# Fast and Accurate Estimation of RFID Tags

Muhammad Shahzad and Alex X. Liu

Abstract—Radio frequency identification (RFID) systems have been widely deployed for various applications such as object tracking, 3-D 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. 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 provably independent of the tag population sizes. ART works with multiple readers with overlapping regions and can estimate sizes of arbitrarily large tag populations. 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.

Index Terms—Estimation, radio frequency identification (RFID), tags.

#### I. INTRODUCTION

#### A. Motivation and Problem Statement

ADIO frequency identification (RFID) systems are widely used in various and widely used in various applications such as object tracking [12], 3-D positioning [21], 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 [16]). 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 ft; 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.

Manuscript received March 29, 2013; revised November 12, 2013; accepted December 03, 2013; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor G. Bianchi. Date of publication January 23, 2014; date of current version February 12, 2015. The preliminary version of this paper, titled "Every Bit Counts—Fast and Scalable RFID Estimation," was published in the Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom), Istanbul, Turkey, August 22–26, 2012 (pp. 365–376). (Corresponding author: A. X. Liu)

The authors are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824, USA (e-mail: shahzadm@cse.msu.edu; alexliu@cse.msu.edu).

Digital Object Identifier 10.1109/TNET.2014.2298039

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 [19]. 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. 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 a quick estimate 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 follows: 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  $P\{|\tilde{t}-t| \leq \beta t\} \geq \alpha$ . When the number of readers is one, we call this problem single-reader estimation; otherwise, we call this problem multireader 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  $\alpha$  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.

## B. Proposed Approach

In this paper, we propose a new scheme called *Average Runbased 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.

## C. 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 VII-C. An estimator with small variance is faster because the Aloha frames need to be repeated fewer times to achieve the required reliability. Furthermore, the estimation time of ART is provably 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 [15] 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.

Paper Organization: Section II describes the prominent related work in RFID estimation. Section III gives an overview of how ART works. It also describes the assumptions in the formal development of ART and their implications. Section IV gives the detailed formal development of ART. Section V derives mathematical expressions to calculate the optimal values of system parameters, which ensure that ART achieves the required reliability in smallest possible time. Section VI describes how ART estimates sizes of arbitrarily large tag populations and how it handles environments with multiple RFID readers. Section VII gives the proof that estimation time of ART is independent of tag population size and discusses its computational complexity. It also presents analytical comparison of ART to existing schemes to mathematically justify the faster speed of ART compared to existing schemes. Section VIII presents results from our experimental evaluation of ART along with side-by-side comparisons to existing schemes. Finally, we conclude the paper in Section IX.

#### II. 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 nonempty 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 [15], 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 noncompliant 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 assumption that frame size can be arbitrarily large, which does not hold in practice. 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 multireader tag estimation scheme that is based on an unrealistic assumption that any tag covered by multiple readers only replies to one reader. In [22], Zanella proposed Collision Set Estimator (CSE) that utilizes maximum likelihood estimation to estimate the number of tags in a population. CSE does not take accuracy requirements ( $\alpha$  and  $\beta$ ) as input and, therefore, cannot achieve any arbitrary required reliability.

## III. ART—SCHEME OVERVIEW

## A. 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 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*.

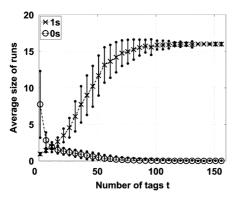


Fig. 1. Average run size of 0s and 1s versus tag population size t. (f = 16).

#### B. 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, 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 size increases, the average run size of 1s increases (and that of 0s decreases). We illustrate this intuition using the simulation results in Fig. 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.

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 populations of large sizes.

## C. 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 in minimum estimation time, we optimize f, p, and the number of rounds n so that the total number of slots  $(f+l) \times n$  is minimized while satisfying

 $P\{|\tilde{t}-t| \leq \beta t\} \geq \alpha$ . Here, l is a constant that represents the C1G2 specified mandatory time delay in terms of number of empty slots between the end of a frame and the start of next frame. Typically, this delay is about 1 ms (i.e.,  $l \approx 3.33$  empty slots) [3], [14].

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 [20]. 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], [15], [17], [18], [20], [23].

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. However, 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. In all the simulations we have presented in this paper, a tag chooses exactly one slot at the start of frame.

#### IV. ART—ESTIMATION ALGORITHM

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

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., 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 substituted for 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

$$q_{b} = \begin{cases} \left(1 - \frac{p}{f}\right)^{t}, & \text{if } b = 0\\ 1 - \left(1 - \frac{p}{f}\right)^{t}, & \text{if } b = 1. \end{cases}$$
 (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

TABLE I SYMBOLS USED IN THE PAPER

$t$ actual tag population size $t_m$ upper bound on # of tags	
$t_m$ upper bound on # of tags	
$L_{tm}, U_{tm}$ bounds within which $t_m/t$ should lie	
$t_M$ maximum # of tags that can be estimated	
$\tilde{t}$ estimated # of tags	
$\alpha$ required reliability	
$\tilde{\alpha}$ expected value of actual reliability	
$\beta$ required confidence interval	
f frame size	
$f_{\rm op}$ optimal frame size	
n # of rounds (i.e., frames)	
p persistence probability	
$\begin{array}{c c} p & \text{persistence probability} \\ \hline R & \text{random seed from reader} \end{array}$	
h(f, R, ID) unform hash function with output in $[1, f]$	
b value of a slot: $b = 0$ or $b = 1$	
$\overline{b}$ 1- $b$	
$q_b$ probability that a slot is $b$	
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	
$q_1, q_1^+, q_1^-$ probability that a slot is 1 when number of tags in	
	y
$\mu\{.\}$ expected value of $X_b$ $\sigma\{.\}$ standard deviation of $X_b$	
number of ways in which $y$ occurrences of $b$	
$\xi\{f,y,r\}$ and $f-y$ occurrences of $\bar{b}$ can be arranged	
in $f$ slots while ensuring that the number of	
runs of $b$ are $r$	
$\eta$ simplification variable: $\eta = (1 - \frac{p}{f})^t$	
z # of bits the reader uses in the mask	$\neg$
constant representing delay between consecutive	
frames in terms of number of empty slots	

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  $X_b$  be the random variable representing the average run size of b in a frame. Next, we calculate the expectation and variance of  $X_b$ . 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 the values of p, n, and f that will ensure that the actual reliability is greater than the required reliability and the estimation time is minimum. 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  $E[Y_b]$ ,  $Var(Y_b)$ ,  $E[R_b]$ ,  $Var(R_b)$ , and  $Cov(Y_b, R_b)$  in Lemmas 2 and 3. Then, we use them to calculate  $E[X_b]$  and  $Var(X_b)$  in Theorem 1. Using (12) in Theorem 1, replacing  $E[X_b]$  by the observed average run size of b from n frames, we obtain an equation with only one unknown t. Finally, we use Brent's method to obtain the numerical solution of this equation. The result is the estimated tag population size  $\tilde{t}$ . 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

$$E[Y_b] = fq_b \tag{2}$$

$$Var(Y_b) = fq_b(1 - q_b) \tag{3}$$

$$E[R_b] = q_b(q_b + f(1 - q_b)) \tag{4}$$

$$Var(R_b) = f(q_b - 4q_b^2 + 6q_b^3 - 3q_b^4) + (3q_b^2 - 8q_b^3 + 5q_b^4).$$
 (5)

*Proof:* Each slot i of frame f has probability  $q_b$  of being b. Therefore,  $Y_b \sim \operatorname{Binom}(f, q_b)$ . Using general formula for expectation and variance of a binomial random variable,  $E[Y_b]$  and  $\operatorname{Var}(Y_b)$  are given by (2) and (3).

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  $\gamma_i$  are identically distributed. Furthermore,  $P\{\gamma_i = b\} = q_b$ . Let  $\bar{b} = 1 - b$  and let  $I_i$  be the indicator random variable whose value is 1 if a run of b begins at  $\gamma_i$ 

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

Thus

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

Because

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

we get

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

As  $R_b$  is sum of f random variables, some of which are correlated, we use the general expression for variance of sum of correlated random variables to obtain the variance of  $R_b$ 

$$\operatorname{Var}\left(R_{b}\right) = \operatorname{Var}\left(\sum_{i=1}^{f} I_{i}\right) = \sum_{i=1}^{f} \operatorname{Var}(I_{i}) + 2\sum_{j=2}^{f} \sum_{\forall i < j} \operatorname{Cov}(I_{i}, I_{j})$$

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  $I_i \sim \text{Bernoulli}(q_b)$ , its variance is that of a Bernoulli random variable given by

$$Var(I_i) = E[I_i](1 - E[I_i]).$$
 (6)

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

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

$$\operatorname{Var}(R_b) = \operatorname{Var}(I_1) + \sum_{j=2}^{f} \operatorname{Var}(I_j) + 2\operatorname{Cov}(I_1, I_2)$$

$$+ 2\sum_{j=3}^{f} \operatorname{Cov}(I_{j-1}, I_j)$$

$$= q_b(1 - q_b) + (f - 1)q_b(1 - q_b) \left\{ 1 - q_b(1 - q_b) \right\}$$

$$- 2q_b^2(1 - q_b) - 2(f - 2)q_b^2(1 - q_b)^2$$

$$= f\left(q_b - 4q_b^2 + 6q_b^3 - 3q_b^4\right) + \left(3q_b^2 - 8q_b^3 + 5q_b^4\right).$$

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

$$Cov (Y_b, R_b) = \sum_{y=0}^{f} \sum_{r=0}^{\lceil \frac{f}{2} \rceil} yr q_b^y (1 - q_b)^{f-y} . \xi \{f, y, r\} - E[Y_b] E[R_b]$$
(7)

where we have the formula shown at the bottom of the page. *Proof:* By definition, we have

Cov 
$$(Y_b, R_b) = \sum_{y=0}^{f} \sum_{r=0}^{f} yr P\{Y_b = y, R_b = r\} - E[Y_b]E[R_b].$$

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

$$P\{Y_b = y, R_b = r\} = P\{R_b = r | Y_b = y\} \times P\{Y_b = y\}.$$
(9)

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

$$P\{Y_b = y\} = \binom{f}{y} q_b^y (1 - q_b)^{f - y}.$$
 (10)

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 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  $\overline{b}$  such that between consecutive partitions of b, the partitions of  $\overline{b}$  can be *interleaved*. For r partitions of b, there are four possible partitions of  $\overline{b}$ .

- 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  $\overline{b}$  and ends with b, implying that there are r partitions of  $\overline{b}$ .
- 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  $\overline{b}$  in  $\binom{f-y-1}{r-1}$  ways. Similarly, we can make r+1 partitions of f-y occurrences of  $\overline{b}$  in  $\binom{f-y-1}{r-1}$  ways and r-1 partitions of f-y occurrences of  $\overline{b}$  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

$$P\{R_b = r | Y_b = y\} = \frac{\xi\{f, y, r\}}{\binom{f}{y}}.$$
 (11)

Substituting values from (10) and (11) in (9) and (8) results in (7).  $\Box$ 

Theorem 1: Given tag population size t, frame size f, and persistence probability p, we have

$$E[X_b] = \frac{E[Y_b]}{E[R_b]} - \frac{\text{Cov}(Y_b, R_b)}{E^2[R_b]} + \frac{E[Y_b]}{E^3[R_b]} \text{Var}(R_b)$$
(12)

$$Var (X_b) = \frac{Var (Y_b)}{E^2[R_b]} - \frac{2E[Y_b]}{E^3[R_b]} Cov(Y_b, R_b) + \frac{E^2[Y_b]}{E^4[R_b]} Var (R_b)$$
(13)

*Proof:* See Appendix-A for proof.  $\Box$ 

Figs. 2 and 3 show the expectation and variance of  $X_1$  calculated using (12) and (13), respectively, with f=16 and p=1. The dots in these figures represent the corresponding values obtained through 100 repetitions of simulation for each tag population size. These figures show that the values given by (12) and (13) track the simulation results very well, which serves as an experimental proof that the assumption "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" practically holds.

$$\xi\{f,y,r\} = \begin{cases} \begin{pmatrix} y-1 \\ r-1 \end{pmatrix} \left[ \begin{pmatrix} f-y-1 \\ r-2 \end{pmatrix} + 2 \begin{pmatrix} f-y-1 \\ r-1 \end{pmatrix} + \begin{pmatrix} f-y-1 \\ r \end{pmatrix} \right], & \text{if } r > 1 \land 0 < y < f \land r \leq y \land r \leq f-y-1 \\ \begin{pmatrix} y-1 \\ r-1 \end{pmatrix} \left[ 2 \begin{pmatrix} f-y-1 \\ r-1 \end{pmatrix} + \begin{pmatrix} f-y-1 \\ r \end{pmatrix} \right], & \text{if } r = 1 \land 0 < y < f \land r \leq y \land r \leq f-y-1 \\ 1, & \text{if } r = 1 \land y = f \\ 1, & \text{if } r = 0 \land y = 0 \\ 0, & \text{otherwise.} \end{cases}$$

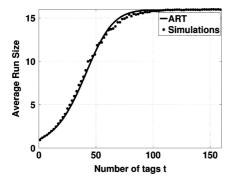


Fig. 2. Expectation of ART estimator.

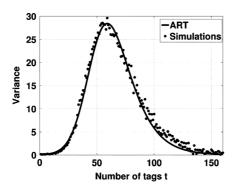


Fig. 3. Variances of ART estimator.

#### V. ART—PARAMETER TUNING

To minimize estimation time while achieving required reliability, next we obtain values of persistence probability p, number of rounds n, and frame size f. As we have three unknowns, we require three equations that can be solved simultaneously. We derive these three equations using following three conditions: 1) the confidence interval should be symmetric around t, i.e.,  $|\tilde{t}-t| \leq \beta t$ ; 2) actual reliability is greater than or equal to the required reliability, i.e.,  $P\{|\tilde{t}-t| \leq \beta t\} \geq \alpha$ ; and 3) estimation time is minimized. We use the first condition to calculate p, the second condition to calculate n, and the last condition to calculate f.

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 in Section VII-C. 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 and that of it being 1 are different.

#### A. Persistence Probability P

We express confidence interval requirement  $|\tilde{t} - t| < \beta t$  as

$$(1-\beta)t < \tilde{t} < (1+\beta)t. \tag{14}$$

Recall from Lemma 1 that we use  $q_1$  to denote the probability that a slot in a frame is 1 when the number of tags in the population are t and the persistence probability is p. Let  $q_1^+$  and  $q_1^-$  denote the probabilities that a slot in a frame is 1 when the

number of tags in the population are  $(1 + \beta)t$  and  $(1 - \beta)t$ , respectively, and the persistence probability is p. Let  $\tilde{q}_1$  represent the estimate of  $q_1$ . Therefore, we have

$$q_{1}^{+} = 1 - \left(1 - \frac{p}{f}\right)^{(1+\beta)t} \Rightarrow -(1+\beta)t = \frac{\ln\left\{1 - q_{1}^{+}\right\}}{\ln\left\{1 - \frac{p}{f}\right\}}$$

$$q_{1}^{-} = 1 - \left(1 - \frac{p}{f}\right)^{(1-\beta)t} \Rightarrow (1-\beta)t = \frac{\ln\left\{1 - q_{1}^{+}\right\}}{\ln\left\{1 - \frac{p}{f}\right\}}$$
(16)

$$\tilde{q}_1 = 1 - \left(1 - \frac{p}{f}\right)^{\tilde{t}} \Rightarrow \tilde{t} = \frac{\ln\left\{1 - \tilde{q}_1\right\}}{\ln\left\{1 - \frac{p}{f}\right\}}.$$
(17)

Substituting values of  $(1 + \beta)t$ ,  $(1 - \beta)t$ , and  $\tilde{t}$  from (15), (16), and (17), respectively, into (14), we get

$$\frac{\ln\left\{1 - q_1^-\right\}}{\ln\left\{1 - \frac{p}{f}\right\}} \le \frac{\ln\{1 - \tilde{q}_1\}}{\ln\left\{1 - \frac{p}{f}\right\}} \le \frac{\ln\left\{1 - q_1^+\right\}}{\ln\left\{1 - \frac{p}{f}\right\}}.$$

As  $\ln\{1-\frac{p}{f}\}<0$ , thus

$$\ln\{1 - q_1^+\} \le \ln\{1 - \tilde{q_1}\} \le \ln\{1 - q_1^-\}.$$

Exponentiating and rearranging, the confidence interval requirement becomes

$$q_1^- \le \tilde{q_1} \le q_1^+$$
.

As  $E[X_1]$  and  ${\rm Var}\,(X_1)$  are functions of  $q_1$ , denoting  $E[X_1]$  by  $\mu\{q_1\}$ ,  ${\rm Var}\,(X_1)$  by  $\sigma^2\{q_1\}$ , and the observed average value of  $X_1$  from the n frames by  $\tilde{X}_1$ , we have  $\tilde{q}_1=\mu^{-1}\{\tilde{X}_1\}$ . Using  $\mu^{-1}\{\tilde{X}_1\}$  to substitute  $\tilde{q}_1$  in the expression above, we get

$$q_1^- \leq \mu^{-1}\{\tilde{X_1}\} \leq q_1^+ \Rightarrow \mu\{q_1^-\} \leq \tilde{X_1} \leq \mu\{q_1^+\}.$$

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  $\frac{\sigma^2\{q_1\}}{n}$ , and the standard deviation of  $X_1$  becomes  $\frac{\sigma\{q_1\}}{\sqrt{n}}$ .

Let Z denote  $\frac{\tilde{X_1} - \mu\{q_1\}}{\sigma\{q_1\}/\sqrt{n}}$ . Thus, the expression above becomes

$$\frac{\mu\{q_1^-\} - \mu\{q_1\}}{\frac{\sigma\{q_1\}}{\sqrt{n}}} \le Z \le \frac{\mu\{q_1^+\} - \mu\{q_1\}}{\frac{\sigma\{q_1\}}{\sqrt{n}}}.$$
 (18)

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. For the confidence interval to be symmetric on both the upper and lower sides of the population size as per the first of the three conditions, the absolute value of the upper and lower limits of Z should be equal. Let k represent the absolute value of these upper and lower limits. Thus, we can represent Z as follows:

$$-k \le Z \le k. \tag{19}$$

From (18) and (19), we get

$$\frac{\mu\left\{q_{1}^{-}\right\} - \mu\left\{q_{1}\right\}}{\frac{\sigma\left\{q_{1}\right\}}{\sqrt{n}}} = -k, \ \frac{\mu\left\{q_{1}^{+}\right\} - \mu\left\{q_{1}\right\}}{\frac{\sigma\left\{q_{1}\right\}}{\sqrt{n}}} = k. \tag{20}$$

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

$$2\mu\{q_1\} - \mu\{q_1^+\} - \mu\{q_1^-\} = 0. \tag{21}$$

The equation above gives the condition that needs to be satisfied to make the confidence interval symmetric around the tag population size. Fig. 4 plots the value of left-hand side (L.H.S.) of this equation as a function of p for three different values of  $\beta$ . We can see that it is a well-behaved function of p, and thus there exists a unique value of p that makes it equal to zero. Furthermore, we also observe that all the curves cross the zero line at the same point, which gives us a hint that the solution to the equation above is independent of  $\beta$ . The solution is given by the following equation:

$$p = f \left\{ 1 - \left(\frac{1}{f - 1}\right)^{\frac{1}{t}} \right\}. \tag{22}$$

The derivation of (22) is given in Appendix-B. Note that this equation does not involve  $\beta$ , which shows that indeed the solution to (21) is independent of  $\beta$  as we had intuitively inferred from Fig. 4.

Equation (22) is first of the three equations that we will solve simultaneously. This equation requires the value of actual tag population size t, which we do not know. Fortunately, we can calculate an upper bound,  $t_m$ , on the actual tag population size and use that in (22) instead of t. We will describe a method to obtain  $t_m$  in Section V-D and also determine how close  $t_m$  has to be to t to ensure that ART achieves the required reliability.

#### B. Number of Rounds n

Using the persistence probability calculated in (22), the two equations in (20) hold. From them, we get

$$\left(\frac{k\sigma\{q_1\}}{\mu\{q_1^+\} - \mu\{q_1\}}\right)^2 = n = \left(\frac{-k\sigma\{q_1\}}{\mu\{q_1^-\} - \mu\{q_1\}}\right)^2. (23)$$

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

$$P\{-k \le Z \le k\} = \Phi(k) - \Phi(-k) = \operatorname{erf}\left\{\frac{k}{\sqrt{2}}\right\}. \tag{24}$$

 $P\{-k \le Z \le k\}$  gives the success probability in terms of the area under the standard normal curve between -k and +k. As per the second of the three conditions, this area should be at least equal to the required reliability  $\alpha$ , i.e.,

$$P\{-k \le Z \le k\} = \alpha. \tag{25}$$

From (24) and (25), we get

$$k = \sqrt{2}\operatorname{erf}^{-1}\{\alpha\}. \tag{26}$$

From (23) and (26), we get

$$\left(\frac{\sqrt{2}\operatorname{erf}^{-1}\{\alpha\} \times \sigma\{q_1\}}{\mu\{q_1^+\} - \mu\{q_1\}}\right)^2 = n$$

$$= \left(\frac{-\sqrt{2}\operatorname{erf}^{-1}\{\alpha\} \times \sigma\{q_1\}}{\mu\{q_1^-\} - \mu\{q_1\}}\right)^2.$$
(27)

Equation (27) is second of the three equations that we will solve simultaneously.

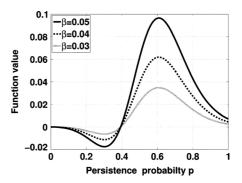


Fig. 4. Equation (21) as a function of p.

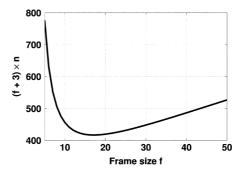


Fig. 5. Total estimation time versus frame size.

#### C. Optimal Frame Size f

As per the third of the three conditions, total estimation time should be minimum. The total estimation time is directly proportional to total number of slots,  $(f+l) \times n$ , which is a convex function of f as seen from Fig. 5. This means that an optimal frame size  $f_{\rm op}$  exists and can be obtained by differentiating  $(f+l) \times n$  with respect to f as follows:

$$\frac{d}{df}\{(f+l)\times n\} = 0. \tag{28}$$

Equation (28) is third of the three equations that we will solve simultaneously.

Required reliability  $\alpha$  and confidence interval  $\beta$  are given constants, and  $t_m$  is calculated using the method proposed in Section V-D. Thus,  $p, q_1, q_1^+$ , and  $q_1^-$  are all functions of f. Consequently, n is a function of f, and therefore  $(f+l) \times n$  is also a function of f with only one unknown, i.e., f. The numerical solution of (28) gives the optimal value of frame size, represented by  $f_{\text{op}}$ .

To numerically solve (28), we substitute the value of n from (27) in (28). As both expressions for n given in (27) have the same values when p is calculated using (22), either of them can be used to calculate n. Substituting n in (28) by the L.H.S. of the expression for n in (27), we get

$$\left[\mu\{q_1^+\} - \mu\{q_1\}\right] \left[\sigma\{q_1\} + 2(f+l)\frac{\partial\sigma\{q_1\}}{\partial f}\right] -2(f+l)\sigma\{q_1\} \left[\frac{\partial\mu\{q_1^+\}}{\partial f} - \frac{\partial\mu\{q_1\}}{\partial f}\right] = 0 \quad (29)$$

 $= \left(\frac{-\sqrt{2}\mathrm{erf}^{-1}\{\alpha\} \times \sigma\{q_1\}}{\mu\{q_1^-\} - \mu\{q_1\}}\right)^2 \text{ where } \frac{\partial \mu\{\cdot\}}{\partial f} \text{ and } \frac{\partial \sigma\{\cdot\}}{\partial f} \text{ are obtained through the differentiation of expressions for } E[X_b] \text{ and } \mathrm{Var}(X_b) \text{ in (12) and (13), respectively. We solve (29) numerically to obtain } f_{\mathrm{op}}.$ 

Summary of Steps to Calculate p, n, and  $f_{\rm op}$ : First, we calculate the value of  $t_m$ , as explained in Section V-D. Second, we

	Tag Population Size									
Accuracy	$10^{2}$			$10^{4}$			106			
Requirement	$f_{ m op}$	n	p	$f_{ m op}$	n	p	$f_{ m op}$	n	p	
$\alpha = 60.0\%, \beta = 40.0\%$	12	1.00E+00	2.84E-01	12	1.00E+00	2.88E-03	12	1.00E+00	2.88E-05	
$\alpha = 70.0\%, \beta = 30.0\%$	14	2.00E+00	3.55E-01	14	2.00E+00	3.59E-03	14	2.00E+00	3.59E-05	
$\alpha = 80.0\%, \beta = 20.0\%$	15	4.00E+00	3.91E-01	15	4.00E+00	3.96E-03	15	4.00E+00	3.96E-05	
$\alpha = 90.0\%, \beta = 10.0\%$	15	2.50E+01	3.91E-01	15	2.50E+01	3.96E-03	15	2.50E+01	3.96E-05	
$\alpha = 95.0\%, \beta = 5.00\%$	15	1.43E+02	3.91E-01	15	1.43E+02	3.96E-03	15	1.43E+02	3.96E-05	
$\alpha = 99.0\%, \beta = 1.00\%$	15	6.24E+03	3.91E-01	15	6.24E+03	3.96E-03	15	6.24E+03	3.96E-05	
$\alpha = 99.9\%, \beta = 0.10\%$	15	1.02E+06	3.91E-01	15	1.02E+06	3.96E-03	15	1.02E+06	3.96E-05	

TABLE II VALUES OF  $f_{\mathrm{op}}, n$ , and p for Different Values of lpha, eta and Tag Population Size

numerically solve (29) to obtain  $f_{\rm op}$ . Third, we put this value of  $f_{\rm op}$  along with  $t_m$  in (22) to obtain the value of p. Last, we put the resulting value of p along with  $f_{\rm op}$  in (27) and obtain the value of p. Note that although (22) does not involve p and p0, p1 still depends on them because it is a function of p2 and the optimal value of p3 depends on p4 and p5.

Table II shows the values of p, n, and  $f_{\rm op}$  for different accuracy requirements and tag population sizes calculated using the steps described above. We observe from this table that for a given tag population size, as the value of  $\alpha$  increases and/or  $\beta$  decreases, the value of n increases to fulfill the more stringent accuracy requirements. We also observe from this table that for a given  $(\alpha,\beta)$  pair, the values of  $f_{\rm op}$  and n are the same for all tag population sizes, which shows that total number of slots,  $(f_{\rm op}+l)\times n$ , depends only on the accuracy requirements and is independent of tag population size. We will formally prove the independence of estimation time from tag population size in Section VII-A. We further observe that as the tag population size increases, the value of p decreases to reduce the number of tags participating in a frame to keep the value of  $f_{\rm op}$  and p independent of tag population size.

## D. Obtaining Population Upper Bound $t_m$

So far, we have assumed the knowledge of an upper bound  $t_m$  on tag population size t. We now present a fast scheme to obtain  $t_m$  based on Flajolet and Martin's probabilistic counting algorithm [5]. Before calculating system parameters p, n, and  $f_{\rm op}$ , 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 ith frame), until the reader gets an empty slot. Suppose the empty slot occurred in the ith frame, then  $t_m=1.2897\times 2^{i-2}$  is an upper bound on t [5], [15]. According to [5],  $t_m$  asymptotically approaches t when instead of using a single value of the first empty slot from one experiment, we use average of values of the first empty slot from a large number of experiment.

Next, we determine how close the upper bound  $t_m$  has to be to the actual tag population size to ensure that ART achieves the required reliability and examine whether  $t_m$  obtained using  $t_m=1.2897\times 2^{i-2}$  lies close enough to t. We derive an expression to calculate the expected value of actual reliability, denoted by  $\tilde{\alpha}$ , as a function of  $t_m$  given that the required reliability  $\alpha$ , confidence interval  $\beta$ , and the actual tag population size t are known.

Equation (27) is obtained using the condition that actual reliability should be greater than or equal to the required reliability. Therefore, we use this equation to derive an expres-

sion for expected value of actual reliability. In (27), we calculate n using  $q_1$ ,  $q_1^+$ , and  $q_1^-$ , which are obtained from (1), (15), and (16), respectively, by putting  $t=t_m$ ,  $f=f_{\rm op}$ , and  $p=f_{\rm op}\{1-(\frac{1}{f_{\rm op}-1})^{\frac{1}{l_m}}\}$ . This gives us

$$q_1 = 1 - \left(\frac{1}{f_{\text{op}} - 1}\right), \ q_1^{\pm} = 1 - \left(\frac{1}{f_{\text{op}} - 1}\right)^{1 \pm \beta}.$$
 (30)

As the number of tags in the population are t and not  $t_m$ , when the reader executes the frames, the actual values of  $q_1$ ,  $q_1^+$ , and  $q_1^-$  represented by  $\hat{q}_1$ ,  $\hat{q}_1^+$ , and  $\hat{q}_1^-$ , respectively, follow the following equations:

$$\hat{q_1} = 1 - \left(\frac{1}{f_{\text{op}} - 1}\right)^{\frac{t}{t_m}}, \quad \hat{q_1}^{\pm} = 1 - \left(\frac{1}{f_{\text{op}} - 1}\right)^{\frac{t}{t_m}(1 \pm \beta)}.$$
(31)

Let  $\tilde{\alpha}$  represent the expected value of actual reliability in n rounds when the population contains t tags and the calculated value of upper bound is  $t_m$ , then the following equality holds:

$$\left(\frac{\sqrt{2}\operatorname{erf}^{-1}\{\tilde{\alpha}\}\times\sigma\{\hat{q}_{1}\}}{\mu\{\hat{q}_{1}^{+}\}-\mu\{\hat{q}_{1}\}}\right)^{2} = n$$

$$= \left(\frac{-\sqrt{2}\operatorname{erf}^{-1}\{\tilde{\alpha}\}\times\sigma\{\hat{q}_{1}\}}{\mu\{\hat{q}_{1}^{-}\}-\mu\{\hat{q}_{1}\}}\right)^{2}.$$

Substituting the value of n from (27) into the equation above and solving for  $\tilde{\alpha}$ , we get

$$\tilde{\alpha} = \operatorname{erf} \left\{ \operatorname{erf}^{-1} \{ \alpha \} \times \frac{\sigma\{q_1\}}{\sigma\{\hat{q}_1\}} \times \frac{\mu\{\hat{q}_1^+\} - \mu\{\hat{q}_1\}}{\mu\{q_1^+\} - \mu\{q_1\}} \right\}$$

$$= \operatorname{erf} \left\{ \operatorname{erf}^{-1} \{ \alpha \} \times \frac{\sigma\{q_1\}}{\sigma\{\hat{q}_1\}} \times \frac{\mu\{\hat{q}_1^-\} - \mu\{\hat{q}_1\}}{\mu\{q_1^-\} - \mu\{q_1\}} \right\}.$$
(32)

The expected actual reliability  $\tilde{\alpha}$  is a convex function of  $\frac{t_m}{t}$  and is equal to  $\alpha$  for two values of  $\frac{t_m}{t}$  represented by  $L_{tm}$  and  $U_{tm}$ . Fig. 6 plots the expected value of actual reliability  $\tilde{\alpha}$  as a function of  $\frac{t_m}{t}$  using (32) with  $\alpha=95\%$  and  $\beta=5\%$ . The dashed horizontal line in the figure marks the required reliability  $\alpha=95\%$ . The actual reliability will be greater than or equal to the required reliability as long as the value of  $\frac{t_m}{t}$  satisfies the following condition:

$$L_{tm} \le \frac{t_m}{t} \le U_{tm}. (33)$$

The values of  $L_{tm}$  and  $U_{tm}$  can be obtained by using  $\tilde{\alpha} = \alpha$  in (32) and solving it for  $t_m$  and dividing it by the tag population

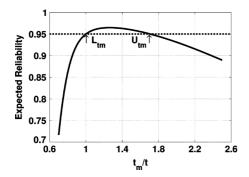


Fig. 6. Expected value of actual reliability versus  $\frac{t_m}{t}$ .

Accuracy	Tag Population Size					
Requirement	$10^{3}$	$10^{4}$	$10^{5}$	$10^{6}$		
$\alpha = 90.00\%, \beta = 10.0\%$	1.83	1.83	1.83	1.83		
$\alpha = 95.00\%, \beta = 5.00\%$	1.71	1.71	1.71	1.71		
$\alpha = 99.00\%, \beta = 1.00\%$	1.66	1.66	1.66	1.66		
$\alpha = 99.90\%, \beta = 0.10\%$	1.64	1.64	1.64	1.64		
$\alpha = 99.99\%, \beta = 0.01\%$	1.64	1.64	1.64	1.64		

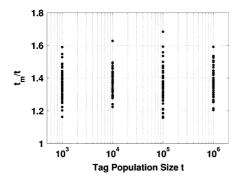


Fig. 7. Experimentally observed values of ratio  $\frac{t_m}{t}$ .

size t. This results in two values of  $\frac{t_m}{t}$  because  $\tilde{\alpha}$  is a convex function of  $\frac{t_m}{t}$  and its maxima is greater than  $\alpha$ . The value of  $L_{tm}$  is always equal to 1 and the value of  $U_{tm}$  is calculated by the numerical solution of (32) using  $\tilde{\alpha} = \alpha$ .

The value of  $U_{tm}$  depends on the required reliability  $\alpha$  and confidence interval  $\beta$ . Table III tabulates the values of  $U_{tm}$  for different population sizes and accuracy requirements. We observe from Table III that the value of  $U_{tm}$  is independent of tag population size. This is because  $U_{tm}$  depends on  $q_1$  for a given  $\alpha$  and  $\beta$  [according to (32)] and  $q_1$  is independent of tag population size as we will discuss in Section VII-A. We also observe that  $U_{tm}$  decreases with increasing accuracy requirements. This makes intuitive sense because the higher the required accuracy, the lesser the error in the upper bound  $t_m$  that can be tolerated. We see from Table III that even for very high accuracy requirements of  $\alpha=99.99\%$  and  $\beta=0.01\%$ , the value of  $t_m$  calculated as  $t_m=1.2897\times 2^{i-2}$  can be up to  $1.64\times t$ .

From simulations, we have observed that the value of  $t_m$  calculated as  $t_m=1.2897\times 2^{i-2}$  always lies within t and  $1.64\times t$ . This is seen in Fig. 7, where we plot the observed values of  $\frac{t_m}{t}$  obtained through 100 runs of simulations using  $t_m=1.2897\times 2^{i-2}$  for different values of tag population size. Within each simulation run, we obtained 10 values of i, av-

eraged them, and replaced i with that average in the equation  $t_m = 1.2897 \times 2^{i-2}$  to obtain  $\frac{t_m}{t}$ .

## VI. ART—PRACTICAL CONSIDERATIONS

In this section, we describe how ART estimates sizes of arbitrarily large tag populations. We also present the method that ART employs to enable the use of multiple RFID readers for estimating the size of a given RFID tag population.

## A. Unbounded Tag Population Size

For a given value of frame size f, Theorem 2 calculates the upper bound  $t_M$  on the number of tags that ART can estimate. This upper bound exists because for tag population sizes larger than  $t_M$ , the system parameters take on values that cannot be implemented practically. After Theorem 2, we describe how we extend ART to estimate sizes of arbitrarily large populations.

Theorem 2: For a given frame size f > 1, the maximum number of tags  $t_M$  that ART can estimate is

$$t_M = -\frac{\ln\{f - 1\}}{\ln\{1 - \frac{1}{2^{15}}\}}. (34)$$

*Proof:* In theory, we can increase the estimation scope of ART to any population size by decreasing the value of p according to (22). In practice, however, f/p has a minimum value of  $2^{15}-1$ . 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 store the result of the hash function is specified to be 15 in the C1G2 standard. Thus, the maximum value of f/p can be  $2^{15}-1$ , i.e.,

$$p > \frac{f}{2^{15}}$$
.

Substituting the value of p from (22) into the equation above, we get

$$f\left\{1 - \left(\frac{1}{f-1}\right)^{\frac{1}{t}}\right\} > \frac{f}{2^{15}}.$$

Rearranging the expression above and solving for t, we get

$$t < -\frac{\ln\{f-1\}}{\ln\left\{1 - \frac{1}{2^{15}}\right\}} = t_M.$$

As an example, with f=15,  $t_M$  is just 86 475. Practically, ART achieves required reliability only for tag populations smaller than  $t_M$ . If population size is larger than  $t_M$ , ART requires  $p \leq \frac{f}{2^{15}}$ , which is practically not possible with C1G2 RFID tags. This limitation exists with all the existing estimation schemes, but has never been addressed before.

Next, we present a strategy to estimate the sizes of arbitrarily large tag populations. The key idea is to first divide the entire population into smaller subpopulations of roughly equal sizes and then estimate the size of each subpopulation independently. At the end, adding the estimated sizes of all subpopulations gives the estimate of number of tags in the entire population. The size of any subpopulation should not require  $\frac{f}{n} \geq 2^{15}$ .

Next, we first calculate the number of subpopulations that ART should divide a given tag population into and then present a strategy to perform this division virtually (i.e., requiring no manual division of tags). Maximum number of tags that a subpopulation can have is given by (34). Therefore, the minimum number of subpopulations that the entire tag population should

be divided into is  $\frac{t_m}{t_M}$ , where  $t_m$  is calculated as explained in Section V-D.

To divide the tag population into subpopulations, we use the SELECT command standardized in the C1G2 standard. The ID of a tag is stored in its memory at a specific memory address. The tag can retrieve any bits stored in its memory by specifying an appropriate address range. Using the SELECT command, a reader can broadcast an address range and a bit mask that specifies which tags should participate in an Aloha frame. Each tag compares the bit mask to the bits in the specified address range in its memory and participates in the frame only if the bit mask matches the specified bits in its memory. To divide the whole population into subpopulations of roughly equal sizes, we leverage the fact that in large populations, the expected number of tags whose IDs have the least significant bit (LSB) of 0 is approximately the same as the expected number of tags whose IDs have the LSB of 1. Similarly, the expected number of tags whose IDs have the two LSBs of 00 is approximately the same as the expected number of tags whose IDs have the two LSBs of 01, 10, or 11, and so on. Therefore, a reader can divide the tag population into  $2^z$  groups of roughly equal sizes by specifying appropriate masks for the address range corresponding to the z LSBs of tag IDs. The value of z is given by  $\lceil \log_2 \{ \frac{t_m}{t_M} \} \rceil$ .

To summarize, a reader first obtains the value of upper bound  $t_m$ . Second, it calculates the value of n and  $f_{op}$ . Third, it calculates the value of  $t_M$  using (34). Fourth, it calculates  $z = \lceil \log_2\{\frac{t_m}{t_{\max}}\} \rceil$ . Fifth, it executes  $2^z$  independent estimation rounds for required reliability  $\alpha$  and confidence interval  $\beta$ , where in each round it uses SELECT command with a unique z bit mask for the z LSBs of the tag IDs. In each independent estimation round, it uses  $p = f_{op}\{1 - (\frac{1}{f_{op}-1})^{\frac{1}{t_m/2^z}}\}$ . Finally, it adds up all  $2^z$  estimates to obtain the estimate of total number of tags in the population.

## B. ART With Multiple Readers

We next discuss how to obtain  $t_m$  and  $\tilde{t}$  using multiple readers with overlapping coverage. 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 with an error tolerance of over  $1.64 \times t$ . Second, deployment of multiple readers in practice often requires site surveys to ensure minimal overlapping between readers.

To obtain t using multiple readers, we adapt the approach proposed by Kodialam et al. in [8], which uses a central controller for all readers. ART parameters  $\beta$ ,  $\alpha$ ,  $t_m$ , p, n, and  $f_{op}$  have the same value across all readers. When a reader transmits seed  $R_i$  in its ith frame, it does not generate  $R_i$  on its own. Rather, it uses the ith seed  $R_i$  issued by the central controller. That is, each reader generates the same sequence of n seeds. In the ith 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 ith frames from all readers and gets a single ith frame as if using a single reader. ART uses the n frames computed by logical OR to estimate the population size. Pseudocode of ART is given in Algorithms 1 and 2 in Appendix-C.

## VII. ART—ANALYSIS

In this section, first we prove that the estimation time of ART is independent of the tag population size. Second, we briefly discuss the computational complexity of ART. Last, we perform an analytical comparison of ART with existing schemes to mathematically justify the faster speed of ART compared to existing schemes.

#### A. Independence of Estimation Time From Tag Population Size

There are three inputs to ART: confidence interval  $\beta$ , required reliability  $\alpha$ , and a population of t tags where t is unknown. The total number of slots of ART,  $(f_{\rm op}+l)\times n$ , actually does not depend on t. Intuitively, the larger t is, the smaller p is according to (22). Although t plays an important role in computing p, n, and f individually, in formula  $(f_{\rm op}+l)\times n$ , the impact of t eventually gets canceled out. Next, we prove this independence.

From (27), we observe that the value of n depends on  $\alpha$ ,  $\beta$ ,  $\mu$ ,  $\sigma$ , and from (29), we observe that the value of  $f_{\rm op}$  depends upon  $\beta$ ,  $\mu$ ,  $\sigma$ . Thus, the total number of slots  $(f_{\rm op}+l)\times n$  depends on  $\alpha$ ,  $\beta$ ,  $\mu$ ,  $\sigma$ . The values of  $\alpha$  and  $\beta$  are given constants, and  $\mu$  and  $\sigma$  are functions of  $q_1$ , as seen from (12) and (13). To prove that  $(f_{\rm op}+l)\times n$  is independent of t, we have to prove that  $q_1$  is independent of t. From (1), we have  $q_1=1-(1-\frac{p}{f})^t$ . As we do not know the value of t, rather we know  $t_m$ , we use  $q_1=1-(1-\frac{p}{f})^{t_m}$ . Substituting the value of p using  $t=t_m$  from (22) into this expression of  $q_1$ , we get

$$q_{1} = 1 - \left(1 - \frac{1}{f} \times f\left\{1 - \left(\frac{1}{f-1}\right)^{\frac{1}{t_{m}}}\right\}\right)^{t_{m}} = \frac{f-2}{f-1}.$$
(35)

Thus the value of  $q_1$  that we use to calculate  $\mu$  and  $\sigma$  and consequently  $f_{\rm op}$  and n is independent of tag population size t or the upper bound on tag population size  $t_m$ . Therefore,  $f_{\rm op}$  and n depend only on  $\alpha$  and  $\beta$  regardless of the value of t or  $t_m$ . The upper bound on tag population size  $t_m$  only affects the value of p. For ART to achieve the required reliability, this upper bound has to satisfy the condition  $L_{tm} \leq t_m \leq U_{tm}$ . If  $t_m > t \times U_{tm}$ , the required reliability will not be achieved because the value of p will become so small that enough number of tags will not participate in the frames. Regardless, the value of  $(f_{\rm op} + l) \times n$  stays the same. We have seen from Fig. 7 that for all practical purposes, the value of  $t_m$  satisfies the requirement  $t_m \leq t_m \leq U_{tm}$  when calculated using the method proposed in Section V-D.

#### B. Computational Complexity

The two most computationally intensive tasks in ART are the numerical solutions of (12) to obtain the estimate  $\tilde{t}$  and of (29) to calculate  $f_{\rm op}$ . Fortunately, these two equations need to be solved numerically only once during the estimation process: (29) before executing the frames, and (12) after executing the frames. Consequently, the runtime complexity of ART is no larger than that of a standardized Aloha protocol. Almost all existing schemes involve numerical solutions of equations to obtain the estimate  $\tilde{t}$ . Therefore, the offline computational complexity of ART is comparable to those of existing estimation schemes.

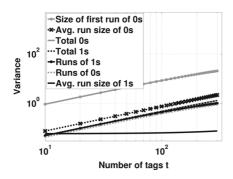


Fig. 8. Variance of different estimators versus RFID tag population size.

#### C. Analytical Comparison of Estimators

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 larger estimation time. Fig. 8 shows the analytical plots of the variances of the ART estimator and the above six 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 nonmonotonic 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 is calculated using (3). The variance of the size of the first run is calculated using (3) in [18] by setting i = 1. The variance of the number of runs of 0s and that of 1s is calculated using (5). We emphasize that plots in Fig. 8 are not based on experimental results, instead, they are based on analytical formulas.

#### VIII. 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 LoF [15] because it is noncompliant with C1G2 and CSE [22] because it does not take accuracy requirements as input. The estimation times for ART reported in this section include the time required to obtain the value of  $t_m$ . 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.

#### A. Estimation Time

The results in Figs. 9–11 show that the estimation time of ART is significantly smaller than all prior schemes. Note that in Figs. 10 and 11, 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 Fig. 9(a)–(c), which shows the estimation time needed by each scheme with population sizes of up to one million tags for different configurations of  $\alpha$  and confidence interval  $\beta$ . 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 Fig. 8. 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 the longer the estimation time.

We make three main observations from Fig. 10(a)–(c), which shows the estimation time of each scheme for 5000 tags with the required reliability  $\alpha$  varying from 90% to 99.9% for different configurations of confidence interval  $\beta$ . 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 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 Fig. 11(a)–(c), which show the estimation time of each scheme for 5000 tags with the confidence interval  $\beta$  varying from 0.1% to 10% for different configurations of  $\alpha$ . 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 fewer number of rounds are needed to achieve the required reliability.

## B. Actual Reliability

Fig. 12(a)–(c) shows the actual reliability of ART versus the number of tags for different configurations of required reliability  $\alpha$  and confidence interval  $\beta$ . We observe that ART always achieves the required reliability. These figures show several ups and downs in the plotted values. These ups and downs are not because of any noise. Rather, we see them because of the magnification level of vertical axis in these figures.

## IX. CONCLUSION

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 of arbitrarily large sizes. Using analytical plots, we show that our estimator has much smaller variance

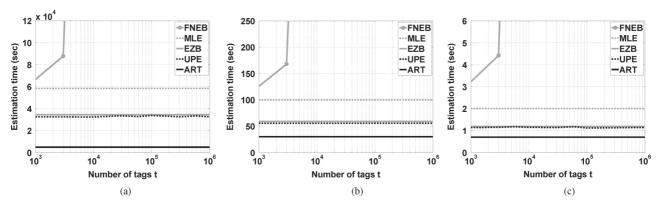


Fig. 9. Estimation time versus tag population size of ART and existing schemes for three different accuracy requirements. (a)  $\alpha = 99.9\%$ ,  $\beta = 0.1\%$ . (b)  $\alpha = 99\%$ ,  $\beta = 1\%$ . (c)  $\alpha = 95\%$ ,  $\beta = 5\%$ .

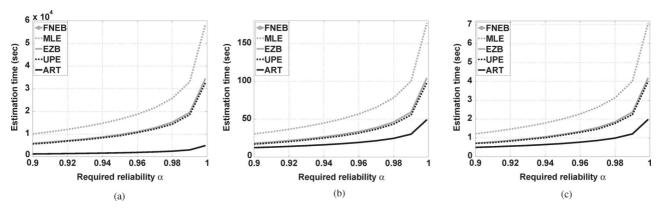


Fig. 10. Estimation time versus required reliability for ART and existing schemes for three different confidence intervals. (a)  $\beta = 0.1\%$ . (b)  $\beta = 1\%$ . (c)  $\beta = 5\%$ .

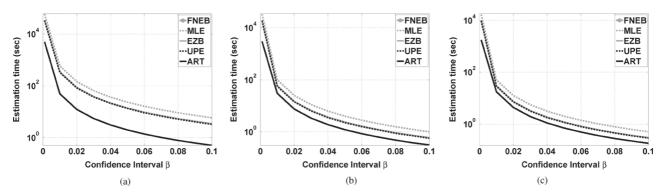


Fig. 11. Estimation time versus confidence interval for ART and existing schemes for three different required reliability values. (a)  $\alpha = 99.9\%$ . (b)  $\alpha = 99\%$ . (c)  $\alpha = 95\%$ .

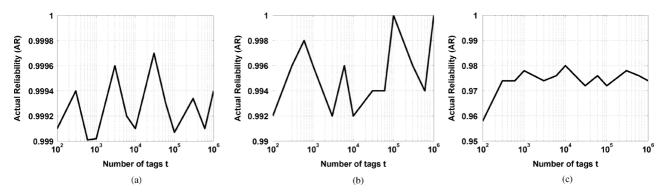


Fig. 12. Actual reliability achieved by ART for three different required reliability values and confidence intervals. (a)  $\alpha = 99.9\%$ ,  $\beta = 0.1\%$ . (b)  $\alpha = 99\%$ ,  $\beta = 1\%$ . (c)  $\alpha = 95\%$ ,  $\beta = 5\%$ .

compared to other estimators including those used in prior work. It is this smaller variance that makes our scheme faster than the previous ones. The key technical depth of this paper is in the mathematical development of the estimation theory using this estimator. ART can estimate arbitrarily large tag populations with arbitrarily high accuracy. It works with single as well as multiple readers. Our experimental results show that ART is significantly faster than all prior RFID estimation schemes. We have shown, both theoretically and experimentally, that the estimation time of ART is independent of the tag population size.

#### **APPENDIX**

## A. Proof of Theorem 1

*Proof*: Let  $g(Y_b,R_b)=X_b=\frac{Y_b}{R_b}$ . The Taylor series expansion of g around  $(\theta_1,\theta_2)$  is given by

$$g(Y_b, R_b) = \sum_{j=0}^{\infty} \left\{ \frac{1}{j!} \left[ (Y_b - \theta_1) \frac{\partial}{\partial Y_b'} + (R_b - \theta_2) \frac{\partial}{\partial R_b'} \right]^j \right.$$

$$\times g(Y_b', R_b') \right\}_{Y_b' = \theta_1}.$$

$$R_b' = \theta_2$$

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  $g(Y_b, R_b)$ :

$$g(Y_b, R_b) = g(\theta_1, \theta_2) + \left[ (Y_b - \theta_1) \frac{\partial g}{\partial Y_b} + (R_b - \theta_2) \frac{\partial g}{\partial R_b} \right]$$

$$+ \frac{1}{2} \left[ (Y_b - \theta_1)^2 \frac{\partial^2 g}{\partial Y_b^2} + 2(Y_b - \theta_1)(R_b - \theta_2) \frac{\partial^2 g}{\partial Y_b \partial R_b} \right]$$

$$+ (R_b - \theta_2)^2 \frac{\partial^2 g}{\partial R_b^2} + O(j^{-1}).$$

Taking the expectation of both sides, we get

$$E[g(Y_b, R_b)] \approx \frac{1}{2} \left[ \text{Var}(Y_b) \frac{\partial^2 g}{\partial Y_b^2} + 2 \text{Cov}(Y_b, R_b) \frac{\partial^2 g}{\partial Y_b \partial R_b} \right] + \text{Var}(R_b) \frac{\partial^2 g}{\partial R_b^2} + g(\theta_1, \theta_2).$$
(36)

Evaluating the partial derivatives of g as required in (36), we get

$$\begin{split} \frac{\partial^2 g(Y_b,R_b)}{\partial Y_b^2} \bigg|_{\substack{Y_b = \theta_1 \\ R_b = \theta_2}} &= 0 \quad \frac{\partial^2 g(Y_b,R_b)}{\partial Y_b \partial R_b} \bigg|_{\substack{Y_b = \theta_1 \\ R_b = \theta_2}} &= -\frac{1}{\theta_2^2} \\ \frac{\partial^2 g(Y_b,R_b)}{\partial R_b^2} \bigg|_{\substack{Y_b = \theta_1 \\ R_b = \theta_2}} &= 2\frac{\theta_1}{\theta_1^3}. \end{split}$$

Putting these values in (36) and using  $\theta_1 = E[Y_b]$  and  $\theta_2 = E[R_b]$ , we get (12).

The variance can be calculated as follows:

$$\operatorname{Var}(g(Y_b, R_b)) = E[\{g(Y_b, R_b) - E[g(Y_b, R_b)]\}^2]. \quad (37)$$

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 (37)

$$E[g(Y_b, R_b)] = E\left[ (Y_b - \theta_1) \frac{\partial g}{\partial Y_b} + (R_b - \theta_2) \frac{\partial g}{\partial R_b} \right]$$

$$+ g(\theta_1, \theta_2) + O(j^{-1})$$

$$= \left[ (0) \frac{\partial g}{\partial Y_b} + (0) \frac{\partial g}{\partial R_b} \right] + g(\theta_1, \theta_2)$$

$$+ O(j^{-1}) \approx g(\theta_1, \theta_2).$$

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 (37), we get

$$\operatorname{Var}\left(g(Y_{b}, R_{b})\right) = E\left[\left\{(Y_{b} - \theta_{1})\frac{\partial g}{\partial Y_{b}} + (R_{b} - \theta_{2})\frac{\partial g}{\partial R_{b}}\right\}^{2}\right] + O(j^{-1})$$

$$\approx \operatorname{Var}(Y_{b})\left(\frac{\partial g}{\partial Y_{b}}\right)^{2} + 2\operatorname{Cov}(Y_{b}, R_{b})\frac{\partial g}{\partial Y_{b}}\frac{\partial g}{\partial R_{b}} + \operatorname{Var}\left(R_{b}\right)\left(\frac{\partial g}{\partial R_{b}}\right)^{2}.$$
(38)

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

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

Putting these values in (38) and using  $\theta_1 = E[Y_b]$  and  $\theta_2 = E[R_b]$ , we get (13).

## B. Derivation of (22)

Applying the first-order Taylor series expansion on  $\mu\{q_1\}$ , we get  $\mu\{q_1\} = E[Y_1]/E[R_1]$ . Using the expressions of  $E[Y_1]$  and  $E[R_1]$  from (2) and (4), respectively, we can express  $\mu\{q_1\}$ ,  $\mu\{q_1^+\}$ , and  $\mu\{q_1^-\}$  as follows:

$$egin{aligned} \mu\{q_1\} &= rac{fq_1}{q_1(q_1+f(1-q_1))} \ \mu\{q_1^+\} &= rac{fq_1^+}{q_1^+(q_1^++f(1-q_1^+))} \ \mu\{q_1^-\} &= rac{fq_1^-}{q_1^-(q_1^-+f(1-q_1^-))} \end{aligned}$$

Substituting these expressions in (21), we get

$$\frac{2}{q_1 + f(1 - q_1)} - \frac{1}{q_1^+ + f(1 - q_1^+)} - \frac{1}{q_1^- + f(1 - q_1^-)} = 0.$$
Substituting the value of  $q_1$ ,  $q_1^+$ , and  $q_1^-$  from (1), (15), and (16), respectively, into the equation above, and to simplify the presentation using  $n = (1 - \frac{p}{2})^t$  we get

presentation, using 
$$\eta=(1-\frac{p}{f})^t$$
, we get 
$$\frac{2}{1-\eta+f\eta}-\frac{1}{1-\eta^{1+\beta}+f\eta^{1+\beta}}-\frac{1}{1-\eta^{1-\beta}+f\eta^{1-\beta}}=0$$
 Next, we do algebraic simplification of the expression above

$$-(1 - \eta + f\eta) \left\{ 1 - \eta^{1+\beta} + f\eta^{1+\beta} + 1 - \eta^{1-\beta} + f\eta^{1-\beta} \right\} +2(1 - \eta^{1+\beta} + f\eta^{1+\beta})(1 - \eta^{1-\beta} + f\eta^{1-\beta}) = 0.$$

Dividing the equation above by  $\eta^{1-\beta}$ , we get

$$-(1 - \eta + f\eta)\{2\eta^{\beta - 1} - \eta^{2\beta} + f\eta^{2\beta} - 1 + f\} + 2(1 - \eta^{1+\beta} + f\eta^{1+\beta})(\eta^{\beta - 1} - 1 + f) = 0.$$

Simplifying the equation above, we get

$$\begin{split} (f-1) + \eta^{2\beta} (-1 + f + 2f\eta - \eta - f^2 \eta) \\ + 2\eta^{\beta} (\eta(f-1)^2 + 1 - f) - \eta(1 - 2f + f^2) &= 0 \\ \Rightarrow (f-1) + \eta^{2\beta} ((f-1) - \eta(f-1)^2) \\ + 2\eta^{\beta} (\eta(f-1)^2 - (f-1)) - \eta(f-1)^2 &= 0. \end{split}$$

Dividing the equation above by f-1 and simplifying, we get

$$\Rightarrow (1 - \eta(f - 1))(1 - \eta^{\beta})^2 = 0.$$

In the equation above, either  $1-\eta(f-1)=0$  and/or  $1-\eta^{\beta}=0$ . The value of  $1-\eta^{\beta}$  equals zero only when  $\beta=0$ , but we know from our problem statement that  $\beta\in(0,1]$ , i.e.,  $\beta\neq0$ .

## **Algorithm 1: EstimateRFIDTagPopulation**( $\alpha, \beta, n_r$ )

```
Input: (1) Required reliability \alpha
              (2) Required confidence interval \beta
              (3) Number of readers n_r
    Output: Estimated tag population size \tilde{t}
 1 t_m := CalculateUpperBound(n_r)
 2 Solve (29) to get f_{op}.
 3 Put the value of f_{op} and t_m in (22) to get p.
 4 Evaluate n by using \alpha, \beta, f_{op}, and p, in (27).
 5 Evaluate t_M using (34).
 6 if t_m > t_M then
 7 \left[ z := \left\lceil \log_2 \left\{ \frac{t_m}{t_M} \right\} \right\rceil \right]
 8 else
    z := 0
10 Reevaluate p by using p = f_{op} \left\{ 1 - \left( \frac{1}{f_{op} - 1} \right)^{\frac{1}{I_m/2^z}} \right\}.
11 \tilde{t} := 0
12 for j := 1 \text{ to } 2^z \text{ do}
          Use SELECT with z bit binary representation of j as mask.
13
         for i := 1 to n do
14
               Provide all readers with f_{op}/p and random seed R_i.
15
                Run Aloha on readers and gather all readers' frames.
16
               Perform slot wise OR on all frames to get one frame.
17
               Obtain \tilde{X}_1(i), the average run size of 1s in this frame.
18
         \begin{array}{l} \tilde{X}_{avg} \leftarrow \sum_{i=1}^n \tilde{X}_1(i)/n \\ \text{Use } E[X_1] := \tilde{X}_{avg} \text{ and solve (12) to obtain an estimate} \end{array}
19
20
          \tilde{t_z} of the number of tags in the current sub-population.
         \tilde{t} := \tilde{t} + \tilde{t_z}
21
```

#### Algorithm 2: CalculateUpperBound $(n_r)$

```
Input: Number of readers n_r
   Output: Upper bound on tag population size t_m
1 \ f := 1
2 for j := 1 to n_r do
       i := 1
       p_i := 1
4
       repeat
            Provide reader j with f/p_i and a random seed R_i.
6
            Run Aloha on reader j and get the response.
            if slot is not empty then
8
                p_{i+1} := p_i/2
             i := i + 1
10
       until slot is empty
11
      t_{m,j} := 1.2897 \times 2^{i-2}
13 t_m := \sum_{j=1}^{n_r} t_{m,j}
14 return t_m
```

Therefore,  $1 - \eta(f - 1) = 0$ . Putting back  $\eta = (1 - \frac{p}{f})^t$  and solving  $1 - (f - 1)(1 - \frac{p}{f})^t = 0$  for p, we get (22).

## C. Pseudocode of ART

22 return  $\tilde{t}$ 

Pseudocode of ART is given in Algorithms 1 and 2.

#### REFERENCES

- [1] C. Bordenave, D. McDonald, and A. Proutiere, "Performance of random medium access control, an asymptotic approach," in Proc ACM SIGMETRICS, 2008, pp. 1–12.
- [2] J.-R. Cha and I. J.-H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," in Proc. Int. Conf. Parallel Distrib. Syst., 2005, pp. 63-67.
- [3] EPCGlobal, Inc., "Radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz-960 MHz," 1.2.0 ed., 2008.
- [4] K. Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication. Hoboken, NJ, USA: Wiley, 2010.

- [5] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for data base applications," J. Comput. Syst. Sci., vol. 31, no. 2, pp. 182-209,
- [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, pp. 1-9.
- [7] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in Proc. ACM MobiCom, 2006, pp.
- [8] M. Kodialam, T. Nandagopal, and W. C. Lau, "Anonymous tracking using RFID tags," in Proc. IEEE INFOCOM, 2007, pp. 1217-1225.
- [9] C. H. Lee and C. W. Chung, "Efficient storage scheme and query processing for supply chain management using RFID," in *Proc. ACM* SIGMOD, 2008, pp. 291–302.
- [10] T. Li, S. Wu, S. Chen, and M. Yang, "Energy efficient algorithms for the RFID estimation problem," in Proc. IEEE INFOCOM, 2010, pp.
- [11] B. Nath, F. Reynolds, and R. Want, "RFID technology and applications," IEEE Pervasive Comput., vol. 5, no. 1, pp. 22-24, Jan.-Mar.
- [12] A. Nemmaluri, M. D. Corner, and P. Shenoy, "Sherlock: Automatically locating objects for humans," in *Proc. ACM MobiSys*, 2008, pp.
- [13] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: Indoor location sensing using active RFID," Wireless Netw., vol. 10, pp. 701–710,
- [14] Philips, Amsterdam, The Netherlands, "SL2 ICS11 I-Code UID smart label IC functional specification datasheet," 2004 [Online]. Available: http://www.advanide.com/datasheets/sl2ics11.pdf
- [15] C. Qian, H. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," in Proc. 6th IEEE PerCom, 2008, pp. 30-39.
- [16] M. Roberti, "A 5-cent breakthrough," RFID J. vol. 5, no. 6, 2006 [Online]. Available: http://www.rfidjournal.com/articles/view?2295
- [17] V. Shah-Mansouri and V. W. Wongm, "Anonymous cardinality estimation in RFID systems with multiple readers," in Proc. IEEE GLOBECOM, 2009, pp. 1-6.
- [18] M. Shahzad and A. X. Liu, "Every bit counts: Fast and scalable RFID estimation," in Proc. ACM MobiCom, 2012, pp. 365–376.
- [19] M. Shahzad and A. X. Liu, "Probabilistic optimal tree hopping for RFID identification," in Proc. ACM SIGMETRICS, 2013, pp. 293-304.
- [20] H. Vogt, "Efficient object identification with passive RFID tags," Per-
- vasive Comput., vol. 2414, pp. 98–113, 2002.

  [21] C. Wang, H. Wu, and N.-F. Tzeng, "RFID-based 3-D positioning schemes," in *Proc. IEEE INFOCOM*, 2007, pp. 1235–1243.
- [22] A. Zanella, "Estimating collision set size in framed slotted aloha wireless networks and RFID systems," IEEE Commun. Lett., vol. 16, no. 3, pp. 300-303, Mar. 2012.
- [23] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for multiple RFID objects identification," IEICE Trans. Commun., vol. 88, pp. 991-999, 2005.



Muhammad Shahzad received the B.E. degree in electrical engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2008, and is currently pursuing the Ph.D. degree in computer science at Michigan State University, East Lansing, MI, USA.

He was with Next Generation Intelligent Networks Research Center, Islamabad, Pakistan, as a Researcher from 2008 to 2010. His research interests include probabilistic protocols, network measurement techniques, user authentication, big

data analytics, and computer and network security.



Alex X. Liu received the Ph.D. degree in computer science from the University of Texas at Austin, Austin, TX, USA, in 2006.

He is currently an Associate Professor with the Department of Computer Science and Engineering, Michigan State University (MSU), East Lansing, MI, USA. His research interests focus on networking, security, and dependable systems.

Dr. Liu received the IEEE and IFIP C. Carter Award in 2004 and an NSF CAREER Award in 2009. He received the MSU College of Engineering

Withrow Distinguished Scholar Award in 2011.