Every Bit Counts - Fast and Scalable RFID Estimation

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 have been widely deployed for various applications such as object tracking, 3D positioning, supply chain management, inventory control, and access control. This paper concerns the fundamental problem of estimating RFID tag population size, which is needed in many applications such as tag identification, warehouse monitoring, and privacy sensitive RFID systems. In this paper, we propose a new scheme for estimating tag population size called *Average Run based Tag estimation* (ART). The technique is based on the average run-length of ones in the bit string received using the standardized framed slotted Aloha protocol. ART is significantly faster than prior schemes because its estimator has smaller variance compared to the variances of estimators of prior schemes. For example, given a required confidence interval of 0.1% and a required reliability of 99.9%, ART is consistently 7 times faster than the fastest existing schemes (UPE and EZB) for any tag population size. Furthermore, ART's estimation time is observably independent of the tag population sizes. ART is easy to deploy because it neither requires modification to tags nor to the communication protocol between tags and readers. ART only needs to be implemented on readers as a software module. ART works with multiple readers with overlapping regions.

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, Estimation

MobiCom'12, August 22–26, 2012, Istanbul, Turkey.
Copyright 2012 ACM 978-1-4503-1159-5/12/08 ...$15.00.

1. INTRODUCTION

1.1 Motivation and Problem Statement

RFID systems are widely used in various applications such as object tracking [12], 3D positioning [19], indoor localization [13], supply chain management [9], inventory control, and access control [4, 11] because the cost of commercial RFID tags is negligible compared to the value of the products to which they are attached (*e.g.*, as low as 5 cents per tag [15]). An RFID system consists of tags and readers. A tag is a microchip 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 are powered up by harvesting the radio frequency energy from readers (as they do not have their own power sources) and have communication range often less than 20 feet; (2) active tags, which have their own power sources and have relatively longer communication range. A reader has a dedicated power source with significant computing power. It transmits a query to a set of tags and the tags respond over a shared wireless medium.

This paper concerns the fundamental problem of estimating the size of a given tag population. This is needed in many applications such as tag identification, privacy sensitive RFID systems, and warehouse monitoring. In tag identification protocols, which read the ID stored in each tag, population size is estimated at the start to guide the identification process. For example, for tag identification protocols that are based on the framed slotted Aloha protocol (standardized in EPCGlobal Class-1 Generation-2 (C1G2) RFID standard [3] and implemented in commercial RFID systems), tag estimation is often used to calculate the optimal frame size [7]. In privacy sensitive RFID systems, such as those used in parks for continuously monitoring the number of visitors in different areas of a park to plan the guided trips efficiently, readers may not have the permission to identify human individuals. In warehouses with RFID-based monitoring systems, managers often need to perform a quick estimation of the number of products left in stock for various purposes such as the detection of employee theft. Note that although tag population size can be accurately measured by tag identification, the speed will be too slow.

We formally define the tag estimation problem as: *given a tag population of unknown size t, a confidence interval $\beta \in (0, 1]$, and a required reliability $\alpha \in [0, 1)$, a set of readers needs to collaboratively compute the estimated number of tags $\tilde{t}$ so that $P\left\{|\tilde{t} - t| \le \beta t\right\} \ge \alpha$.* When the number of readers is one, we call this problem *single-reader estimation*; otherwise, we call this problem *multi-reader estimation*.

A tag estimation scheme should satisfy the following three requirements:

1. *Reliability*: The actual reliability should always be greater than or equal to the required reliability. The reliability α given as input is called the *required reliability*. The reliability that an estimation scheme achieves is called its *actual reliability*.
2. *Scalability*: The estimation time needs to be scalable to large population sizes because in many applications, the number of passive tags can be very large due to their low cost, easy disposability, and powerless operation.
3. *Deployability*: The estimation scheme needs to be compliant with the C1G2 standard and should not require any changes to tags.

1.2 Proposed Approach

In this paper, we propose a new scheme called *Average Run based Tag estimation* (*ART*), which satisfies all of the above three requirements. The communication protocol used by ART is the standardized framed slotted Aloha protocol, in which 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 randomly picks a time slot in the frame and replies during that slot. Thus, the reader gets a binary sequence of 0s and 1s by representing a slot with no tag replies as 0 and a slot with one or more tag replies as 1. The key idea of ART is to estimate tag population size based on the average run size of 1s in the binary sequence. We show that the average run size of 1s in a frame monotonously increases with the increase in the size of tag population. Thus, average run size of 1s is an indicator of tag population size.

1.3 Advantages of ART over Prior Art

ART is advantageous in terms of speed and deployability. For speed, ART is faster than all prior schemes. For example, given a confidence interval of 0.1% and the required reliability of 99.9%, ART is consistently 7 times faster than the fastest existing schemes (*i.e.*, UPE [7] and EZB [8]) for any tag population size. The reason behind ART being faster than prior schemes is that the new estimator that we propose in this paper, namely the average run size of 1s, has significantly smaller variance compared to the estimators used in prior schemes (such as the total number of 0s [7, 8] and the location of the first 1 in the binary sequence [6]), as we analytically show in Section 4.2. An estimator with small variance is faster because the estimation process needs to be repeated fewer times to achieve the required reliability. *The intuitive reason that our estimator has smaller variance than prior estimators is that our estimator uses more information available in the bit sequence.* Furthermore, ART estimation time is observably independent of tag population sizes. In contrast, as tag volume increases, the estimation time of some prior schemes (*e.g.*, FNEB [6]) increases.

For deployability, ART neither requires modification to the tags nor to the communication protocol between tags and readers. ART only needs to be implemented on the reader side as a software module without any hardware modifications. ART also does not demand any unpractical system parameters beyond the C1G2 standard. In contrast, some prior schemes require modification to tags and some demand unrealistic system parameters. For example, the scheme in [14] requires each tag to store thousands of hash functions, which is not practical to implement on passive tags and is not compliant with the C1G2 standard. As another example, the scheme in [6] uses increasingly large frame sizes as population size increases (*e.g.*, the frame size required by the scheme in [6] is greater than half of tag population size), which soon exceeds the maximum limit allowed by the C1G2 Standard.

2. RELATED WORK

The first tag estimation scheme, called Unified Probabilistic Estimator (UPE), was proposed by Kodialam and Nandagopal in 2006 [7]. UPE uses the framed slotted Aloha protocol and makes estimation based on either the number of empty slots or that of collision slots in a frame. Besides this estimator having larger variance than ART, UPE requires the differentiation among empty, single, and collision slots, which takes significantly more time than differentiating between empty and non-empty slots. According to C1G2, a reader requires $300\mu s$ to detect an empty slot, $1500\mu s$ to detect a collision, and $3000\mu s$ to complete a successful read. In [8], Kodialam *et al.* proposed an improved framed slotted Aloha protocol based estimation scheme called Enhanced Zero Based (EZB) estimator, which performs estimation based on the total number of 0s in a frame. While UPE estimates population size in each round and averages the estimated sizes when all rounds are finished, EZB only records the total number of 0s in each frame and at the end of all rounds, EZB first averages the recorded values and then uses it to do estimation.

In [14], Qian *et al.* proposed an estimation scheme called Lottery Frame (LoF). Compared to UPE and EZB, LoF is faster; however, it is impractical to implement as it requires each tag to store a large number (*i.e.*, the number of bits in a tag ID times the number of frames, which can be in the scale of thousands) of unique hash functions. LoF needs to modify both tags and the communication protocol between readers and tags, which makes it non-compliant with C1G2. Han *et al.* proposed a tag estimation scheme called First Non Empty Based (FNEB) estimator, which is based on the size of the first run of 0s in a frame [6]. FNEB is based on an unrealistic assumption that frame size can be arbitrarily large. Li *et al.* proposed an estimation scheme called Maximum Likelihood Estimator (MLE) for active tags with the goal of minimizing power consumption of active tags [10]. In [17], Shah and Wong proposed a multi-reader tag estimation scheme which is based on an unrealistic assumption that any tag covered by multiple readers only replies to one reader.

3. ART — SCHEME OVERVIEW

3.1 Communication Protocol Overview

ART uses the framed slotted Aloha protocol specified in C1G2 as its MAC layer communication protocol. In this protocol, the reader first tells tags the frame size f, which is typically no more than 512 slots for practical reasons [16], and a random seed number R. Later in the paper, we will see how a simple use of seed number R will make it straightforward to extend our estimation scheme to use multiple readers with overlapping regions. Each tag within the transmission range of the reader then uses f, R, and its ID to select a slot in the frame by evaluating a hash function $h(f, R, ID)$ whose result is in $[1, f]$ following a uniform distribution. Each tag has a counter initialized with the slot number it chose to reply. After each slot, the reader first transmits an end of slot signal and then each tag decrements its counter by one. In any given slot, all the tags whose counters are equal to 1 respond to the reader. In essence, each tag picks a random slot from 1 to f following a uniform distribution. If no tag replies in a slot, it is called an *empty slot*; if exactly one tag replies, it is called a *singleton slot*; and if two or more tags reply, it is called a *collision slot*.

3.2 Estimation Scheme Overview

At the end of a frame, the reader obtains a sequence of 0s and 1s by representing an empty slot with 0 and a singleton or collision slot with 1. In this binary sequence, a *run* is a subsequence where all bits in this subsequence are 0s (or 1s) but the bits before and after the subsequence are 1s (or 0s), if they exist. For example, frame 011100 has 3 runs: 0, 111, and 00.

ART uses the average run size of 1s to estimate tag population size. The intuition is that as tag population increases, the average run size of 1s increases (and that of 0s decreases). We illustrate this intuition using the simulation results in Figure 1, which shows that the average run size of 1s increases as tag population size increases from 0 to 160. The markers in this figure are the average of 100 runs. The lines above and below each marker show the standard deviation of the experiments. This figure shows that given a tag population size and a frame size, there is a distinct expected value of the average run size of 1s. The expected value of the average run size of 1s is a monotonic function of the number of tags, which means that a unique inverse of this function exists. Thus, given the observed average run size of 1s, using the inverse function, we can get the estimated value $\tilde{t}$ of tag population size t. Similar to other tag estimation schemes, ART also uses multiple frames obtained from multiple rounds of the framed slotted Aloha protocol to reduce its estimation variance and therefore increase its estimation reliability. Using different seed values for different frames, in each frame, the same tag will choose a different slot to respond.

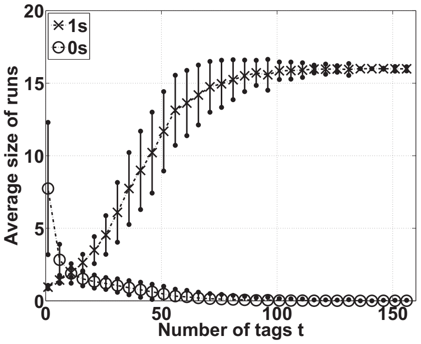


Figure 1: Average run size vs. t ($f = 16$)

To scale to large tag population sizes, ART uses a persistence probability p by which a tag decides whether it should reply to the reader in a given frame. The persistence probability was first introduced in [7]. To avoid making any modification to tags, this probability is implemented by "virtually" extending frame size $1/p$ times, *i.e.*, the reader announces a frame size of f/p but terminates the frame after the first f slots. According to C1G2, the reader can terminate a frame at any point. By adjusting p, ART is able to estimate tag population of arbitrarily large sizes.

3.3 Formal Development: Overview and Assumptions

To formally develop an estimator, we first need to derive the equation for the expected value of average run size of 1s as a function of frame size f, tag population size t, and persistence probability p. We then use the inverse of this function to get the estimated value $\tilde{t}$ from the observed value of the average run size of 1s. To achieve the required reliability and minimized estimation time, we optimize f, p, and the number of rounds n so that the total number of slots $(f+3)\times n$ is minimized while satisfying $P\{|\tilde{t}-t| \leq \beta t\} \geq \alpha$. We add 3 to f because at the start of each frame, the reader waits for about $1ms$ ($\approx$ 3 empty slots) to let the tags get energized [3, 16]. We use $\overline{f}$ to denote $f + 3$.

To make the formal development tractable, we assume that instead of picking a single slot to reply at the start of frame of size f, a tag independently decides to reply in each slot of the frame with probability $1/f$ regardless of its decision about previous or forthcoming slots. Vogt first used this assumption for the analysis of framed slotted Aloha protocol for RFID and justified its use by recognizing that this problem belongs to a class of problems known as "occupancy problems", which deals with the allocation of balls to urns [18]. Ever since, the use of this assumption has been a norm in the formal analysis of all Aloha based RFID protocols [2, 6–8, 10, 14, 17, 18, 20].

The implication of this assumption is that when a tag independently chooses a slot to reply, it can end up choosing more than one slots in the same frame or even not choosing any at all, which is not in accordance with C1G2 standard that requires a tag to pick exactly one slot in a frame. Note that even with the independence assumption, the expected number of slots that a tag chooses in a frame is still one. As we draw our estimate from a large number of frames to achieve required reliability, we can expect to observe this expected number. Therefore, the analysis with the assumption of independence is asymptotically the same as that without the independence assumption. Bordenave *et al.* further explained in detail why this independence assumption in analyzing Aloha based protocols provides results just as accurate as if all the analysis was done without this assumption [1]. Note that this independence assumption is made only to make the formal development tractable. The simulations in Section 6 are based on the C1G2 standard where a tag chooses exactly one slot at the start of frame.

4. ART — ESTIMATION ALGORITHM

Next, we first focus on the single-reader version of ART. In Section 5.6, we present a method to extend ART to handle multiple-readers with overlapping regions. Table 1 lists the symbols used in this paper.

4.1 Average Run Based Tag Estimation

For ART, in each round of the Aloha protocol, we calculate the average run size of b. For example, the average run size of 1 in frame 01110011 (which has two runs of 1, *i.e.*,

Table 1: Symbols used in the paper

Symbol	Description
t	actual tag population size
t_m	upper bound on # of tags
t_M	maximum # of tags that can be estimated
$\tilde{t}$	estimated # of tags
α	required reliability
β	required confidence interval
f	frame size
f_{op}	optimal frame size
$\overline{f}$	$f+3$ to include delay between two consecutive frames
n	# of rounds (*i.e.*, frames)
n_{op}	optimal # of rounds (*i.e.*, frames)
p	persistence probability
p_{op}	optimal persistence probability
R	random seed from reader
$h(f, R, ID)$	unform hash function in $[1, f]$
b	value of a slot: $b = 0$ or $b = 1$
$\overline{b}$	1-b
q_b	probability that a slot is b
$S_{b,i}$	random variable for run size of b starting at i
$E[.]$	expected value
Var(.)	variance
Cov(.)	covariance
Y_b	random variable for # of b slots in frame
y	element of sample space of Y_b
R_b	random variable for # of runs of b in frame
r	element of sample space of R_b
X_b	random variable for average run size of b in a frame
$\mu\{.\}$	expected value of X_b
$\sigma\{.\}$	standard deviation of X_b
$\xi\{f, y, r\}$	number of ways in which y occurrences of b and $f - y$ occurrences of $\overline{b}$ can be arranged in f slots while ensuring that the number of runs of b are r

111 and 11) is $(3 + 2)/2 = 2.5$. After n rounds, we obtain n average run sizes of b and then calculate the average of these n values. This final value is then used as the expected value of the average run size of b in a frame to estimate the tag population size.

The probability that a slot in a frame is b, where $b = 0$ or 1, can be calculated using Lemma 1.

LEMMA 1. *Let t be the actual tag population size, f be the frame size, p be the persistence probability (*i.e., *the probability that a tag participates in a frame), and q_b be the probability that a slot in a frame is b. Thus:*

$$q_b = \begin{cases} (1 - \frac{p}{f})^t & \text{if } b = 0 \\ 1 - (1 - \frac{p}{f})^t & \text{if } b = 1 \end{cases} \tag{1}$$

PROOF. The probability that a tag chooses a given slot in a frame is p/f. The probability that it does not choose that slot is $1 - \frac{p}{f}$. The probability that none of the tags choose that slot is $(1 - \frac{p}{f})^t$, which is the value of q_0. As the tags choose the slots independently, q_b is the same for each slot of the frame. The probability that a slot is chosen by at least one tag is $1 - q_0$, which is the value of q_1. □

Let $S_{b,i}$ be the random variable representing the run size of b starting at location i in a frame. If the i-th slot is not the starting point of a run, then $S_{b,i} = 0$. The expectation and variance of $S_{b,i}$ can be calculated using Theorem 1.

THEOREM 1. *Let $S_{b,i}$ be the random variable representing the run size of b starting at location i in a frame of size f. Let $a = f - i + 1$. The expectation and variance of $S_{b,i}$ are:*

$$E[S_{b,i}] = \frac{q_b}{1 - q_b}(1 - q_b^a) \tag{2}$$

$$\text{Var}(S_{b,i}) = \frac{q_b + (2a + 1)(q_b^{a+2} - q_b^{a+1}) - q_b^{2(a+1)}}{(1 - q_b)^2} \tag{3}$$

PROOF. To calculate the expected value of $S_{b,i}$, we first need its probability density function $P\{S_{b,i} = s\}$. The probability that a run starting at location i will be of length s is the product of the probability that s consecutive slots are b and $s + i^{th}$ slot is $\overline{b}$, where $\overline{b} = 1 - b$. Thus,

$$P\{S_{b,i} = s\} = \begin{cases} q_b^s(1 - q_b) & \text{if } s < a \\ q_b^s & \text{if } s = a \\ 0 & \text{if } s > a \end{cases}$$

where $1 \leq i \leq f$ and $1 \leq s \leq a$. To derive the expected value and variance of $S_{b,i}$, we use the moment generating function:

$$\phi(\tau) = E[e^{\tau S_{b,i}}] = \sum_{s=1}^{a} e^{\tau s} P\{S_{b,i} = s\}$$
$$= (q_b e^\tau)^a + (1 - q_b) \sum_{s=1}^{a-1} (q_b e^\tau)^s$$

Differentiating both side w.r.t. τ, we get

$$\phi'(\tau) = a(q_b e^\tau)^a + (1 - q_b)(q_b e^\tau) \sum_{s=1}^{a-1} s(q_b e^\tau)^{s-1}$$
$$= a(q_b e^\tau)^a + (1 - q_b)(q_b e^\tau) \frac{d}{d(q_b e^\tau)} \Big(\frac{1 - (q_b e^\tau)^a}{1 - q_b e^\tau} - 1 \Big)$$
$$= a(q_b e^\tau)^a + \frac{(1 - q_b)\Big((a - 1)(q_b e^\tau)^{a+1} - a(q_b e^\tau)^a + q_b e^\tau\Big)}{(1 - q_b e^\tau)^2}$$

Evaluation of $\phi'(\tau)$ at $\tau = 0$ gives us Equation (2). To find $\text{Var}(S_{b,i})$, we calculate $E[S_{b,i}^2]$ by taking the second derivative of $\phi'(\tau)$ and setting $\tau = 0$. Thus,

$$E[S_{b,i}^2] = \frac{q_b + q_b^2 + (2a - 1)q_b^{a+2} - (2a + 1)q_b^{a+1}}{(1 - q_b)^2}$$

Evaluating $E[S_{b,i}^2] - E^2[S_{b,i}]$ gives Equation (3). □

Let X_b be the random variable representing the average run size of b in a frame. Next, we calculate the expectation and variance of X_b using the expectation and variance of $S_{b,i}$ from Theorem 1. The expectation of X_b will be used to estimate the tag population size and the variance of X_b will be used to calculate optimal values for f, p, and n. Let Y_b be the random variable representing the number of times b occurs in a frame and R_b be the random variable representing the number of runs of b in a frame. By definition, $X_b = \frac{Y_b}{R_b}$ holds for any frame. Next, we first calculate $E[Y_b]$, $\text{Var}(Y_b)$, $E[R_b]$, $\text{Var}(R_b)$, and $\text{Cov}(Y_b, R_b)$ in Lemmas 2 and 3. Then, we use them to calculate $E[X_b]$ and $\text{Var}(X_b)$ in Theorem 2. Using Equation (14) in Theorem 2, replacing $E[X_b]$ by the average run size of b from n frames, we obtain an equation with only one variable t. Finally, we use Brent's method to obtain the numerical solution of this equation. The result is the estimated tag population size. Since ART uses X_b to estimate the tag population size, we call X_b the *estimator* of ART.

Lemma 2. *Let Y_b be the random variable representing the number of times b occurs in a frame and R_b be the random variable representing the number of runs of b in a frame. Given tag population size t, frame size f, and persistence probability p, we have:*

$$E[Y_b] = fq_b \tag{4}$$

$$\mathrm{Var}(Y_b) = fq_b(1-q_b) \tag{5}$$

$$E[R_b] = q_b\big(q_b + f(1-q_b)\big) \tag{6}$$

$$\mathrm{Var}(R_b) = f(q_b - 4q_b^2 + 6q_b^3 - 3q_b^4) + (3q_b^2 - 8q_b^3 + 5q_b^4) \tag{7}$$

Proof. The expected size of a run of b starting at location i is $E[S_{b,i}]$. A run of b starting at location i means that the slot $i-1$ is $\overline{b}$. Thus,

$$\begin{aligned}
E[Y_b] &= E[S_{b,1}] + \sum_{i=2}^{f} E[S_{b,i}] \times P\left\{\text{slot } i-1 \text{ is } \overline{b}\right\} \\
&= \frac{q_b}{1-q_b}(1-q_b^f) + \sum_{i=2}^{f} \frac{q_b}{1-q_b}(1-q_b^{f-i+1}) \times (1-q_b) \\
&= \frac{q_b}{1-q_b}(1-q_b^f) + q_b\Big[(f-1) - q_b^{f+1}\Big(\frac{q_b^{-2} - q_b^{-(f+1)}}{1-q_b^{-1}}\Big)\Big] \\
&= fq_b
\end{aligned}$$

Each slot i of frame f has probability q_b of being b. So $Y_b \sim \mathrm{Binom}(f, q_b)$. Thus, $\mathrm{Var}(Y_b)$ is simply the variance of a binomial distribution with parameters f and q_b:

$$\mathrm{Var}(Y_b) = fq_b(1-q_b)$$

Let $\gamma_1, \gamma_2, \ldots, \gamma_f$ represent the sequence of binary random variables representing the value of each slot in a frame of size f. Since each tag randomly and independently picks a slot in the frame, all γ_i are identically distributed. Furthermore, $P\{\gamma_i = b\} = q_b$. Let I_i be the indicator random variable whose value is 1 if a run of b begins at γ_i.

$$I_i = \begin{cases} 1 & \text{if } (\gamma_i = b, i = 1) \vee (\gamma_i = b \wedge \gamma_{i-1} = \overline{b}, i > 1) \\ 0 & \text{otherwise} \end{cases}$$

Thus,

$$R_b = \sum_{i=1}^{f} I_i$$

Because

$$E[I_i] = \begin{cases} P\{\gamma_i = b\} = q_b & \text{if } i = 1 \\ P\left\{\gamma_{i-1} = \overline{b}, \gamma_i = b\right\} = q_b(1-q_b) & \text{if } i > 1 \end{cases}$$

we get

$$E[R_b] = \sum_{i=1}^{f} E[I_i] = q_b + \sum_{i=2}^{f} q_b(1-q_b) = q_b\big(q_b + f(1-q_b)\big)$$

As R_b is the sum of f identically distributed random variables, we use the general expression for the variance of the sum of identically distributed random variables to obtain the variance of R_b.

$$\begin{aligned}
\mathrm{Var}(R_b) &= \mathrm{Var}(\sum_{i=1}^{f} I_i) \\
&= \sum_{i=1}^{f} \mathrm{Var}(I_i) + 2\sum_{j=2}^{f} \sum_{\forall i<j} \mathrm{Cov}(I_i, I_j)
\end{aligned}$$

Here we used the fact that the frame size is always greater than 1 during the estimation process whenever the information about runs is used. As $I_i \sim \mathrm{Bernoulli}(q_b)$, its variance is that of a bernoulli random variable given by

$$\mathrm{Var}(I_i) = E[I_i](1 - E[I_i]) \tag{8}$$

Note that I_i and I_j are dependent on each other if and only if (iff) $i = j-1$ because I_{j-1} and I_j can not both be 1 in the same frame. Other than that, $\forall i < j-1$, I_i and I_j are independent. Thus,

$$\mathrm{Cov}(I_i, I_j) = \begin{cases} 0 & \text{if } i < j-1 \\ -E[I_i]E[I_j] = -E[I_i]q_b(1-q_b) & \\ & \text{if } i = j-1 \end{cases}$$

Hence we have:

$$\begin{aligned}
\mathrm{Var}(R_b) &= \mathrm{Var}(I_1) + \sum_{j=2}^{f} \mathrm{Var}(I_j) + 2\mathrm{Cov}(I_1, I_2) \\
&\quad + 2\sum_{j=3}^{f} \mathrm{Cov}(I_{j-1}, I_j) \\
&= q_b(1-q_b) + (f-1)q_b(1-q_b)\{1 - q_b(1-q_b)\} \\
&\quad -2q_b^2(1-q_b) - 2(f-2)q_b^2(1-q_b)^2 \\
&= f(q_b - 4q_b^2 + 6q_b^3 - 3q_b^4) + (3q_b^2 - 8q_b^3 + 5q_b^4) \qquad \square
\end{aligned}$$

Lemma 3. *Given tag population size t, frame size f, and persistence probability p, we have:*

$$\begin{aligned}
\mathrm{Cov}(Y_b, R_b) = &\sum_{y=0}^{f} \sum_{r=0}^{\lceil \frac{f}{2} \rceil} yr q_b^y (1-q_b)^{f-y} . \xi\{f, y, r\} \\
&- E[Y_b]E[R_b]
\end{aligned} \tag{9}$$

where

$$\xi\{f, y, r\} = \begin{cases} \binom{y-1}{r-1}\Big[\binom{f-y-1}{r-2} + 2\binom{f-y-1}{r-1} + \binom{f-y-1}{r}\Big] \\ \text{if } r > 1 \wedge 0 < y < f \wedge r \le y \wedge r \le f-y-1 \\ \\ \binom{y-1}{r-1}\Big[2\binom{f-y-1}{r-1} + \binom{f-y-1}{r}\Big] \\ \text{if } r = 1 \wedge 0 < y < f \wedge r \le y \wedge r \le f-y-1 \\ \\ 1 \quad \text{if } r = 1 \wedge y = f \\ \\ 1 \quad \text{if } r = 0 \wedge y = 0 \\ \\ 0 \quad \text{otherwise} \end{cases}$$

Proof. By definition, we have

$$\mathrm{Cov}(Y_b, R_b) = \sum_{y=0}^{f} \sum_{r=0}^{f} yrP\{Y_b = y, R_b = r\} - E[Y_b]E[R_b] \tag{10}$$

Here $P\{Y_b = y, R_b = r\}$ represents the probability that exactly y out of f slots in the frame are b and at the same time the number of runs of b is r. This probability is difficult to evaluate directly, but conditioning on Y_b simplifies the task.

$$P\{Y_b = y, R_b = r\} = P\{R_b = r | Y_b = y\} \times P\{Y_b = y\} \tag{11}$$

As $Y_b \sim \mathrm{Binom}(f, q_b)$, we have:

$$P\{Y_b = y\} = \binom{f}{y} q_b^y (1-q_b)^{f-y} \tag{12}$$

Now we calculate $P\{R_b = r|Y_b = y\}$ *i.e.*, the probability of having r runs of b in a frame of size f given that y out of f slots are b. As tags choose the slots independently, each occurrence with r runs having y slots of b is equally likely. Therefore, we determine the total number of ways, denoted by $\xi\{f, y, r\}$, in which y occurrences of b and $f - y$ occurrences of $\bar{b}$ can be arranged such that the number of runs of b is r. We treat this as an ordered partition problem. First, we separate all the y occurrences of b from the frame and make r partitions of these y occurrences. Then, we create appropriate number of partitions of $f - y$ occurrences of $\bar{b}$ such that between consecutive partitions of b, the partitions of $\bar{b}$ can be *interleaved*. For r partitions of b, there are 4 possible partitions of $\bar{b}$.

1. The frame starts with b and ends with b, implying that there are $r-1$ partitions of $\bar{b}$, each interleaved between adjacent partitions of b.

2. The frame starts with b and ends with $\bar{b}$, implying that there are r partitions of $\bar{b}$.

3. The frame starts with $\bar{b}$ and ends with b, implying that there are r partitions of $\bar{b}$.

4. The frame starts with $\bar{b}$ and ends with $\bar{b}$, implying that there are $r+1$ partitions of $\bar{b}$.

We can make r partitions of y occurrences of b in $\binom{y-1}{r-1}$ ways and r partitions of $f - y$ occurrences of $\bar{b}$ in $\binom{f-y-1}{r-1}$ ways. Similarly, we can make $r+1$ partitions of $f-y$ occurrences of $\bar{b}$ in $\binom{f-y-1}{r}$ ways and $r-1$ partitions of of $f-y$ occurrences of $\bar{b}$ in $\binom{f-y-1}{r-2}$ ways. The equation of $\xi\{f, y, r\}$ in the lemma statement follows from this discussion. The total number of ways in which y zeros can be arranged among f slots is $\binom{f}{y}$. Thus, we get

$$P\{R_b = r|Y_b = y\} = \frac{\xi\{f, y, r\}}{\binom{f}{y}} \tag{13}$$

Substituting values from Equations (12) and (13) in (11) and (10) results in Equation (9). □

Theorem 2. *Given tag population size t, frame size f, and persistence probability p, we have:*

$$E[X_b] = \frac{E[Y_b]}{E[R_b]} - \frac{\text{Cov}(Y_b, R_b)}{E^2[R_b]} + \frac{E[Y_b]}{E^3[R_b]}\text{Var}(R_b) \tag{14}$$

$$\text{Var}(X_b) = \frac{\text{Var}(Y_b)}{E^2[R_b]} - \frac{2E[Y_b]}{E^3[R_b]}\text{Cov}(Y_b, R_b) + \frac{E^2[Y_b]}{E^4[R_b]}\text{Var}(R_b) \tag{15}$$

Proof. Let $g(Y_b, R_b) = X_b = \frac{Y_b}{R_b}$. The Taylor series expansion of g around (θ_1, θ_2) is given by:

$$g(Y_b, R_b) = \sum_{j=0}^{\infty}\Big\{\frac{1}{j!}\Big[(Y_b - \theta_1)\frac{\partial}{\partial Y'_b} + (R_b - \theta_2)\frac{\partial}{\partial R'_b}\Big]^j \times g(Y'_b, R'_b)\Big\}_{\substack{Y'_b = a_1 \\ R'_b = a_2}}$$

According to Bienaymé-Chebyshev inequality, we have $\theta_1 = E[Y_b]$ and $\theta_2 = E[R_b]$. Therefore, we get the following expansion of the Taylor series of $g(Y_b, R_b)$:

$$\begin{aligned} g(Y_b, R_b) = g(\theta_1, \theta_2) + \Big[(Y_b - \theta_1)\frac{\partial g}{\partial Y_b} + (R_b - \theta_2)\frac{\partial g}{\partial R_b}\Big] \\ + \frac{1}{2}\Big[(Y_b - \theta_1)^2\frac{\partial^2 g}{\partial Y_b^2} + 2(Y_b - \theta_1)(R_b - \theta_2)\frac{\partial^2 g}{\partial Y_b \partial R_b} \\ + (R_b - \theta_2)^2\frac{\partial^2 g}{\partial R_b^2}\Big] + O(j^{-1}) \end{aligned}$$

Taking the expectation of both sides, we get

$$\begin{aligned} E[g(Y_b, R_b)] \approx \frac{1}{2}\Big[\text{Var}(Y_b)\frac{\partial^2 g}{\partial Y_b^2} + 2\text{Cov}(Y_b, R_b)\frac{\partial^2 g}{\partial Y_b \partial R_b} \\ + \text{Var}(R_b)\frac{\partial^2 g}{\partial R_b^2}\Big] + g(\theta_1, \theta_2) \end{aligned} \tag{16}$$

Evaluating the partial derivatives of g as required in Equation (16), we get

$$\frac{\partial^2 g(Y_b, R_b)}{\partial Y_b^2}\Big|_{\substack{Y_b = \theta_1 \\ R_b = \theta_2}} = 0$$

$$\frac{\partial^2 g(Y_b, R_b)}{\partial Y_b \partial R_b}\Big|_{\substack{Y_b = \theta_1 \\ R_b = \theta_2}} = -\frac{1}{\theta_2^2}$$

$$\frac{\partial^2 g(Y_b, R_b)}{\partial R_b^2}\Big|_{\substack{Y_b = \theta_1 \\ R_b = \theta_2}} = 2\frac{\theta_1}{\theta_1^3}$$

Putting these values in Equation (16) and using $\theta_1 = E[Y_b]$ and $\theta_2 = E[R_b]$, we get Equation (14).

The variance can be calculated as follows:

$$\text{Var}\big(g(Y_b, R_b)\big) = E\big[\{g(Y_b, R_b) - E[g(Y_b, R_b)]\}^2\big] \tag{17}$$

Considering that $E[g(Y_b, R_b)]$ is being squared in the expression above, we use first order Taylor series expansion to get the value of $E[g(Y_b, R_b)]$ and substitute it in Equation (17).

$$\begin{aligned} E[g(Y_b, R_b)] &= E\Big[(Y_b - \theta_1)\frac{\partial g}{\partial Y_b} + (R_b - \theta_2)\frac{\partial g}{\partial R_b}\Big] \\ &\quad + g(\theta_1, \theta_2) + O(j^{-1}) \\ &= \Big[(0)\frac{\partial g}{\partial Y_b} + (0)\frac{\partial g}{\partial R_b}\Big] + g(\theta_1, \theta_2) + O(j^{-1}) \approx g(\theta_1, \theta_2) \end{aligned}$$

Substituting the value of $E[g(Y_b, R_b)]$ and using the first order Taylor series expansion of $g(Y_b, R_b)$ in (17), we get

$$\begin{aligned} \text{Var}\big(g(Y_b, R_b)\big) = E\big[\{(Y_b - \theta_1)\frac{\partial g}{\partial Y_b} + (R_b - \theta_2)\frac{\partial g}{\partial R_b}\}^2\big] \\ + O(j^{-1}) \\ \approx \text{Var}(Y_b)(\frac{\partial g}{\partial Y_b})^2 + 2\text{Cov}(Y_b, R_b)\frac{\partial g}{\partial Y_b}\frac{\partial g}{\partial R_b} + \text{Var}(R_b)(\frac{\partial g}{\partial R_b})^2 \end{aligned} \tag{18}$$

Evaluating the partial derivatives of g as required in the equation above, we get

$$\frac{\partial g(Y_b, R_b)}{\partial Y_b}\Big|_{\substack{Y_b = \theta_1 \\ R_b = \theta_2}} = \frac{1}{\theta_2}$$

$$\frac{\partial g(Y_b, R_b)}{\partial R_b}\Big|_{\substack{Y_b = \theta_1 \\ R_b = \theta_2}} = -\frac{\theta_1}{\theta_2^2}$$

Putting these values in Equation (18) and using $\theta_1 = E[Y_b]$ and $\theta_2 = E[R_b]$, we get Equation (15). □

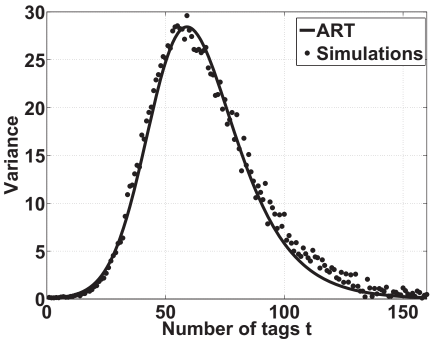


Figure 2: Variances of ART estimator

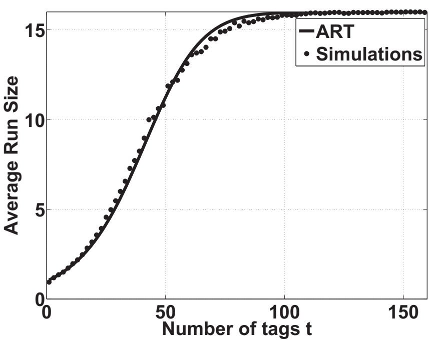


Figure 3: Expectation of ART estimator

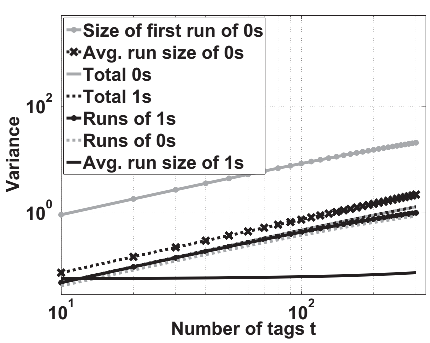


Figure 4: Variance of different estimators vs. t

Figure 2 contains a line plotting the variance of X_1 using Equation (15), where $f = 16$ and $p = 1$. This figure also contains many dots where each dot represents the variance of X_1 that we obtained through 100 times of simulation for each tag population size. This figure shows that for the variance of X_1, simulation results match with the variance calculated from Equation (15). Figure 3 contains a line plotting the expectation of X_1 using Equation (14), where $f = 16$ and $p = 1$, and many dots where each dot represents the average value of X_1 that we obtained through 100 times of simulation for each tag population. This figure shows that the expected value calculated from Equation (14) tracks the simulation results very well.

4.2 Analytical Comparison of Estimators

Although both X_0 and X_1 can be used to estimate the tag population size, we choose X_1 for ART because the tag population size estimation calculated from X_1 has smaller variance compared to X_0 as we show below. It is worth noting that X_0 and X_1 are not equivalent estimators. The average run size of 0s cannot be inferred from the average run size of 1s, and vice versa. For example, 1100011 and 1100110 have the same average run size of 1s, but they have different average run size of 0s. Fundamentally, X_0 and X_1 are not equivalent estimators because for any slot, the probability of it being 0 (which means no tag chooses this slot) and that of it being 1 (which means one or more tags choose this slot) are different.

Next, we show that the ART estimator, namely the average run size of 1s, has less variance than many other framed slotted Aloha based estimators, namely (1) the size of the first run of 0s (used by FNEB [6]), (2) the average run size of 0s, (3) the total number of 0s (used by UPE [7] and EZB [8]), (4) the total number of 1s, (5) the total number of runs of 0s, and (6) the total number of runs of 1s. The higher the variance of an estimator is, the more number of rounds n are needed to improve reliability, and more rounds means more estimation time. Figure 4 shows the analytical plots of the variances of the ART estimator and the above 6 other estimators with frame size $f = 16$ versus tag population sizes. This figure shows that *the variance of ART estimator is significantly lower than all other estimators*. Runs of 1s and runs of 0s have smaller variance compared to ART for very small tag population sizes. This observation, however, is insignificant because both these quantities are non-monotonic functions of tag population size and therefore, cannot be used alone for estimation. The variances of these estimators are calculated as follows. The variance of the total number of 0s and 1s can be calculated using Equation (5). The variance of the size of the first run can be calculated using Equation (3) by setting $i = 1$. The variance of the number of runs of 0s and that of 1s can be calculated using Equation (7). We emphasize that plots in Figure 4 are not based on experimental results, instead, they are based on analytical formulas.

4.3 Unbounded Tag Population Size

For fixed values of required reliability α, frame size f, and persistence probability p, Theorem 3 calculates the upper bound t_M on the number of tags that ART can estimate; that is, for tag population sizes larger than t_M, all the slots in each frame are expected to be 1 and thus ART cannot make an estimate because the tag population size can be infinitely large.

THEOREM 3. *For given required reliability $\alpha \in [0, 1)$, frame size $f > 1$, persistence probability $p > 0$, the maximum number of tags t_M that ART can estimate is:*

$$t_M = \frac{\log\{1 - (1-\alpha)^{\frac{1}{f}}\}}{\log\{1 - \frac{p}{f}\}}$$

PROOF. ART fails if and only if all the slots are 1. To achieve the required reliability α, the failure probability of ART $P\{Y_1 = f\} = q_1^f$ has to be less than $1 - \alpha$. Using the value of q_1 calculated in Lemma 1, we have

$$P\{Y_1 = f\} = q_1^f = \left[1 - \left(1 - \frac{p}{f}\right)^t\right]^f < 1 - \alpha$$

Thus, by taking the log, we have

$$\log\left\{1 - (1-\alpha)^{\frac{1}{f}}\right\} < t\log\left\{1 - \frac{p}{f}\right\}$$

As $\log\left\{1 - \frac{p}{f}\right\} < 0$, dividing both sides by $\log\left\{1 - \frac{p}{f}\right\}$ changes the direction of inequality and results in:

$$t < \frac{\log\{1 - (1-\alpha)^{\frac{1}{f}}\}}{\log\{1 - \frac{p}{f}\}} = t_M \quad \Box$$

From this theorem, we observe that as $p/f \to 0$ we have $t_M \to \infty$. Figure 5 shows the plot of t_M for increasing values of p and 4 different values of f with required reliability $\alpha = 99\%$. In theory, for any given required reliability α, we can increase the estimation scope of ART to any value by decreasing the value of p/f. In practice, however, p/f has

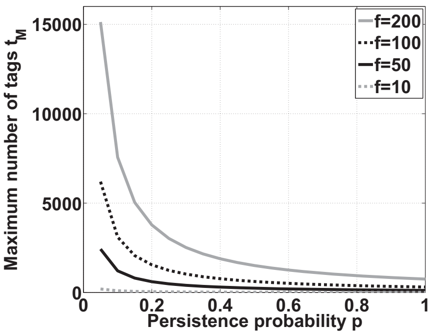


Figure 5: t_M vs. p and f

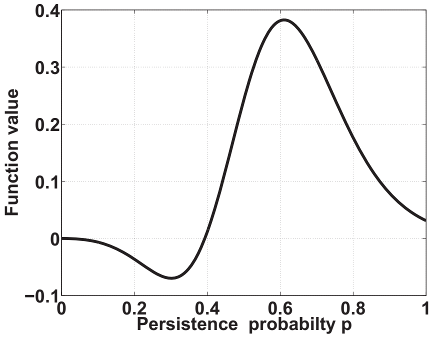


Figure 6: Eq. (19) as func. of p

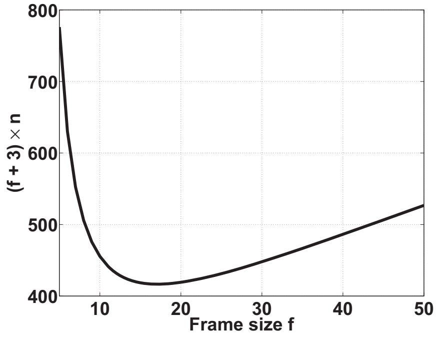


Figure 7: $(f+3)\times n$ vs. f

a minimum value of $1/2^{32}$ due to the 32-bit limitation of the hash functions used by passive tags as specified in the C1G2 standard. Recall that in ART, the reader announces a virtual frame size of f/p (although terminates the frame after the first f slots) and each tag uses the result of a hash function h to select a slot in the range $[1, f/p]$. The number of bits to encode the hash function result is specified to be 32 in the C1G2 standard. Thus, the maximum value of f/p is 2^{32}. Therefore, the maximum number of tags that ART can estimate based on the C1G2 standard is:

$$t_{M_{C1G2}} = \frac{\log\{1-(1-\alpha)^{\frac{1}{f}}\}}{\log\{1-\frac{1}{2^{32}}\}}$$

Here $t_{M_{C1G2}}$ is large enough for all practical applications. For example, with $f = 512$ and $\alpha = 90\%$, $t_{M_{C1G2}}$ is 2.3221e+10, which is over 23 billion; with $f = 512$ and $\alpha = 99\%$, $t_{M_{C1G2}}$ is 2.0254e+10, which is over 20 billion.

5. ART — PARAMETER OPTIMIZATION

To minimize estimation time while achieving required reliability, next, we optimize the three ART parameters: frame size f, persistence probability p, and number of rounds n.

5.1 Optimizing Persistence Probability p

Theorem 4 gives the condition that p needs to satisfy so that the actual reliability is equal to its lower bound, which is the required reliability α. We first prove this theorem and then show how to calculate the optimal value for p.

THEOREM 4. *Given confidence interval β, tag population size t, and frame size f, denoting $E[X_1]$ by $\mu\{t\}$, the optimal persistence probability p_{op} satisfies the following equation.*

$$2\mu\{t\} - \mu\{(1-\beta)t\} - \mu\{(1+\beta)t\} = 0 \tag{19}$$

PROOF. To find the optimal value p_{op} for p, we first find the conditions that p_{op} needs to satisfy so that the actual reliability $P\left\{|\tilde{t}-t| \le \beta t\right\}$ is equal to its lower bound, which is the required reliability α.

$$P\left\{(1-\beta)t \le \tilde{t} \le (1+\beta)t\right\} = \alpha \tag{20}$$

Denoting the observed average value of X_1 from the n frames by $\tilde{X_1}$, $E[X_1]$ by $\mu\{t\}$, and $\mathrm{Var}(X_1)$ by $\sigma^2\{t\}$, we have $\tilde{t} = \mu^{-1}\{\tilde{X_1}\}$. Using $\mu^{-1}\{\tilde{X_1}\}$ to substitute $\tilde{t}$ in (20), we get

$$\begin{aligned} &P\left\{(1-\beta)t \le \mu^{-1}\{\tilde{X_1}\} \le (1+\beta)t\right\} \\ &\quad = P\left\{\mu\{(1-\beta)t\} \le \tilde{X_1} \le \mu\{(1+\beta)t\}\right\} = \alpha \end{aligned} \tag{21}$$

Based on the fact that the variance of a random variable is reduced by n times if the same experiment is repeated n times, by running n rounds and getting n frames, the variance of X_1 becomes $\frac{\sigma^2\{t\}}{n}$ and the standard deviation of X_1 becomes $\frac{\sigma\{t\}}{\sqrt{n}}$. Let Z denote $\frac{\tilde{X_1}-\mu\{t\}}{\sigma\{t\}/\sqrt{n}}$. Thus, (21) becomes

$$P\left\{\frac{\mu\{(1-\beta)t\}-\mu\{t\}}{\frac{\sigma\{t\}}{\sqrt{n}}} \le Z \le \frac{\mu\{(1+\beta)t\}-\mu\{t\}}{\frac{\sigma\{t\}}{\sqrt{n}}}\right\} = \alpha \tag{22}$$

By the central limit theorem, Z approximates a standard normal random variable. The area under the standard normal curve gives the success probability, which is the required reliability in our context. As our confidence interval requirement is symmetric on both the upper and lower sides of the population size, we can represent the required reliability α in terms of a constant k as follows:

$$P\{-k \le Z \le k\} = \alpha \tag{23}$$

From Equations (22) and (23), we get

$$\frac{\mu\{(1-\beta)t\}-\mu\{t\}}{\frac{\sigma\{t\}}{\sqrt{n}}} = -k, \qquad \frac{\mu\{(1+\beta)t\}-\mu\{t\}}{\frac{\sigma\{t\}}{\sqrt{n}}} = k \tag{24}$$

As the absolute values of the right hand size (R.H.S.) of both equations above are k, we get

$$2\mu\{t\} - \mu\{(1-\beta)t\} - \mu\{(1+\beta)t\} = 0 \quad \square \tag{25}$$

Next, we show how to calculate the optimal value for p using Theorem 4, which shows that p_{op} only depends on confidence interval β, tag population size t, and frame size f. For now, we first assume that we know an upper bound on the tag population size denoted by t_m. Later we give a method to obtain t_m automatically. Second, we assume that we know the optimal frame size f_{op}. Later we give a method to calculate f_{op}. Replacing t by t_m and f by f_{op}, left hand side (L.H.S) of Equation (19) in Theorem 4 becomes a well behaved function of p as shown in Figure 6. The numerical solution of this equation gives the optimal value p_{op}.

5.2 Minimizing Number of Rounds n

Using optimal persistence probability p_{op}, the two equations in (24) hold. From them, we get

$$\left(\frac{k\sigma\{t\}}{\mu\{(1+\beta)t\}-\mu\{t\}}\right)^2 = n = \left(\frac{-k\sigma\{t\}}{\mu\{(1-\beta)t\}-\mu\{t\}}\right)^2 \tag{26}$$

Let Φ be the cumulative distribution function of a standard normal distribution and erf$\{.\}$ be the standard error function, we get

$$P\{-k \le Z \le k\} = \Phi(k) - \Phi(-k) = \text{erf}\left\{\frac{k}{\sqrt{2}}\right\} \tag{27}$$

From Equations (23) and (27), we get

$$k = \sqrt{2}\ \text{erf}^{-1}\{\alpha\} \tag{28}$$

From Equations (26) and (28), we get

$$\left(\frac{\sqrt{2}\ \text{erf}^{-1}\{\alpha\}\times\sigma\{t\}}{\mu\{(1+\beta)t\}-\mu\{t\}}\right)^2 = n = \left(\frac{-\sqrt{2}\ \text{erf}^{-1}\{\alpha\}\times\sigma\{t\}}{\mu\{(1-\beta)t\}-\mu\{t\}}\right)^2 \tag{29}$$

Based on this equation, with β, α, $p = p_{op}$, $t = t_m$, and $f = f_{op}$, we can calculate the minimal value n_{op} for n.

5.3 Optimizing Frame Size f

Our goal of optimizing ART parameters, namely p, n, and f, is to minimize the total estimation time, which is $(f+3)\times n_{op}$. Because β, α, and t_m are known and p_{op} is a function of f, n_{op} is essentially a function of f. Thus, $(f+3)\times n_{op}$ is a function of f. Next, we show how to find the optimal frame size f_{op} so that $(f+3)\times n_{op}$ is minimized.

Function $(f+3)\times n_{op}$ is a convex function of f as seen from Figure 7. This means that an optimal frame size f_{op} exists and can be obtained by differentiating $(f+3)\times n_{op}$ with respect to f as shown in the following equation:

$$\frac{d}{df}\{(f+3)\times n_{op}\} = 0 \tag{30}$$

As both expressions for n given in Equation (29) have the same value when $p = p_{op}$, either of them can be used to calculate the value of n_{op}. By substituting n_{op} in Equation (30) by the L.H.S of the n_{op} expression in Equation (29), we get

$$\Big[\mu\{(1+\beta)t_m\} - \mu\{t_m\}\Big]\Big[\sigma\{t_m\} + 2\overline{f}\frac{\partial\sigma\{t_m\}}{\partial f}\Big] \\ -2\overline{f}\sigma\{t_m\}\Big[\frac{\partial\mu\{(1+\beta)t_m\}}{\partial f} - \frac{\partial\mu\{t_m\}}{\partial f}\Big] = 0 \tag{31}$$

where $\frac{\partial\mu\{.\}}{\partial f}$ and $\frac{\partial\sigma\{.\}}{\partial f}$ are obtained through the differentiation of expressions for $E[X_b]$ and $\text{Var}(X_b)$ in Equations (14) and (15), respectively.

To obtain of f_{op} from Equation (31), we need the value of p_{op}, while to obtain the value of p_{op} from Equation (19), we need the value of f_{op}. Therefore, we have two simultaneous equations, (19) and (31), with two unknowns, f_{op} and p_{op}. These equations can be numerically solved simultaneously using Levenberg-Marquardt Algorithm to obtain the values of f_{op} and p_{op}.

Note that the estimation scheme will still work if we do not use $f = f_{op}$, but in that case the value of n will be such that the $(f+3)\times n$ will not be minimum. If we make f arbitrarily small, theoretically, it will still be possible to obtain the estimate accurately; however, in this case, n can be prohibitively large making it practically impossible to obtain the estimate as per the accuracy requirements.

5.4 Observably Constant Estimation Time

There are three inputs to ART: confidence interval β, required reliability α, and a population of t tags where t is unknown. The total estimation time of ART is $(f_{op}+3)\times n_{op}$, which from our experiments we observe to be dependent only on β and α, *i.e.*, $(f_{op}+3)\times n_{op}$ is constant with respect to (w.r.t.) t. Because $(f_{op}+3)\times n_{op}$ is highly complex due to complex components such as $\text{Cov}(Y_b, R_b)$ as expressed in Equation (9), we have not yet formally proven that $(f_{op}+3)\times n_{op}$ is independent of t, although we hypothesize that this is mathematically true. Intuitively, the larger t is, the smaller p_{op} is. Although t plays an important role in computing p_{op}, n_{op}, and f_{op} individually, in formula $(f_{op}+3)\times n_{op}$ the impact of t gets canceled out. From Equations (29) and (31), we observe that the value of n_{op} depends upon α, β, μ, σ and value of f_{op} depends upon β, μ, σ. Here $(f_{op}+3)\times n_{op}$ is independent of t because α and β are given values and μ and σ are functions of q_1, if q_1 is independent of t. Indeed q_1 is intuitively independent of t because $q_1 \propto 1/t$ and p_{op} obtained from Equation (19) decreases as the value of t_m increases, as shown in Figure 8. A constant value of q_1 means that the probability of any slot in a frame being non-empty is the same, which in turn implies that average run size in a frame will also be the same regardless of t. Figure 9 plots the value of q_1 w.r.t. t using Equations (19) and (1). We observe that the value of q_1 stays perfectly constant w.r.t. t. Figure 10 shows that the total estimation time of ART stays constant w.r.t. t for given β and α.

5.5 Obtaining Population Upper Bound t_m

So far we have assumed the knowledge of an upper bound t_m on tag population size t. In some applications, t_m is readily available. For example, the number of TVs in a TV warehouse can be reasonably estimated by the warehouse size and minimum TV size. However, this may not be available for some applications where the tag population size changes in large magnitude. We next present a fast scheme to obtain t_m based on Flajolet and Martin's probabilistic counting algorithm used in databases [5]. Before running ART, the reader uses this scheme to obtain t_m. In this scheme, the reader keeps issuing single-slot frames, where the persistence probability p follows a geometric distribution starting from $p = 1$ (*i.e.*, $p = \frac{1}{2^{i-1}}$ in the i-th frame), until the reader gets an empty slot. Suppose the empty slot occurred in the i-th frame, then $t_m = 1.2897 \times 2^{i-2}$ is an upper bound on t [5,14].

5.6 ART with Multiple Readers

We next discuss how to obtain t_m and $\tilde{t}$ using multiple readers whose covered regions may overlap. To obtain t_m using multiple readers, we can let each reader obtain the t_m value on its own and then sum them up as the final overall t_m because of two reasons. First, our requirement on t_m is only a rough upper bound estimate. Second, deployment of multiple readers in practice often requires site surveys to ensure minimal overlapping between readers. To use multiple readers to obtain $\tilde{t}$ more precisely, we adapt the approach

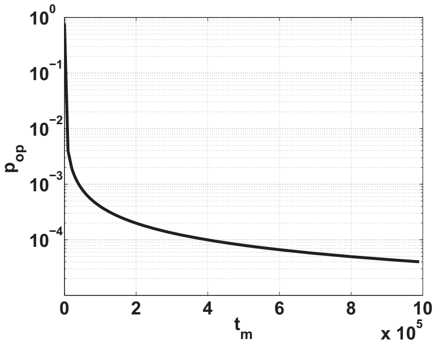


Figure 8: p_{op} vs. t_m

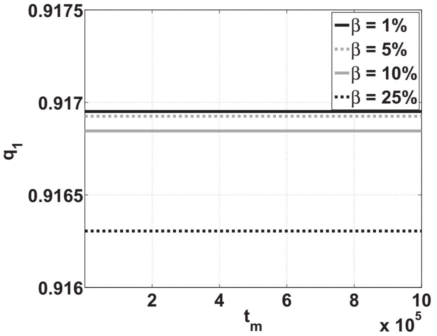


Figure 9: q_1 vs. t_m

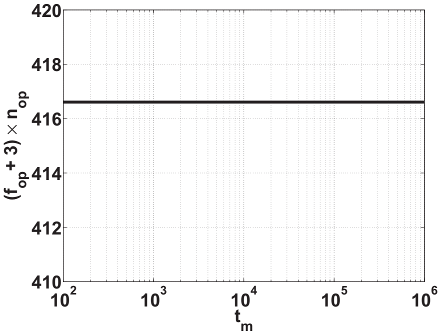


Figure 10: $(f_{op}+3)\times n_{op}$vs.t_m

proposed by Kodialam *et al.* in [8], which uses a central controller for all readers. ART parameters β, α, t_m, f_{op}, p_{op}, and n_{op} have the same value across all readers. When a reader transmits seed R_i in its i-th frame, it does not generate R_i on its own, rather it uses the i-th seed R_i issued by the central controller. That is, each reader generates the same sequence of n_{op} seeds. In the i-th frames from different readers, because all readers use the same seed R_i, the slot number that a given tag chooses is the same (*i.e.*, $h(f, R_i, ID)$) in the frame of each reader covering this tag. Once a reader has completed its frame, it sends the frame to the central controller. The controller applies the logical OR on all the i-th frames from all readers, and gets the same i-th frame as if using a single reader. ART uses the n_{op} frames computed by logical OR to estimate the population size.

5.7 ART - Pseudocode

Algorithm 1 shows the pseudocode of ART.

Algorithm 1: EstimateRFIDTagPopulation(α, β, n_r)

```
Input: (1) Required reliability $\alpha$
       (2) Required confidence interval $\beta$
       (3) Number of readers $n_r$
Output: Estimated tag population size $\tilde{t}$
$t_m$ := GetMaximumTags($n_r$)
Solve (19) and (31) to get $f_{op}$ and $p_{op}$.
Evaluate $k$ using (28).
Evaluate $n_{op}$ by using $\alpha, \beta, k, f_{op}, p_{op}$, and $t_m$ in (29).
for $i := 1$ to $n_{op}$ do
    Provide all readers with $f_{op}/p_{op}$ and a random seed $R_i$.
    Run Aloha on readers and gather all readers' frames.
    Perform slot wise OR on all frames to obtain one frame.
    Obtain $\tilde{X}_1(i)$, the average run size of 1s in this frame.
$\tilde{X}_{avg} \leftarrow \sum_{i=1}^{n_{op}} \tilde{X}_1(i)/n_{op}$
Use $E[X_1] := \tilde{X}_{avg}$ and solve (14) to obtain an estimate $\tilde{t}$ of $t$.
return $\tilde{t}$
GetMaximumTags($n_r$)
$f := 1$
for $j := 1$ to $n_r$ do
    $i := 1$
    $p_i := 1$
    repeat
        Provide reader $j$ with $f/p_i$ and a random seed $R_i$.
        Run Aloha on reader $j$ and get the response.
        if slot is not empty then
            $p_{i+1} := p_i/2$
            $i := i + 1$
    until slot is empty
    $t_{m,j} := 1.2897 \times 2^{i-2}$
$t_m := \sum_{j=1}^{n_r} t_{m,j}$
return $t_m$
```

6. PERFORMANCE EVALUATION

We numerically evaluated in Matlab our ART scheme as well as four prior RFID estimation schemes: UPE [7], EZB [8], FNEB [6], and MLE [10]. We did not evaluate the other estimation scheme LoF [14] because it is non-compliant with C1G2. In terms of implementation complexity, the number of lines of code required to implement ART were almost the same as all four prior protocols. To ensure compliance with the C1G2 standard, in all our simulations, each tag picks up exactly one slot at the start of frame as soon as the reader broadcasts the frame size.

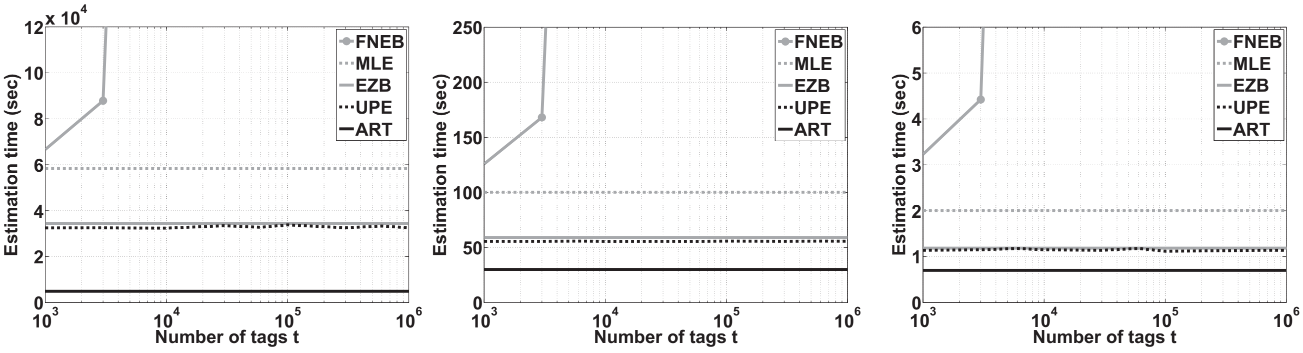
Next, we first conduct a side-by-side comparison on estimation time between ART and the four prior schemes. Then, we conduct experiments to show that ART indeed achieves the required reliability.

6.1 Estimation Time

The results in Figures 11, 12, and 13 show that *the estimation time of ART is significantly smaller than all prior schemes*. Note that in Figures 12 and 13, the plots for FNEB are out of the range of the vertical axes, and the plots of UPE and EZB are almost overlapping.

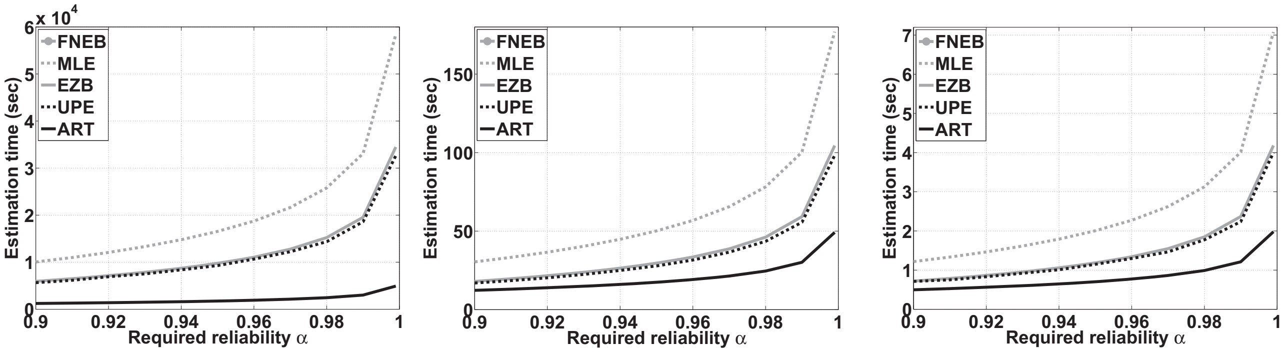
We make three main observations from Figures 11 (a), (b), and (c), which show the estimation time needed by each scheme with population sizes of up to one million tags for different configurations of α and confidence interval β. First, we observe that ART is faster than all four prior schemes in all these configurations. For $\alpha = 99.9\%$ and $\beta = 0.1\%$, ART is 7 times faster than the fastest prior estimation schemes, which are UPE [7] and EZB [8]. For $\alpha = 99\%$ and $\beta = 1\%$, ART is 1.96 times faster than UPE and EZB. For $\alpha = 95\%$ and $\beta = 5\%$, ART is 1.68 times faster than UPE and EZB. Second, we observe that ART, UPE, EZB, and MLE perform estimation in constant time, which attributes to the use of persistence probabilities. Third, we observe that FNEB, whose estimator is the size of the first run of 0s, is the slowest. This concurs with our analytical analysis in Figure 4, where we show that FNEB has the largest variance. The larger the variance of an estimator, the more the rounds of execution needed to achieve the required reliability, and therefore the longer the estimation time.

We make three main observations from Figures 12 (a), (b), and (c), which show the estimation time for 5000 tags required by each scheme with the required reliability α varying from 90% to 99.9% for different configurations of confidence interval β. First, we observe that ART is faster than all four prior estimation schemes in all these configurations. Second, the difference between the estimation time of ART



(a) $\alpha = 99.9\%, \beta = 0.1\%$ (b) $\alpha = 99\%, \beta = 1\%$ (c) $\alpha = 95\%, \beta = 5\%$

Figure 11: Time vs. t



(a) $\beta = 0.1\%$ (b) $\beta = 1\%$ (c) $\beta = 5\%$

Figure 12: Time vs. α

and those of prior schemes increases as the required reliability increases. For example, for $\beta = 5\%$ and $\alpha = 95\%$, ART is 1.68 times faster than UPE and EZB while for $\beta = 0.1\%$ and $\alpha = 99.9\%$, it is 7 times faster. This shows that ART becomes more and more advantageous over existing schemes when the required reliability increases. Third, for all schemes, the estimation time increases as the required reliability increases because more number of rounds are needed to achieve the required reliability. We further observe that ART's estimation time increases at the lowest rate as the required reliability increases because its estimator has the smallest variance.

We make three main observations from Figures 13 (a), (b), and (c), which show the estimation time for 5000 tags required by each scheme with the confidence interval β varying from 0.1% to 10% for different configurations of α. First, we observe that ART is faster than all estimation schemes in all these configurations. Second, for all schemes, the estimation time decreases as the confidence interval increases because lesser number of rounds are needed to achieve the required reliability.

6.2 Actual Reliability

The subfigures in Figure 14 show the actual reliability of ART versus the number of tags for different configurations of required reliability α and confidence interval β. We observe that *ART always achieves the required reliability.*

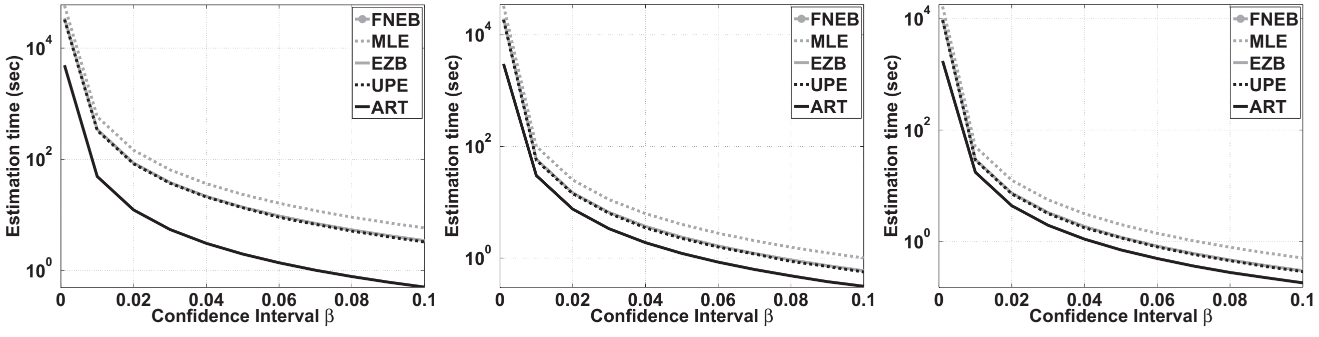
7. CONCLUSIONS AND FUTURE WORK

The key technical novelty of this paper is in proposing the new estimator, the average run size of 1s, for estimating RFID tag population size. Using analytical plots, we show that our estimator has much smaller variance compared to other estimators including those used in prior work. It is this smaller variance that makes our scheme faster than the previous ones. In future work, we will work on mathematically proving that our estimator has smaller variance compared to other estimators. The key technical depth of this paper is in the mathematical development of the estimation theory using this estimator. Our experimental results show that our scheme ART is significantly faster than all prior RFID estimation schemes. Furthermore, both the analytical and experimental results show that the estimation of ART is independent of tag population size. In future work, we will work on mathematically proving this independence.

8. REFERENCES

[1] C. Bordenave, D. McDonald, and A. Proutiere. Performance of random medium access control, an asymptotic approach. In *Proc. of Int. Conf. on Measurements and Modeling of Computer Systems SIGMETRICS*, 2008.

[2] J.-R. Cha and I. Jae-Hyun Kim. Novel anti-collision algorithms for fast object identification in rfid system. In *Proc. of Int. Conf. on Parallel and Distributed Systems*, 2005.

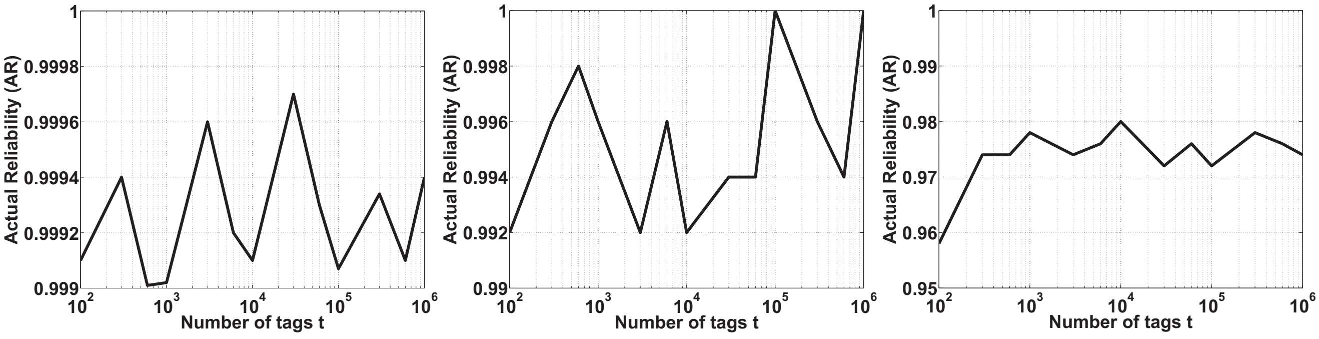


(a) $\alpha = 99.9\%$

(b) $\alpha = 99\%$

(c) $\alpha = 95\%$

Figure 13: Time vs. β



(a) $\alpha = 99.9\%, \beta = 0.1\%$

(b) $\alpha = 99\%, \beta = 1\%$

(c) $\alpha = 95\%, \beta = 5\%$

Figure 14: Actual reliability

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

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

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

[6] 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.

[7] M. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in RFID systems. In *Proc. 12th MobiCom*, pages 322–333, 2006.

[8] M. Kodialam, T. Nandagopal, and W. C. Lau. Anonymous tracking using RFID tags. In *Proc. IEEE INFOCOM*, 2007.

[9] C. H. Lee and C. W. Chung. Efficient storage scheme and query processing for supply chain management using RFID. In *Proc. ACM SIGMOD*, pages 291–302, 2008.

[10] T. Li, S. Wu, S. Chen, and M. Yang. Energy efficient algorithms for the RFID estimation problem. In *Proc. IEEE INFOCOM*, 2010.

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

[12] A. Nemmaluri, M. D. Corner, and P. Shenoy. Sherlock: Automatically locating objects for humans. In *Proc. MobiSys*, pages 187–198, 2008.

[13] 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.

[14] C. Qian, H. Ngan, and Y. Liu. Cardinality estimation for large-scale RFID systems. In *Proc. 6th IEEE PerCom*, pages 30–39, 2008.

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

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

[17] V. Shah-Mansouri and V. W. Wong. Cardinality estimation in RFID systems with multiple readers. In *Proc. IEEE GLOBECOM*, 2009.

[18] H. Vogt. Efficient object identification with passive RFID tags. *Pervasive Computing*, 2414:98–113, 2002.

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

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