Collision-Aware Churn Estimation in Large-Scale Dynamic RFID Systems

Qingjun Xiao, *Member, IEEE, ACM*, Bin Xiao, *Senior Member, IEEE, Member, ACM*, Shigang Chen, *Fellow, IEEE*, and Jiming Chen, *Senior Member, IEEE*

***Abstract*— RFID technology has been widely adopted for real-world applications, such as warehouse management, logistic control, and object tracking. This paper focuses on a new angle of applying RFID technology—monitoring the temporal change of a tag set in a certain region, which is called churn estimation. This problem is to provide quick estimations on the number of new tags that have entered a monitored region, and the number of pre-existing tags that have departed from the region, within a predefined time interval. The traditional cardinality estimator for a single tag set cannot be applied here, and the conventional tag identification protocol that collects all tag IDs takes too much time, especially when the churn estimation needs to perform frequently to support real-time monitoring. This paper will take a new solution path, in which a reader periodically scans the tag set in a region to collect their compressed aggregate information in the form of empty/singleton/collision time slots. This protocol can reduce the time cost of attaining pre-set accuracy by at least 35%, when comparing with a previous work that uses only the information of idle/busy slots. Such a dramatic improvement is due to our awareness of collision slot state and the full utilization of slot state changes. Our proposed churn estimator, as shown by the extensive analysis and simulation studies, can be configured to meet any pre-set accuracy requirement with a statistical error bound that can be made arbitrarily small.**

***Index Terms*— RFID, cardinality estimation, churn estimation, departed tags, new tags, random hashing.**

I. Introduction

RFID (radio frequency identification) technology has a wide range of applications in real-world business operations [1]–[4], such as warehouse management, logistic control, asset tracking and automatic payment. RFID tags, each of which carries a unique ID, are attached to medical devices, retail products, library books, or car plates, allowing an RFID reader to remotely access the information of each individual tag, or collect the aggregate statistical information of a group of tags. When comparing with other wireless identification or sensing devices, RFID tags have the advantage of removing the need for batteries, whose recharge, replacement and recycling prove to be troublesome in practice [5], [6]. RFID tags can operate using only the small amount of energy harvested from the continuous radio waves emitted by nearby RFID readers.

A common basic functionality of RFID systems is *cardinality estimation*, whose aim is to estimate the number of tagged objects in a particular physical region as efficiently as possible [1], [2], [7]–[14]. This basic functionality can be used to monitor the inventory level of a warehouse, the sales in a retail store, and the popularity of attractions in a park [1]. In addition to its direct utility, RFID cardinality estimation can work as a pre-processing step to optimize the parameter of ID collection protocol from a group of tags [15]–[18]. Besides the high efficiency, another key advantage of cardinality estimation is that it avoids to identify any tags, and hence it will not raise privacy concerns, particularly in sensitive scenarios where the party performing the operation (such as warehouse or port authority) does not own the tagged items.

Most existing solutions (as referenced above) are designed to count the number of tags in a single static tag set. However, many RFID systems are inevitably dynamic, since as time passes, tagged items may enter or leave the surveillance region, e.g., a warehouse or a metro station. It is practically useful to estimate the number of tagged objects that are moved into the monitored region and the number of tagged objects that are moved out, within a certain time interval. This problem is called *churn estimation*, which has many applications, such as monitoring how the inventory level in a warehouse changes over time, and how the number of passengers carried by a subway train fluctuates after passing through each stop. Churn estimation is a different problem from *continuous scanning*, which is to identify the IDs of new tags or departed tags [19], [20]. Estimating the numbers of such tags is an operation that can be made much more efficient than identifying their IDs.

Can the traditional cardinality estimators do this job? We may use them to estimate the number of tags in the system at pre-set times. For example, a cardinality estimator may tell

Manuscript received December 27, 2014; revised November 12, 2015 and March 31, 2016; accepted June 6, 2016; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. X. Liu. Date of publication August 15, 2016; date of current version February 14, 2017. This work was supported in part by the Hong Kong Polytechnic University under Grant 4-BCB9, in part by the U.S. National Science Foundation under Grant CNS-1115548, and Grant CNS-1409797, in part by the China National Natural Science Foundation under Grant CNSF-61472385, Grant 61222305, Grant 61502098, Grant 61532013, and Grant 61320106007, in part by the National Program for Special Support of Top-Notch Young Professionals, in part by the Jiangsu Provincial Natural Science Foundation of China under Grant BK20150629, and in part by the Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grant 93K-9. The preliminary version of this paper titled "Differential Estimation in Dynamic RFID Systems" was published in proceedings of the IEEE INFOCOM Miniconference, pp. 295–299, April 14–19, 2013.

Q. Xiao is with the Key Laboratory of Computer Network and Information Integration, Education Ministry, Southeast University of China, Nanjing 211189, China (e-mail: csqjxiao@seu.edu.cn).

B. Xiao is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csbxiao@comp.polyu.edu.hk).

S. Chen is with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: sgchen@cise.ufl.edu).

J. Chen is with the Department of Control, Zhejiang University, Hangzhou 310027, China (e-mail: jmchen@iipc.zju.edu.cn).

Digital Object Identifier 10.1109/TNET.2016.2586308

us that there are 5,000 tags in the system at time 1, and 7,500 tags at time 2. We know the difference is 2,500. However, this does not tell us how many new tags are moved into the system between time 1 and time 2, and nor does it inform us how many existing ones are moved out within the time interval.

For the churn estimation problem, another possible solution is to use a tag identification protocol [15]–[18] to collect all the tag IDs at time 1 and then at time 2. By comparing the two tag sets, we can easily identify the new tags that appear only at time 2, and the departed tags that exist at time 1 but no longer at time 2. However, each tag ID must be 96 bits long to ensure the global uniqueness [21]. The time cost of collecting each 96-bit ID from a tag set is far more expensive than estimating their number [1], [2], [7]–[14]. It is desirable to avoid ID collection for saving protocol time cost, since long execution time of RFID system has many negative impacts, e.g., may disrupt workers' normal warehouse inventory operations. Moreover, collecting tag IDs could raise privacy concerns in certain applications. Imagine a subway system: If a passenger carries his electronic ticket everywhere and his surrounding RFID system collects the ticket ID frequently, then his whereabout can be tracked with fine details.

For the churn estimation problem, we design solutions based on the following intuition. Suppose an RFID reader uses framed ALOHA protocol to query a set of tags, and each tag responds by selecting a slot at random in the time frame. The reader will record the state of each time slot: emptiness (no tag transmitting), singleton (one tag transmitting), or collision (more than one tag transmitting). The combined state of all slots is referred to as the aggregate information of the tag set, which is collected periodically after each pre-set time interval. When the tag set is unchanged, its aggregate information will stay the same: Empty (singleton, or collision) slots will remain to be empty (singleton, or collision). When the tag set changes, it will cause some slots to have different states. By measuring the numbers of changed slots in the aggregate information, we can derive the numbers of new/departed tags.

Our preliminary work in [2] considers only two slot states: empty (no tag transmitting) and busy (one or more tags transmitting). In this paper, we also exploit the ability of RFID readers to detect radio collision among tags. According to the EPCglobal RFID standard [21]–[23], the function of collision detection will increase per-slot time cost roughly by 27%. Based on our analysis, thanks to the additional information contributed from tag collisions, our new solution with three states (empty/singleton/collision) can reduce the needed number of time slots by about half. Overall, we can reduce the protocol execution time by about 36.5%.

The contributions of this paper are summarized below.

- We define a problem called churn estimation for monitoring dynamic RFID tag sets, which has practical values.
- We propose three different kinds of churn estimators, each of which is designed based on a unique set of state changes in the collected aggregate information.
- The accuracy and time cost of the proposed churn estimators is analyzed based on Cramér-Rao lower bound.
- We use simulations to study the impact of various system parameters (e.g., frame load factor, observed slot number, and ratio of departed/new tags) on estimation accuracy.

The rest of this paper is organized as follows: Section II introduces the related work, and section III defines the problem of churn estimation. Section IV describes the raw input data of the problem. Section V defines the probability functions for a time slot to stay in empty/singleton/collision state. Sections VI, VII and VIII present three churn estimators which use different sets of slot pair observations. Section IX analyzes the mean and variance of proposed estimators, based on which section X compares them theoretically. Section XI uses simulations to verify the theoretical results, and section XII concludes the paper.

II. Related Work

Much existing RFID work focuses on how to efficiently read the IDs of a group of tags, which is called *tag identification* problem. Since the communications between tags and a reader is by wireless, collisions will happen when multiple tags respond simultaneously. Collision arbitration protocols fall into two categories: tree-based protocols [16], [17], and frame slotted ALOHA protocols [15], [18], [21]. The de-facto EPCglobal C1G2 standard falls in the category of ALOHA protocol [21].

Instead of collecting the IDs of individual tags, another branch of RFID research studies how to efficiently estimate the number of tags in a tag set. For reducing the time overhead of this counting process, a plethora of protocols have been developed, including unified probabilistic estimator (UPE) [7], which is based on ALOHA frames, lottery frame protocol (LoF) [9], first busy slot based estimation [10], probabilistic estimating tree (PET) [11], average run based tag estimation [13], and zero-one estimator [14]. The best known protocol is called two-phase simple RFID counting (SRC). It argues that, because a tag set can be scanned easily for multiple rounds, it is better to adopt a two-phase protocol design: The first phase makes a rough estimation quickly (e.g., by LoF or PET protocol), while the second phase generates estimation with finer accuracy by UPE [7].

Most of the estimation work also considers the case of counting the tags in a large region that needs multiple readers to cover [1], [9], [12], [24]. Essentially, their work estimates the *union of multiple tag sets*, each covered by one reader.

There is extensive work on the detection and identification of missing tags (which can be considered as unexpectedly departed tags, for example, due to theft). Missing-tag detection is to detect the event of missing tags [25]. Missing-tag identification is to collect the IDs of missing tags [26]–[28]. In the missing-tag line of research, we are not aware of work specifically for estimating the number of missing tags. Missing-tag identification will certainly give the number of missing tags, but it is well known that individual tag identification is unnecessary (and costly) for count estimation. Moreover, many approaches designed for missing tags assume that there are no new tags entering the system. For example, in [27], the reader maps (assigns) the set of tags under surveillance to a frame of time slots based on the hash values

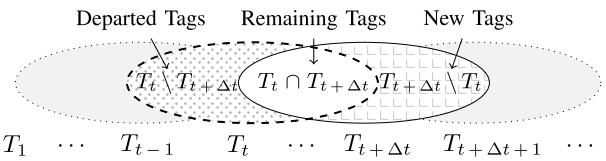


Fig. 1. A dynamic set of tags monitored by an RFID reader.

of the tag IDs. Now if the tags transmit in their assigned slots and the reader finds that a slot that should be busy turns out to be empty, then it knows that the tags assigned to the slot must be missing. However, if there are new tags entering the system, this approach will fail. Therefore, the approaches that are designed for missing-tag detection/identification may not work in an environment where there are both missing and new tags.

A similar argument can be made for the work that detects/identifies unknown (new) tags [6]. There also exists work that continuously identifies both the new tags entering the system and the pre-existing tags departing from the system [19], [20]. Again, identifying the IDs of individual tags is more costly than estimating the number of such tags.

The first work specifically designed for estimating the number of departed, remaining, and new tags is our 2013 INFOCOM mini-conference paper [2], which only uses empty slots. It is more general than the past missing-tag (or unknown-tag) work that assumes no new (existing) tags will enter (depart from) the system and more efficient than the past work that identifies missing (or new) tags. There is a follow-up work by other researchers in 2014 INFOCOM [20], which also uses only the information of empty/busy channel (for estimating the Jaccard similarity of two tag sets). This journal submission should be considered as a major extension to [2] by exploiting all types of slots: empty, singleton and collision.

We acknowledge that much related work on classical cardinality estimation can exploit the collision information [7], [29], [30]. The first paper by Kodialam and Nandagopal [7], which started this line of research exploiting tag collisions, uses such information to estimate the cardinality of a single tag set. In contrast, we use collision slots to estimate the difference of two tag sets.

III. Problem Formulation

In this section, we define the problem of *churn estimation*.

A. System Model

Consider an RFID system where a reader covers a region containing a large number of tags. As tags enter or leave this region, the set of tags that can be scanned by the reader changes over time. Such a dynamic tag set can be modeled as a series of tag sets T_1, T_2, …, T_t, … distributing over time domain, where T_t denotes the tag set interrogated by the reader at time t. The time series is illustrated in Fig. 1, where each ellipse represents a tag set at one particular time point.

We study how to monitor the change that a dynamic tag set experiences between two arbitrary time points. Let t and $t+\Delta t$ be the two time points assigned by users. The tag sets that are scanned by reader at these two time points are denoted by T_t and $T_{t+\Delta t}$, where T_t is called the "previous set" and $T_{t+\Delta t}$ is called the "current set". As shown in Fig. 1, T_t is drawn as a dashed ellipse and $T_{t+\Delta t}$ is depicted as a solid ellipse. We divide their union set $T_t \cup T_{t+\Delta t}$ into three disjoint subsets of tags, which are explained as follows.

- *Departed tags* are the tags found in the previous set but no longer in the current set. In Fig. 1, they are denoted by $T_t \setminus T_{t+\Delta t}$ and depicted as a dotted shadow region.
- *Remaining tags* are the tags found in both the previous set and the current set. They are denoted by $T_t \cap T_{t+\Delta t}$ and illustrated as a blank region in the figure.
- *New tags* are the tags that are not found in the previous set but exist in the current set. They are denoted by $T_{t+\Delta t} \setminus T_t$ and are depicted as a grid shadow region.

We want to measure the number of departed/remaining/new tags. Let n_1 be the number of departed tags, n_2 be the number of remaining tags, and n_3 be the number of new tags. Then,

$$n_1 = |T_t \setminus T_{t+\Delta t}| \quad n_2 = |T_t \cap T_{t+\Delta t}| \quad n_3 = |T_{t+\Delta t} \setminus T_t|.$$

Further, we denote the cardinality of the union of the two sets by $n_u = |T_t \cup T_{t+\Delta t}|$, and hence we have $n_u = n_1 + n_2 + n_3$.

B. Churn Estimation Problem

In many applications, we do not need the exact values of the numbers of departed/remaining/new tags. Their approximately estimated values with bounded error can already satisfy users' need. Let $\hat{n_1}$, $\hat{n_2}$, and $\hat{n_3}$ be the estimated values of churn numbers n_1, n_2, and n_3, respectively. We define their *relative estimation error* as the normalized differences between estimated values and ground truths: $\frac{\hat{n_1}-n_1}{n_1}$, $\frac{\hat{n_2}-n_2}{n_2}$, and $\frac{\hat{n_3}-n_3}{n_3}$.

For the number of departed tags n_1, we would like to bound its relative error within threshold $\pm\epsilon$ at a probability of at least α, i.e., $Prob\{-\epsilon < \frac{\hat{n_1}-n_1}{n_1} < \epsilon\} \geq \alpha$. By simple conversion,

$$Prob\{(1-\epsilon)n < \hat{n_1} < (1+\epsilon)n_1\} \geq \alpha. \tag{1}$$

It states that the confidence interval of departed tag estimation $\hat{n_1}$ is $\big((1-\epsilon)n_1, (1+\epsilon)n_1\big)$, and the confidence level should be at least α. Typically, we fix α to be 95% or 97.5%. Then, the smaller the ϵ value ($\epsilon \in (0,1)$) is predefined, the better the accuracy we have to implement for departed tag estimation.

Similarly, for the remaining tag estimation $\hat{n_2}$ and the new tag estimation $\hat{n_3}$, we can define their confidence intervals as $\big((1-\epsilon)n_2, (1+\epsilon)n_2\big)$ and $\big((1-\epsilon)n_3, (1+\epsilon)n_3\big)$, respectively.

$$Prob\{(1-\epsilon)n_2 < \hat{n_2} < (1+\epsilon)n_2\} \geq \alpha \tag{2}$$

$$Prob\{(1-\epsilon)n_3 < \hat{n_3} < (1+\epsilon)n_3\} \geq \alpha \tag{3}$$

The problem of *churn estimation* is to design efficient protocol that can satisfy the above constraints on estimation accuracy of departed/remaining/new tags, and meanwhile minimize the protocol execution time to attain the accuracy requirement.

Later, we will show that the relative estimation errors of n_1, n_2 and n_3 are affected by the ratios of departed/remaining/new tags in the union set $T_t \cup T_{t+\Delta t}$, which are denoted as $\gamma_1 = \frac{n_1}{n_u}$, $\gamma_2 = \frac{n_2}{n_u}$, and $\gamma_3 = \frac{n_3}{n_u}$, respectively. Since the sum of the three ratios is equal to one, we can use two of them, e.g., γ_1 and γ_3,

to specify the relation between previous tag set and current tag set. For instance, when $\gamma_1 = 0.15$ and $\gamma_3 = 0.2$, there are 15% departed tags, 65% remaining tags, and 20% new tags.

IV. System Model and Input Data

In this section, we introduce the underlying communication protocol that allows a reader to collect aggregate compressed information from the set of tags in its radio range. The aggregate information is the raw input data for churn estimation.

A. ALOHA-Based Communication Protocol

An RFID reader uses the following *framed slotted ALOHA protocol* to communicate with its surrounding tags. It is a popular anti-collision MAC (medium access control) protocol used by many previous studies [1], [2], [7], [9], [12], [24]. It is partially compliant with EPCglobal C1G2 standard [21].

- Firstly, the reader starts an ALOHA frame consisting of f time slots by broadcasting a `Query` command with two parameters $\langle f, r\rangle$, where r is a random seed shared by all tags in the frame. After hearing the command, each tag selects a time slot randomly to respond, using the hash function $h(ID \oplus r) \bmod f$, where ID is the tag identifier, r is the random seed, $\oplus$ is the bitwise XOR operator that mixes ID and r, and h is a pseudorandom hash function which is typically 16 bits long.
- Secondly, the reader divides the frame into f time slots, by broadcasting a `QueryRep` command at the boundary of each two neighboring slots (in order to terminate the current slot and initiate the next). As the reader broadcasts more `QueryRep` commands, the index of time slot increases. The tags whose generated hash values are consistent with the current slot will send their responses.

By executing this slotted ALOHA protocol, the responses of tags are distributed uniformly in a ALOHA frame, which can be treated as the aggregate information about the tag set at the time of reader scanning.

A sampling mechanism can be incorporated into the above ALOHA protocol, due to which only p $(0 < p \leq 1)$ fraction of tags respond in the frame and the rest of them keep silent. The following is an implementation of sampling mechanism which does not need tags with simple circuits to manipulate float numbers. Each tag has already generated a random number for slot selection after receipt of the `Query` command, i.e., $h(ID \oplus r) \bmod f$. The reader treats the tag as sampled if its random number is smaller than a sampling threshold f', i.e., $h(ID \oplus r) \bmod f < f'$. It is not difficult to find that, when f' is set to the integer $\lfloor p\, f \rfloor$, the sampling probability is $\lfloor p\, f \rfloor / f$, which is close to p when f is no small. Due to this sampling mechanism, the reader only needs to complete the transmission of the leading f' time slots in an ALOHA frame of size f, $0 < f' \leq f$. We call it a "truncated frame", which reduces the protocol execution time to the leading f' slots.

B. Empty/Singleton/Collision Slots and Their Time Expenses

In an ALOHA frame, the RFID reader monitors the state of each time slot transmitted. A time slot is said to be *empty* if there are no tag responses in that slot, or *busy* if it contains at least one tag responses. A busy slot can be further classified as a *singleton* slot if it contains exactly one tag response, and a *collision* slot if it contains two or more tag responses. An empty or singleton slot is called a *non-collision* slot.

If a reader only wants to distinguish busy slots from empty ones, then each tag reply needs to be just one bit long to indicate a busy channel. If the reader wants to further distinguish whether a busy slot is in singleton or collision state, each tag response needs be extended to 16 bits as defined by EPCglobal standard [21], such that the probability for two tags (among 10,000 tags) to reply with exactly the same content is less than 0.1%. The difference in tag reply contents can help the reader to detect the radio collision of multiple tags from their overlapped waveform.

Much previous work assumes that each tag reply contains only one bit for differentiating empty and busy slots [2], [19], [27]. Suppose the time cost for a tag to send each bit is 4μs.[1] It is natural to think that, since each tag response must be increased to 16 bits long for implementing collision detection, the time cost of a tag reply will grow by 16 times to 64μs. If the function of collision detection truly leads to the increase of per-slot time cost by 16 times, then it should not be considered.

However, as specified by the EPCglobal RFID standard [21], no matter how many bits are included in a tag response, there is considerable overhead for initiating a time slot, which is approximately 152μs.[2] The key reason is that the starting and terminating of a time slot is not controlled by the synchronized clock of tags, but by the reader's `QueryRep` command, which forces the current slot to finish and the next slot to start. The motivation of such a design is to simplify the circuit of passive tags and reduce the time cost of transmitting an empty slot [22], [31]. Therefore, the time cost of an empty slot is 113μs[3] for transmitting the `QueryRep` command.

The time cost of a busy slot should be calculated as follows, depending on whether we need to detect collisions. If each tag responds with one bit to indicate a busy channel, then the time cost of a busy slot is $152\mu\text{s} + 4\mu\text{s} \times 1 = 156\mu$s. If each tag responds with sixteen bits to facilitate the collision detection at the reader side, then the time cost is $152\mu\text{s} + 4\mu\text{s} \times 16 = 216\mu$s.

Also considering that the time slots of an ALOHA frame often have one third to be empty (when the number of tags and the number of slots are roughly equal as shown in Fig. 2),

- if each tag responds with only one bit, then the per-slot time cost is roughly $113 \cdot 1/3 + 156 \cdot 2/3 \approx 142.7\mu$s;
- if the function of collision detection is enabled, then the per-slot time cost is approximately $113 \cdot 1/3 + 216 \cdot 2/3 \approx 181.7\mu$s, which increases only by 27%.

[1] Suppose the tag-to-reader data rate is 256kbps as in [23], which lists typical settings of the parameters of UHF EPCglobal protocol, such as reader-to-tag data rate, FM0 encoding with TRext configured to zero, and so on. Then, the time cost for the tag to transmit each bit is about 4μs.

[2] The overhead of a busy slot consists of at least three parts: the transmission of QueryRep command (about 74μs, since QueryRep has 4 bits and reader-to-tag data rate is 54.23kbps), the waiting time from reader transmission to tag response T_1 (about 39μs), and the waiting time vice verse T_2 (39μs) [22], [23]. The Frame-Sync before QueryRep command is ignored for simplicity.

[3] The time expense of an empty slot is the transmission time of QueryRep command plus the waiting time T_1, i.e., $74 + 39 = 113\mu$s, if assuming T_3 is zero (the time a reader waits, after T_1, before it issues another command) [21].

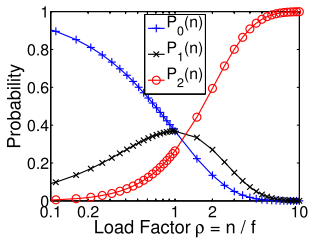


Fig. 2. Probabilities for 0, 1 or at least 2 tags to respond in a slot ($n = 3000$).

The low cost of detecting collisions in time slots has inspired us to study further how to exploit such information to improve the churn estimation accuracy.

Afterwards, we use 0, 1 and 2 to denote the empty, singleton and collision states of a time slot, respectively. By executing the slotted ALOHA protocol, an RFID reader can compress a tag set into an array of integers, where each integer represents the state of a corresponding time slot in the frame.

C. Input Data Collected From Two Frames

A system user may specify two arbitrary time points t and $t + \Delta t$ for estimating the number of departed/remaining/new tags within the period. In this subsection, we define the input data collected by the reader to accomplish this churn estimation task. An RFID reader can use the previously described ALOHA protocol to scan the tag set in its radio zone periodically.

The ALOHA frame at time t is called the previous frame, and the frame at $t + \Delta t$ is called the current frame. We make pairwise comparison between their time slots, and we define a *slot pair* as a slot in previous frame and its corresponding slot in current frame. We use the symbol x (or y) to denote the state of the ith slot in previous (or current) frame. Then, the state of a slot pair can be represented by a tuple $\langle x, y\rangle$. This tuple has nine possible states, because both x and y have three possible states $\{0, 1, 2\}$. We use symbol $X^{(i)}_{x,y}$, $0 \le i < f$, to denote the event that the ith slot pair is in the $\langle x, y\rangle$ state.

The reader counts the number of slot pairs that stay in $\langle x, y\rangle$ state, which is denoted by $N_{x,y}$. Let $1_{X^{(i)}_{x,y}}$ be the indicator function of event $X^{(i)}_{x,y}$, which equals one when the event happens and equals zero otherwise. Then, we have

$$N_{x,y} = \sum\nolimits_{i=0,1,\ldots,f'-1} 1_{X^{(i)}_{x,y}}, \tag{4}$$

where f' is the number of leading slots in the truncated frame both at time t and $t + \Delta t$. There are totally nine slot pair numbers $N_{x,y}$ with $0 \le x, y \le 2$. Our problem of churn estimation is to use these input data to generate the estimations $\hat{n_1}$, $\hat{n_2}$ and $\hat{n_3}$ satisfying the accuracy constraints in (1) and (3).

We define the number of "wildcard" slot pairs $N_{*,y}$ and $N_{x,*}$, which will be used later. In particular, $N_{*,0}$, $N_{*,1}$ and $N_{*,2}$ are the numbers of empty, singleton and collision slots in the current frame; $N_{x,*}$ can be similarly defined for the previous frame. Their relation with $N_{x,y}$ in (4) is as follows.

$$N_{*,y} = \sum\nolimits_{x=0,1,2} N_{x,y} \quad N_{x,*} = \sum\nolimits_{y=0,1,2} N_{x,y}$$

V. Cardinality Estimation for a Single Tag Set

In this section, we introduce the traditional algorithm for estimating the cardinality of a single tag set, which is scanned by reader using ALOHA protocol. Although this issue has been thoroughly investigated by previous work [7], we would like to give a brief introduction here, because the defined probability functions will be reused later in subsequent sections.

Let f be the number of time slots in the frame. Focus on a single slot of the frame. The probability for a tag to select the slot is $1/f$. Then, among n tags, the number of tags that will pick the slot follows a binomial distribution: $Binom(n, \frac{1}{f})$. Let P_0, P_1, and P_2 be the probabilities for this slot to hold 0, 1 and at least 2 tags, respectively, which correspond to the empty, singleton and collision states of a slot. Then, we have

$$\begin{aligned} P_0(n) &= (1 - \frac{1}{f})^n \\ P_1(n) &= n\,\frac{1}{f}\,(1 - \frac{1}{f})^{n-1} \\ P_2(n) &= 1 - P_0(n) - P_1(n). \end{aligned} \tag{5}$$

By applying the approximation $(1 - \frac{1}{f})^n \approx e^{-\frac{n}{f}}$ that works for a large f value, the above probability functions can be simplified as $\tilde{P}_0(\rho) \approx e^{-\frac{n}{f}} = e^{-\rho}$, $\tilde{P}_1(\rho) \approx \frac{n}{f} e^{-\frac{n-1}{f}} \approx \rho\, e^{-\rho}$, and $\tilde{P}_2(\rho) = 1 - \tilde{P}_0(\rho) - \tilde{P}_1(\rho)$. These probability functions have only one parameter $\rho = \frac{n}{f}$, which is called the *load factor* (or tag density) of the frame. Here, the upper tilde above symbol P indicates that the function parameter has been changed from n to the load factor ρ.

We plot the probability functions $\tilde{P}_0$, $\tilde{P}_1$ and $\tilde{P}_2$ against the load factor ρ in Fig. 2. It shows that $\tilde{P}_0$ and $\tilde{P}_2$ are monotonic functions of the load factor $\rho = n/f$. Since the frame length f is always known by the RFID reader, P_0 and P_2 can also be regarded as monotonic functions of the number of tags n.

We mentioned before that in an ALOHA frame, only the leading f' time slots with $f' = \lfloor pf \rfloor$ are transmitted, to realize the sampling of tags with probability p. Let z be the fraction of empty slots in this truncated frame with leading f' slots. When f' is large enough, z can be regarded as a good approximation for the probability of a slot to be empty, i.e., $z \approx P_0(n)$. Since function P_0 is monotonic, we can use its inverse function $P_0^{-1}(z)$ to generate an estimation of the tag number n.

$$\hat{n} = P_0^{-1}(z) = \log_{1-\frac{1}{f}}(z) \approx -f \log(z) \tag{6}$$

Let c be the fraction of collision slots in the truncated frame constituted by the preceding f' slots. Given the observation of c, we have $c \approx P_2(n)$. Since the function P_2 is also monotonic, we can use the inverse function $P_2^{-1}(c)$ to estimate the number of tags n. However, there is no closed-form formula to directly calculate P_2^{-1}. This inverse function has to be approximated by some numerical methods, e.g., bisection root finding.

VI. Empty-Slot Churn Estimator

In this section, we present our first estimator for the numbers of departed/new tags, called empty-slot churn estimator (ECE). It utilizes the observations about the numbers of slot pairs with at least one empty slot, i.e., $N_{0,*}$, $N_{*,0}$ and $N_{0,0}$.

From the number of empty slots $N_{*,0}$ in the current frame, we can estimate the number of tags in the current frame as

$$\widehat{n_2+n_3} = P_0^{-1}(N_{*,0}/f'). \tag{7}$$

We briefly explain this equation as follows. In the preceding f' slots of the current frame, the expected fraction of empty slots is $E(N_{*,0}/f') = P_0(n_2+n_3)$. According to (4), $N_{*,0}/f'$ is the arithmetic mean of a large number of independent variables. By applying central limit theorem, it approximates a Gaussian distribution whose variance is proportional to $1/f'$. When f' is large enough, we have $N_{*,0}/f' \approx P_0(n_2+n_3)$. Since P_0 has an inverse function in (6), we estimate n_2+n_3 as in (7).

Similarly, n_1+n_2 is the number of tags in the previous frame, and we can use $N_{0,*}$ to estimate it as $\widehat{n_1+n_2} = P_0^{-1}(N_{0,*}/f')$. Here, we have substituted the expected value of $N_{x,y}/f'$ by its instance value to generate an estimator. We will prove the unbiasedness of the generated estimator later.

In order to give estimations of n_1, n_2 and n_3, we need to know one more tag number. We will estimate the number of tags $n_u = n_1+n_2+n_3$ in the union of two sets, from the number of $\langle \text{empty}, \text{empty}\rangle$ slot pairs $N_{0,0}$.

Since the previous frame and the current frame use the same frame size f, each remaining tag in n_2 will select the same time slot in the two frames to respond (see the hash function for slot selection in Section IV-A). Hence, the probability for a slot pair to stay in the $\langle \text{empty}, \text{empty}\rangle$ state is $(1-\frac{1}{f})^{n_1}(1-\frac{1}{f})^{n_2}(1-\frac{1}{f})^{n_3} = (1-\frac{1}{f})^{n_1+n_2+n_3} = P_0(n_u)$. This probability is approximately equal to $N_{0,0}/f'$, i.e., the proportion of slot pairs that are $\langle \text{empty}, \text{empty}\rangle$ in the preceding f' slot pairs. Therefore, we estimate the union cardinality as $\hat{n_u} = P_0^{-1}(N_{0,0}/f')$.

Finally, the number of departed tag can be estimated as

$$\hat{n_1} = \hat{n_u} - \widehat{n_2+n_3} = P_0^{-1}(N_{0,0}/f') - P_0^{-1}(N_{*,0}/f').$$

Since P_0^{-1} is given in (6), we have $\hat{n_1} = -f\log(N_{0,0}/N_{*,0})$. Similarly, we can derive the estimators for the number of remaining tags n_2 and the number of new tags n_3, which have been presented in Definition 1. We will analyze the mean and variance of ECE estimators later in Section IX.

Definition 1 (Empty-Slot Churn Estimator): By observing the three slot pair numbers $N_{0,0}$, $N_{0,*}$ and $N_{*,0}$, the RFID reader can estimate the tag numbers n_1, n_2 and n_3 as follows.

$$\begin{aligned}
\hat{n_1} &= P_0^{-1}(N_{0,0}/f') - P_0^{-1}(N_{*,0}/f') \approx -f\log(N_{0,0}/N_{*,0}) \\
\hat{n_2} &= P_0^{-1}(N_{0,0}/f')\hat{n_1} - \hat{n_3} \\
&\approx -f\log\left(N_{0,*}\,N_{*,0}/(f'N_{0,0})\right) \\
\hat{n_3} &= P_0^{-1}(N_{0,0}/f') - P_0^{-1}(N_{0,*}/f') \approx -f\log(N_{0,0}/N_{0,*})
\end{aligned}$$

This ECE estimators were proposed in the preliminary conference version [2], which assumes RFID readers are able to differentiate empty slots from busy ones. Its shortcoming is that, when a frame receives too many tag responses, very few slots in the frames will remain empty, and the estimation accuracy of ECE severely degrades. Meanwhile, as specified by EPCglobal standard [21], RFID reader is capable of differentiating between singleton and collision states. We need to take benefit of the collision detection capacity, to improve the accuracy of churn estimation when dealing with dense frames.

VII. Collision-Slot Churn Estimator

In this section, we present the collision-slot churn estimator (CCE), which utilizes the numbers of slot pairs with at least one collision slots, i.e., $N_{*,2}$, $N_{2,*}$, and $N_{2,2}$.

For the fraction of collision slots in previous frame $N_{2,*}/f'$ and for the fraction of collision slots in current frame $N_{*,2}/f'$, their expected values are as follows.

$$E(N_{2,*}/f') = P_2(n_1+n_2) \quad E(N_{*,2}/f') = P_2(n_2+n_3) \tag{8}$$

Here, P_2 is the collision probability of a slot defined in (5), and it is a monotonic function as depicted in Fig. 2. Hence, we can solve the above equations and obtain the following estimators.

$$\widehat{n_1+n_2} = {P_2}^{-1}(N_{2,*}/f') \quad \widehat{n_2+n_3} = {P_2}^{-1}(N_{*,2}/f')$$

To estimate churns, we still need to know one more tag number. We will estimate the number of remaining tags n_2 from the number of slot pairs $N_{2,2}$ that are collision in both frames. How to implement the estimation is not straightforward. As to the tags that are contained by a slot pair in $\langle \text{collision}, \text{collision}\rangle$ state, there are three possible combinations, listed as follows:

1) at least two remaining tags,
2) one remaining tag, one or more departed tags, and one or more new tags,
3) no remaining tags, two or more departed tags, and two or more new tags.

The probability of the first case is the chance for a slot pair to contain at least two remaining tags, which is $P_2(n_2)$, where P_2 have been defined in (5). The probability of the second case is $\big(1-P_0(n_1)\big)P_1(n_2)\big(1-P_0(n_3)\big)$. The probability of the third case is $P_2(n_1)P_0(n_2)P_2(n_3)$. Therefore, the expected value of the fraction of $\langle \text{collision}, \text{collision}\rangle$ slot pairs $N_{2,2}/f'$ is

$$\begin{aligned}
E(N_{2,2}/f') = {} & P_2(n_2) + P_2(n_1)\,P_0(n_2)\,P_2(n_3) \\
& + \big(1-P_0(n_1)\big)\,P_1(n_2)\,\big(1-P_0(n_3)\big).
\end{aligned} \tag{9}$$

Since (9) has three unknown parameters n_1, n_2 and n_3, we cannot use (9) alone to estimate the number of remaining tags n_2. We must jointly consider the three equations in (8) and (9), which essentially constitute an equation set that puts three constraints over the three unknown parameters n_1, n_2 and n_3. A difficulty of solving this equation set is that the inverse function ${P_2}^{-1}$ has no closed-form formula. We have to resort to numerical iterative optimization method that fine tunes the churn estimations $\hat{n_1}$, $\hat{n_2}$ and $\hat{n_3}$ to minimize the error of the three equations in (8) and (9). We present the numerical optimization procedure in Algorithm 1.

VIII. All-Slot Pair Churn Estimator

The aforementioned ECE and CCE algorithms share an inadequacy that they partially utilize the numbers of slot pairs. ECE relies on the observation of $N_{*,0}$, $N_{0,*}$ and $N_{0,0}$, and CCE utilizes the observation of $N_{*,2}$, $N_{2,*}$ and $N_{2,2}$. In this section, we will present our best estimator, called the all-slot pair churn estimator (ACE), which exploits the observations of

Algorithm 1 Collision-Slot Churn Estimator

input : Slot number observations $N_{2,*}$, $N_{*,2}$ and $N_{2,2}$
output: Churn estimates $\hat{n_1}$, $\hat{n_2}$ and $\hat{n_3}$

1 Initialize the guesses of $\hat{n_1}$, $\hat{n_2}$ and $\hat{n_3}$ to zero.
2 Derive the current estimates of N_{202} and $N_{\bar{0}1\bar{0}}$: $\hat{N}_{202} = f' P_2(\hat{n_1}) P_0(\hat{n_2}) P_2(\hat{n_3})$ and $\hat{N}_{\bar{0}1\bar{0}} = f'\big(1 - P_0(\hat{n_1})\big) P_1(\hat{n_2}) \big(1 - P_0(\hat{n_3})\big)$, where $\hat{N}_{202}$ is the estimated number of slot pairs (i.e., case 3) and $\hat{N}_{\bar{0}1\bar{0}}$ is the estimated number of slot pairs (i.e., case 2).
3 Revise the remaining tag estimate $\hat{n_2}$ by $\hat{n_2} = P_2^{-1}\big(\frac{N_{2,2} - \hat{N}_{202} - \hat{N}_{\bar{0}1\bar{0}}}{f'}\big)$, where $N_{2,2} - \hat{N}_{202} - \hat{N}_{\bar{0}1\bar{0}}$ is the estimated number of slot pairs with at least two remaining tags, i.e., case 1
4 Revise the estimates of departed tags $\hat{n_1}$ and new tags $\hat{n_3}$: $\hat{n_1} = P_2^{-1}(\frac{N_{2,*}}{f'}) - \hat{n_2}$ and $\hat{n_3} = P_2^{-1}(\frac{N_{*,2}}{f'}) - \hat{n_2}$.
5 Go to step 2, if the revisions (to $\hat{n_1}$, $\hat{n_2}$ and $\hat{n_3}$) by steps 3 and 4 are larger than a threshold. Otherwise, program terminates

all the nine kinds of slot pairs $N_{x,y}$, $0 \le x, y \le 2$. Benefiting from the increase of available information, this new estimator can potentially achieve higher accuracy than ECE and CCE.

A. Maximum Likelihood Estimation (MLE)

We analyze the probability for a single slot pair to contain z_1 departed tags, z_2 remaining tags, and z_3 new tags, with $0 \le z_1, z_2, z_3 \le 2$. When z_1, z_2 or z_3 equals to 2, we mean at least two tags. Assuming there are totally n_1 departed tags, n_2 remaining tags, and n_3 new tags, the probability can be modelled as $P_{z_1}(n_1)\, P_{z_2}(n_2)\, P_{z_3}(n_3)$, where P_0, P_1 and P_2 are defined in (5). Note that the probability does not need to be $P_{z_1}(n_1)\, \big(P_{z_2}(n_2)\big)^2\, P_{z_3}(n_3)$, since a remaining tag always selects the same slot to reply in two frames by using the same hash function and one term $P_{z_2}(n_2)$ can be omitted.

Then, we analyze $P_{x,y}(n_1, n_2, n_3)$ — the probability for a slot pair to be in the state of $\langle x, y\rangle$, when there are totally n_1 departed tags, n_2 remaining tags, and n_3 new tags.

$$
\begin{aligned}
P_{0,0} &= P_0(n_1)\, P_0(n_2)\, P_0(n_3)\\
P_{0,1} &= P_0(n_1)\, P_0(n_2)\, P_1(n_3)\\
P_{0,2} &= P_0(n_1)\, P_0(n_2)\, P_2(n_3)\\
P_{1,0} &= P_1(n_1)\, P_0(n_2)\, P_0(n_3)\\
P_{1,1} &= P_1(n_1)\, P_0(n_2)\, P_1(n_3) + P_0(n_1)\, P_1(n_2)\, P_0(n_3)\\
P_{1,2} &= P_1(n_1)\, P_0(n_2)\, P_2(n_3) + P_0(n_1)\, P_1(n_2)\, \big(1 - P_0(n_3)\big)\\
P_{2,0} &= P_2(n_1)\, P_0(n_2)\, P_0(n_3)\\
P_{2,1} &= P_2(n_1)\, P_0(n_2)\, P_1(n_3) + \big(1 - P_0(n_1)\big)\, P_1(n_2)\, P_0(n_3)\\
P_{2,2} &= P_2(n_2) + \big(1 - P_0(n_1)\big)\, P_1(n_2)\, \big(1 - P_0(n_3)\big)\\
&\quad + P_2(n_1) P_0(n_2) P_2(n_3) \qquad (10)
\end{aligned}
$$

The definition of $P_{2,2}$ has been proved before in (9), which is the sum of the probabilities of three cases. We explain how to obtain the probability $P_{2,1}$. There are two cases for a slot pair to stay in the state $\langle$collision, singleton$\rangle$. It contains either (1) exactly one remaining tag, at least one departed tags and no new tag, or (2) no remaining tag, at least two departed tags and only one new tag. The probability of the first case is $\big(1 - P_0(n_1)\big)\, P_1(n_2)\, P_0(n_3)$ and the probability of the other case is $P_2(n_1)\, P_0(n_2)\, P_1(n_3)$. By summing up the two probabilities, we can obtain $P_{2,1}$ in (10). The other seven probabilities $P_{0,0}$, $P_{0,1}$, ..., $P_{1,1}$ and $P_{2,0}$ in (10) can be derived similarly.

The RFID reader, after scanning the preceding f' slots in the two frames, obtain the number of slot pairs $N_{x,y}$ that stay in the state $\langle x, y\rangle$, $0 \le x, y \le 2$, which was previously formulated into equation (4). The probability for the reader to make such observation, under the precondition of totally n_1 departed tags, n_2 remaining tags and n_3 new tags, is

$$Prob\{N_{x,y} | n_1, n_2, n_3\} = \prod\nolimits_{0 \le x,y \le 2} \big[P_{x,y}(n_1, n_2, n_3)\big]^{N_{x,y}}.$$

We explain it as follows. Given the probability $P_{x,y}$ in (10) for a slot pair to stay in $\langle x, y\rangle$ state, the probability for $N_{x,y}$ slot pairs to stay in $\langle x, y\rangle$ state is $[P_{x,y}(n_1, n_2, n_3)]^{N_{x,y}}$, because these slot pairs can be treated as approximately independent. By jointly considering all the nine kinds of slot pairs $\langle x, y\rangle$, $0 \le x, y \le 2$, we derive the probability of observing $N_{x,y}$ slot pairs for each state by multiplication rule of probability.

The above probability equation is also known as the *likelihood function* of the unknown churn numbers n_1, n_2 and n_3.

$$\mathcal{L}(n_1, n_2, n_3 \mid N_{x,y}) = \prod\nolimits_{0 \le x,y \le 2} \big[P_{x,y}(n_1, n_2, n_3)\big]^{N_{x,y}} \qquad (11)$$

We obtain the optimized churn estimations $\hat{n_1}$, $\hat{n_2}$ and $\hat{n_3}$, by maximizing log-likelihood function $\log \mathcal{L}(n_1, n_2, n_3 | N_{x,y})$.

$$\hat{n_1},\, \hat{n_2},\, \hat{n_3} = \underset{n_1, n_2, n_3}{\arg\max} \log \mathcal{L}(n_1, n_2, n_3 \mid N_{x,y}) \qquad (12)$$

Since this equation exploits all the nine slot pair states $\langle x, y\rangle$, $0 \le x, y \le 2$, we call it all-slot pair churn estimator (ACE). Later in Section X, we will analyze how much accuracy improvement ACE can achieve, as compared with ECE and CCE. To our best knowledge, it is the first work that improves the churn estimation accuracy by taking full advantage of RFID reader's function in detecting radio collisions.

B. Numerical Solution for MLE

For the likelihood maximization problem in (12), there is no closed-form solution, and instead we adopt a numerical iterative method, which is introduced as follows.

Firstly, we generate initial guesses about the churn numbers, using a hybrid estimator combining ECE and CCE (i.e., we apply ECE first to obtain churn number estimations $\hat{n_1}$, $\hat{n_2}$, $\hat{n_3}$, and if the estimated union load factor $(\hat{n_1} + \hat{n_2} + \hat{n_3})/f$ is smaller than one, we use the estimated results by ECE directly, and otherwise, we apply CCE instead to obtain another set of churn estimations), which will be explained in Section X-B.

Secondly, starting from the initial guesses, we search for the optimal churn estimations to minimize the log-likelihood function as in (12). The search is guided by the first-order derivative of the log-likelihood function $\langle \frac{\partial \log \mathcal{L}}{\partial n_1}, \frac{\partial \log \mathcal{L}}{\partial n_2}, \frac{\partial \log \mathcal{L}}{\partial n_3} \rangle$, where n_1, n_2, n_3 represent the current estimation of

the number of departed/remaining/new tags. We terminate the iterative optimization process, if the magnitude of the gradient vector becomes smaller than a given threshold.

The expression of the partial derivative $\frac{\partial \log \mathcal{L}}{\partial n_i}$ is as follows.

$$\begin{aligned}\frac{\partial \log \mathcal{L}(n_1,n_2,n_3|N_{x,y})}{\partial n_i} &= \frac{1}{\mathcal{L}(\ldots)}\frac{\partial}{\partial n_i}\prod\nolimits_{0\le x,y\le 2} P_{x,y}(n_1,n_2,n_3)^{N_{x,y}} \\ &= \sum\nolimits_{0\le x,y\le 2} \frac{N_{x,y}}{P_{x,y}(n_1,n_2,n_3)}\frac{\partial P_{x,y}(n_1,n_2,n_3)}{\partial n_i}\end{aligned}$$

In some boundary situations, the probability $P_{x,y}(n_1,n_2,n_3)$ may unfortunately become close to zero, which will drive the partial derivative $\frac{\partial \log \mathcal{L}}{\partial n_i}$ to infinity. If that happens, we will terminate the iterative optimization process instantly.

The above equation needs the partial derivative $\frac{\partial P_{x,y}}{\partial n_i}$, which is complicated and has twenty-seven cases, due to the existence of nine slot pair states $\langle x,y\rangle$ and three tag numbers n_i. We show three cases of partial derivative $\frac{\partial P_{x,y}}{\partial n_i}$ as follows.

$$\begin{aligned}\frac{\partial P_{2,2}}{\partial n_1} &= -\frac{\partial P_0(n_1)}{\partial n_1}P_1(n_2)\big(1-P_0(n_3)\big) \\ &\quad + \frac{\partial P_2(n_1)}{\partial n_1}P_0(n_2)P_2(n_3) \\ \frac{\partial P_{2,2}}{\partial n_2} &= \frac{P_2(n_2)}{\partial n_2} + \big(1-P_0(n_1)\big)\frac{\partial P_1(n_2)}{\partial n_2}\big(1-P_0(n_3)\big) \\ &\quad + P_2(n_1)\frac{\partial P_0(n_2)}{\partial n_2}P_2(n_3) \\ \frac{\partial P_{2,2}}{\partial n_3} &= -\big(1-P_0(n_1)\big)P_1(n_2)\frac{\partial P_0(n_3)}{\partial n_3} \\ &\quad + P_2(n_1)P_0(n_2)\frac{\partial P_2(n_3)}{\partial n_3} \end{aligned} \tag{13}$$

For simplicity, the other twenty-four cases of $\frac{\partial P_{x,y}}{\partial n_i}$ are omitted, which can be easily inferred from $P_{x,y}$ in (10).

The above equations need the probability $P_x(n)$, $0 \le x \le 2$, which can be found in (5). The above equations also require the first-order derivatives $\frac{\partial P_x(n)}{\partial n}$, which are given as follows.

$$\begin{aligned}\frac{\partial P_0(n)}{\partial n} &= (1-\frac{1}{f})^n \log(1-\frac{1}{f}) \\ \frac{\partial P_1(n)}{\partial n} &= \frac{1}{f}(1-\frac{1}{f})^{n-1} + n\frac{1}{f}(1-\frac{1}{f})^{n-1}\log(1-\frac{1}{f}) \\ \frac{\partial P_2(n)}{\partial n} &= -\frac{\partial P_0(n)}{\partial n} - \frac{\partial P_1(n)}{\partial n}\end{aligned} \tag{14}$$

Note that both (5) and (14) require the input number of tags n to be non-negative for generating valid results. Hence, if any of the current churn estimations $\hat{n_1}$, $\hat{n_2}$ and $\hat{n_3}$ is negative during the iterative optimization, it should be rounded to zero.

IX. Analysis of Proposed Estimators

In this section, we analyze the expected values and variances of the proposed three churn estimators, which paves the road for theoretical comparison of their accuracy and time cost.

A. Bias Analysis

We prove as follows that our ECE, CCE and ACE estimators are asymptotically unbiased. It is well-known that a maximum likelihood estimator (MLE) is asymptotically unbiased, if the independent and identically distributed experiments it observes has a sufficiently large number [32]. The ACE estimator is based on MLE, and it treats each slot pair as an independent experiment, whose number f' is often large enough. Therefore, ACE is asymptotically unbiased.

The ECE and CCE estimators are also asymptotically unbiased, since they are based on the well-known estimators P_0^{-1} and P_2^{-1}, which are able to derive the number of tags involved in an ALOHA frame in an asymptotically unbiased fashion. Take ECE estimator as an example. When it handles the number of departed tags, it uses $P_0^{-1}(\frac{N_{0,0}}{f'})$ to estimate the size of union set n_u, and $P_0^{-1}(\frac{N_{*,0}}{f'})$ to estimate the number tags in current frame n_2+n_3. Since both estimates are asymptotically unbiased, their difference is also asymptotically unbiased when it is used as an estimation of the departed tag number.

B. Variance Analysis

We will analyze the variances of proposed estimators in a generic theoretical framework called Cramér-Rao lower bound (CRLB) [32]. CRLB expresses a lower bound on the variance of estimators (or covariance matrix in multivariate scenario). It states that the variance of any unbiased estimator is at least as high as the inverse of the *Fisher information*. Meanwhile, for an unbiased maximum likelihood estimator, it will achieve the CRLB when the number of independent experiments it observes tends to infinity. Therefore, for an unbiased maximum likelihood estimator with a large number of independent observations, CRLB is often used to approximate its variance.

Fisher Information Matrix. The Fisher information matrix is a generic statistical tool to measure the amount of information that observable random variables carry about unknown parameters. In our churn estimation problem, the unknown system parameters are the numbers of departed/remaining/new tags n_1, n_2, n_3, and the observable random variables are the states of the leading f' slot pairs. Each of the three churn estimators concerns with a different set of states of these slot pairs.

- The ACE estimator uses all the nine kinds of slot pair states $\langle x,y\rangle$ with $0 \le x,y \le 2$, and it observes the number of slot pairs staying in each $\langle x,y\rangle$ state, which is denoted by $N_{x,y}$.
- The ECE estimator exploits the numbers of slot state transitions between empty and busy, i.e., $N_{0,0}$, $N_{0,\not 0}$, $N_{\not 0,0}$, $N_{\not 0,\not 0}$, which are equivalent to the numbers of state changes $N_{0,0}$, $N_{0,*}$, $N_{*,0}$, for $N_{0,*} = N_{0,0} + N_{0,\not 0}$ and $N_{*,0} = N_{0,0} + N_{\not 0,0}$.
- The CCE estimator exploits the number of state transitions between collision and non-collision: $N_{2,2}, N_{2,\not 2}, N_{\not 2,2}, N_{\not 2,\not 2}$.

The Fisher information matrix $\mathbf{F}$ can be calculated as follows.

Theorem 1 (Fisher Information Matrix): For a churn estimator that depends on the observation of slot pair numbers

$N_{x,y}$, its Fisher information matrix $\mathbf{F}$ is

$$\mathbf{F}_{ij} \approx f'^2 \cdot \Big(\sum\nolimits_{\langle x_1,y_1\rangle,\, \langle x_2,y_2\rangle} \frac{\partial P_{x_1,y_1}}{\partial n_i} \frac{\partial P_{x_2,y_2}}{\partial n_j} \Big), \quad (15)$$

where $\mathbf{F}_{ij}$ is the element of Fisher information matrix $\mathbf{F}$ at the ith row and the jth column ($1 \le i, j \le 3$), f' is the number of leading slots in a truncated frame that are observed, and $P_{x,y}$ is the probability for a slot pair to stay in the $\langle x, y\rangle$ state.

Proof: Please check the appendix for proof. ■

In (15), the $\sum$ operator applies summation over all the combinations of slot pair state $\langle x_1, y_1\rangle$ and another slot pair state $\langle x_2, y_2\rangle$. Since each churn estimator has its own set of observations, the number of terms of the summation $\sum$ differs.

- For the ACE estimator, there are eighty-one terms, because both $\langle x_1, y_1\rangle$ and $\langle x_2, y_2\rangle$ have nine possible slot pair states, with x and y picked from empty, singleton and collision.
- For the ECE estimator, there are sixteen terms for the summation, because it concerns with four slot pair states $\langle x, y\rangle$, with x and y picked from empty and busy.
- The CCE estimator also has sixteen terms for the summation, because it concerns with four slot pair states $\langle x, y\rangle$, with x and y picked from collision and non-collision.

Equation (15) needs the partial derivative $\frac{\partial P_{x,y}}{\partial n_i}$, which are described as follows for each churn estimator.

- For the ACE estimator, it observes the nine numbers of slot pairs $N_{x,y}$, $0 \le x, y \le 2$, and there are twenty-seven cases for the partial derivative $\frac{\partial P_{x,y}}{\partial n_i}$, which are shown in (13).
- The ECE estimator cares about four numbers of slot pairs: $N_{0,0}, N_{0,\bar{0}}, N_{\bar{0},0}, N_{\bar{0},\bar{0}}$. The probabilities of the four slot pair states are $P_{0,0}, P_{0,\bar{0}}, P_{\bar{0},0}, P_{\bar{0},\bar{0}}$, which can be easily derived from $P_{x,y}$ ($0 \le x, y \le 2$) in (10), using $P_{\bar{0},0} = P_{1,0} + P_{2,0}$, $P_{0,\bar{0}} = P_{0,1} + P_{0,2}$ and $P_{\bar{0},\bar{0}} = P_{1,1} + P_{1,2} + P_{2,1} + P_{2,2}$. Hence, for ECE, the partial derivative $\frac{\partial P_{x,y}}{\partial n_i}$ has twelve cases: $\frac{\partial P_{0,0}}{\partial n_i}$, $\frac{\partial P_{\bar{0},0}}{\partial n_i}$, $\frac{\partial P_{0,\bar{0}}}{\partial n_i}$, $\frac{\partial P_{\bar{0},\bar{0}}}{\partial n_i}$ with $0 \le i \le 2$, which can be derived from $\frac{\partial P_{x,y}}{\partial n_i}$ of the ACE estimator.

$$\frac{\partial P_{0,\bar{0}}}{\partial n_i} = \frac{\partial P_{0,1}}{\partial n_i} + \frac{\partial P_{0,2}}{\partial n_i} \quad \frac{\partial P_{\bar{0},0}}{\partial n_i} = \frac{\partial P_{1,0}}{\partial n_i} + \frac{\partial P_{2,0}}{\partial n_i}$$
$$\frac{\partial P_{\bar{0},\bar{0}}}{\partial n_i} = \frac{\partial P_{1,1}}{\partial n_i} + \frac{\partial P_{1,2}}{\partial n_i} + \frac{\partial P_{2,1}}{\partial n_i} + \frac{\partial P_{2,2}}{\partial n_i}$$

- For the CCE estimator, its observations are the four numbers of slot pairs: $N_{2,2}, N_{2,\bar{2}}, N_{\bar{2},2}, N_{\bar{2},\bar{2}}$. We can derive the probabilities of the four slot pairs $P_{2,2}$, $P_{2,\bar{2}}$, $P_{\bar{2},2}$, $P_{\bar{2},\bar{2}}$, using $P_{x,y}$ ($0 \le x, y \le 2$) in (10). The partial derivative $\frac{\partial P_{x,y}}{\partial n_i}$ has twelve cases: $\frac{\partial P_{2,2}}{\partial n_i}$, $\frac{\partial P_{\bar{2},2}}{\partial n_i}$, $\frac{\partial P_{2,\bar{2}}}{\partial n_i}$, $\frac{\partial P_{\bar{2},\bar{2}}}{\partial n_i}$ ($0 \le i \le 2$). Their formula can be derived from $\frac{\partial P_{x,y}}{\partial n_i}$ of the ACE estimator.

$$\frac{\partial P_{2,\bar{2}}}{\partial n_i} = \frac{\partial P_{2,0}}{\partial n_i} + \frac{\partial P_{2,1}}{\partial n_i} \quad \frac{\partial P_{\bar{2},2}}{\partial n_i} = \frac{\partial P_{0,2}}{\partial n_i} + \frac{\partial P_{1,2}}{\partial n_i}$$
$$\frac{\partial P_{\bar{2},\bar{2}}}{\partial n_i} = \frac{\partial P_{0,0}}{\partial n_i} + \frac{\partial P_{0,1}}{\partial n_i} + \frac{\partial P_{1,0}}{\partial n_i} + \frac{\partial P_{1,1}}{\partial n_i}$$

Due to the high complexity of $\frac{\partial P_{x,y}}{\partial n_i}$, the equation (15) that produces the element $\mathbf{F}_{ij}$ of Fisher information matrix is complicated. Although we have implemented the whole equation in MATLAB, we have to omit it in this paper to save space.

Cramér-Rao Lower Bound. According to CRLB in multivariate scenario, the covariance matrix of an unbiased estimator is at least the inverse of Fisher information matrix. Hence, the covariance matrix of $\hat{n_1}$, $\hat{n_2}$ and $\hat{n_3}$ satisfies the following inequality, no matter which churn estimator we adopt here:

$$Cov(\hat{n_1}, \hat{n_2}, \hat{n_3}) \ge \mathbf{F}^{-1}(n_1, n_2, n_3 \,|\, N_{x,y}), \quad (16)$$

where $\mathbf{F}$ is the Fisher information that measures the amount information carried by observations $N_{x,y}$ about the unknown parameters n_1, n_2, n_3. Due to the three churn estimations, the Fisher information takes the form of 3×3 matrix.

As can be derived from (16), the variance of churn estimation $Var(\hat{n_i})$ is at least the corresponding diagonal element $[\mathbf{F}^{-1}]_{ii}$ in the inverse of Fisher information matrix $\mathbf{F}^{-1}$.

$$Var(\hat{n_i}) \ge \big[\mathbf{F}^{-1}(n_1, n_2, n_3 \,|\, N_{x,y})\big]_{ii}$$

It is well known that, for an arbitrary maximum likelihood estimator, if its observed number of independent experiments tends to infinity, its variance becomes close to or even achieves CRLB [32]. For our churn estimation problem, our independent observations are the slot pairs whose number f' is counted by hundreds or thousands. Hence, we can approximate the variances of churn estimators by their CRLB.

$$Var(\hat{n_i}) \approx \big[\mathbf{F}^{-1}(n_1, n_2, n_3 \,|\, N_{x,y})\big]_{ii} \quad (17)$$

For example, the estimation variance of departed tag number $Var(\hat{n_1})$ can be approximated by the element $[\mathbf{F}^{-1}]_{11}$ at the left-top corner of the inverse of Fisher information matrix.

C. Transformation of Accuracy Constraints

Our churn estimators have been proved asymptotically unbiased, and they approximate Gaussian distributions, when the number of leading slot pairs f' observed is sufficiently large. Hence, the accuracy requirement in (1)-(3) can be rewritten as

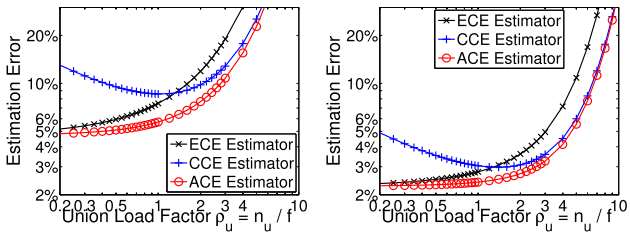
$$Z_\alpha \sqrt{Var(\hat{n_i})} \,/\, n_i \le \epsilon, \qquad Var(\hat{n_i}) \le (\epsilon n_i / Z_\alpha)^2, \quad (18)$$

where Z_α is the $\frac{1+\alpha}{2}$ percentile of standard Gaussian distribution, and for instance, $Z_{95\%} = 1.96$. For any accuracy constraint specified by confidence interval ϵ and confidence level α, (18) tells us to reduce $Var(\hat{n_i})$ smaller than $(\epsilon n_i / Z_\alpha)^2$.

After obtaining the information matrix $\mathbf{F}$ by (15), we can derive the variance of each churn estimator by (17). We denote the variance of ACE estimator by $Var_A(\hat{n_i})$, and the variance of ECE by $Var_E(\hat{n_i})$, and the variance of CCE by $Var_C(\hat{n_i})$. For each churn estimator, we should substitute Var in (18) by its own variance Var_A, Var_E or Var_C.

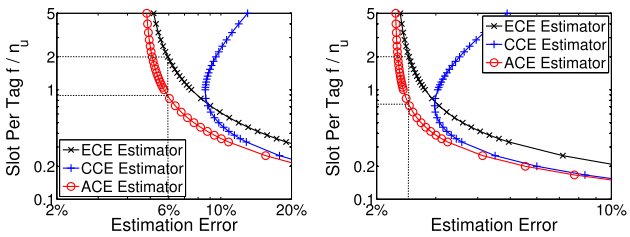
X. Theoretical Comparison of Churn Estimators

In this section, we will compare the performance of the three proposed churn estimators. We will show that, when they are given the same protocol execution time, ACE provides the best accuracy among them, and when they are configured to attain the same accuracy, ACE requires the lowest protocol time cost.



(a) (b)

Fig. 3. Comparison of relative standard error, when the sampling probability f'/f is fixed to one, and the number of tags in the union n_u set to 3000. (a) Departed tag estimation, with departed tag ratio $\gamma_1 = 0.15$. (b) Remaining tag estimation, with remaining tag ratio $\gamma_2 = 0.65$.



(a) (b)

Fig. 4. Compare protocol time cost when sampling probability $f'/f = 1$. (a) Departed tag estimation, with departed tag ratio $\gamma_1 = 0.15$. (b) Remaining tag estimation, with remaining tag ratio $\gamma_2 = 0.65$.

A. Performance Metrics and Protocol Parameters

The number of tags in the union set is fixed to 3000. The fractions of departed/remaining/new tags in the union set are configured to $\gamma_1 = 15\%$, $\gamma_2 = 65\%$ and $\gamma_3 = 20\%$, by default. We will adopt other combinations of tag fractions as well.

Performance Metrics. The performance of a churn estimator is evaluated by two metrics: churn estimation accuracy, and protocol time overhead required for encoding a tag set. Since the reader transmits only the leading f' slots of a frame (to record the numbers of slot pairs $N_{x,y}$ in $\langle x, y\rangle$ state), we can use f' to measure protocol execution time. For simplicity, we ignore the difference of empty/busy slots in per-slot time cost. The larger the tag ratio is, the smaller the relative error we will receive.

The estimation accuracy of churn number n_i, $1 \le i \le 3$, if performing simulation studies, can be quantified by the relative estimation error $|\hat{n_i} - n_i|/n_i$. But for the theoretical analysis in this section, the accuracy must be measured by the expected relative error $\sqrt{Var(\hat{n_i})}/n_i$, which is supposed to be smaller than ϵ/Z_α to satisfy accuracy constraint as shown in (18).

Protocol Parameters. There are two system parameters that affect the churn estimation accuracy, i.e., the union load factor $\rho_u = n_u/f$, and the tag sampling probability $p = f'/f$. The union load factor ρ_u is the load factor of the bitwise OR of previous frame and current frame. Once ρ_u is known, the load factor of previous frame is $(1-\gamma_3)\cdot\rho_u$ and the load factor of current frame is $(1-\gamma_1)\cdot\rho_u$, where γ_3 is the fraction of new tags in union tag set, and γ_1 is the fraction of departed tags.

B. Varied Protocol Execution Time f'

In this subsection, we ignore any constraint on the protocol time cost f', and configure f' equal to the frame size f, which implies that the sampling probability $p = f'/f$ is fixed to one.

Accuracy of Churn Estimators. In Fig. 3, we plot the relative estimation error $\sqrt{Var(\hat{n_i})}/n_i$ against the union load factor $\rho_u = n_u/f$ (note: if the three estimators are configured with the same ρ_u, it means they are given the same execution time f). Subfigure (a) depicts the estimation error of departed tags, while subfigure (b) shows the estimation error of remaining tag. The result of new tag estimation is omitted here, which is similar to departed tags. It is easy to find that the estimation error of remaining tags is smaller than that of departed tags. This is because the ratio of the remaining tags $\gamma_2 = 0.65$ is higher than the ratio of departed tags $\gamma_1 = 0.15$. The larger the tag ratio is, the smaller the relative error we will receive. We will evaluate the impact of tag ratios later in Section XI.

Figure 3(a) shows that, when the union load factor is smaller than one, ECE is much more accurate than CCE, but when the union load factor is larger than two, the accuracy of CCE becomes better. This is because the two estimators are complementary to each other: ECE uses the empty/busy slots, while CCE depends on the collision/non-collision slots. Since ECE and CCE use different information and they perform better in their respective ranges, we can easily combine them and build a hybrid estimator: We firstly apply ECE to obtain the churn number estimations $\hat{n_1}$, $\hat{n_2}$, $\hat{n_3}$, and if the estimated union load factor $(\hat{n_1} + \hat{n_2} + \hat{n_3})/f$ is smaller than one, we use the estimated results by ECE directly; otherwise, we apply CCE instead to obtain another set of churn estimations.

However, this hybrid estimator combining ECE and CCE is not the best way to exploit the abundant information buried in radio collision detection. As compared with both ECE and CCE, our proposed ACE estimator exploits the information of empty, singleton and collision slots simultaneously, and hence it greatly improves the estimation accuracy, which can be appreciated from Fig. 3. More precisely, ACE wisely exploits all the nine kinds of slot pair states $\langle x, y\rangle$, with x and y coming from empty/singleton/collision, and it is based on a solid math tool — maximum likelihood estimation.

It may appear that we can totally abandon ECE and CCE, and only depend on ACE to generate high-quality estimations. However, this is not the case. ECE and CCE, although less accurate than ACE, have the advantage of simplicity. ECE uses closed-form equations based on P_0^{-1} for generating churn estimations (see Definition 1), and CCE only invokes the function P_2^{-1} for ten or twenty rounds (see Algorithm 1). By contrast, ACE is based on the maximum likelihood computing, which is implemented by numerical iterative optimization. If the optimization algorithm wants fast convergence, it needs good initial guesses about churn numbers, which can be generated using the hybrid algorithm that combines ECE and CCE. How to implement the maximum likelihood calculation of ACE has been described previously in Section VIII-B.

Compare Protocol Time Cost. Considering that ACE is more accurate than other two estimators when given the same number of slots, our next experiment is to investigate how much protocol execution time can be saved by using ACE, if all the estimators are to achieve the same accuracy. Since the

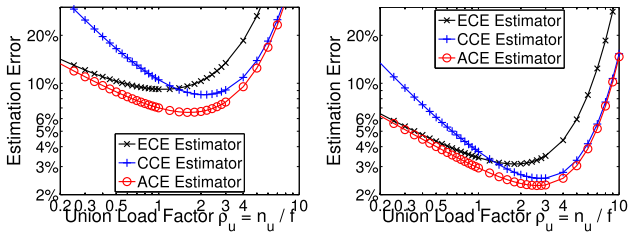


Fig. 5. Comparison of estimation error with fixed time cost $f' = 2000$. If f reduces to be smaller than 2000, then we set $f' = f$. (a) Departed tag estimation, with departed tag ratio $\gamma_1 = 0.15$. (b) Remaining tag estimation, with remaining tag ratio $\gamma_2 = 0.65$.

time cost f' is configured equal to the frame size f in this subsection, we can divide f by the union tag set n_u to obtain the average number of slots per tag. We plot f/n_u against the relative estimation error in Fig. 4, which shows that ACE reduces the number of slots by at least half when it is compared with ECE. In Fig. 4(a), to reduce the estimation error to 6%, ECE needs about 2 time slots per tag. In contrast, ACE can use only 0.9 slots per tag to attain the same accuracy, which saves the number of slots by at least half. A similar phenomenon can be observed for remaining tag estimation in Fig. 4(b).

C. Fixed Protocol Execution Time f'

In this subsection, we acknowledge that there exists a constraint on the protocol time cost, and configure the number of transmitted slots f' to a predefined value. When f' is fixed, as the frame size f reduces, both the tag sampling probability $p = f'/f$ and the union load factor $\rho_u = n_u/f$ will increase.

Impact of Load Factor ρ_u. We evaluate the impact of union load factor ρ_u on the relative standard error in Fig. 5, where the protocol time cost f' is fixed to 2000. The remaining tags in subfigure (b) exhibit a similar trend with the departed tags in (a): When f' is fixed, the curves of estimation error are no longer monotonic in Fig. 3, and there is an optimal value of union load factor that minimizes the estimation error. For ACE, the optimal value is $\rho_u = 1.6$ in (a), and $\rho_u = 2.5$ in (b).

We explain this new trend of relative estimation error as follows. When ρ_u is greater than the optimal value and increases, the estimation accuracy degrades rapidly as shown in Fig. 5, since the ALOHA frames becomes overly dense with too much slot pairs containing at least two remaining tags and staying in $\langle \text{collision}, \text{collision} \rangle$ state, which can not reflect the existence of departed/new tags. When $\rho_u = n_u/f$ is smaller than the optimal value and continues to reduce, the estimation accuracy degrades, because it causes the decreasing of sampling probability $p = f'/f$, which brings larger sampling error.

XI. Simulation Results

In this section, we use simulations to study the performance of the proposed churn estimators. We will compare our churn estimators with a recent work named ACOS (Adaptive COntinuous Scanning) scheme [20], which encodes a set of tags by their k-smallest hash values. We will evaluate how much improvement of estimation accuracy can be achieved, if we encode each tag set into a slotted ALOHA frame and exploit the information buried in collision slots.

Simulation Settings. The settings are largely the same with those in Section X-A. We just explain how to measure the protocol time cost for a churn estimator. ECE, CCE and ACE encode a tag set into an ALOHA frame with f time slots, and only the leading f' slots are actually transmitted. Hence, their time cost can be quantified by f'. In contrast, ACOS [20] encodes each tag set by the k minimum hash values, and thus its time cost is proportional to k. Later in Section XI-D, we will explain that the time cost of ACOS is roughly $18k$ rounds of communication between the reader and its nearby tags, which is equivalent to a frame with $f' = 18k$ time slots.

A. Impact of Union Load Factor When $f' = f$

In this subsection, we configure the number of observed slots f' equal to the frame size f, and thus the sampling probability always equals one. The only protocol parameter that influences the estimation accuracy is the union load factor ρ_u. We evaluate its impact, and plot the simulation results in Fig. 6(a)(b)(c) for departed/remaining/new tags, respectively.

Figures 6(a)(b)(c) show that basic trends for the three kinds of tags are the same. The major difference is that the accuracy of estimating remaining tags is better than the accuracy of departed tags or new tags. This is because the ratio of remaining tags (i.e., $\gamma_2 = 0.65$) is higher than the ratios of the other two kinds of tags (i.e., $\gamma_1 = 0.15$ and $\gamma_3 = 0.2$). The larger the tag ratio is, the smaller the relative error we will receive.

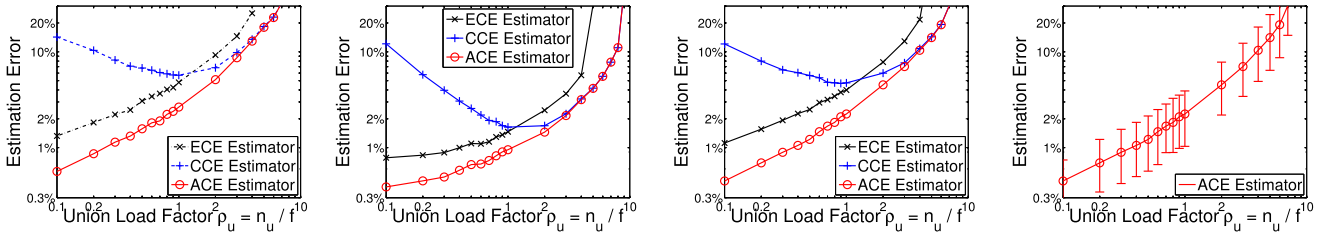
Figure 6(a) shows that, when the union load factor ρ_u is smaller than 1 roughly, ECE estimator is more accurate than CCE estimator. When ρ_u exceeds this bound, both ECE and CCE estimators degrade in accuracy. But the degradation speed of CCE estimator is more graceful than ECE estimator. This is consistent with our analysis result in Fig. 3.

Figure 6(a) also shows that ACE provides the best accuracy among the three estimators. Its advantage is particularly prominent when the load factor is smaller than 2. The reason, as stated before, is that ACE estimator exploits the observations of all the nine numbers of slot pairs $N_{x,y}$, $0 \le x, y \le 2$. More information brings about higher estimation accuracy.

Different from plots (a)(b)(c), Fig. 6(d) depicts both the average value and the deviation of estimation error. It shows that both the average and deviation of ACE estimation error reduces, as the union load factor decreases. For example, when the load factor is as large as 4, the estimation error deviates between 4.9% and 18%, which shows that the estimation error is quite unstable. When the load factor is reduced to 0.3, the estimation error deviates between 0.4% and 1.5%.

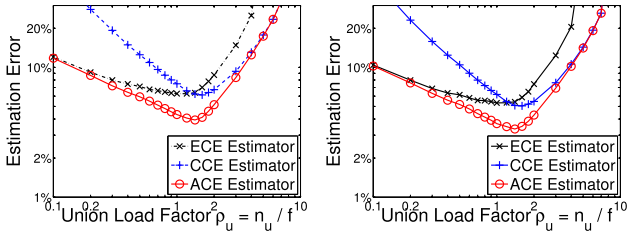
B. Impact of Union Load Factor When f' Is Fixed

In this subsection, we fix the number of observed slots f' to 2000, and investigate the impact of union load factor on estimation accuracy. We plot the simulation results in Fig. 7, with subfigure (a) is for departed tags and (b) for new tags. The figure shows that, when the number of observed slots f' is fixed, the estimation error of ECE is no longer a monotonic function of union load factor ρ_u, and the best ρ_u value for accurate estimation is close to one. In either plot, there is a crossing point of ECE and CCE estimators, exceeding



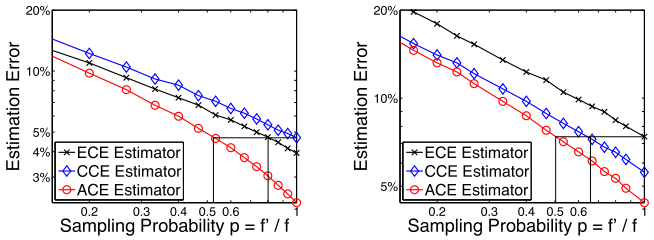
(a) (b) (c) (d)

Fig. 6. Impact of union load factor ρ_u on estimation error, when the sampling probability f'/f equals one, and assuming $\gamma_1 = 15\%$, $\gamma_2 = 65\%$, $\gamma_3 = 20\%$. (a) Departed tag estimation. (b) Remaining tag estimation. (c) New tag estimation. (d) New tag estimation (errorbars).



(a) (b)

Fig. 7. Impact of union load factor ρ_u, when the number of observed time slots f' is fixed to 2000, and assuming $\gamma_1 = 15\%$, $\gamma_2 = 65\%$, $\gamma_3 = 20\%$. (a) Departed tag estimation. (b) New tag estimation.



(a) (b)

Fig. 8. Impact of sampling ratio f'/f with fixed union load factor ρ_u. (a) New tag estimation with $\rho_u = 1$. (b) New tag estimation with $\rho_u = 2$.

which CCE performs better than ECE. These simulation results are consistent with our theoretical analysis result in Fig. 5. From Fig. 7, it is quite clear that ACE estimator provides the best accuracy among the three protocols. By comparing subfigures (a) and (b), the estimation accuracy of new tags is slightly better than the accuracy of departed tags, since the ratio of departed tags 20% is larger than the ratio of new tags 15%.

C. Impact of Tag Sampling Probability p

In this subsection, we keep the union load factor ρ_u fixed, and vary the protocol time cost f' (or the sampling probability $p = f'/f$), to evaluate the impact of p on estimation accuracy.

Firstly, we theoretically analyze the impact of the sampling probability p. By (15), each element in the Fisher information matrix $\mathbf{F}$ is proportional to f'^2. Hence, the square root of inverse Fisher information $\sqrt{[\mathbf{F}^{-1}]_{ii}}$, which determines $\sqrt{Var(\hat{n}_i)}$ (see Eq. (17)), is proportional to $1/f'$. Therefore, we can know that the standard relative error $\sqrt{Var(\hat{n}_i)}/n_i$ is inversely proportional to f', and to the sampling probability p.

Secondly, we plot the simulation results in Fig. 8, where subfigure (a) is for new tag estimation when $\rho_u = 1$ and subfigure (b) is for new tag estimation when $\rho_u = 2$. Figure 8 shows that, for all the estimators, their error decreases linearly as the growth of sampling probability $p = f'/f$. This is consistent with the analysis result about the impact of p.

Thirdly, Fig. 8(a) also shows that the time cost of ACE is much lower than the other two estimators. If the same accuracy (depicted by a horizontal line) is to be attained, ACE needs the sampling probability to be about 0.5, while the ECE and CCE estimators need the sampling probability to be 0.8 and 1, respectively. The same phenomenon can be witnessed in Fig. 8(b), where the union load factor increases two and CCE becomes more accurate than ECE. To achieve the estimation accuracy depicted as a horizontal line, ACE only needs the sampling probability to be 0.5, while ECE and CCE requires the sampling probability to be 1 and 0.67, respectively.

In summary, the number of slots needed by ECE is roughly two times larger than ACE to attain the same accuracy, which is consistent with our analysis result in Figures 4.

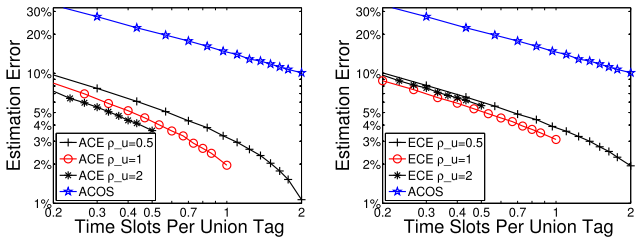
D. Accuracy Comparison of ECE, ACE and ACOS

This subsection compares the estimation accuracy of ECE and ACE with a recently proposed protocol named ACOS [20], when all of them are given the same protocol execution time.

We explain the protocol time cost of ACOS as follows. To encode a tag set, ACOS broadcasts a command asking each tag to generate a 24-bit hash value,[4] and then attempts to collect the k smallest ones. The entire hash space can be organized as a sorted binary tree, whose depth is 24. In such a tree structure, identifying the smallest hash value is equivalent to finding a path starting from the tree root to the leftmost leaf node. The time cost of reaching the leftmost leaf node is $24 \cdot (0.5 + 2 \cdot 0.5) = 36$ rounds of communication, assuming there is always half of chance for the leaf to hide on the left branch (need one round of communication) and another half of chance on the right (need two rounds of communication). Since the collection of the k smallest hash values is performed one by one, the overall time cost can be approximated by $36k$.

ACOS also proposes an optimization to reduce the time cost, utilizing the prior knowledge of union set size, e.g., n_u falls in the range $(2048, 4096]$. Then, by dividing the hash space with 2^{24} possible values equally into $2^{\lceil \log_2(n_u) \rceil} = 2^{12}$ intervals, there is a good chance for each interval to contain one tag's hash value, due to the uniform distribution of hash values of

[4]ACOS [20] assumes that any two tags in the union set have negligibly small probability to experience hash collision. Since the union set has been configured to have 3000 tags, according to the birthday paradox, each hash value must be at least $\lceil \log_2(3000 \times 3000) \rceil = 24$ bits long to avoid collision.



(a) (b)

Fig. 9. Accuracy comparison of ECE, ACE and ACOS, when given the same protocol execution time, and assuming $\gamma_1 = 30\%$, $\gamma_2 = 55\%$, $\gamma_3 = 15\%$. (a) Compare ACE and ACOS on the estimation accuracy of departed tags. (b) Compare ECE and ACOS on the estimation accuracy of departed tags.

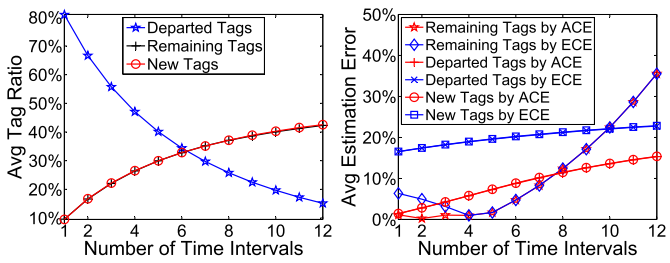
all tags. Each interval can construct an aforementioned binary tree, whose depth reduces to $24 - 12 = 12$. To retrieve the k minimum hash values, ACOS only applies the tree-based binary search to the smallest k intervals, and the expected time cost decreases to $k \cdot 12 \cdot (0.5 + 2 \cdot 0.5) = 18k$ time slots.

In Fig. 9, we plot the estimation error against the number of time slots per union tag, i.e., $18k/n_u$ for ACOS, or f'/n_u for ECE and ACE. The subfigure (a) shows that ACE is much more accurate than ACOS, no matter whether the union load factor ρ_u is configured to 0.5, 1 or 2 for ACE. This is because we encode each tag set efficiently into a slotted ALOHA frame, rather than by k smallest hash values. Similarly, the subfigure (b) shows that ECE is better than ACOS. By comparing (a) and (b), it is clear that ACE is more accurate than ECE, since ACE exploits the information buried in collision slots.

E. Discrete Event Simulations

For a group of tags, we simulate a series of tag arrival/departure events. We begin with 3000 tags and produce events of new tag arrivals and existing tag departures with a Poisson process. We collect the aggregate information of the tag set after each time interval. The rate of tag arrival/departure events is configured to 300 per interval. Hence, after the elapse of each time interval, 10% of existing tags will depart and a similar number of new tags will arrive. We estimate the numbers of new/remaining/departed tags from the beginning to the end of the first/second/third/... time interval. Namely, the measurement period is one/two/three/... time intervals long. Note that the tag ratios (as defined in Section III-B) will change after more time intervals elapse. As shown in Fig. 10(a), after one time interval, the departed/new tag ratios are 10%, but after 12 time intervals, the departed/new tag ratios increase to 42.5%, while the ratio of remaining tags drops from 80% to 15%. Because the departed/new tag ratios are always equal, the size of the tag population in the system stays the same as its initial value. However, the number of all tags involved during the process, including the new/remaining/departed tags, increases gradually from 3000 to 4275 (i.e., 3000 current tags added by $1275 = 3000 \times 42.5\%$ departed tags) during the 12 time intervals. The length of the ALOHA frame used in the simulation is set to 20000.

In Fig. 10(b), we present the churn estimation errors with respect to the measurement period in number of time intervals. Let's first focus on the performance of ACE. The plot shows that the estimation error of departed/new tags increases



(a) (b)

Fig. 10. Estimation accuracy against the number of time intervals elapsed. (a) Change of tag ratio. (b) Change of estimation error.

from 2% to 15% as we increase the measurement period from one time interval to 12 time intervals. In the meantime, the estimation error of remaining tags increases significantly from 1% to 36%. This is because the remaining tag ratio is reduced from 80% to 15% according to subfigure (a). The smaller the tag ratio is, the greater the difficulty we will encounter in providing high estimation accuracy. These simulation results show that estimation errors increase with the measurement period, which in turn means that the measurement period should be limited in practice if there exist specific accuracy requirements.

In Fig. 10(b), we also compare the churn estimation accuracies of ECE and ACE. The plot shows that the estimation error of ACE is much smaller than that of ECE, especially for departed/new tags, since ACE can take full advantage of tag collision information.

XII. Conclusion

This paper studies a problem called *RFID churn estimation*, which is to count the numbers of departed/remaining/new tags between two arbitrary time points, without the time-consuming collection of tag IDs. We have proposed three solutions for this problem: ECE estimator that observes the slot state transitions between empty and busy, CCE estimator that observes the slot state transitions between collision and non-collision, and ACE estimator that exploits the information buried in all possible state transitions between empty, singleton and collision. For the proposed churn estimators, we have investigated their bias and variance, by both simulation studies and theoretical analysis based on Cramér-Rao lower bound. We show that ACE can achieve the best accuracy among all estimators, and it can save protocol execution time by 35% as compared with ECE if they are to attain the same accuracy. Meanwhile, we discover that, although ECE and CCE are less accurate than ACE, they own the advantage of simplicity and can be combined to generate the good-quality initial guesses of churn numbers, which are required by the iterative optimization procedure of ACE.

Acknowledgement

The authors would like to thank the reviewers for their constructive comments. Any opinions, findings, conclusions and recommendations in the paper are those of the authors and do not necessarily reflect the views of funding agencies.

References

[1] B. Chen, Z. Zhou, and H. Yu, "Understanding RFID counting protocols," in *Proc. ACM MobiCom*, 2013, pp. 291–302.

[2] Q. Xiao, B. Xiao, and S. Chen, "Differential estimation in dynamic RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 295–299.
[3] S. He *et al.*, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1931–1942, Oct. 2013.
[4] Y. Zhang, S. He, and J. Chen, "Data gathering optimization by dynamic sensing and routing in rechargeable sensor networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1632–1646, Jun. 2016.
[5] X. Liu, S. Zhang, B. Xiao, and K. Bu, "Flexible and time-efficient tag scanning with handheld readers," *IEEE Trans. Mobile Comput.*, vol. 15, no. 4, pp. 840–852, Apr. 2016.
[6] X. Liu, B. Xiao, S. Zhang, and K. Bu, "Unknown tag identification in large RFID systems: An efficient and complete solution," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1775–1788, Jun. 2015.
[7] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MobiCom*, 2006, pp. 322–333.
[8] M. Kodialam, T. Nandagopal, and W. C. Lau, "Anonymous tracking using RFID tags," in *Proc. IEEE INFOCOM*, May 2007, pp. 1217–1225.
[9] C. Qian, H. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," in *Proc. IEEE PerCom*, Mar. 2008, pp. 30–39.
[10] H. Han *et al.*, "Counting RFID tags efficiently and anonymously," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[11] Y. Zheng, M. Li, and C. Qian, "PET: Probabilistic estimating tree for large-scale RFID estimation," in *Proc. IEEE ICDCS*, Jun. 2011, pp. 37–46.
[12] V. Shah-Mansouri and V. W. S. Wong, "Cardinality estimation in RFID systems with multiple readers," *IEEE Trans. Wireless Commun.*, vol. 10, no. 5, pp. 1458–1469, May 2011.
[13] M. Shahzad and A. X. Liu, "Every bit counts: Fast and scalable RFID estimation," in *Proc. ACM MobiCom*, 2012, pp. 365–376.
[14] Y. Zheng and M. Li, "ZOE: Fast cardinality estimation for large-scale RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 908–916.
[15] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. Pervasive*, Aug. 2002, pp. 98–113.
[16] N. Bhandari, A. Sahoo, and S. Iyer, "Intelligent query tree (IQT) protocol to improve RFID tag read efficiency," in *Proc. IEEE ICIT*, Dec. 2006, pp. 46–51.
[17] J. Myung and W. Lee, "Adaptive splitting protocols for RFID tag collision arbitration," in *Proc. ACM MobiHoc*, 2006, pp. 202–213.
[18] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. IEEE MOBIQUITOUS*, Jul. 2005, pp. 166–174.
[19] B. Sheng, Q. Li, and W. Mao, "Efficient continuous scanning in RFID systems," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[20] H. Liu, W. Gong, X. Miao, K. Liu, and W. He, "Towards adaptive continuous scanning in large-scale RFID systems," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 486–494.
[21] *EPC Radio-Frequency Identity Protocols C1 G2 UHF RFID Protocol for Communications at 860MHz–960MHz V1.2.0*, EPCglobal Inc., 2008.
[22] M. Buettner and D. Wetherall, "An empirical study of UHF RFID performance," in *Proc. ACM MobiCom*, 2008, pp. 223–234.
[23] I. Farris, A. Iera, L. Militano, and S. C. Spinella, "Performance evaluation of RFID tag-based 'virtual' communication channels," in *Proc. IEEE ICC*, Jun. 2014, pp. 2897–2902.
[24] W. Gong, K. Liu, X. Miao, and H. Li, "Arbitrarily accurate approximation scheme for large-scale RFID cardinality estimation," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 477–485.
[25] W. Luo, S. Chen, T. Li, and Y. Qiao, "Probabilistic missing-tag detection and energy-time tradeoff in large-scale RFID systems," in *Proc. ACM MobiHoc*, 2012, pp. 95–104.
[26] C. C. Tan, B. Sheng, and Q. Li, "How to monitor for missing RFID tags," in *Proc. IEEE ICDCS*, Jun. 2008, pp. 295–302.
[27] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. ACM MobiHoc*, 2010, pp. 1–10.
[28] Y. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast identification of the missing tags in a large RFID system," in *Proc. SECON*, Jun. 2011, pp. 278–286.
[29] L. Kang, K. Wu, J. Zhang, and H. Tan, "Decoding the collisions in RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 351–355.
[30] Y. Hou, J. Ou, Y. Zheng, and M. Li, "PLACE: Physical layer cardinality estimation for large-scale RFID systems," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 1957–1965.
[31] O. Bang, J. H. Choi, D. Lee, and H. Lee, "Efficient novel anti-collision protocols for passive RFID tags," Auto-ID Labs White Paper WP-HARDWARE-050, Mar. 2009.
[32] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Belmont, CA, USA: Duxbury Press, 2002.

Qingjun Xiao (A'10–M'12) received the B.Sc. degree from the Computer Science Department, Nanjing University of Posts and Telecommunications, China, in 2003, the M.Sc. degree from the Computer Science Department, Shanghai JiaoTong University, China, in 2007, and the Ph.D. degree from the Computer Science Department, The Hong Kong Polytechnic University, in 2011. He joined the Georgia State University and the University of Florida and worked as a Post-Doctoral Researcher. He is currently an Assistant Professor with Southeast University, China. His research interests include protocol and algorithm design in wireless sensor networks, RFID systems, and network traffic measurement. He is a member of the IEEE and ACM.

Bin Xiao (S'01–M'04–SM'11) received the B.Sc. and M.Sc. degrees in electronics engineering from Fudan University, China, and the Ph.D. degree in computer science from The University of Texas at Dallas, USA. He is currently an Associate Professor with the Department of Computing, The Hong Kong Polytechnic University. He joined the Department of Computing, The Hong Kong Polytechnic University, as an Assistant Professor. He has authored more than 100 technical papers in international journals and conferences. His research interests include distributed wireless systems, network security, and software-defined networks. He is an Associate Editor of the *Journal of Parallel and Distributed Computing* and *Security and Communication Networks*. He is a Senior Member of the IEEE and a member of ACM.

Shigang Chen (M'02–SM'12) received the B.S. degree in computer science from the University of Science and Technology of China, Hefei, China, in 1993, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana–Champaign, USA, in 1996 and 1999, respectively. He was with Cisco Systems, San Jose, CA, USA, for three years. He joined the University of Florida, Gainesville, FL, USA, in 2002. He is currently a Professor with the Department of Computer and Information Science and Engineering, University of Florida. He served on the Technical Advisory Board of Protego Networks from 2002 to 2003. He published more than 100 peer-reviewed journal/conference papers. He holds 11 U.S. patents. His research interests include computer networks, Internet security, wireless communications, and distributed computing.

Dr. Chen is an Associate Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING, *Computer Networks*, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He served on the Steering Committee of the IEEE IWQoS from 2010 to 2013. He received the IEEE Communications Society Best Tutorial Paper Award in 1999 and the NSF CAREER Award in 2007.

Jiming Chen (M'08–SM'11) received the B.Sc. and Ph.D. degrees in control science and engineering from Zhejiang University in 2000 and 2005, respectively. He was a Visiting Researcher with INRIA in 2006, the National University of Singapore in 2007, and the University of Waterloo from 2008 to 2010. He is a Full Professor with the Department of Control Science and Engineering, and the Co-Ordinator of Group of Networked Sensing and Control, State Key laboratory of Industrial Control Technology, Zhejiang University, China. His research interests are estimation and control over sensor network, sensor and actuator network, coverage, and optimization in sensor network.

Dr. Chen serves as a Co-Chair for Ad hoc and Sensor Network Symposium, IEEE Globecom 2011, the General Symposia Co-Chair of ACM IWCMC 2009 and 2010, the WiCON 2010 MAC track Co-Chair, Publicity Co-Chair for IEEE MASS 2011, IEEE DCOSS 2011, and IEEE ICDCS 2012, and TPC member for IEEE ICDCS 2010, IEEE MASS 2010, IEEE SECON 2011, IEEE INFOCOM 2011 and 2012, and IEEE ICDCS 2012. He currently serves as Associate Editor for several international journals including the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS. He is a Guest Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Computer Communication*, *Wireless Communication and Mobile Computer*, and the *Journal of Network and Computer Applications*. He is a Senior Member of the IEEE.