Fast and Reliable Estimation Schemes in RFID Systems

Murali Kodialam
Bell Labs, Lucent Technologies
Holmdel, NJ 07733. USA
muralik@lucent.com

Thyaga Nandagopal
Bell Labs, Lucent Technologies
Holmdel, NJ 07733. USA
thyaga@lucent.com

ABSTRACT

RFID tags are being used in many diverse applications in increasingly large numbers. These capabilities of these tags span from very dumb passive tags to smart active tags, with the cost of these tags correspondingly ranging from a few pennies to many dollars. One of the common problems that arise in any RFID deployment is the problem of quick estimation of the number of tags in the field up to a desired level of accuracy. Prior work in this area has focused on the identification of tags, which needs more time, and is unsuitable for many situations, especially where the tag set is dense. We take a different, more practical approach, and provide very fast and reliable estimation mechanisms. In particular, we analyze our estimation schemes and show that the time needed to estimate the number of tags in the system for a given accuracy is much better than schemes presented in related work. We show that one can estimate the cardinality of tag-sets of any size in near-constant time, for a given accuracy of estimation.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication; C.2.8 [**Mobile Computing**]: Algorithm Design and Analysis

General Terms

Algorithms, Design, Theory, Measurement, Performance

Keywords

Algorithms, RFID, ALOHA, Estimation, Tags

1. INTRODUCTION

Radio-frequency identification (RFID) tags are increasingly being used in everyday scenarios, ranging from inventory control and tracking, to medical-patient management. The key driver behind this widespread adoption is the simplicity of the tags, which enables very low (nearly zero) cost at high volumes. The tags themselves vary significantly in their capabilities, from dumb tags which merely transmit a particular bit-string when probed by a reader, to smart tags which have their own CPU, memory and power supply. Most of these tags are designed to have a very long life, and hence do not use any existing energy sources for transmitting data. Rather, they derive the energy needed for transmission from a probe signal sent by a reader node. This probe can performed via magnetic coupling (called near-field), or electro-magnetic coupling (called far-field). The latter has a much larger range, and is designed to read hundreds of tags at a time, while the former has a range less than 1 meter and hence is used to read less than 1 – 5 tags at a time.

RFID tags can be generally classified into *passive tags*, *semi-passive tags*, and *active tags*. Active and semi-passive tags have their own power source, typically in the form of a battery. However, semi-passive tags do not use their power source for transmission, but use it primarily to drive other on-board circuitry. Nearly all current RFID deployments around the world involve passive and semi-active tags. A sensor mote can be classified as an active tag. More information about RFID technology and its taxonomy can be found in [1, 2, 5, 6, 7]. A collection of white papers describing RFID technology, latest news and vendors can also be found in [8].

RFID tags are used to label items, and hence, identifying this information is the main goal of any RFID system. The general idea is as follows: the reader probes a set of tags, and the tags reply back. There are many algorithms that enable identification, and these can be classified into two categories: (a) *probabilistic*, and (b) *deterministic*. Since RFID devices are very simple, and operate in the wireless medium, collisions will result whenever a reader probes a set of tags. The identification algorithms use *anti-collision* schemes to resolve collisions.

In *probabilistic identification* algorithms [25, 3, 18, 19, 20, 22, 23, 24, 28, 34], a framed ALOHA scheme [25] is used where the reader communicates the frame length, and the tags pick a particular slot in the frame to transmit. The reader repeats this process until all tags have transmitted at least once successfully in a slot without collisions. In semi-active and active tag systems, the reader can acknowledge tags that have succeeded at the end of each frame, and hence those tags can stay silent in subsequent frames, reducing the probability of collisions thereby shortening the overall identification time. In passive tags, all tags will continue to transmit in every frame, which lengthens the total time needed to identify all tags.

Deterministic identification algorithms typically use a slot-

MobiCom'06, September 23–26, 2006, Los Angeles, California, USA.

ted ALOHA model, where the reader identifies the set of tags that need to transmit in a given slot, and tries to reduce the contending tag set in the next slot based on the result in the previous slot. These algorithms fall into the class of *tree-based identification* algorithms [21, 26, 27, 37, 38, 39, 40, 43, 44, 29, 30] with the tags classified on a binary tree based on their id, and the reader moving down the tree at each step to identify all nodes. Deterministic algorithms are typically faster than probabilistic schemes in terms of actual tag response slots used, however, they suffer from large reader overhead since the reader has to specify address ranges to isolate contending tag subsets using a probe at the beginning of each slot.

The common requirement for both classes of identification algorithms is an estimate of the actual number of tags t in the system. This estimate is used to set the optimal frame size in framed ALOHA and to guide the tree-based identification process for computing the expected number of slots needed for identification. Hence, it is important to have a quick estimate that is as accurate as possible. One could potentially combine estimation and identification together to save time, however, the drawback is that the initial steps rely on inaccurate estimates of the number of tags. Hence, the estimation process should be able to use non-identifiable information, such as a string of bits used by all tags, to compute the size of the tag set t.

Estimation of the cardinality of the tag set is also important in other problems pertaining to RFID tags. Due to privacy constraints, it may not be acceptable for readers to query the tags for their identification in certain instances. In such instances, tags could send out non-identifiable information, which could still be used to compute estimates of the cardinality. Another set of problems arise when the tag set is changing so fast to make identification of all tags impossible (e.g., an airplane flying over a field of sensors, trying to get an estimate of the number of active sensors left in the field). An efficient cardinality estimation scheme should be able to work in such environments as well. We would like to point out that in these instances, having an active tag does not confer any special advantages to the estimation problem over a passive tag from an energy management perspective.

Our goal is to develop an efficient and fast estimation scheme that work extremely well in a wide variety of circumstances. In this paper, we describe a method that will enable us to compute the cardinality of a tag set in a very small amount of time when compared to the time needed for identification. The proposed algorithms require very simple modifications to current RFID tags and are easily implementable using available technology with very little incremental cost. Our key contributions are:

- We propose two estimation algorithms for a static tag set, and demonstrate their properties through analysis and simulations. We show that the two algorithms are complementary to each other.

- We describe a single unified estimation algorithm that allows us to estimate the cardinality of a static tag set with a desired level of accuracy, and show, via analysis and simulations, the performance of the unified algorithm.

- Using a *probabilistic framed-ALOHA* model, we provide even better estimation algorithms that can achieve the desired performance in substantially lesser time than any known algorithm. We also show that the estimation range of this algorithm spans many orders of magnitude (i.e., from tens of tags to tens of thousands of tags).

The rest of this paper is organized as follows. In Section 2, we present the RFID system model. Section 3 describes estimation algorithms based on framed-ALOHA, and Section 4 describes the estimation algorithm based on probabilistic-framed ALOHA. We compare our work with related work in this area in Section 5. We summarize our results and discuss future work in Section 6.

2. SYSTEM MODEL

The RFID system considered in this paper consists of a set of readers and many tags. We adopt a *Listen-before-Talk* model for the RFID tags [2], where the tags listen to the reader's request before they talk back. We assume that there exists a separate estimation phase for computing the cardinality of the tag set that precedes any identification process, and that this phase uses the framed-slotted ALOHA model for tags to transmit back to the reader.[1] Given a frame of size f slots, tags randomly pick a slot based on a uniform probability distribution, and transmit in that slot. Tags cannot sense the channel, and hence, they merely transmit in the chosen slot. Slot synchronization is provided by the reader's energizing probe/request.

When probed by the reader in the estimation phase, we assume that tags respond with a bit-string that contains some error-detection (such as CRC) embedded in the string. The length of this common bit-string is defined as the minimum length string such that the reader can detect collisions when multiple tags transmit the same string in a given slot. This string need not be unique across tags, and therefore is much smaller than the length of the unique tag identifier. The reader can thus detect collisions in the estimation phase, and identify a successful transmission in any slot by only one tag. If none of the tags choose a time slot, then the reader will recognize that this time slot is idle. The entire system uses a single wireless channel/band for operation. The *load factor* of the system is defined as the ratio of the number of tags to the number of time slots in a frame. We denote the load factor by $\rho = t/f$.

When tags choose to transmit, have two degrees of freedom: (a) choosing a slot in a frame of size f, and (b) the probability of transmission p in any given frame. Current tag systems already allow variable frame sizes, albeit from a limited set of choices, for both passive and active tags[9]. Accordingly, the reader's transmission request can contain one or both options: (a) a desired frame size to be used by all tags, and (b) probability of transmission to be used by tags for transmitting in a given frame. Given both parameters, the tag first decides whether to participate in the frame with probability p, and then picks a slot at random in a frame of size f. Each of these parameters can be varied across frames, resulting in four possible combinations (fixed/variable f, fixed/variable p), which will be analyzed in this paper.

In this paper, we do not consider the identification problem, also referred to as collision resolution or conflict resolu-

[1]This phase can potentially be combined with the identification phase for probabilistic identification algorithms.

tion in related work. Our goal is simply to provide a reliable estimate of the cardinality of the tag set in as little time as possible.

Since the estimation scheme is probabilistic in nature, we specify the accuracy requirement for the estimation process, specified using two parameters, the error bound $\beta > 0$ and failure probability $0 < \alpha < 1$.

The problem we solve is the following: Given a set of t tags in the system, the reader has to *estimate the number of tags* in the system with an confidence interval of width β, i.e., we want to obtain an estimate $\hat{t}$ such that $\frac{\hat{t}}{t} \in \left(1 - \frac{\beta}{2}, 1 + \frac{\beta}{2}\right)$ with probability greater than α. In other words, we need maximum error to be at most $\pm\frac{\beta t}{2}$ with probability greater than α. A sample problem would be to estimate the number of tags within $\pm 1\%$ of the actual number of tags with probability greater than 99.99 %.

Measuring Performance: We measure the performance of the estimator in terms of the number of slots needed to perform the estimation to the desired accuracy level. Typically, in order to achieve the specified accuracy level, multiple measurements have to be made. The performance is measured in terms of the total number of slots, summed over all the measurements. The goal is to achieve the desired performance in as little time as possible. In other words, if it takes l_e slots to compute $\hat{t}$, the estimate of t tags with a certain accuracy, and l_i slots to uniquely identify $\hat{t}$ tags, then, we need $s_e l_e \ll l_i s_i$, where s_e and s_i are the sizes of the bit strings transmitted during the estimation and identification phases respectively.

Throughout this paper, we use Z_α to denote the α percentile for the unit normal distribution. If $\alpha = 99.9\%$ then $Z_\alpha = 3.2$.

Before we proceed further, we briefly describe a current implementation of random slot selection in a frame that can be extended easily to accommodate the probability of transmission in a given frame. In the Phillips I-Code system[9], a frame size f (typically a power of 2) is sent by the readers along with a seed value which is a 16-bit number. Each tag uses this seed information along with its identifier to hash into an integer in the range $[1, f]$, which specifies the slot in which the frame will contend. The reader sends a different seed in each frame to ensure tags do not necessarily select the same slot in each frame. Note that tag/reader implementations by other RFID vendors follow similar principles for slot selection. This scheme can be extended to support variable contention probability p. The reader now sends three parameters in each probe: (a) the seed, (b) the frame size f, and (c) the integer $\lceil \frac{f}{p} \rceil$. The tag hashes the combined seed/identifier value into the range $[1, \lceil \frac{f}{p} \rceil]$. If the hashed value is greater than f, then the tag does not transmit in this frame, else, it transmits in the computed slot, thereby resulting in a frame transmission probability of $\frac{f}{f/p} = p$. We implement this model in our simulations, except that we use the *drand()* function for the hashing scheme.

Based on the I-Code system, we set the estimator slot to be 10 bits long[2], to achieve a rate of 4000 estimation slots per second. The unique tag id field is 56 bits long, including the CRC, and the maximum tag identification rate in I-Code is 200 tags per second, using a 56 kbps bit-rate. Thus the estimation slot is much smaller than the identification slot.

[2]This is an over-estimate, since it is possible to detect collisions using even smaller number of bits.

3. BASIC ESTIMATION ALGORITHMS

In this section, we develop two different estimators for t, the cardinality of the tag set, assuming that all tags transmit in all frames during the estimation process. The two estimators complement each other well and combining them gives an estimation algorithm that performs well for a wide range of tag set cardinalities.

3.1 System Description

The reader probes the tags with the frame size f and the tags pick a slot j in the frame uniformly at random and transmit in that slot. We use the indicator random variable X_j for the event that there is no transmission in slot j. In other words, $X_j = 1$ if no tag transmits in slot j and $X_j = 0$ otherwise. Similarly, we set $Y_j = 1$ if and only if there is exactly one tag that transmits in slot j and $V_j = 1$ if and only if there are multiple tags that transmit in slot j. Note that $X_j + Y_j + V_j = 1$ for all slots j. If slot j has no transmissions in it, i.e, $X_j = 1$ then we refer to this slot j as an *empty slot* or a *zero slot*. If exactly one tag transmits in slot j, i.e, $Y_j = 1$, then we refer to slot j as a *singleton slot*. If multiple tags transmit in slot j creating a collision, i.e., $V_j = 1$, then we refer to slot j as a *collision slot*. Let $N_0 = \sum_{j=1}^{f} X_j$ denote the total number of empty slots, $N_1 = \sum_{j=1}^{f} Y_j$ denote the total number of singleton slots and $N_c = f - N_0 - N_1$ denote the number of collision slots. Note that $N_0, N_1,$ and N_c are random variables. Let $n_0, n_1,$ and n_c represent the values that are observed by the reader in a particular instance. The reader has to estimate t based on the (n_0, n_1, n_c). Toward this end, we first give the following result.

Lemma 1. *Let (N_0, N_1, N_c) represent the number of time slots with no transmissions, one transmission and collision respectively in a system with t tags and frame size f. Let $\rho = t/f$. Then*

$$\begin{aligned} E[N_0] &\approx f\, e^{-\rho} \\ E[N_1] &\approx f\, \rho\, e^{-\rho} \\ E[N_c] &\approx f\left(1 - (1+\rho)e^{-\rho}\right) \end{aligned}$$

Proof. Slot j will be empty if none of the tags transmit in that slot. Therefore,

$$Pr[X_j = 1] = \left(1 - \frac{1}{f}\right)^t \approx e^{-\rho}.$$

This implies that $E[N_0] = \sum_{j=1}^{f} Pr[X_j = 1] \approx f\, e^{-\rho}$. Similarly,

$$Pr[Y_j = 1] = t\frac{1}{f}\left(1 - \frac{1}{f}\right)^{t-1} \approx \rho\, e^{-\rho}$$

and $E[N_1] = \sum_{j=1}^{f} Pr[Y_j = 1] \approx f\ \rho\, e^{-\rho}$. Since $X_j + Y_j + V_j = 1$ for all j,

$$E[N_c] = \sum_{j=1}^{f} Pr[V_j = 1] \approx f - f\,\rho\, e^{-\rho} - f\, e^{-\rho}.$$

This completes the proof. □

3.2 Obtaining the Estimators

The reader measures (n_0, n_1, n_c). From Lemma 1, we know that the expected number of empty slots is $f\ e^{-\rho}$, or

Estimator	Problem to be Solved
ZE: Zero Estimator t_0	$e^{-(t_0/f)} = n_0/f$
SE: Singleton Estimator t_1	$(t_1/f)\ e^{-(t_1/f)} = n_1/f$
CE: Collision Estimator t_c	$1-(1+(t_c/f))\ e^{-(t_c/f)} = n_c/f$

Table 1: Estimators for t

the fraction of empty slots is $e^{-\rho}$. From the current measurement the reader observes that the fraction of empty slots is n_0/f. Equating the expected value and the observed value, the reader now determines ρ_0 that solves $e^{-\rho_0} = n_0/f$ and sets $t_0 = f\rho_0$. Similarly, the reader can get estimates for t from the singleton slots as well as the collisions. We show the three estimates in Table I.

It is easy to solve for the estimator t_0 in closed form but the other two estimators involve solving a non-linear equation in one variable. A simple bisection search or Newton's method can be used to solve the equation, since the estimation functions shown above are well behaved and therefore both these methods converge very quickly. We can also use the fact that the estimate has to be an integer to terminate the search once we know the interval of uncertainty is less than one. We use the bisection search method for the results in this paper.

The three estimators have very different characteristics. In Figure 1 we plot the normalized expected values, $E[N_0]/f$, $E[N_1]/f$ and $E[N_c]/f$ as functions of the load factor ρ. Note that the curves for empty slots and collision slots are monotonic in ρ but singleton slots is non-monotonic. Intuitively, when the load factor is very low, there are many empty slots but very few singleton or collision slots. As the load factor increases, the number of empty slots decreases with a corresponding increase in the number of singleton and collision slots. The expected number of singleton slots attains a maximum when the load factor $\rho = 1$, a fact widely used in identification algorithms to optimize the number of successful identifications in a single frame. From this point on, as the load factor increases, there are many more collision slots and the number of singleton slots decreases. Thus, when the reader solves for ρ_1 in $\rho_1\ e^{-\rho_1} = n_1/f$, the solution is not unique for $\rho \neq 1$. This suggests that the singleton slots cannot be used alone for estimating the number of tags. Therefore, in the rest of the paper, we focus on the zero estimator (ZE) with estimate denoted by $\hat{t} = t_0$ and the collision estimator (CE) with estimate denoted by $\hat{t} = t_c$.

3.3 Operating Range for the Estimators

When the number of tags $t \gg f$, then all slots in the frame will encounter collisions with high probability, resulting in $n_0 = n_1 = 0$ and $n_c = f$. In such cases, both the zero estimator and the collision estimator will not have finite estimates, i.e., $t_0 = t_c = \infty$. Since the number of tags is assumed to be fixed, as the frame size increases (the load factor decreases), the probability that estimators are finite will increase. Rephrasing this, given a frame size f, there is an upper bound on the number of tags that can be estimated reliably using a given estimator.

Definition 1. *Given a frame size f, and a probability $\theta < 1$, the operating range for an estimator is defined as the maximum number of tags t for which the estimator has a finite solution with probability greater than θ.*

The definition of the operating range[3] simply ensures that we can get a finite estimate. For a fixed f, the objective for the zero estimator is to determine the maximum number of tags that will result in no empty slots with probability less $1-\theta$ and for CE is to determine the maximum number of tags that will result in collisions in all slots with probability less than $1-\theta$. We will use the following classical result due to von Mises (see Feller [10]) on the distribution of N_0 and N_1.

Lemma 2. *Let t tags each pick a slot randomly among f slots and transmit in that slot. Let $t, f \to \infty$ while maintaining $t/f = \rho$, then the number of empty slots, N_0 approaches a Poisson random variable with parameter $\lambda_0 = fe^{-\rho}$ and the number of singleton slots, N_1 is distributed approximately as a Poisson random variable with parameter $\lambda_1 = f\rho e^{-\rho}$ where $\rho = t/f$ is the load factor.*

Using the above result, the probability that the reader fails to get a finite t_0 is the probability that $N_0 = 0$. Since N_0 is distributed as a Poisson variable with parameter λ_0,

$$\Pr[N_0 = 0] = e^{-\lambda_0}.$$

Hence, requiring the failure probability to be less than $(1-\theta)$ is equivalent to setting $\lambda_0 \leq -\log(1-\theta)$. If we set $\theta = 0.99$, this corresponds to setting $\lambda_0 \leq 5$. (When $\lambda_0 = 5$, the failure probability is about 0.007.)

In the case of the CE estimator, the estimation process fails only if $N_c = f$, i.e., there are no empty or singleton slots. This probability is given by

$$\Pr[N_0 = 0, N_1 = 0] \approx e^{-\lambda_0-\lambda_1}.$$

Again using $\theta = 0.99$, we see that as long as the load factor ensures that $\lambda_0 + \lambda_1 \leq 5$ then the collision based estimator fails with probability less than 0.007. Figure 2 compares the operating ranges for the two estimators ZE and CE for $\theta = 0.99$, i.e., a failure probability of less than 1%. The x-axis gives the number of slots and the the y-axis gives the maximum number of tags t_0 and t_c that can be estimated using the ZE and CE estimators respectively. Note that the range for CE estimator is greater than the range of ZE. This difference increases with the frame size. For example, the operating range for CE is about 180 higher than ZE for $f = 100$ slots and is about 11600 higher when $f = 5000$ slots. Therefore, the collision-based estimator can operate at higher load factors than the empty slots based estimator.

In Figure 3, we show the experimental performance of collision-based and Zero Estimators when $f = 100$ and the number of tags are increased from 0 to 1000. The x-axis shows the actual number of tags and the y-axis shows the estimated number of tags, t_0 and t_c. The ideal curve is the 45 degree line shown in the plot. Each point in the plot represents the average of 100 experiments. Note that the performance of t_0 starts deteriorating when the number of tags t is about 350 and the performance of t_c starts deteriorating when t is about 550.

[3]The operating range does not say anything about the accuracy of the estimate when the estimate is finite. We discuss the accuracy of the estimate in Section 3.4.

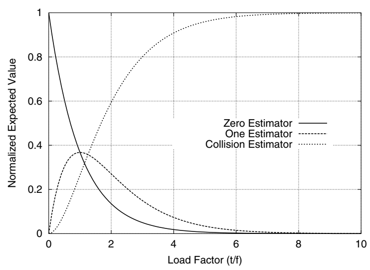

Figure 1: Normalized Expected Number of Slots

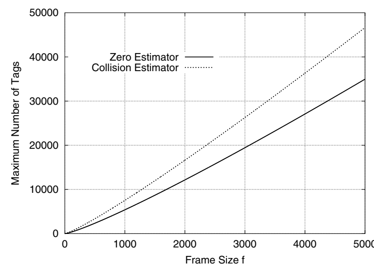

Figure 2: Operating Range

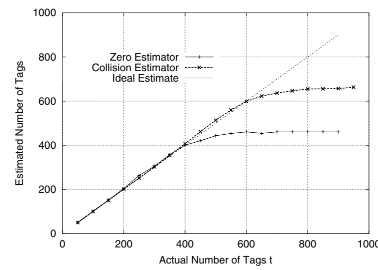

Figure 3: Experimental Testing of the Operating Range

The key observation is that with increasing frame size, the operating range expands for both estimators, with a bigger range for the Collision Estimator than for the Zero Estimator. This also implies that the Collision Estimator works well for a greater range of load factors than the Zero Estimator.

3.4 Accuracy of the Estimators

In this section, we derive the variance of the two estimators. Computing the variance serves two purposes. First, it lets us compare the accuracy of the two estimators. Second, it helps us decide how to use the estimators to get the desired accuracy level for the estimates. Previously, we used the fact that N_0 and N_1 can be approximated as Poisson random variables. The reason for preferring the Poisson distribution in the last section is that it is a discrete distribution and it gives a probability mass function value at zero for N_0 and N_1. It turns out that N_0 and N_1 can also be asymptotically approximated as normal distributions, a fact we use for analyzing the variance of the estimators. We denote a normal distribution with mean a and variance b with $\mathcal{N}[a, b]$.

THEOREM 1. *Let t tags each pick randomly among f slots and transmit in this slot. N_0 represents the number of empty slots, and N_c represents the number of collision slots. If $f, t \to \infty$ while maintaining $t/f = \rho$, then*

$$N_0 \sim \mathcal{N}\left[\mu_0, \sigma_0^2\right]$$

where

$$\mu_0 = fe^{-\rho}, \quad \sigma_0^2 = fe^{-\rho}\left(1 - (1+\rho)\, e^{-\rho}\right)$$

and

$$N_c \sim \mathcal{N}\left[\mu_c, \sigma_c^2\right]$$

where

$$\mu_c = f\left(1 - e^{-\rho}(1+\rho)\right) \quad \textit{and}$$

$$\sigma_c^2 = fe^{-\rho}\left((1+\rho) - \left(1 + 2\rho + \rho^2 + \rho^3\right) e^{-\rho}\right).$$

PROOF. See Feller [10] for a proof of the normality. The expressions for the mean of the random variables were derived in the proof of Lemma 1. We focus on the computation of the variance. Note that

$$Var[N_0] \;=\; E\left[\left(\sum_{j=1}^{f} X_j\right)^2\right] - \left(E\left[\sum_{j=1}^{f} X_j\right]\right)^2$$

Note that for $i \neq j$

$$E[X_i X_j] = Pr[X_i = 1, X_j = 1] = \left(1 - \frac{2}{f}\right)^t.$$

Plugging this result in the expression for the variance, and using the fact that $E[X_j^2] = E[X_j]$, we get

$$\begin{aligned} Var[N_0] &= f(f-1)(1-\frac{2}{f})^t + f(1-\frac{1}{f})^t - f^2(1-\frac{1}{f})^{2t} \\ &\approx -te^{-\rho} + f\left(e^{-\rho} - e^{-2\rho}\right), \textit{(Appendix:Lemma 3)} \\ &= fe^{-\rho}\left(1 - (1+\rho)\, e^{-\rho}\right) \end{aligned}$$

The proof for the computation of $Var[N_c]$ for a more general case is given in [13]. □

In Figure 4, we show the experimental distribution of the number of collision slots, superimposed on the normal distribution with the mean and variance computed as in Theorem 1.

Ultimately, we want to measure instances of N_0 and N_c and use these to estimate t. Note that we view μ_0 and μ_c as (non-linear) functions of the number of tags t, i.e, as $\mu_0(t)$ and $\mu_c(t)$. From Lemma 1 and Figure 1, we know that both $\mu_0(t)$ and $\mu_c(t)$ are monotonic continuous functions of t. ($\mu_0(t)$ is increasing in t and $\mu_c(t)$ is decreasing in t.) Since they are monotonic and continuous, both these functions have unique inverses, denoted by $g_0()$ and $g_c()$ respectively. In other words, $g_0(\mu_0(t)) = t$ and $g_c(\mu_c(t)) = t$.

THEOREM 2. *Let $t, f \to \infty$ while maintaining $t/f = \rho$. Then*

$$\left[g_0(N_0) - g_0(\mu_0(t))\right] \sim \mathcal{N}[0, \delta_0]$$

and

$$\left[g_c(N_c) - g_c(\mu_c(t))\right] \sim \mathcal{N}[0, \delta_c]$$

where

$$\delta_0 = t\frac{(e^{\rho} - (1+\rho))}{\rho} \tag{1}$$

and

$$\delta_c = t\frac{(1+\rho)\, e^{\rho} - \left(1 + 2\rho + \rho^2 + \rho^3\right)}{\rho^3}. \tag{2}$$

PROOF. See Appendix for the proof of this theorem. □

In order to compare the variances of the two estimators, we first define the notion of a normalized variance which is the ratio of the estimator variance to the number of tags.

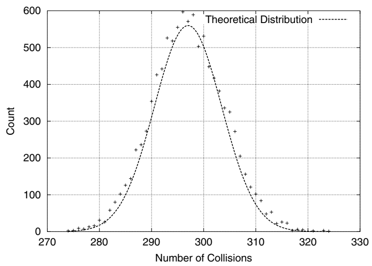

Figure 4: Distribution of Number of Empty Slots

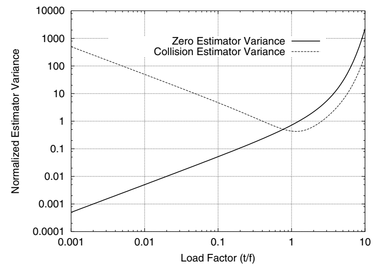

Figure 5: Normalized Estimator Variance

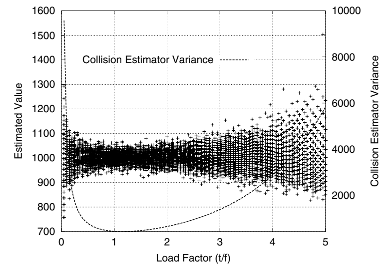

Figure 6: Variance of t_c

The normalized variance of ZE is δ_0/t and for CE is δ_c/t. From the expressions for the estimator variance shown above, the normalized variance is just a function of the load factor ρ. Figure 5 plots the normalized variance as a function of the load factor ρ. There are two factors that stand out in this plot.

1. For $\rho < 1$, the variance of t_0 decreases while the variance of t_c increases. In other words, if we fix the number of tags t and increase the number of slots, the ZE estimator gets more accurate while the collision based CE estimator gets less accurate.

2. As the load factor increases, the variance of the collision based estimator is significantly less than the variance of the empty slots based estimator.

3. ZE can achieve arbitrarily low normalized variance, while CE's normalized variance is always at least 0.425.

These observations suggest that for estimating a given tag set with cardinality t, one can use the ZE with a very large frame size to obtain any desired accuracy in a single frame, assuming such a frame size is allowed. On the other hand, with CE, one has to investigate other methods of reducing the variance. We also note that the two estimators are complementary to each other. For load factors greater than a threshold, CE performs better, while the ZE estimator performs better at lower load factors.

3.5 Reducing the Variance of the Simple Estimators

We have the expressions for the variance of the ZE and CE, given in Equations 1 and 2. A straightforward way of reducing the variance of an estimator is to repeat the experiment multiple times and take the average of the estimates. If the final estimate is the average of m independent experiments each with an estimator variance of σ^2, then the variance of the average is σ^2/m. We can also manipulate the variance in our case by changing the frame size or perform a weighted average of the estimates. Before choosing a method for reducing the variance, we first need to understand the characteristics of the two estimators.

- The variance of ZE can be reduced by reducing ρ or equivalently, increasing the frame size f. We can show having a frame size of mf and performing the reading once, gives a lower variance for t_0, than having a frame size f and averaging the results of m experiments. If we want the variance to be less than σ^2, for a given estimate $\hat{t}$, we first set

$$\hat{t}\,\frac{(e^{\rho} - (1+\rho))}{\rho} \leq \sigma^2$$

  and solve for ρ. We can then set $f \geq \hat{t}\rho$.

- From Figure 6 we see that t_c attains minimum variance when $\rho = 1.15$. We obtain this by evaluating the minima of Equation 2 with respect to ρ. Note that $\rho = 1.15$ is equivalent to setting $f = (1/1.15)\ t = 0.87\ t$. The minimum variance when $\rho = 1.15$ is $0.425\ t$. If we fix $f = 0.87\ t$ and repeat the experiment m times, then average the m estimates, then the variance of the final estimate is reduced by a factor of about m, i.e., the variance will be $0.87\ t/m$. This suggests that if we want the final variance to be less than σ^2, then we have to repeat the measurement at least $\lceil 0.87\ t/\sigma^2 \rceil$ times.

There are two other practical issues that we have to address:

- *Maximum Frame Size*
  The frame size that arises from the computation for a desired variance may be quite large especially in the case of ZE. In practice, all systems have some maximum frame size restriction. Therefore, if the frame size computation above leads to a size larger than the maximum permitted, then we use the maximum permitted frame size instead. This implies that we may have to perform multiple experiments in order to reduce the variance, even for ZE. We use $f_{\max}$ to denote the maximum frame size.

- *Frame Overhead*
  There is typically some overhead associated with each frame, primarily the time/energy that is needed to energize the tags. We assume that the frame overhead is specified in terms of the number of slots that are needed to initialize a frame and we denote this quantity by τ. Therefore, a frame size f actually uses up $f + \tau$ slots.

Based on our earlier observation that the two estimators are complementary to each other, we can devise a unified Simple Estimation Algorithm, which uses both estimators, depending on the frame size and the estimated number of tags. For a given frame size and tag set estimates from the two estimators, we choose the value with the lowest variance, and use it to refine subsequent estimates. This unified algorithm is described in Figure 7.

The simple estimators presented in this section are capable of estimating the value of t within a 20% confidence interval ($\beta = 20$) in a single frame, assuming that the frame size is appropriately chosen, once, to accomodate the operating range of measurements. We demonstrate this by the experimental results in Figure 8, in which we plot the various estimated values in different experimental runs over a single frame, against the tag set cardinality, t. The frame size is selected to be 1000 slots. For comparison, we also plot the 20% confidence interval represented by the upper and lower lines with slopes 1.1 and 0.9 respectively. We also show the results of the tag set size estimator in [18] where a lower bound is computed as $\hat{t} \geq n_1 + 2 * n_c$. Clearly, the estimators presented in this paper are far superior and are able to provide a high confidence estimate in as few as 1000 slots.

With our proposed estimation schemes[4] in the I-Code system[9], we can estimate the size of a 4500-tag set with 20% accuracy in 0.25 seconds. To achieve the same accuracy with the identification scheme in [9], we need 18 seconds. Even if we had only 500 tags in the system, our scheme will still estimate the size within ± 100 in 0.25 seconds, while [9] requires 2 seconds.

We earlier mentioned that with m multiple experimental estimates, averaging the estimates will reduce the variance by m. We can reduce the variance even further by weighted averaging of the estimates. We use the following well-known statistical result.

Theorem 3. *Let $e_1, e_2, \dots e_k$ be k estimates for t with variances $v_1, v_2, \dots, v_k$. For any set $\{\alpha_i\}$ with $0 \leq \alpha_i \leq 1$ and $\sum_i \alpha_i = 1$, $\sum_{i=1}^{k} \alpha_i e_i$ is an estimator for t with variance $\sum_{i=1}^{k} \alpha_i^2 v_i$. The optimal choice of α_i that minimizes the variance of the linear combination is*

$$\alpha_i = \frac{\frac{1}{v_i}}{\sum_{i=1}^{k} \frac{1}{v_i}}$$

and the minimum variance is $1/\sum_{i=1}^{k} \frac{1}{v_i}$.

Proof. Minimizing the weighted variance function subject to the $\sum \alpha_i$ constraints is a standard convex optimization problem. Solving the Kuhn-Tucker conditions gives the above optimal solution. □

We use weighted statistical averaging to compute the final variance of the sampled estimates in our simulations.

Using the Combined Simple Estimators, we measure the number of slots needed to estimate various tag sets with set sizes ranging from 5 to 50,000. In order to accomodate this large operating range, we need to set the initial frame size as described in Figure 7, which turns out to be 6984 slots. Note that subsequent frames can be of different sizes. For various levels of accuracy, we find the number of slots needed for estimation using simulations, and the results are shown in Table II. The results show that for a tag sizes well within the operating range, the algorithm easily estimates the number of tags to within a couple of frames with an accuracy of greater than 0.05%. However, there is very little tunability of the algorithm for various levels of accuracy. In addition, as the desired accuracy increases or as the number of tags

INPUT
1. Upper bound $\bar{t}$ on the number of tags t
2. Confidence Interval Width β
3. Error probability α

ESTIMATION PROCEDURE

1. Compute the desired variance, $\sigma^2 = \frac{Z_\alpha^2}{\beta^2}$.
2. Compute the initial frame size f by solving $fe^{-(\bar{t}/f)} = 5$.
3. Energize the tags and get n_0 and n_c.
4. Compute t_0 as in Table 1 and the variance of this estimate δ_0 using Equation 1.
5. Compute t_c as in Table 1 and the variance of this estimate δ_c using Equation 2.
6. If $\delta_0 < \delta_c$ then set $\hat{t} \leftarrow t_0$ else $\hat{t} \leftarrow t_c$.
7. Compute the frame size f_{ZE} needed for ZE by solving for ρ in
$$\hat{t}\frac{(e^\rho - (1+\rho))}{\rho} = \sigma^2$$
and setting $f_{ZE} = \hat{t}/\rho$ and set the number of repetitions $m = 1$.
8. If $f_{ZE} > f_{\max}$ then set $f_{ZE} = f_{\max}$. Now set $\rho = \hat{t}/f_{\max}$ in Equation 1 and obtain the variance $\hat{\sigma^2}$. The number of repetitions needed is $m = \hat{\sigma^2}/\sigma^2$.
9. Therefore the total number of slots for ZE denoted by T_{ZE} including the frame overhead is $m(\tau + f_{ZE})$.
10. Let $f_{CE} = 1.15 * \hat{t}$. If $f_{CE} > f_{\max}$ then set $f_{CE} = f_{\max}$.
11. Set $\rho = \hat{t}/f_{\max}$ in Equation 2 and obtain the variance $\hat{\sigma^2}$. The number of repetitions needed is $m = \hat{\sigma^2}/\sigma^2$.
12. Therefore the total number of slots for CE denoted by T_{CE} including the frame overhead is $m(\tau + f_{ZE})$.
13. If $T_{ZE} < T_{CE}$, then we use ZE else we use CE and use the appropriate number of repetitions and average the estimate over all the repetitions.

Figure 7: Computing the Estimate using Combined Simple Estimators

reaches the upper end of the operating range, the algorithm takes a large number of slots to obtain the desired estimate, mainly due to the large frame sizes involved. But, it is worth pointing out that the algorithm can obtain an estimate of 50,000 tags within an confidence interval of ± 500 tags in 4.5 seconds and within ± 50 tags in 16 seconds, while the identification time can be more than 100 seconds.

In Table III, we show the impact of the operating range on the number of slots needed to estimate 500 tags with a confidence interval of ± 5 tags. This shows that there is a tradeoff between operating range and estimation time, when the number of tags is well within the operating range.

4. A SCALABLE ESTIMATION ALGORITHM

As described in the combined algorithm in Section 3, estimation schemes that allow all tags to contend in every frame have a specific operating range that is dependent on the frame size chosen. Hence, in order to estimate any tag

[4] See slot size assumptions in Section 2.

Number of tags	Slots needed for confidence interval (in %) of			
	0.2	0.1	0.05	0.02
5	6984	6984	6984	6984
50	7028	7256	7826	14837
500	7498	7498	8666	15674
5000	12736	12736	12736	18558
50000	65378	65378	182306	883874

Table 2: Unified Simple Estimation Algorithm: Large Operating Range

Slots needed when operating range is set to				
500	1000	5000	10000	50,000
714	870	1554	2302	7498

Table 3: Impact of Operating Range on Estimation Time

set size, an upper bound on the size t has to be assumed, and the frame size fixed such that the operating range stretches up to the upper bound. In addition, the optimal frame sizes for computing a low variance estimate are lower bounded by the tag size t for both the Zero and Collision Estimators. This requirement can be impractical in actual systems for large values of t.

In this section, we address the case when the number of tags can be orders of magnitude larger than the maximum frame size f_{max}, where f_{max} can be very small. We show that extending the framed ALOHA to include probabilistic contention increases the range and improves the accuracy of both the estimates when the load factor is large. To this end, we define the *probabilistic framed ALOHA (PFA) protocol.*

DEFINITION 2. *The* probabilistic framed ALOHA (PFA) *protocol is defined as the framed ALOHA protocol model with a frame size f and an additional contention probability p. A node in the PFA protocol decides to contend in a frame with probability p, and if it decides to contend, it picks one of the f slots to transmit.*

Note that the PFA protocol is just a simple extension of probabilistic ALOHA to the framed model. When $p = 1$, PFA protocol becomes the Framed Aloha protocol. Our goal is analyze the behavior of a probabilistic estimator in such a model. The next theorem is analogous to Theorem 1 and it gives the mean and the variance for the number of empty slots and collision slots for the PFA scheme.

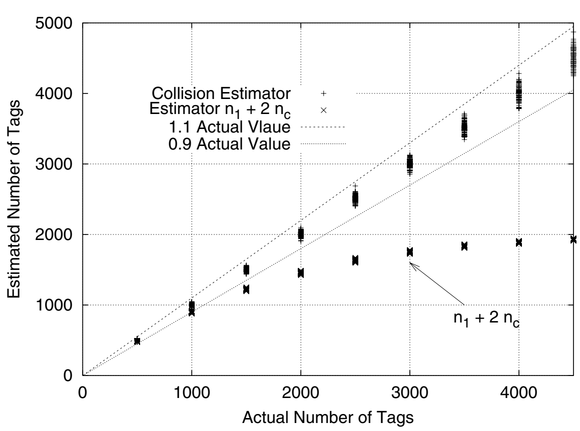

Figure 8: Estimation over a Single Frame

Estimator	Problem to be Solved
PZE: Empty Slot Estimator t_0	$e^{-(pt_0/f)} = n_0/f$
PCE: Collision Estimator t_c	$1 - (1 + (pt_c/f))\ e^{-(pt_c/f)} = n_c/f$

Table 4: Estimators for t

THEOREM 4. *Let t tags each pick randomly among f slots and transmit in that slot if they choose to contend with probability p. Then*

$$N_0 \sim \mathcal{N}\left[\mu_0, \sigma_0^2\right]$$

$$N_c \sim \mathcal{N}\left[\mu_c, \sigma_c^2\right]$$

where
$\mu_0 = fe^{-p\rho}, \quad \sigma_0^2 = fe^{-p\rho}\left(1 - \left(1 + p^2\rho\right)e^{-p\rho}\right),$
$\mu_c = f\left(1 - e^{-p\rho}\left(1 + p\rho\right)\right),$ *and*
$\sigma_c^2 = fe^{-p\rho}\left(\left(1 + p\rho\right) - \left(1 + 2p\rho + p^2\rho^2 + p^4\rho^3\right)e^{-p\rho}\right).$

PROOF. The proof is similar to the proof of Theorem 1. □

The estimators t_0 and t_c for the probabilistic case are computed as shown in Table 4. We use the abbreviations PZE and PCE to denote the estimators in the case of probabilistic framed aloha.

We now give the variance of the estimators, and the derivation is similar to the framed ALOHA derivation in Theorem 2. Let $g_0(x)$ and $g_c(x)$ be the estimator functions for the PFA model, with $g_0(\mu_0) = g_c(\mu_c) = t$, where μ_0 and μ_t are defined as in Theorem 4.

THEOREM 5. *N_0 and N_c are the number of empty slots and number of collision slots, respectively. We have,*

$$\left[g_0\left(N_0\right) - g_0\left(\mu_0\right)\right] \sim \mathcal{N}\left[0, \delta_0\right]$$

and

$$\left[g_c\left(N_c\right) - g_c\left(\mu_c\right)\right] \sim \mathcal{N}\left[0, \delta_c\right]$$

where

$$\delta_0 = t\frac{\left(e^{p\rho} - \left(1 + p^2\rho\right)\right)}{\rho p^2}$$

and

$$\delta_c = t\frac{\left(1 + p\rho\right)e^{p\rho} - \left(1 + 2p\rho + p^2\rho^2 + p^4\rho^3\right)}{\rho^3 p^4}.$$

PROOF. The proof of this theorem can be found in [13]. □

The main use of the probabilistic scheme is for handling cases where the number of tags is large and it is not feasible to increase the frame size to accommodate the tags. Therefore, we temporarily assume that the values of t and f are fixed and the load factor $\rho = t/f$ is large.

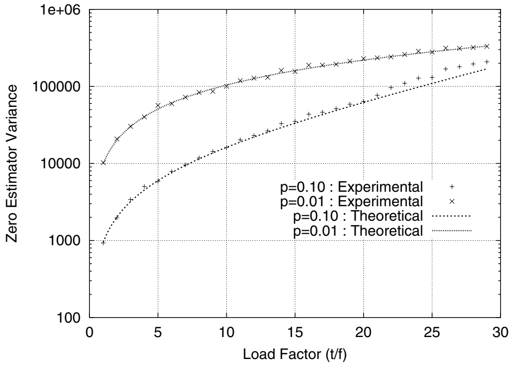

Figure 9: Experimental plot of variance of PZE

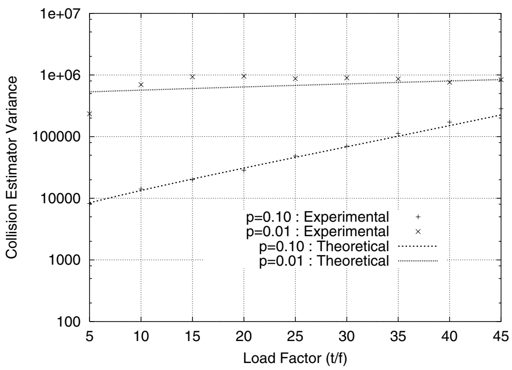

Figure 10: Experimental plot of variance of PCE

4.1 Choosing the Optimal Contention Probability

We now address the problem of choosing the optimal contention probability given the load factor ρ. We do this for both PZE and PCE estimators. In the case of PZE, we compute the partial derivative of δ_0 with respect to p and set it to zero to get the minimum variance.

$$\frac{\partial \delta_0}{\partial p} = \frac{t(p\rho e^{p\rho} - 2e^{p\rho} + 2)}{p^4 \rho^2} = 0$$

In order to obtain the minimum variance, we have to solve for p in

$$p\rho e^{p\rho} - 2e^{p\rho} + 2 = 0.$$

Since $p\rho$ occurs together in the expression, it is easy to show numerically that the minimum is attained when p is chosen such that $p\rho = 1.59$. Therefore, the optimal $p = 1.59/\rho$. If $\rho < 1.59$ then the optimal value for $p = 1$. In Figure 9, we show how the variance changes with p for two different values of ρ. In the case of PCE, we seek the minima of the function $\delta_c(p, \rho)$ with respect to p to obtain the value of p that gives minimum variance. The optimal p has to satisfy

$$e^{p\rho}\left(p^2\rho^2 - 2p\rho - 4\right) + 2\left(p^2\rho^2 + 3p\rho + 2\right) = 0$$

Once again, note that p and ρ occur together, therefore it is easy to solve this numerically to show that the minimum is attained when $p\rho = 2.59$. Therefore $p = 2.59/\rho$ and $p = 1$ if $\rho < 2.59$.

INPUT
Frame Size f, confidence Interval Width β and error probability α
OUTPUT
An estimator $\hat{t}$ for t that satisfies the accuracy requirement.
INITIALIZATION

Set $p = 1$, $n_c = 0$.
While $n_c = 0$ do
Read Tags (f, p)
If $p = 1$ and $n_0 > 0$ get t_0
If $t_0 < f$ then
$e_1 = t_0$, $v_1 = \delta_0$, $\hat{t} = e_1$ and $\hat{v} = v_1$
PROCEDURE ZERO ESTIMATE
Else
$e_1 = t_c$, $v_1 = \delta_c$, $\hat{t} = e_1$ and $\hat{v} = v_1$
PROCEDURE COLLISION ESTIMATE
Else If $n_c = 0$, Then Set $p \leftarrow 0.1\ p$
Else set $e_1 = t_c$, $v_1 = \delta_c$, $\hat{t} = t^1$ and $\hat{v} = v^1$
PROCEDURE COLLISION ESTIMATE
End While

PROCEDURE COLLISION ESTIMATE

While $\left(\hat{v} > \frac{\beta^2 \hat{t}^2}{Z_\alpha^2}\right)$ do
Set $k \leftarrow k + 1$
Let $p = \min\{1, \frac{2.6f}{t}\}$
Read Tags (f, p) and get estimate t_c
Compute estimator variance δ_c
Set $e_k = t_c$, $v_k = \delta_c$
Compute new estimate $\hat{t} = \left(\sum_{i=1}^{k} \frac{e_i}{v^i}\right) / (\sum_{i=1}^{k} \frac{1}{v_i})$
Compute new variance $\hat{v} = (\sum_{i=1}^{k} \frac{1}{v_i})^{-1}$
End While

PROCEDURE ZERO ESTIMATE

While $\left(\hat{v} > \frac{\beta^2 \hat{t}^2}{Z_\alpha^2}\right)$ do
Set $k \leftarrow k + 1$
Let $p = \min\{1, \frac{1.5f}{t}\}$
Read Tags (f, p) and get estimate t_0
Compute estimator variance δ_0
Set $e_k = t_0$, $v_k = \delta_0$
Compute new estimate $\hat{t} = \left(\sum_{i=1}^{k} \frac{e_i}{v^i}\right) / (\sum_{i=1}^{k} \frac{1}{v_i})$
Compute new variance $\hat{v} = (\sum_{i=1}^{k} \frac{1}{v_i})^{-1}$
End While

Figure 11: Unified Probabilistic Estimation Algorithm

We also see that the collision based estimator is robust over a larger range than the empty slots based estimator. Figure 10 shows how the variance changes with respect to p for two different values of ρ. What the probabilistic contention mechanism lets us do is to dramatically reduce the frame size needed, even when t is large. If the variance for a single estimate is too large, then averaging multiple estimates reduces variance.

We combine the PZE and PCE estimators, using the same approach as in Section 3, to obtain a unified Probabilistic Estimation Algorithm, described in Figure 11.

One of the advantages of the PFA protocol is that the total estimation time for given accuracy level is independent of the cardinality t of the tag set. We illustrate this using some simulation results next. As in Section 3, we measure the number of slots needed to estimate various tag sets with set sizes ranging from 5 to $50,000$. We do not need to

Number of tags	Slots needed for confidence interval (in %) of			
	0.2	0.1	0.05	0.02
5	90	390	1440	8970
50	60	210	840	5190
500	180	540	2220	12420
5000	210	600	2220	13590
50000	240	660	2280	13560

Table 5: Results from Unified Probabilistic Estimation Algorithm

worry about the operating range here, since the probability of contention will be adapted dynamically depending on the number of tags present. We set the frame size to be a constant, $f = 30$ slots, and vary the probability of contention in a frame.

For various levels of accuracy, we find the number of slots needed for estimation using simulations, and the results are shown in Table 4.1. The main observation that stands out is that for a given confidence level, the number of slots needed is nearly independent of the tag size. The second observation is that the estimation slots needed is easily orders of magnitude smaller than those needed for the unified simple estimation algorithms (Table 3.5). As the accuracy requirement increases by a factor of x, the estimation time increases by a factor of x^2. This is expected, since the variance is related to the square of the confidence levels. It can be seen that the estimation time for tag set size of 50 is much better than the time for tag set of size 5. This is because of the frame size of 30, which results in near-optimal variance estimate for the size of 50, when compared to 5. In terms of actual time, assuming a rate of 4000 estimation slots per second, we can claim that one can estimate any tag size (not necessarily restricted to 50,000), with confidence interval of 0.05 within 1 second. This is an entirely unique result.

5. RELATED WORK

The basic RFID standards are covered in [4] and in [3]. where [3] proposes a *Q-algorithm* which attempts to set the frame-size to be equal to the number of unidentified tags. Every slot is ACK-ed in this model, and the frame size is multiplied (divided) by β, $(1.07 \leq \beta \leq 1.41)$, for each detected collision (zero) in a slot. Vogt [18] presents a simple estimation algorithm for estimating the number of tags using the number of ones and collisions in a given frame. The algorithm estimates the number of tags using the following equation $N_{est} = c_1 + 2c_M$, where c_1 and c_M are the number of ones and collisions in a given frame respectively. It then provides an expression to compute the number of slots needed to detect $\alpha.N$ tags, where $\alpha = 0.99$ in the simulations.

Hernandez [19] and Zhen [20] use a continuous-time model to evaluate the probability that a particular tag is not identified. The time between tag attempts to transmit is exponentially distributed in [19], and in [20], the time taken for reading N tags such that the probability of not reading a particular tag is minimized is computed as $18.76N$. All tags are assumed to transmit for every probe, and N is assumed to be known. In this paper, we compute the probability that *all tags are identified with high certainty*, while estimating N too.

When tags can be silenced based on feedback/input from the reader, then [22], [25], [23] and [24] provide estimation and identification algorithms. In [22] and [25], the number of unidentified tags is 2.39 times the number of collisions in the current frame, and sets the frame size to be equal to the number of unidentified tags. This number is derived by attempting to maximize the number of successful tag transmissions in a given frame, assuming frame size can be dynamically varied. Floerkemeier [23, 24] uses a Bayesian probability estimation scheme, based on the number of 1s and 0s in current frame, to maximize throughput (i.e., number of 1s per frame).

Zhen [28] uses Schoute's [25] estimation algorithm, and analyzes the minimum number of slots needed to identify all tags such that the expected number of identified tags is N. It claims that $1.4N$ is the optimal frame size (through simulations), and it takes 6.6 frames for total read time if the tags repeat transmission for every probe (passive tags). For active tags, they identify $0.65N$ as the ideal frame size, and claim that it takes 3 frames for identifying all tags. In addition, they also analyze the capture effect in tags for both Rician and Rayleigh fading channels. Lee et. al. [34] show that the optimal frame size is equal to number of unidentified tags, and hence, for a given frame size, they use a modulo operation to restrict the number of responding tags to around the frame size. In [42], an analysis of TDMA, random access and pseudo-random protocols is done regarding the trade off in the average energy consumed per slot versus the average delay per packet sent from the tag, in the context of active tags with a Poisson arrival process.

In an early work, Wieselthier et. al. [33] provide an analysis for framed ALOHA with and without the capture effect. They consider the various states in which a frame can evolve, and provide results that characterize the probability of each such state with a static frame size as well as dynamic frame sizes. In the dynamic case, the frame size is optimized to achieve maximum throughput. They also consider the case where an unsuccessful node transmits with different probabilities in each frame, until it succeeds. They show through simulations that per-frame throughput with constant packet transmission probability performs just as well as variable probabilities. In addition, they also show that if successful nodes stay quiet, then the optimal frame size is equal to N. In this paper, our goal is not just to maximize ALOHA throughput, but to estimate the number of nodes, optimize the frame sizes, and adjust the contention probabilities as well to minimize the number of slots needed such that all tags are detected with high probability.

Deterministic RFID-tag identifications algorithms follow a tree-based mechanism wherein the range of addresses of the RFID tags are selectively narrowed down along a tree. These schemes assume that each tag can understand and respond to complex commands from the reader, such as responding only if the ID is within an address range specified by the reader. Examples of such algorithms can be found in [21, 26, 27, 37, 38, 39, 40, 43, 44, 29, 30]. While these algorithms deterministically resolve conflicts, they often discount the amount of time that readers spend in probing and isolating different segments of the tag population in computing the total time taken to resolve all conflicts. They also assume a slotted model, and not a framed model, wherein the reader responds before and after every slot, adding to the resolution overhead.

The HIPERLAN standard [31] and the Sift protocol [32, 36] attempt to minimize collisions in a CSMA protocol wherein all the nodes can hear each other, using a truncated geometric distribution for contention probabilities. The goal there is to have only one winner per slot, and assumes that the winner will not contend immediately again. Moreover, the nodes have to contend in every slot, and must have the ability to listen to the result of each slot.

Estimation using the number of zeroes in a frame is similar to hash-based estimators[11, 45]. The key difference is that hash estimators use a single-bit per slot to identify if there is a *zero* or a *hit* in that slot, where a hit includes a 1 or a *collision*. In the wireless model, one can additionally differentiate between a 1 and collision without additional effort, and hence one can design better estimators using this extra information.

Estimating the intersection of multiple hashed sets is a well-known problem in database literature[11]. Our approach that is outlined in this paper is a variant of that approach that incorporates averaging to improve accuracy. Further, as in the case of estimating cardinality, we are exploring extending collision based approaches to develop better intersection cardinality algorithms. Also, the use of temporal intersection to track dynamics appears to be a new application in the wireless context.

The problem of synchronizing multiple RFID readers is addressed in [14]. Engels and Sarma [16] propose using well-known graph coloring based algorithms on a bipartite graph to eliminate reader collisions. Waldrop et. al. [17] present a randomized channel assignment algorithms for readers to minimize collisions between them. Kim et.al. consider the fairness aspects when multiple readers are present [41] by trying to balance the coverage of the readers so that the number of transmitting tags covered by each reader is approximately equal.

6. CONCLUSION

In this paper, we present several estimation algorithms that enable us to identify a set of RFID tags in a very short period of time. We believe that our collision-based estimator and the probabilistic estimators are entirely novel, and we perform a detailed performance evaluation of all the estimators described in this paper via thorough analysis and simulation. We present two unified estimation algorithms that have complementary properties: the Unified Simple Estimator provides a high level of accuracy within a single frame, while the Unified Probabilistic Estimator has a running-time that is independent of the size of the estimated tag set, for a given level of accuracy. We believe that the techniques proposed in this paper will be of great use in other areas such as neighborhood estimation problems in wireless networks, the multiple RFID reader problem, and privacy related issues in RFID networks. Our ongoing research is focused on improved probabilistic identification algorithms which can be guarantee much faster resolution of tag identities in a dense tag environment.

7. REFERENCES

[1] "RFID: Wikipedia entry", *http://en.wikipedia.org/wiki/RFID*, 2006.

[2] K. Finkenzeller, "RFID handbook: Radio frequency identification fundamentals and applications", *John Wiley & Sons*, 2000.

[3] EPCglobal, "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version 1.0.9", 2005.

[4] ISO/IEC 15963, "Information technology – Radio frequency identification for item management – Unique identification for RF tags", 2004.

[5] L. Burdet, "RFID multiple access methods", *Technical Report*, ETH Zurich, 2004.

[6] R. Want, "An introduction to RFID technology", *IEEE Pervasive Computing*, Jan. 2006.

[7] T. Hassan and S. Chatterjee, "A taxonomy for RFID", *Hawaii International Conference on System Sciences*, Kauai, HI, Jan 2006.

[8] RFID Journal, *http://www.rfidjournal.com*, 2006.

[9] Philips Semiconductors, "I-CODE Smart Label RFID Tags", *http://www.semiconductors.philips.com/acrobat_download/other/identification/SL092030.pdf*

[10] Feller, W., "An Introduction to Probability Theory and Its Applications",*Vol 1*, John Wiley, 1968.

[11] K. Whang, B. Vander-Zanden, and H. Taylor, "A Linear Time Probabilistic Counting Algorithm for Database Applications", *ACM Transactions on Database Systems*, June, 1990.

[12] Rao, C.R., "Linear Statistical Inference and its Applications",

[13] M. Kodialam and T. Nandagopal, "Fast Estimation Schemes in RFID Systems", *Bell Labs Technical Report*, March 2006. (Available at *http://www.bell-labs.com/user/thyaga*).

[14] K.S. Leong, M.L. Ng and P.H. Cole, "Synchronization of RFID readers for dense RFID reader environments", *International Symposium on Applications and Internet Workshops*, Phoenix, AZ. Jan 2006.

[15] S. Chen and V. Thomas, "Optimization of inductive RFID technology", *IEEE International Symposium on Electronics and Environment*, 2001.

[16] D. Engels and S. Sarma, "The reader collision problem", *IEEE Systems, Man and Cybernetics*, 2002.

[17] J. Waldrop, D. Engels and S. Sarma, "Colorwave: An anticollision algorithm for the reader collision problem", *IEEE ICC*, 2002.

[18] H. Vogt, "Efficient Object Identification with Passive RFID Tags", *Lecture Notes in Computer Science*, Springer-Verlag, vol. 2414, 2002.

[19] P. Hernandez, J. D. Sandoval, F. Puente and F. Perez, "Mathematical Model for a Multi-read Anti-collision Protocol", *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2001.

[20] B. Zhen, M. Kobayashi and M. Shimizu, "To read transmitter-only RFID tags with confidence", *IEEE PIMRC '04*, Barcelona, Spain. Sept 2004.

[21] H. Choi, J. R. Cha and J. H. Kim, "Fast wireless anti-collision algorithm in ubiquitous ID system", *IEEE VTC '04*, Los Angeles, CA. Sep 2004.

[22] J. R. Cha and J.H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system", *IEEE International Conference on Parallel and Distributed Systems*, 2005.

[23] C. Floerkemeier and M. Wille, "Comparison of Transmission Schemes for Framed ALOHA based RFID Protocols", *Workshop on RFID and Extended Network Deployment of Technologies and Applications*, Phoenix, AZ. Jan 2006.

[24] C. Floerkemeier, "Transmission control scheme for fast RFID object identification", *IEEE PerCom Workshop on Pervasive Wireless Networking*, Mar 2006.

[25] F. C. Schoute, "Dynamic framed length ALOHA", *IEEE Transactions on Communications*, vol. 31(4), Apr 1983.

[26] A. Micic, A. Nayak, D. Simplot-Ryl and I. Stojmenovic, "A

hybrid randomized protocol for RFID tag identification", *IEEE International Workshop on Next Generation Wireless Networks*, Goa, India. 2005.
[27] I. Cidon and M. Sidi, "Conflict multiplicity estimation and batch resolution algorithms", *IEEE Transactions on Information Theory*, vol. 34(1), Jan 1988.
[28] B. Zhen, M. Kobayashi and M. Shimizu, "Framed ALOHA for multiple RFID objects identification", *IEICE Transactions on Communications*, vol. E88-B(3), Mar 2005.
[29] P.F. Greenberg and R.E. Ladner, "Estimating the multiplicities of conflict to speed their resolution in multiple access channels", *J. ACM*, vol. 34(2), Apr. 1987.
[30] P. Popovski, F.H.P. Fitzek, and R. Prasad, "Batch Conflict Resolution Algorithm with Progressively Accurate Multiplicity Estimation", *ACM DIALM-POMC Workshop*, Philadephia, PA. Oct. 2004.
[31] K. Cho, H. Shin and J. Lee, "Performance analysis of HIPERLAN channel access control protocol", *IEICE Transactions on Communications*, vol. E85-B(10),Oct 2002.
[32] Y.C. Tay, K. Jamieson and H. Balakrishnan, "Collision-minimizing CSMA and its applications to wireless sensor networks", *IEEE Journal on Selected Areas in Communications*, vol. 22(6), Aug 2004.
[33] J. Wieselthier, A. Ephremides and L. Michaels, "An exact analysis and performance evaluation of framed ALOHA with capture", *IEEE Transactions on Communications*, vol. (37(2), Feb 1989.
[34] S. Lee, S. Joo and C. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification", *IEEE MobiQuitous*, 2005.
[35] F. Cali, M. Conti and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit", *IEEE/ACM Transactions on Networking*, vol. 8(6), Dec 2000.
[36] K. Jamieson, H. Balakrishnan, and Y. C. Tay, "Sift: a MAC Protocol for Event-Driven Wireless Sensor Networks", Proceedings of the Third European Workshop on Wireless Sensor Networks, Feb 2006.
[37] M. Nanjundaiah and V. Chaudhary, "Improvement to the anticollision protocol specification for 900 MHz class 0 RFID tag", *International Conference on Advanced Information Networking and Applications*, 2005.
[38] F. Zhou, C. Chen, D. Jin, C. Huang and H. Min, "Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems", *ACM ISLPED '04*, Aug 2004.
[39] D. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration", *IEEE Symposium on Information Theory*, Cambridge, MA. Aug 1998.
[40] L. Bolotnyy and G. Robins, "Randomized pseudo-random function tree walking algorithm for secure radio-frequency identification", *IEEE Workshop on Automatic Identification Advanced Technologies*, Oct 2005.
[41] J. Kim, W. Lee, J. Yu, E. Kim and C. Lee, "Effect of localized optimal clustering for reader anti-collision in RFID networks: fairness aspects to the readers", *IEEE ICCCN*, 2005.
[42] I. Chlamtac, C. Petrioli and J. Redi, "Energy conserving access protocols for identification networks", *IEEE/ACM Transaction on Networking*, vol. 7(1), Feb 1999.
[43] C. Law, K. Lee, and K. Siu, "Efficient Memoryless Protocol for Tag Identification", *ACM DIAL-M Workshop*, Aug 2000.
[44] J. Myung and W. Lee, "An Adaptive Memoryless Tag Anti-Collision Protocol for RFID Networks", *IEEE ICC*, 2005.
[45] A. Krohn, M. Beigl and S. Wendhack, "SDJS: Efficient Statistics in Wireless Networks", *IEEE ICNP*, Berlin, Germany, Oct 2004.

8. APPENDIX

We use the following asymptotic results in deriving the mean and the variance of the estimators. These are a general form of the expressions in [11].

LEMMA 3. *If A, B, w and m are constants, then, for large m,*

$$A\left(1-\frac{B}{m}\right)^{w} \approx A\, e^{-Bw/m}$$

$$A\left(1-\tfrac{2}{m}\right)^{w} - B\left(1-\tfrac{1}{m}\right)^{2w} \approx \left[(A-B)+(B-2A)\tfrac{w}{m^2}\right] e^{-\frac{2w}{m}}$$

$\diamond$

Whenever there is an $\approx$ sign in the derivation of the expressions, we have used one of the two results to get the asymptotics.

We use the following standard inversion result from statistics to get error bounds on the estimators..

THEOREM 6. *Let X_n be a sequence of statistics such that*

$$\sqrt{n}\left[\frac{X_n}{n}-\theta\right] \to X \sim \mathcal{N}[0,\sigma^2(\theta)].$$

Let f be a differentiable function of one variable. Then

$$\sqrt{n}\left[f\left(\frac{X_n}{n}\right)-f(\theta)\right] \to f(X) \sim \mathcal{N}[0,\sigma^2(\theta)(f'(\theta))^2].$$

PROOF. See Rao [12] for the details of the proof. $\square$

We now give a proof of Theorem 2.

PROOF. Note that N_0 and N_c are normally distributed [10]. Moreover, from the definition of $g_0()$, note that

$$g_0(\mu_0(t)) = t.$$

Differentiating this equation with respect to t we get,

$$g_0'(\mu_0(t))\, \mu_0'(t) = 1.$$

Therefore,

$$g_0'(\mu_0(t)) = \frac{1}{\mu_0'(t)}.$$

From Theorem 6, the variance of the zero estimator of t,

$$\delta_0 = \sigma_0^2(t)\left[g'(\mu_0(t))\right]^2 = \frac{\sigma_0^2(t)}{\left[\mu_0'(t)\right]^2} \quad (3)$$

From Theorem 1, we know that $\mu_0 = fe^{-\rho}$ and $\sigma_0^2 = fe^{-\rho}\left(1-(1+\rho)\,e^{-\rho}\right)$. Differentiating $\mu_0()$ with respect to t

$$\frac{\partial \mu_0}{\partial t} = e^{-\rho}.$$

Substituting the above expression and that of σ_0^2 in Equation 3 gives us the result. The argument for the variance of the collision estimator follows the same steps. $\square$