Furthermore, subdividing the population before the actual identification is in itself very time consuming.

3. OPTIMAL TREE HOPPING

After quick population size estimation using Flajolet and Martin's algorithm [6], TH needs to find the optimal level to hop to. First, we derive an expression to calculate the expected number of queries (*i.e.*, the number of nodes that TH will visit) if it starts DFTs from the nodes on level γ, assuming that tags are uniformly distributed in the ID space. Second, as the derived expression is too complex to calculate the optimal value of γ that minimizes the expected number of queries by simply differentiating the expression with respect to γ, we present a numerical method to calculate the optimal level γ_{op}. If tags are not uniformly distributed, each time when the DFT on a node is completed, as stated in Section 1.4, TH re-estimates the total population size based on the initial estimate and the number of tags that have been identified, re-calculates the new optimal level, and finds the hopping destination node. Table 1 summarizes the symbols used in this paper.

Table 1: Notations

Symbol	Description
b	# of bits in tag ID, which is 64 for C1G2 tags
n	size of the whole ID space, which is 2^b
(l, p)	node whose top-to-down vertical level is $1 \leq l \leq b$ and left-to-right horizontal position is $0 \leq p \leq 2^b - 1$
z	estimated number of tags in the population
γ	level from which TH performs DFTs
γ_{op}	optimal level to perform DFTs
q	tag response probability used in estimation
$I(l, p)$	indicator random variable, 1 if (l, p) is visited
T	random variable for total # of nodes visited
$E[T]$	expected # of nodes visited to identify all tags in the population
$P_l\{(l, p)\}$	probability of visiting (l, p) if it is left child
$P_r\{(l, p)\}$	probability of visiting (l, p) if it is right child
m	size of ID space covered by the parent of the current node being visited
k	# of tags covered by the parent of the current node being visited
P_s, P_c, P_e	probabilities of successful read, collision, or empty read at parent of the current node
β	repetitions of query to handle unreliable channel
g	actual probability of reading a tag
u	required probability of reading a tag
V	maximum # of nodes visited to identify all tags
θ_0	# of subtrees covering no tags
θ_1	# of subtrees covering 1 tag

3.1 Average Number of Queries

Let random variable T denote the total number of nodes that TH visits to identify all tags. Note that each node visit corresponds to one reader query. We next calculate $E[T]$. Let $I(l, p)$ be an indicator random variable whose value is 1 if and only if node (l, p) is visited. Thus, T is the sum of $I(l, p)$ for all l and all p.

$$T = \sum_{l=1}^{b} \sum_{p=0}^{2^l - 1} I(l, p) \tag{1}$$

Let $P\{(l, p)\}$ be the probability that TH visits node (l, p). Thus, $E[T]$ can be expressed as follows:

$$E[T] = \sum_{l=1}^{b} \sum_{p=0}^{2^l - 1} P\{(l, p)\} \tag{2}$$

Next, we focus on expressing $P\{(l, p)\}$ using variable γ, where γ denotes the level that TH hops to. Recall that TH skips all nodes on levels from 1 to $\gamma - 1$ and performs DFT on each of the 2^γ nodes on level γ, where $1 \leq \gamma \leq b$. Note that the root node of the whole binary tree is always meaningless to visit as it corresponds to a query of length 0. Here $P\{(l, p)\}$ is calculated differently depending on whether node (l, p) is the left child of its parent or the right. Let $P_l\{(l, p)\}$ and $P_r\{(l, p)\}$ denote the probability of visiting (l, p) when (l, p) is the left and right child of its parent, respectively. If the estimated total number of tags z is zero, then $P_l\{(l, p)\} = P_r\{(l, p)\} = 0$ for all l and p. Below we assume $z > 0$. As TH skips all nodes from levels 1 to $\gamma - 1$, we have

$$P_l\{(l, p)\} = P_r\{(l, p)\} = 0 \textbf{ if } 1 \leq l < \gamma \tag{3}$$

As TH performs DFT from each node on level γ, it visits each node on this level. Thus, we have

$$P_l\left\{(l,p)\right\} = P_r\left\{(l,p)\right\} = 1 \textbf{ if } l = \gamma \tag{4}$$

For each remaining level $\gamma < l \leq b$, when (l,p) is the left child of its parent, $P_l\left\{(l,p)\right\}$ is equal to the probability that the parent of (l,p) is a collision node. When (l,p) is the right child of its parent, if the parent is a collision node and $(l,p-1)$ is an empty read node, then (l,p) will also be a collision node. Thus, instead of visiting (l,p), TH should directly hop to the left child of (l,p). Therefore, $P_r\left\{(l,p)\right\}$ is equal to the probability that the parent of (l,p) is a collision node and $(l,p-1)$ is not an empty read node.

Let k denote the number of tags *covered* by the parent of node (l,p) (*i.e.*, the number of tags that are in the subtree rooted at the parent of (l,p)). Let $m = 2^{b-l+1}$ denote the maximum number of tags that the parent of (l,p) can cover and $n = 2^b$ denote the number of tags in the whole ID space. The probability that the parent of (l,p) covers k of z tags follows a hypergeometric distribution:

$$P\left\{\#\text{tags} = k\right\} = \frac{\binom{m}{k}\binom{n-m}{z-k}}{\binom{n}{z}} \tag{5}$$

Let P_e be the probability that the parent of (l,p) is an empty read. Thus,

$$P_e = P\left\{\#\text{tags} = 0\right\} = \frac{\binom{n-m}{z}}{\binom{n}{z}} \tag{6}$$

Let P_s be the probability that the parent of (l,p) is a successful read. Thus,

$$P_s = P\left\{\#\text{tags} = 1\right\} = \frac{m\binom{n-m}{z-1}}{\binom{n}{z}} \tag{7}$$

Let P_c be the probability that the parent of (l,p) is a collision node. Thus,

$$P_c = 1 - (P_e + P_s) = 1 - \frac{\binom{n-m}{z}}{\binom{n}{z}} - \frac{m\binom{n-m}{z}}{\binom{n}{z}} \tag{8}$$

Next we calculate $P_l\left\{(l,p)\right\}$ and $P_r\left\{(l,p)\right\}$ for $\gamma < l \leq b$ for the following three cases: $n-m < z-1$, $n-m = z-1$, and $n-m > z-1$. Note that $n-m$ is the size of the ID space that is not covered by the parent of (l,p), and $z-k$ is the remaining number of tags that are not covered by the parent of (l,p). Thus, $z-k \leq n-m$.

Case 1: $n - m < z - 1$.

In this case, $z-k \leq n-m < z-1$, which means $k \geq 2$. Thus, as the parent of (l,p) covers at least two tags, it must be a collision node, *i.e.* $P_c = 1$. Thus, if (l,p) is the left child of its parent, TH for sure visits it:

$$P_l\left\{(l,p)\right\} = 1 \tag{9}$$

If (l,p) is the right child of its parent, TH visits it if and only if node $(l,p-1)$, which is the left sibling of (l,p), is not an empty read. If $(l,p-1)$ is an empty read, as its parent is a collision node, (l,p) must also be a collision node, which means that TH will directly visit the left child of (l,p) instead of (l,p). The size of the ID space covered by $(l,p-1)$ is $\frac{m}{2}$. If $n - \frac{m}{2} \leq z-1$, then node $(l,p-1)$ covers at least one tag, which means that $(l,p-1)$ is not an empty read and TH for sure visits (l,p), *i.e.*, $P_r\left\{(l,p)\right\} = 1$. If $n - \frac{m}{2} > z-1$, then the probability that TH visits (l,p) is equal to the probability that $(l,p-1)$ is not an empty read, which is $1 - \binom{n-\frac{m}{2}}{z}/\binom{n}{z}$ based on Equation (6). Finally, we have

$$P_r\left\{(l,p)\right\} = \begin{cases} 1 - \frac{\binom{n-\frac{m}{2}}{z}}{\binom{n}{z}} & \textbf{if } n - \frac{m}{2} > z-1 \\ 1 & \textbf{if } n - \frac{m}{2} \leq z-1 \end{cases} \tag{10}$$

Case 2: $n - m = z - 1$.

In this case, $z-k \leq n-m = z-1$, which means $k \geq 1$. As the parent of (l,p) covers $k \geq 1$ tags, the probability of the parent of (l,p) being an empty read is 0 and the probability of the parent of (l,p) being a successful read is $m\binom{n-m}{z-1}/\binom{n}{z} = m\binom{z-1}{z-1}/\binom{n}{z} = m/\binom{n}{z}$ based on Equation (7). If (l,p) is the left child of its parent, then TH visits it if and only if the parent of (l,p) is a collision node. Thus, the probability of visiting (l,p) is equal to the probability of the parent of (l,p) being a collision node, which is equal to $1 - P_e - P_s$. Thus, we have

$$P_l\left\{(l,p)\right\} = 1 - P_e - P_s = 1 - \frac{m}{\binom{n}{z}} \tag{11}$$

If (l,p) is the right child of its parent, then TH visits it if and only if both the parent of (l,p) is a collision node and $(l,p-1)$ is not an empty read. The probability that the parent of (l,p) is a collision node is $1 - m/\binom{n}{z}$ as calculated above. Given that the parent of (l,p) is a collision node, the probability that $(l,p-1)$ is an empty read is $\left(\binom{n-\frac{m}{2}}{z} - \frac{m}{2}\right)/\left(\binom{n}{z} - m\right)$.

$$P_r\left\{(l,p)\right\} = \left[1 - \frac{m}{\binom{n}{z}}\right].\left[1 - \frac{\binom{n-\frac{m}{2}}{z} - \frac{m}{2}}{\binom{n}{z} - m}\right] \tag{12}$$

Case 3. $n - m > z - 1$.

In this case, $k \geq 0$. Similar to the above calculations, based on Equations (6) and (7), we have:

$$P_l\left\{(l,p)\right\} = 1 - P_e - P_s = 1 - \frac{\binom{n-m}{z} + m\binom{n-m}{z-1}}{\binom{n}{z}} \tag{13}$$

$$P_r\left\{(l,p)\right\} = \left[1 - \frac{\binom{n-m}{z} + m\binom{n-m}{z-1}}{\binom{n}{z}}\right] \times \left[1 - \frac{\binom{n-\frac{m}{2}}{z} - \left\{\binom{n-m}{z} + \frac{m}{2}\binom{n-m}{z-1}\right\}}{\binom{n}{z} - \left\{\binom{n-m}{z} + m\binom{n-m}{z-1}\right\}}\right] \tag{14}$$

Finally, Equations (3) through (14) completely define the probabilities $P_l\left\{(l,p)\right\}$ and $P_r\left\{(l,p)\right\}$. Note that as tags are uniformly distributed, the probability of visiting node (l,p) is independent of the horizontal position p.

The expected number of queries can now be calculated using Theorem 1.

THEOREM 1. *For a population of z tags uniformly distributed in the ID space, where each tag has an ID of b bits, if TH hops to level γ to perform DFT from each node on this level, the expected number of queries for identifying all z tags is:*

$$E[T] = 2^{\gamma} + \sum_{l=\gamma+1}^{b} 2^{l-1}\left[P_l\left\{(l,p)\right\} + P_r\left\{(l,p)\right\}\right] \tag{15}$$

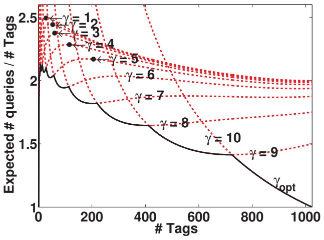


Figure 3: Norm. $E[T]$ vs. population size $\forall\gamma$

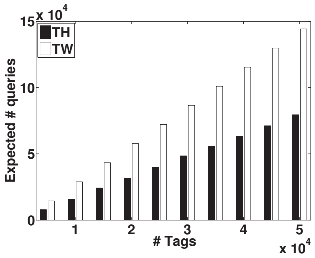


Figure 4: Expected # queries: TH vs. TW

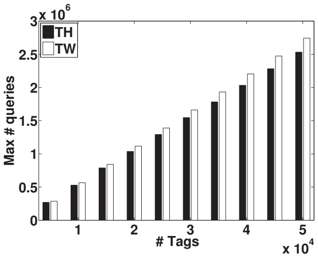


Figure 5: Maximum # queries: TH vs. TW

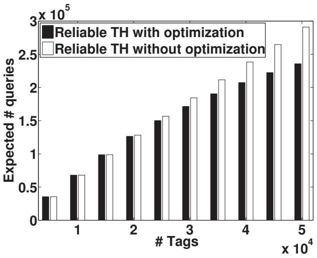


Figure 6: Expected # queries of Reliable TH

PROOF. First, on level γ, all the 2^γ nodes are visited by TH. Second, on any level l where $\gamma+1 \leq l \leq b$, the probabilities of left and right nodes being visited are $P_l\{(l,p)\}$ and $P_r\{(l,p)\}$ respectively. As there are 2^{l-1} pairs of left and right nodes on level l, the expected number of nodes visited by TH on level l is $2^{l-1}[P_l\{(l,p)\} + P_r\{(l,p)\}]$. □

When $\gamma = 1$, Equation (15) is also the analytical model for calculating expected number of queries of TW protocol.

3.2 Calculating Optimal Hopping Level

Equation (15) shows that $E[T]$ is a function of γ as $n = 2^b$, $m = 2^{b-l+1}$, and b is given. For any given z, we want to find the optimal level $\gamma = \gamma_{\text{op}}$ so that $E[T]$ is minimal. The conventional approach to finding the optimal variable value that minimizes a given function is to differentiate the function with respect to that variable, equate the resulting expression to zero, and solve the equation to obtain the optimal variable value. However, it is very difficult, if not impossible, to use this approach to find the optimal level because Equation (15) for calculating $E[T]$ is too complex.

Next, we present a numerical method to find the optimal level. First, we define *normalized* $E[T]$ as the ratio of $E[T]$ to tag population size. Figure 3 shows the plots of normalized $E[T]$ vs. the number of tags for different γ values ranging from 1 to b (here we used $b = 10$ for illustration). From this figure, we observe that for any tag population size, there is a unique optimal value of γ. For example, for a population of 600 tags, $\gamma_{\text{op}} = 9$. Second, we define *crossover points* as follows: *for a given ID length b, the crossover points are the tag population sizes* $c_0 = 0, c_1, c_2, \cdots, c_{b+1} = 2^b$ *such that for any tag population size in* $[c_i, c_{i+1})$ $(0 \leq i \leq b)$, $\gamma_{op} = i$. These crossover points are essentially the x-coordinates of the intersection points of the normalized $E[T]$ curves of consecutive values of γ in Figure 3. Thus, the value of c_i can be obtained by putting $z = c_i$ and numerically solving $E[T, \gamma = i-1] = E[T, \gamma = i]$ for c_i using the bisection method. Once c_i is calculated for each $1 \leq i \leq b$, γ_{op} for a given z can be obtained by simply identifying the unique interval $[c_i, c_{i+1})$ in which z lies and then using $\gamma_{\text{op}} = i$. The solid line in Figure 3 is plotted using the values of γ_{op} obtained using the proposed strategy. As values of c_i only depend on b, it is a one time cost to calculate them. Table 2 tabulates the values of c_i obtained using this strategy for $b = 64$.

We next conduct an analytical comparison between the expected number of queries for TH and that for TW. Figure 4 shows the expected number of queries for TH, which is calculated using Equation (15) using $\gamma = \gamma_{\text{op}}$, and that for TW, which is calculated using Equation (15) using $\gamma = 1$, for 64 bit tag IDs. We observe that TH significantly outperforms TW for the expected number of queries. For example, for a population of 10K tags, the expected number of queries for TH is only 54% of that for TW. We will present detailed experimental comparison between TH and other identification protocols in Section 5.

Table 2: All crossover points for 64-bit tag IDs

c_0	0.00E+00	c_{22}	3.84E+06	c_{44}	1.61E+13
c_1	2.00E+00	c_{23}	7.68E+06	c_{45}	3.22E+13
c_2	4.00E+00	c_{24}	1.54E+07	c_{46}	6.45E+13
c_3	7.00E+00	c_{25}	3.07E+07	c_{47}	1.29E+14
c_4	1.50E+01	c_{26}	6.12E+07	c_{48}	2.58E+14
c_5	2.90E+01	c_{27}	1.22E+08	c_{49}	5.16E+14
c_6	5.90E+01	c_{28}	2.42E+08	c_{50}	1.03E+15
c_7	1.17E+02	c_{29}	4.76E+08	c_{51}	2.06E+15
c_8	2.35E+02	c_{30}	9.22E+08	c_{52}	4.13E+15
c_9	4.69E+02	c_{31}	1.73E+09	c_{53}	8.25E+15
c_{10}	9.38E+02	c_{32}	3.04E+09	c_{54}	1.65E+16
c_{11}	1.88E+03	c_{33}	8.59E+09	c_{55}	3.30E+16
c_{12}	3.75E+03	c_{34}	1.72E+10	c_{56}	6.59E+16
c_{13}	7.51E+03	c_{35}	3.01E+10	c_{57}	1.32E+17
c_{14}	1.50E+04	c_{36}	6.44E+10	c_{58}	2.63E+17
c_{15}	3.00E+04	c_{37}	1.25E+11	c_{59}	5.24E+17
c_{16}	6.00E+04	c_{38}	2.53E+11	c_{60}	1.04E+18
c_{17}	1.20E+05	c_{39}	5.03E+11	c_{61}	2.04E+18
c_{18}	2.40E+05	c_{40}	1.01E+12	c_{62}	3.96E+18
c_{19}	4.80E+05	c_{41}	2.01E+12	c_{63}	7.42E+18
c_{20}	9.61E+05	c_{42}	4.03E+12	c_{64}	1.30E+19
c_{21}	1.92E+06	c_{43}	8.06E+12	c_{65}	1.84E+19

3.3 Maximum Number of Queries

Although the primary goal of our TH protocol is to minimize the average number of queries, next, we analyze the maximum number of queries of TH and analytically show that it is still smaller than that of TW. The maximum number of queries that TH may need to identify z tags with b-bit IDs is shown in Theorem 2.

THEOREM 2. *Let V denote the number of queries that TH may need to identify a population of $z \geq 2$ tags with b-bit IDs using $\gamma = \gamma_{op}$. We have*

$$V \leq z(b - \gamma_{\text{op}} + 1) - 2^{\gamma_{\text{op}}} + 2\theta_0 - \theta_1(b - \gamma_{\text{op}} - 1) \quad (16)$$

where

$$\theta_0 = 2^{\gamma_{\text{op}}} - \left\lceil \frac{z}{2^{b-\gamma_{\text{op}}}} \right\rceil$$

$$\theta_1 = \left\lceil \frac{z}{2^{b-\gamma_{\text{op}}}} \right\rceil - \left\lceil \frac{z-1}{2^{b-\gamma_{\text{op}}}} \right\rceil \left\lceil 1 - \frac{\gamma_{\text{op}}}{b} \right\rceil$$

PROOF. Let V_{TW} denote the number of queries that TW may need to identify $z \geq 2$ tags with b-bit IDs. The upper bound of V_{TW} is given as follows (proven in [12]):

$$V_{TW} \leq z(b + 1 - \log \frac{z}{2}) - 1 \quad (17)$$

Because $z \geq 2$, we have $V_{TW} \leq z(b+1) - 1$.

When z tags are uniformly distributed in the ID space, TH essentially performs TW on all subtrees rooted at nodes on level γ_{op}. Let θ_0 and θ_1 denote the number of subtrees covering 0 and 1 tags, respectively. For these $\theta_0+\theta_1$ subtrees, TH only visits the roots, which are at level γ_{op}. Let α denote the number of remaining subtrees (*i.e.*, $\alpha = 2^{\gamma_{\text{op}}} - \theta_0 - \theta_1$) and T_i denote a subtree covering $z_i \geq 2$ tags. For each subtree T_i, the maximum number of nodes that TH visits is $z_i(b - \gamma_{\text{op}} + 1) - 1$. Summing all $2^{\gamma_{\text{op}}}$ subtrees, we have

$$\begin{aligned} V &\leq \sum_{i=0}^{\alpha-1} \Big(z_i(b - \gamma_{\text{op}} + 1) - 1 \Big) + \theta_0 + \theta_1 \\ &= z(b - \gamma_{\text{op}} + 1) - 2^{\gamma_{\text{op}}} + 2\theta_0 - \theta_1(b - \gamma_{\text{op}} - 1) \quad (18) \end{aligned}$$

The right hand side (RHS) of Equation (18) is maximized when θ_0 is maximized and θ_1 is minimized, which happens when all z tag IDs are contiguous and they start from the left most leaf of a subtree at level γ_{op}. In this case, the number of subtrees with tags are $\left\lceil \frac{z}{2^{b-\gamma_{op}}} \right\rceil$ and therefore $\theta_0 = 2^{\gamma_{\text{op}}} - \left\lceil \frac{z}{2^{b-\gamma_{op}}} \right\rceil$. Furthermore in this case, when $\gamma_{\text{op}} \leq b-1$, there is at most one subtree at level γ_{op} that has exactly one tag *i.e.*, $\theta_1 = \left\lceil \frac{z}{2^{b-\gamma_{\text{op}}}} \right\rceil - \left\lceil \frac{z-1}{2^{b-\gamma_{\text{op}}}} \right\rceil$; when $\gamma_{\text{op}} = b$, θ_1 equals z. Combining the two cases of $\gamma_{\text{op}} \leq b-1$ and $\gamma_{\text{op}} = b$, we have $\theta_1 = \left\lceil \frac{z}{2^{b-\gamma_{op}}} \right\rceil - \left\lceil \frac{z-1}{2^{b-\gamma_{op}}} \right\rceil \left\lceil 1 - \frac{\gamma_{op}}{b} \right\rceil$. □

The proof above gives us the insight that *TH requires fewer queries when the tag IDs are distributed more uniformly in the ID space.* Intuitively, this makes sense because the more the tag IDs are distributed uniformly, the fewer the number of collisions encountered by TH. Experimentally, our results shown in Figures 7(a) and 7(b) in Section 5 also confirm this insight: for the same number of tags, the number of queries needed by TH when tags are uniformly distributed is less than that when tags are non-uniformly distributed.

We now conduct an analytical comparison between the maximum number of queries for TH and that for TW. Figure 5 shows the maximum number of queries for TH, which is calculated using the RHS of Equation (16), and that for TW, which is calculated using the RHS of Equation (17), for 64 bit tag IDs. We observe that TH again outperforms TW for the maximum number of queries, although slightly. For example, for a population of 10K tags, the maximum number of queries for TH is 93% of that for TW.

Algorithms 1 through 4 show the pseudocode of TH.

4. DISCUSSION

4.1 Reliable Tag Identification

So far we have assumed that the communication channel between the reader and tags is reliable, which means that each tag can receive the query from the reader and the reader can receive either the response if only one tag responds or the collision if more than one tag respond. However, this assumption often does not hold in reality because wireless communication medium is inherently unreliable. There are two existing schemes for making tag identification reliable.

Algorithm 1: IdentifyRFIDTags(b)

```
Input: Tag ID size $b$ in bits
Output: IDs of all the tags $IDs$
$z_{\text{init}}$ := EstimateNumberofTags()
$i_{\text{ID}} := 0$
$s := 0$ /*counts number of leaf nodes covered*/
$x := 0$ /*counts number of tags identified*/
$z := z_{\text{init}}$
$l := 0$
$p := 0$
while $s \leq 2^b$ do
    $\gamma_{\text{op}}$ := Calculate $\gamma_{\text{op}}$ using the method of Section 3.2
    $(l, p)$ := FindHopDest($\gamma_{\text{op}}$, $b$, $(l, p)$)
    [$\tilde{x}$, $\tilde{\text{IDs}}$] := TreeWalking($(l, p)$)
    $\text{IDs}[i_{\text{ID}} : i_{\text{ID}} + |\tilde{\text{IDs}}| - 1] := \tilde{\text{IDs}}$
    $i_{\text{ID}} := i_{\text{ID}} + |\tilde{\text{IDs}}|$
    $x := x + \tilde{x}$
    $s := s + 2^{b-l}$
    $z := \frac{z_{\text{init}} - x}{2^b - s} \times 2^b$
return $IDs$
```

Algorithm 2: EstimateNumberofTags()

```
Input: None
Output: Initial estimate $z_{\text{init}}$ of population size
$f := 1$
$i := 1$
$p_i := 1$
repeat
    Provide the reader with $f/p_i$ and a random seed $R_i$.
    Run Aloha on reader and get the response.
    if slot is not empty then
        $p_{i+1} := p_i/2$
        $i := i + 1$
until slot is empty
$z_{\text{init}} := 1.2897 \times 2^{i-2}$
return $z_{\text{init}}$
```

Backes *et al.* proposed the scheme of letting each tag store the IDs of several other tags [1]. When the reader queries a tag, the tag transmits back its own ID as well as the IDs of other tags stored in it. When identification completes, the reader compares the set of IDs of tags that responded with

Algorithm 3: FindHopDest(γ_{op}, $b, (l, p)$)

```
Input: (1) Optimal level $\gamma_{\text{op}}$
       (2) Tag ID size $b$ in bits
       (3) Current node location $(l, p)$
Output: Hop destination node location $(\tilde{l}, \tilde{p})$
if $\gamma_{op} < l$ then
    if $p + 1 > \left( \left\lfloor \frac{p}{2^{l-\gamma_{op}}} \right\rfloor + 1 \right)(2^{l-\gamma_{op}}) - 1$ then
        $\tilde{l} = \gamma_{\text{op}}$
        $\tilde{p} = \left\lfloor \frac{p}{2^{l-\gamma_{\text{op}}}} \right\rfloor$
    else
        $\tilde{l} = l$
        $\tilde{p} = p$
else if $\gamma_{op} > l$ then
    $\tilde{l} = \gamma_{\text{op}}$
    $\tilde{p} = (p + 1)(2^{l-\gamma_{\text{op}}}) - 1$
else if $\gamma_{op} == l$ then
    $\tilde{l} = l$
    $\tilde{p} = p$
$\tilde{p} = \tilde{p} + 1$
return $(\tilde{l}, \tilde{p})$
```

Algorithm 4: TreeWalking((l,p))

```
Input: Current node location $(l, p)$
Output: (1) Number of tags $\tilde{x}$ identified in the subtree whose root node is $(l, p)$
        (2) Identifiers $\tilde{\text{IDs}}$ of the tags identified in the subtree whose root node is $(l, p)$
$\tilde{l} := l$
$\tilde{p} := p$
$i_{\text{ID}} := 0$
$\tilde{x} := 0$
$\tilde{\text{IDs}} := [\,]$
while $\tilde{p} < (p+1) \times 2^{\tilde{l}-l}$ do
    $[status, tagID]$ = type of response to query $(\tilde{l}, \tilde{p})$ /*a query $(\tilde{l}, \tilde{p})$ means $\tilde{l}$ bit representation of $\tilde{p}$*/
    if $status$ == collision then
        $\tilde{p} := 2\tilde{p}$
        $\tilde{l} := \tilde{l} + 1$
    else if $status$ == successful then
        $\tilde{x} := \tilde{x} + 1$
        $\tilde{\text{IDs}}[i_{\text{ID}}] := tagID$
        $i_{\text{ID}} := i_{\text{ID}} + 1$
        $q := \sum_{i=0}^{\log_2 \tilde{p}} \left( \lfloor \frac{\tilde{p}}{2^i} \rfloor - \lfloor \frac{\tilde{p}-1}{2^i} \rfloor \right)$
        $\tilde{l} := \tilde{l} - q + 1$
        $\tilde{p} := \frac{\tilde{p}}{2^{q-1}}$
return $[\tilde{x}, \tilde{\text{IDs}}]$
```

the union of sets of IDs of other tags reported by each responding tag. If the sets are not equal, the whole process is repeated again to ensure that the missed tags are identified. This scheme has two weaknesses. First, this scheme does not comply with the C1G2 standard. Second, it assumes that the tag population remains static for the lifetime of tags as each tag is hard coded with some other tags' IDs. The second scheme is to run an identification protocol on the same population several times until probability of missing a tag falls below a threshold [7,10]. They estimate the probability of missing a tag based upon the number of tags that were identified in some runs of the protocol but not in others.

While we can use the C1G2 compliant scheme proposed in [7,10] to make TH reliable, *i.e.*, repeatedly run TH until the required reliability is achieved. We observe that in this scheme, the leaf nodes in the binary tree are queried multiple times. This is wasteful of time for the nodes that the reader successfully reads. To eliminate such waste, we propose *to query each node multiple times, instead of querying the whole binary tree multiple times.* We define the *reliability of successfully reading a tag* to be *the probability that both the tag receives the query from the reader and the reader receives the response from the tag.* For this, we calculate the maximum number of times the reader should transmit a query, which is denoted by β. Let g and u be the *given* and *required* reliability of successfully reading a tag, respectively. Thus, the probability of successfully identifying a tag is $1-(1-g)^{\beta}$. Equating it to u, we get:

$$\beta = log_{(1-g)}(1-u) \tag{19}$$

Our scheme of reliable tag identification works as follows: for each non-terminal node in the binary tree that TH needs to visit, TH transmits a query corresponding to that node β times; for each terminal node, TH keeps transmitting the query corresponding to that node until either that query has been transmitted β times or the reader successfully receives the tag ID.

The optimization technique of stop transmitting the query corresponding to a terminal on a successful read significantly reduces the total number of queries. Figure 6 plots the expected number of queries per tag for the reliable TH protocol with and without this optimization. For example, for a population of 50000 tags, the number of queries per tag are reduced by 24%.

4.2 Continuous Scanning

In some applications, the tag population may change over time (*i.e.*, tags leave and join the population dynamically). We adapt the continuous scanning strategy proposed by Myung *et al.* in [14]. In the first scanning of the whole tag population, TH records the queries that resulted in successful or empty reads. If the tag population does not change, by perfoming DFTs on the subtrees rooted at successful and empty read nodes of the previous scan, TH experiences no collision. If some new tags join the population, some of the successful read nodes of the previous scan can now turn into collision nodes and some empty read nodes can turn into successful or collision nodes. If some old tags leave the population, some successful read nodes will become empty read nodes. If any of the new empty read nodes happens to be a sibling of another empty read node, then TH discards these two nodes from the record and stores the location of their parent because the parent is also an empty read node. This strategy works well when the tag population size either remains static or increases. However, when the tag population decreases, the best choice is to re-execute TH for the subsequent scan.

4.3 Multiple Readers

An application with a large number of RFID tags requires multiple readers with overlapping regions because a single reader can not cover all tags due to the short communication range of tags (usually less than 20 feet). The use of multiple readers introduces several new types of collisions such as reader-reader collisions and reader-tag collisions. Such collisions can be handled by reader scheduling protocols such as those proposed in [3,25,26,29]. TH is compatible with all of these reader scheduling protocols.

5. PERFORMANCE COMPARISON

We implemented TH and all the 8 prior tag identification protocols in Matlab, namely the 3 nondeterministic protocols (Aloha [28], BS [2], and ABS [14]), the 3 deterministic protocols (TW [12], ATW [24], and STT [19]), and the 2 hybrid protocols (MAS [15] and ASAP [20]). As ATW starts DFTs from the level of log z which may not be a whole number, we present results for ATW by both ceiling and flooring the values of log z and representing them with ATW-c and ATW-f respectively. In terms of implementation complexity, TH and all the 8 prior protocols are implemented in the similar number of lines of code. We performed extensive testing, both manually and automatically, to ensure the correctness of each protocol implementation.

We performed the side-by-side comparison with TH, although this comparison is not completely fair for TH for two reasons. First, 3 of these 8 protocols (*i.e.*, BS, ABS, and ASAP) require modification to tags and thus do not work with standard C1G2 tags, whereas TH is fully compliant with C1G2. Second, for the framed slotted Aloha, to its best advantage, we choose the frame size to be the ideal

Table 3: Comparison with Prior C1G2 Compliant Protocols (TH/Prior Art)

		Prior Nondeterministic Protocol (=Aloha)			Prior Deterministic Protocols				Prior Hybrid Protocol (=MAS)		
		Max	Min	Mean	Best prior	Max	Min	Mean	Max	Min	Mean
Uniform	#queries/tag	0.24	0.10	0.18	ATW-f	0.51	0.50	0.50	0.39	0.38	0.39
	query time/tag	0.84	0.71	0.76	ATW-c	0.92	0.89	0.90	0.81	0.78	0.79
	#responses/tag	0.85	0.59	0.69	ATW-c	0.85	0.67	0.70	0.64	0.24	0.38
	response fairness	1.15	1.10	1.13	TW	1.12	1.07	1.11	1.12	1.07	1.10
Non-Uni	#queries/tag	0.27	0.20	0.22	ATW-f	0.75	0.68	0.74	0.47	0.18	0.33
	query time/tag	0.44	0.34	0.37	ATW-f	0.69	0.61	0.63	0.31	0.16	0.22
	#responses/tag	0.85	0.42	0.68	ATW-c	0.86	0.58	0.74	0.59	0.30	0.48
	response fairness	1.38	1.25	1.35	ATW-c	1.03	1.00	1.02	1.05	0.95	1.02

size, which is equal to the tag population size, disregarding the practical limitations on the frame sizes. We choose tag ID length to be the C1G2 standard 64 bits. We performed the comparison for both the uniform case (where the tag population is uniformly distributed in the ID space) and the non-uniform case (where the tag population is not uniformly distributed in the ID space). For the uniform case, we range tag population sizes from 100 to $100,000$ to evaluate the scalability of these protocols. For the non-uniform case, we distribute tag populations in blocks where each block is a continuous sequence of tag IDs. We range block sizes from 5 to 1000. Our motivation for simulating non-uniform distribution in blocks is that in some applications, such as supply chains, tag IDs often come in such blocks when they are manufactured. For each tag population size, we run each protocol 100 times and report the mean. We compare TH with prior protocols from both reader and tag perspectives.

5.1 Reader Side Comparison

For the reader side, we compared TH with the 8 prior protocols based on the following two metrics: (1) normalized reader queries and (2) identification speed. Normalized reader queries is the ratio of the number of queries that the reader transmits to identify a tag population divided by the number of tags in the population. Similarly, identification speed is the total time that the reader takes to identify a tag population divided by the number of tags in that population.

In general, more queries implies more identification time. However, identification time is not strictly in proportion to the number of queries because different queries may take different amounts of time. According to [8] and [9], the time for a successful read t_s, an empty read t_e, and a collision t_c are $3ms$, $0.3ms$, $1.5ms$, respectively.

For each metric, in Table 3, we show the value of TH divided by that for the best prior C1G2 compliant protocol for this metric in the corresponding category of nondeterministic, deterministic, or hybrid. Note that the only prior C1G2 compliant nondeterministic tag identification protocol is the framed slotted Aloha and the only prior C1G2 compliant hybrid tag identification protocol is MAS. There are 3 prior C1G2 compliant deterministic tag identification protocols: TW, ATW, and STT. We report min, max, and mean for these ratios for tag populations ranging from 100 to $100,000$.

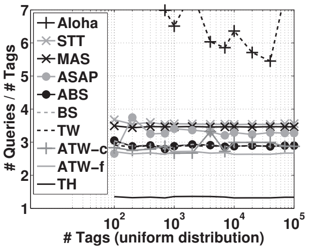
For the two metrics defined above, the absolute performance of TH and all prior 8 tag identification protocols is shown in Figures 7(a) to 8(b), for both uniform and non-uniform distributions. Note that for non-uniform distributions, we fix the tag population size to be 5000 and range the block size from 5 to 1000.

5.1.1 Normalized Reader Queries

TH reduces the normalized reader queries of the best prior C1G2 compliant nondeterministic, deterministic, and hybrid tag identification protocols by an average of 82%, 50%, *and* 61%, *respectively, for uniformly distributed tag populations, and by an average of* 78%, 26%, *and* 67%, *respectively, for non-uniformly distributed tag populations.* Figures 7(a) and 7(b) show the normalized reader queries of all protocols for uniformly and non-uniformly distributed populations, respectively. Based on these two figures, we make the following two observations from the perspective of normalized reader queries for both uniform and non-uniform distributions. First, among all the 8 prior protocols, the traditional ATW protocol turns out to be the best. Second, the framed slotted Aloha in the C1G2 standard performs the worst even when we disregard the practical limitations on the frame sizes. Although BS is the best among the 3 prior nondeterministic tag identification protocols, it is not compliant with C1G2. Similarly, although ASAP is the best among the 2 prior hybrid tag identification protocols, it is not compliant with C1G2.

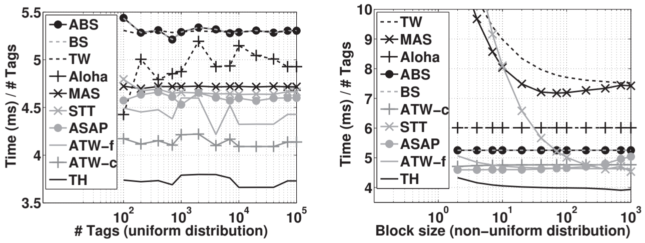
5.1.2 Identification Speed

TH improves the identification speed of the best prior C1G2 compliant nondeterministic, deterministic, and hybrid tag identification protocols by an average of 24%, 10%, *and* 21%, *respectively, for uniformly distributed tag populations, and by an average of* 63%, 37%, *and* 78%, *respectively, for non-uniformly distributed tag populations.* Figures 8(a) and 8(b) show the identification speed of all protocols for uniformly and non-uniformly distributed tag populations, respectively. Based on these two figures, we make the following two observations from the perspective of identification speed. First, among all 8 prior protocols, the traditional ATW protocol turns out to be the best for both uniform and non-uniform distributions. Second, although framed slotted Aloha is the worst in terms of normalized reader queries, its identification speed is not the worst. This is because in our experiments we allow it to use unrealistically large frame sizes, which leads to many empty slots and empty read is much faster than successful read and collision.



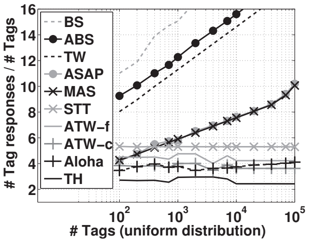
(a) Uniform distribution (b) Non-unif. distribution

Figure 7: Normalized Queries



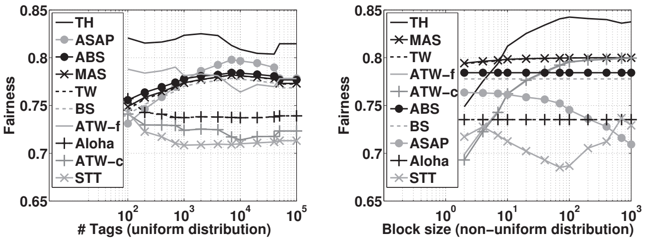
(a) Uniform distribution (b) Non-unif. distribution

Figure 8: Identification Speed



(a) Uniform distribution (b) Non-unif. distribution

Figure 9: Normalized Responses



(a) Uniform distribution (b) Non-unif. distribution

Figure 10: Response Fairness

5.2 Tag Side Comparison

On the tag side, we compare TH with the 8 prior protocols based on the following four metrics: (1) normalized tag responses, (2) response fairness, (3) normalized collisions, and (4) normalized empty reads. Normalized tag responses is the ratio of sum of responses of all tags during the identification process to the number of tags in the population. Response fairness is the Jain's fairness index given by $\frac{(\sum_{i=1}^{z} x_i)^2}{z \cdot \sum_{i=1}^{z} x_i^2}$ where x_i is the total number of responses by tag i [11]. Normalized collisions is the ratio of total number of collisions during the identification process to the number of tags in the population. Normalized empty reads is the ratio of total number of empty reads during the identification process to the number of tags in the population.

The first two metrics are important for active tags because active tags are powered by batteries. Lesser number of normalized tag responses mean lesser power consumption for active tags. Response fairness measures the variance in the number of responses per tag. Less fairness results in the depletion of the batteries of some tags more quickly compared to others. In large scale tag deployments, it is often nontrivial to identify tags with depleted batteries and replace them. Using an absolutely fair tag identification protocol, the batteries of all tags deplete at the same time and therefore all can be replaced at the same time. We use the Jain's fairness metric defined in [11]. For z tags, the fairness value is in the range $[\frac{1}{z}, 1]$. The higher this fairness value is, the more fair the protocol is. The second two metrics are important for understanding these identification protocols.

For normalized tag responses and response fairness, in Table 3, we show the value of TH divided by that for the best prior C1G2 compliant protocol in the corresponding category of nondeterministic, deterministic, or hybrid. The absolute performance of TH and all prior 8 tag identification protocols is shown in Figures 9(a) to 11(b), for both uniform and non-uniform distributions.

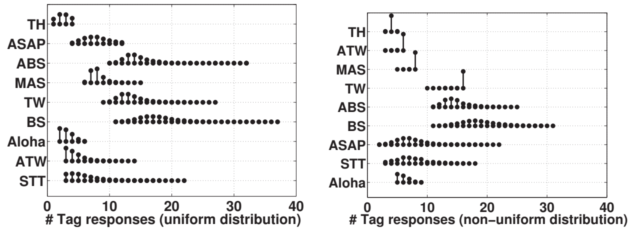
5.2.1 Normalized Tag Responses

TH reduces the normalized tag responses of the best prior C1G2 compliant nondeterministic, deterministic, and hybrid tag identification protocols by an average of 31%, 30%, *and* 62%, *respectively, for uniformly distributed tag populations, and by an average of* 32%, 26%, *and* 52%, *respectively, for non-uniformly distributed tag populations.* Figures 9(a) and 9(b) show the normalized tag responses of all protocols for uniformly and non-uniformly distributed tag populations, respectively. We make following 3 observations from these two figures. First, the normalized tag responses of BS, ABS, TW, MAS, and ASAP increase with increasing tag population size. Second, for non-uniformly distributed tag populations, the normalized tag responses of nondeterministic protocols is not affected by the block size because their performance is independent of tag ID distribution. In contrast, the normalized tag responses of deterministic protocols slightly increase with increasing block size. Third, among all 8 prior protocols, Aloha has the smallest number of normalized tag responses. This is because of the the unlimitedly large frame sizes that we used for Aloha. With large frame sizes, tags experience lesser collisions and thus reply fewer times.

5.2.2 Tag Response Fairness

TH improves the tag response fairness of the best prior C1G2 compliant nondeterministic, deterministic, and hybrid tag identification protocols by an average of 13%, 11%, *and* 10%, *respectively, for uniformly distributed tag populations, and by an average of* 35%, 2%, *and* 2%, *respectively, for non-uniformly distributed tag populations.* Figures 10(a) and 10(b) show the tag response fairness of all protocols for uniformly and non-uniformly distributed tag populations, respectively. We observe that among all 8 prior protocols, ASAP and ATW are the best for uniformly and non-uniformly distributed populations, respectively.

Figures 11(a) and 11(b) show the distribution of the number of tag responses for each protocol for uniformly and non-uniformly distributed tag populations, respectively. For any protocol, the wider the horizontal span of its distribution is, the larger the range of the number of responses per tag it has. We observe that TH has the smallest range among all protocols for the number of responses per tag.



(a) Uniform distribution (b)Non-unif. distribution

Figure 11: Distribution of tag responses

5.2.3 Normalized Collisions

TH incurs smaller number of collisions than all 8 prior protocols for uniformly and non-uniformly distributed tag populations. Figures 12(a) and 12(b) show the normalized collisions for all protocols for uniformly and non-uniformly distributed tag populations, respectively. From these figures we make following 2 observations. First, Aloha incurs the smallest number of normalized collisions among all 8 prior protocols because of the unlimitedly large frame sizes that we used for it. Second, TW mostly incurs the largest number of normalized collisions for both types of populations.

5.2.4 Normalized Empty Reads

For uniformly distributed tag populations, TH incurs a smaller number of empty reads than all 8 prior protocols. For non-uniformly distributed tag populations, for small block sizes, TH incurs smaller number of empty reads than all prior C1G2 compliant protocols; for larger block sizes, TW and MAS incur slightly lesser empty reads compared to TH. Figure 13(a) and 13(b) show the normalized empty reads of all protocols for uniformly and non-uniformly distributed tag populations, respectively. From these figures, we observe that although the two prior C1G2 compliant protocols, TW and MAS, have fewer empty reads compared to TH for large block sizes, they have much larger number of collisions compared to TH, which makes their overall identification time much larger than TH. Note that the slightly larger number of empty reads for TH for large block sizes is immaterial because the time for an empty read is 5 times lesser than that for a collision and 10 times lesser than that for a successful read. Therefore, reducing collisions is more important than reducing empty reads.

Note that the collisions and empty reads shown in Figures 12(a) and 13(a), respectively, are consistent with the reader queries shown in Figure 7(a) as well as the identification speed shown in Figure 8(a). Similarly, the collisions and empty reads shown in Figures 12(b) and 13(b), respectively, are consistent with the reader queries shown in Figure 7(b) as well as the identification speed shown in Figure 8(b). For example, Figure 12(a) shows that TW has more collisions than Aloha, but 7(a) shows that Aloha has more queries than TW. This is because Aloha has much more empty reads than TW as shown in Figure 13(a). Although Aloha has more queries than TW, Figure 8(a) also shows that Aloha requires less identification time than TW. This is because an empty read is 5 times faster than a collision for a reader.

6. CONCLUSION

The technical novelty of this paper lies in that it represents the first effort to formulate the Tree Walking process mathematically and propose a method to minimize the expected number of queries. The significance of this paper in terms of impact lies in that the Tree Walking protocol is a fundamental multiple access protocol and has been standardized as an RFID tag identification protocol. Besides static optimality, our Tree Hopping protocol dynamically chooses a new optimal level after each subtree is traversed. We also presented methods to make our protocol reliable, to continuously scan tag populations that are dynamically changing, and to work with multiple readers with overlapping regions. Another key contribution of this paper is that we conducted a comprehensive side-by-side comparison of our protocol with eight major prior tag identification protocols that we implemented. Our experimental results show that our protocol significantly outperforms all prior tag identification protocols, even those that are not C1G2 compliant, for metrics such as the number of reader queries per tag, the identification speed, and the number of responses per tag.

7. REFERENCES

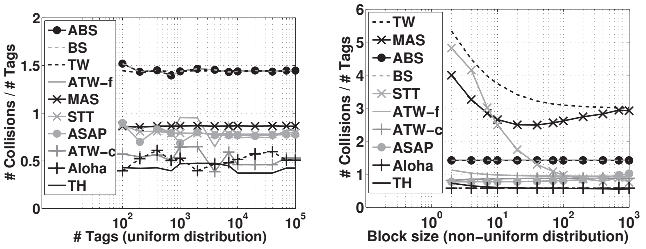
[1] M. Backes, T. R. Gross, and G. Karjoth. Tag identification system, 2008.

[2] J. I. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Transactions on Information Theory*, 25:505–515, 1979.

[3] B. Carbunar, M. K. Ramanathan, M. Koyuturk, C. Hoffmann, and A. Grama. Redundant reader elimination in RFID systems. In *Proc. IEEE Communications Society Conf. on SECON*, pages 576–580, 2005.

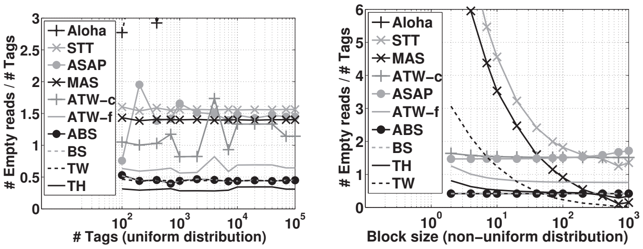
[4] R. Dorfman. The detection of defective members of large populations. *Annals of Mathematical Statistics*, 14:436–440, 1943.

[5] K. Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication.* Wiley, 2010.



(a) Uniform distribution (b) Non-unif. distribution

Figure 12: Normalized Collisions



(a) Uniform distribution (b) Non-unif. distribution

Figure 13: Normalized Empty Reads

[6] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

[7] K. Fyhn, , R. M. Jacobsen, P. Popovski, and T. Larsen. Fast capture – recapture approach for mitigating the problem of missing rfid tags. *IEEE Transactions on Mobile Computing*, 11(3):518–528, 2012.

[8] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu. Counting RFID tags efficiently and anonymously. In *Proc. IEEE INFOCOM*, 2010.

[9] E. Inc. *Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz*. EPCGlobal Inc, 1.2.0 edition, 2008.

[10] R. Jacobsen, K. F. Nielsen, P. Popovski, and T. Larsen. Reliable identification of rfid tags using multiple independent reader sessions. In *Proc. IEEE Int. Conf. on RFID*, pages 64–71, 2009.

[11] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical report, Digital Equipment Corporation, 1984.

[12] C. Law, K. Lee, and K.-Y. Siu. Efficient memoryless protocol for tag identification. In *Proc. 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2000.

[13] C. H. Lee and C.-W. Chung. Efficient storage scheme and query processing for supply chain management using RFID. In *Proc. ACM Conf. on Management of data*, pages 291–302, 2008.

[14] J. Myung and W. Lee. Adaptive splitting protocols for RFID tag collision arbitration. In *Proc. 7th ACM Int. Symposium on Mobile Ad Hoc Networking and Computing*, pages 202–213, 2006.

[15] V. Namboodiri and L. Gao. Energy-aware tag anticollision protocols for RFID systems. In *Proc. 5th IEEE Int. Conf. on Pervasive Computing and Communications*, pages 23–36, 2007.

[16] B. Nath, F. Reynolds, and R. Want. RFID technology and applications. *Proc. IEEE Pervasive Computing*, 5:22–24, 2006.

[17] A. Nemmaluri, M. D. Corner, and P. Shenoy. Sherlock: Automatically locating objects for humans. In *Proc. Int. Conf. on Mobile Systems, Applications, and Services*, pages 187–198, 2008.

[18] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. Landmarc: Indoor location sensing using active RFID. *Wireless networks*, 10:701–710, 2004.

[19] L. Pan and H. Wu. Smart trend-traversal: A low delay and energy tag arbitration protocol for large RFID systems. In *Proc. IEEE INFOCOM*, 2009.

[20] C. Qian, Y. Liu, H. Ngan, and L. M. Ni. ASAP: Scalable identification and counting for contactless RFID systems. In *Proc. 30th IEEE Int. Conf. on Distributed Computing Systems*, pages 52–61, 2010.

[21] M. Roberti. A 5-cent breakthrough. *RFID Journal*, 5(6), 2006.

[22] W. A. Rosenkrantz and D. Towsley. On the instability of slotted aloha multiaccess algorithm. *IEEE Transactions on Automatic Control*, 28(10):994–996, 1983.

[23] P. Semiconductors. *SL2 ICS11 I.Code UID Smart Label IC Functional Specification Datasheet http://www.advanide.com/datasheets/sl2ics11.pdf*, 2004.

[24] A. S. Tanenbaum. *Computer Networks*. Prentice-Hall, 2002.

[25] S. Tang, J. Yuan, X.-Y. Li, G. Chen, Y. Liu, and J. Zhao. Raspberry: A stable reader activation scheduling protocol in multi-reader RFID systems. In *Proc. IEEE Int. Conf. on Network Protocols*, pages 304–313, 2009.

[26] J. Waldrop, D. W. Engels, and S. E. Sarma. Colorwave: A MAC for RFID reader networks. In *Proc. IEEE Wireless Communications and Networking*, pages 1701–1704, 2003.

[27] C. Wang, H. Wu, and N.-F. Tzeng. RFID-based 3-D positioning schemes. In *Proc. IEEE INFOCOM*, pages 1235–1243, 2007.

[28] B. Zhen, M. Kobayashi, and M. Shimizu. Framed ALOHA for multiple RFID objects identification. *IEICE Transactions on Communications*, 88:991–999, 2005.

[29] Z. Zhou, H. Gupta, S. R. Das, and X. Zhu. Slotted scheduled tag access in multi-reader RFID systems. In *Proc. IEEE Int. Conf. on Network Protocols*, pages 61–70, 2007.