A Protocol for Simultaneously Estimating Moments and Popular Groups in a Multigroup RFID System

Qingjun Xiao, *Member, IEEE, ACM*, Shigang Chen, *Fellow, IEEE*, Jia Liu, *Member, IEEE*, Guang Cheng, *Senior Member, IEEE*, and Junzhou Luo, *Senior Member, IEEE, Member, ACM*

***Abstract*— Radio frequency identification (RFID) technology has rich applications in cyber-physical systems, such as warehouse management and supply chain control. Often in practice, tags are attached to objects belonging to different groups, which may be different product types/manufacturers in a warehouse or different book categories in a library. As RFID technology evolves from single-group to multiple-group systems, there arise several interesting problems. One of them is to identify the popular groups, whose numbers of tags are above a pre-defined threshold. Another is to estimate arbitrary moments of the group size distribution, such as sum, variance, and entropy for the sizes of all groups. In this paper, we consider a new problem which is to estimate all these statistical metrics simultaneously in a time-efficient manner without collecting any tag IDs. We solve this problem by a protocol named generic moment estimator (GME), which allows the tradeoff between estimation accuracy and time cost. According to the results of our theoretical analysis and simulation studies, this GME protocol is several times or even orders of magnitude more efficient than a baseline protocol that takes a random sample of tag groups to estimate each group size.**

***Index Terms*— RFID, multi-group tagged system, randomized algorithm, popular groups, moments of group size distribution.**

I. Introduction

RADIO-FREQUENCY identification (RFID) tags, each carrying a unique ID, are attached to physical objects and can be scanned by RFID readers from several meters away. In future, cheap battery-free tags may be pervasively embedded in or attached to objects in our daily living or working environment, to help realize the vision of Internet-of-Things. In the early period of RFID research, people pay more attention to individual tags. Tremendous research efforts have been devoted to identifying all tag IDs in the interrogation zone of one or multiple RFID readers as fast as possible, avoiding the signal collision among tags [1], [17]. The key reason is that RFID is traditionally a technique applied to the item-level asset management, for example, book management in a library [12], inventory management in a warehouse or a large retailer store. If some supposed-to-exist tags can no longer be found (may be stolen) or some tags are placed at wrong places, alerts should be triggered immediately, so that the managers of these facilities can trace back the stolen products and keep all products orderly placed.

Motivation: A recent trend is that RFID tags have been deployed in highly dynamic environments, where the tags are not owned by a single authority. For example, in a shipping port, RFID tags are used to identify the products owned by different exporter/importers. RFID tags may also be attached to car plates, so that the car registration numbers can be read by road-side equipment even when the lighting condition is inadequate. RFID tags may also be attached to cattle ears, bird wings, implanted under the skins of horses, or carried by humans, so that some agency can track their migration. In these scenarios, monitoring each individual tag and periodically collecting all tag IDs is time-consuming, as RFID systems work in low-rate channels. More importantly, in these scenarios, in which tag-carrying objects fast move in and out of a monitoring region (e.g., a tourist park, a metro station, a city center, or a shipping port), collecting tag IDs may violate the privacy of the tag owner as the moving trajectory of the tag can be tracked with fine details. Therefore, administrators of these places may shift their attention from item-level individual tags to the *aggregated statistics* of the tag set in the monitoring region.

The most well-known aggregated statistics is the number of tags, or called the *size* of a tag set, under the radio coverage of an RFID system. The prior research has explored novel statistical methods of estimating this metric, without having to collect any tag ID [4], [5], [7], [16], [25]. However, the total number of tags alone is coarse-grained and cannot provide abundant information about the tag population. Fortunately, in many application scenarios, a tag population can be naturally divided into different *groups* (i.e., non-overlapping subsets): Cars have different manufacturers, prices and mileage, and are registered in different states; Products belong to different categories, have different brands, production locations and expiration dates; Animals have different species, birth dates and genders; People have different source/destination stops in a subway system, or have visited different attractions in a theme park. The group ID of a tag will be embedded/recorded in its tag ID. We define the *size* of a group as the number of tags in the

Manuscript received December 7, 2017; revised August 31, 2018; accepted November 13, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor X.-Y. Li. Date of publication January 11, 2019; date of current version February 14, 2019. This work was supported in part by the National Key Research and Development Program of China under Grants 2017YFB1003000 and 2017YFB0801703, in part by the National Natural Science Foundation of China under Grants 61872080, 61502098, and 61602114, in part by the Jiangsu Provincial Natural Science Foundation of China under Grant BK20150629, in part by the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China under Grant 93K-9, and in part by the National Science Foundation of United States under Grant CNS-1718708. *(Corresponding author: Qingjun Xiao.)*

Q. Xiao and G. Cheng are with the Jiangsu Key Laboratory of Computer Networking Technology, School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: csqjxiao@seu.edu.cn; chengguang@seu.edu.cn).

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. Liu is with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: jialiu@nju.edu.cn).

J. Luo is with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: jluo@seu.edu.cn).

Digital Object Identifier 10.1109/TNET.2018.2884961

group. By carefully dividing a tag set into multiple groups and determining the size of each group, we can better understand a large tag population.

In a RFID system with a large number of groups, it is time-consuming and also unnecessary to determine the size of each group. We can certainly apply the tag set estimation protocols [4], [5], [7], [16], [25] to determine the size of one group at a time. However, while they are time-efficient for large tag groups, these protocols are in fact very inefficient for small tag groups [15]. Researchers have proposed protocols to identify popular groups whose sizes are beyond a pre-defined threshold or the top-k largest [13], [15], [18]. These protocols leave out potentially numerous non-popular groups in order to save execution time. Only knowing the group IDs and sizes of the popular groups may not be sufficient for administrating a large multi-group RFID system.

Our Problem: Besides the popular groups, we propose to also measure the moment statistics, which help in characterizing numerous non-popular groups as a whole. The first moment gives the sum of the sizes of all groups. From the number of groups, we can determine the mean group size. The second moment gives the sum of the size squares of all groups, from which we can derive the variance among groups. The entropy is a special type of moment (which will be formally given later). It is commonly used to characterize the diversity of a distribution and in our case the group distribution. With the knowledge of popular groups, we can easily remove them from the moment measurements and derive the mean group size, variance and entropy among non-popular groups. The moment measurements, together with the popular groups, provide useful information for management efficiency improvement, for example, by aligning the warehouse configuration and storage allocation according to the overall group-size distribution and the specifics of popular groups.

Consider a logistic distribution center where products from numerous vendors and manufacturers are moved in and out frequently. The center needs a simple, yet efficient way to monitor each type of products, which may be shoes of different brands, books from different publishers, construction materials for different builders, or even boxes of nails from different sources. Products in each type are naturally categorized into different groups, based on brands, publishers, builders and sources as in the above examples. It is useful to identify the popular groups (whose sizes pass a pre-defined threshold), and these are the big customers that the distribution center may want to know and make sure that they will stay in its business. For the remaining small groups, it may not be necessary to learn their detailed information, but some overall statistics will be helpful in storage management: What is the total number of products in these groups? Together with the number of groups, we will know the average number of products in each group. What are the overall characteristics of the group size distribution? They include the variance of group sizes and the entropy, which measures the diversity of the group-size distributions and provides the basis for identifying trend over time [2], [6].

Moment can be regarded as aggregated statistics for a multi-group RFID system, which include the sum, variance and entropy of the size distribution of all groups. Our goal is to find new ways to measure the different moments of group sizes, as well as the sizes of popular groups, without having to estimate the individual sizes of the potential numerous small groups. Instead of designing a separate protocol for each type of information as the prior work does, e.g., [13], [15], and [18] for popular groups and [4], [5], [7], [16], and [25] for the total number of products, we want to design a single protocol to simultaneously measure all the aforementioned information, including the popular groups, the variance and the entropy of group sizes, which the prior art has not investigated. An ideal protocol should push most complexity to the RFID reader while keeping the tasks of tags simple. This one-protocol-multi-purposes design is appealing in the context of RFID systems because tags are simple hardware with very limited resources which prohibit them from implementing many different protocols simultaneously. We also want to stress that our problem has other applications beyond the distribution-center example. For instance, a delivery company may want to automatically collect information at each of its local storage facilities to find out the popular sub-divisions (where more items are delivered) and statistics about the numerous other sub-divisions not having that many deliveries. This information will help the company align its delivery resources accordingly.

Our Solution: In this paper, we propose a protocol named GME (Generic Moment Estimator), which is time-efficient and is scalable to a large number of tag groups. This protocol can perform the moment estimation with accuracy preset to any desired level, allowing the tradeoff between accuracy and time cost. Our key technique is that, each time before performing the moment estimation for a sampled set of groups, we identify the popular groups among them. Since popular groups are large enough to occupy a significant portion of moment, we can use their size information to improve the accuracy of moment estimation.

We use an example to explain how GME protocol works in practice. Suppose in a warehouse with tens of thousands of tag groups, its manager wants to know the mean and variance of the sizes of all groups. To answer the query, a naive method is to run a tag cardinality estimation protocol to determine each group size. Clearly, this method will be very time-consuming, and to reduce the cost, an often used optimization is the group-level sampling that selects only a small fraction of groups to determine their sizes. With the known sizes of sampled groups, the moments of all groups can be derived. However, such moment estimations will be highly variant, not only because of the well-known sampling error, but also due to the existence of popular groups whose sizes are much larger than the rest. The random events whether the popular groups are sampled will cause the moment estimation results to fluctuate a lot. To tame the sampling error of popular groups, we propose to firstly identify them and exclude them from the group-level sampling process, which can appreciably improve estimation quality.

Although a popular group identification protocol can help improve the accuracy of moment estimation, we also discover that its time cost skyrockets exponentially as the threshold of popular groups reduces. Thus, we can not count on the setting of a ultra small threshold to attain satisfactory moment estimation accuracy, which will have prohibitively high time cost. This motivates us to construct multiple group-sampling layers with their sampling probabilities reducing exponentially. Then, the set of sampled groups on a layer is always a subset of the sampled groups on its previous layer, such that the moment estimation of a layer is a sub-problem of the moment estimation of its previous layer. In order to improve moment estimation accuracy, among the set of sampled groups on each

layer, we will identify the popular groups whose sizes exceed a properly configured threshold.

The main contributions of our paper are as follows.

- We design a time-efficient protocol named GME (Generic Moment Estimator) with multiple sampling layers, to estimate arbitrary moments for a multigroup RFID system.
- We theoretically analyze the estimation accuracy of GME protocol, and study how to properly configure its protocol parameters to attain desired accuracy of moment estimation.
- We introduce a protocol named TBC (threshold-based classification) for identifying popular groups, and analyze how to configure its parameters to satisfy accuracy constraints.

The rest of the paper is organized as follows. Section II discusses the related work. Section III presents the system model and research problem. Section IV proposes a moment estimation protocol named GME, and analyzes the accuracy of GME protocol for estimating moments. To make our paper self-contained, Section V introduces a TBC protocol for identifying popular groups. Section VI evaluates our protocols by simulations. Section VII concludes the paper.

II. Related Work

For RFID systems, an important application is to use an RFID reader to remotely collect the IDs of a group of tags in its radio range, which is called the *tag identification* problem. Since the tags communicate with a reader through wireless medium, inevitably collisions will occur when multiple tags respond to the same reader simultaneously. Collision arbitration protocols mainly fall into two categories, i.e., tree-based protocols [17], and framed slotted ALOHA protocols [1]. EPCglobal C1G2 protocol, as de-facto industrial standard, is a variant of the slotted ALOHA protocol [1]. Its idea is to construct an ALOHA frame with multiple time slots and distribute tags uniformly in the frame, in order to reduce the chance for two tags to pick a same slot and have signal collision.

In certain application scenarios, collecting tag IDs is not required, and it is also very time consuming to collect all tag IDs in a large-scale RFID system. Hence, another branch of RFID research considers the *cardinality estimation* problem, which is to efficiently estimate the total number of tags without ID collection. A plethora of protocols have been developed, such as UPE [7], LoF [16], FNEB [5], PET [25], and SRC [4].

Recent RFID research began to consider the extended scenario of multiple readers distributed at different locations, which is commonly seen in warehouses or logistic supply chains. Some researchers study the *multi-reader scheduling* protocols, in order to mitigate the radio collision among readers [20]. Several other works focus on the *joint cardinality estimation* problem, which is to count the number of tags moved from the coverage area of one reader to another [22]. Some other works study the *multiset joint estimation*, which counts the number of tags in an arbitrary set expression that connects multiple tag sets at different places by the operators of set union, intersection and complement [9], [21]. Researchers also investigate the time-efficient *monitoring of missing tags* by comparing the tag sets at two different time points [8], [19].

Recently, researchers also consider a complex RFID system deployed in a large warehouse with multiple groups of tags. A recent paper studies the *histogram collection* problem, which is to efficiently estimate the size of each group without any piori knowledge of group IDs [24]. Another interesting problem is *popular group identification* — among a large number of groups with known IDs, identify the popular groups whose sizes are above a predefined threshold [15], [18]. Another work studies the *top-k group identification* problem, which identifies both the k-largest groups and the k-smallest groups [13].

Although a lot of previous work exists solving the problem of popular group identification [13], [15], [18], this problem is totally different from our moment estimation problem. Its objective is to determine the IDs and sizes of the popular groups (whose sizes pass a pre-defined threshold). The individual sizes of the potential numerous small groups are left unknown. With only the sizes of popular groups, it is impossible to quickly calculate the aggregated information of group size distribution, such as sum, entropy and variances.

As far as we know, this paper is probably the first to investigate the problem of *moment estimation* in RFID domain, which estimates an arbitrary moment for a multigroup RFID system at low time cost. The moment can be regarded as statistical aggregated information of the group size distribution, and can be used to detect the abnormal change in a RFID system or perform trend analysis. This paper points out that we do not need to determine the size of each group in order to estimate the moment, and we can exploit the result of popular group identification to improve the accuracy of moment estimation.

III. Problem and System Model

In this section, we introduce the RFID system model with multiple groups of tags, and then formulate the two problems of moment estimation and popular group identification.

A. System Model

To be compliant with EPC C1G2 [1], we assume the communication between a reader and its nearby tags adopts slotted ALOHA protocol: A reader broadcasts a query command to start an ALOHA frame with a number of time slots. When receiving the command, each tag randomly picks a time slot to send its response. In a time slot of the frame, the reader may receive multiple tag replies, which is called tag collision.

In this paper, we assume that each tag makes only a short response in a time slot (for example, using the 16-bit RN16 command as specified in EPC C1G2 [1]). The reader can detect whether the time slot is busy, by sensing whether it is occupied by any tag responses. Then, from the reader's perspective, the state of a time slot can be represented by a bit, i.e., using a '1' bit to record a busy slot and a '0' bit to record an empty slot. Thus, an ALOHA frame with multiple time slots can be represented as a bit vector.

Due to the limited range of RFID reader (typically less than ten meters when scanning commercial battery-less tags), it is impossible to use a single reader to cover a large region, like a warehouse. Multiple readers are often deployed to attain the proper coverage. In the scenario of dense reader deployment, these readers may take turns to transmit request to avoid interference, or a more sophisticated scheduling algorithm may be used to allow readers that do not interfere to transmit simultaneously [20]. Thanks to reader scheduling, each reader can work independently without interferences. The scanning result of each reader about its surrounding tags can be represented by a bit vector. By bitwise ORing the scanning

results of all readers, we can construct a snapshot of the tags in the entire warehouse. From this perspective, the multiple readers can be treated as one big 'virtual' reader that monitors the warehouse.

Consider a warehouse with tens of thousands of product items. An RFID tag is attached to each item, for communication with an RFID reader (can be either physical or virtual) deployed in the warehouse. The tags are divided into different groups based on certain properties, e.g., product type, brand, manufacturer and production date/place. To support grouping, each tag ID has two components: a group identifier (gid) and an item identifier (iid). By concatenating the two identifiers, a tag's complete ID can be obtained ($id = gid|iid$). Clearly, all tags in a group must carry the same group ID, while tags in different groups will carry different group IDs.

We assume the RFID reader knows all the group IDs in the system. When such knowledge is unavailable, we can collect the group ID information by running a group identification protocol [10], [11]. The protocol [10] is time efficient because it exploits the fact that all tags in a same group have the same group ID embedded in their tag IDs. It does not need to interrogate each tag. One tag's response in a group will suffice. After the collection of all group IDs, the knowledge will not become outdated unless sufficiently long time passes.

We formalize the multi-group RFID system model. Suppose there are m groups. Without loss of generality, we assume the group IDs are sorted and relabelled from 1 to m. We define the size of a group as the number of tags in the group. Let n_i be the size of the ith group. From the reader's perspective, a product inventory with multiple groups of tags can be specified by a group size distribution vector $N = \langle n_1, n_2, \ldots, n_m \rangle$. Let n be the total number of tags. Clearly, we have $n = \sum_{1 \le i \le m} n_i$. Note that in our paper, the distribution vector N has not been normalized for the simplicity of presentation. When used in practice, the vector will be divided by the number of tags n.

B. Metric Definition

For a product inventory, we consider to measure two types of aggregate statistical information — moments and popular groups, which can assist the management of the inventory.

Moment: The xth-order moment is defined as

$$L_x = \sum\nolimits_{1 \le i \le m} {n_i}^x. \tag{1}$$

Three typical kinds of xth-order moments are as follows.

- L_0 is the zero-order moment when $x = 0$. It is equal to the number of groups in the RFID system, and $L_0 = m$.
- L_1 is the first-order moment, which is equal to the total number of tags for all m groups combined. Thus, $L_1 = n$.
- L_2 is the second-order moment. It is also called surprise number, which can help calculate the variance and measure how uneven the group size distribution vector N is.

The notion of moment can be extended to the sum of vector N, after each of its entry has been applied with a function g:

$$F_g = \sum\nolimits_{1 \le i \le m} g(n_i), \tag{2}$$

where g could be any monotonic function bounded by $O({n_i}^2)$. For example, if g is $g(n_i) = n_i \log n_i$, then F_g is the entropy of group size distribution N; if $g(n_i) = n_i \log_2 n_i$, then F_g is the Shannon entropy of group size distribution N, which has been used to measure the diversity of an inventory [2], [6].

The aim of moment metrics is to quantify different attributes of a group size distribution $N = \langle n_1, n_2, \ldots, n_m \rangle$. It can be regarded as an aggregated statistics which is more generic than the total number of tags n. The first-order moment is the sum of sizes of all groups, which equals n. The second-order moment is used to quantify the variance of vector N. The entropy moment is to model the diversity of distribution vector N. There exist previous works that seek to determine the entire distribution vector N, called histogram collection problem [24], which however incurs much higher communication cost between tags and readers. Our paper only estimates the aggregate statistics of the vector N to save protocol running time.

We illustrate the meanings of moments by an example. Imagine a warehouse with $m + 1$ tag groups whose size distribution is a vector $\langle m, 1, 1, \ldots, 1 \rangle$. It tells that the first group is a popular group whose size is m, and the sizes of other groups are all 1s. Then, using the function $g(n_i) = n_i$, the entropy is calculated as $m \log m + m$, and using the function $g(n_i) = {n_i}^2$, the second-order moment is calculated as $m^2 + m$.

Tags' moving in/out of a monitoring region will change the group size distribution vector N, which in turn will affect the moment metrics. From the spatial perspective, different types of moment metrics reflect how the missing/newly-arrived tags distribute among different groups. If the group size changes are concentrated in the popular groups, the second-order moment can better reflect the change, since it greatly amplifies the contribution of popular groups. If the group size changes are dispersed to many different groups, the entropy is a better metric to reflect the change, since it is commonly used to measure the diversity of a multi-group population. For example, suppose m new tags move into a warehouse whose group size vector is $\langle m, 1, 1, \ldots, 1 \rangle$. If the m new tags are all in the first popular group, the *relative change* of the entropy is $\frac{2m \log(2m) - m \log m}{m \log m + m} = \frac{\log m + 2 \log 2}{\log m + 1}$. The relative change of second-order moment is $\frac{3\ m^2}{m^2 + m} = \frac{3m}{m+1}$, which is much larger than the entropy. If the m new tags are evenly spread over the m non-popular groups, doubling the size of each non-popular group, the entropy grows by $\frac{(2 \log 2 - 1)m}{m \log m + m} = \frac{2 \log 2 - 1}{\log m + 1}$, and the second-order moment grows by $\frac{3m}{m^2 + m} = \frac{3}{m+1}$. In this case, entropy can better reflect the change.

From the temporal perspective, a warehouse administrator can accumulate the moment statistical data over time, which is useful for analyzing the short-term or long-term change of the warehouse inventory. The change rate of a group size distribution vector N can be quantified by the relative growth rate (RGR) of a moment metric (i.e., $\frac{\text{relative change of a moment}}{\text{time passed}}$). If N changes quickly in short term, then the RGR will exceeds a threshold for some type of moment, which triggers an alert to report the change. If the distribution vector N evolves very slowly, the moment metrics can form time series, which provide the basis for identifying the long-term trend [2], [6].

Popular Groups: Generally speaking, a group i is a popular group with respect to function g, if the fluctuation of its size n_i strongly affects the moment F_g. More formally, the popular group is defined as any group ID i whose size n_i satisfies

$$g(n_i) \ge \alpha F_g, \tag{3}$$

where α is a pre-defined ratio which is between zero and one. The information about popular groups is quite useful, since when we detect a rapid change of moment values, we may find

its root cause to be the size change of one or several popular groups. Let H_g be the set of all popular groups. Then,

$$H_g = \{i | g(n_i) \geq \alpha F_g\}. \quad (4)$$

Typically, the number of popular groups $|H_g|$ is a small value as compared with the number of groups m, i.e., $|H_g| \ll m$.

Note that, for H_g in equation (4), the number of popular groups $|H_g|$ is not a fixed value but a variable that may change with F_g. Hence, our popular group definition in (4) is different from the top-k group identification problem [13], which finds exactly k largest groups. Our popular group definition is also different from [15] and [18], which considers only the L_1-popular groups satisfying $n_i \geq \alpha n$. What we study is the identification of F_g-popular groups satisfying $g(n_i) \geq \alpha F_g$, for an arbitrary monotonic function g bounded by $O({n_i}^2)$.

C. Metric Estimation

For a warehouse inventory with a large number of tag groups, it is time-consuming and also unnecessary to determine the exact values for moment F_g and size of each popular group n_i, $i \in H_g$. In many applications, we only need to collect their approximated values. Hence, in following, we define the approximation models for popular groups and moments.

Popular Group Identification Problem: For the set of popular groups H_g defined in (4), let $\hat{H}_g$ be its estimation, or call it the set of reported popular groups. The probability for $\hat{H}_g$ to include all the actual popular groups H_g must be at least $1-\delta$.

$$Pr\{H_g \subseteq \hat{H}_g\} \geq 1 - \delta \quad (5)$$

For each reported popular group i in $\hat{H}_g$, we need to obtain an estimation $\hat{n_i}$ of its group size n_i, and its relative estimation error $\frac{g(\hat{n_i}) - g(n_i)}{g(n_i)}$ must be bounded by $\pm\epsilon$ at a probability $1-\delta$.

$$\forall i \in \hat{H}_g, \quad Pr\big\{|g(\hat{n_i}) - g(n_i)| \leq \epsilon g(n_i)\big\} \geq 1 - \delta \quad (6)$$

Moment Estimation Problem: For the moment F_g defined in (2), let $\hat{F}_g$ be its estimated value, and its estimation error $\frac{\hat{F}_g - F_g}{F_g}$ must be bounded by $\pm\gamma$ at a probability of at least $1-\eta$.

$$Pr\big\{|\hat{F}_g - F_g| \leq \gamma F_g\big\} \geq 1 - \eta \quad (7)$$

Our objective is to design a protocol to address these two problems, and meanwhile minimize the communication time cost. Also, we require that the protocol for popular group identification and the protocol for moment estimation are not totally separated. Otherwise, due to the separation, the overall time cost will be increased by multiple folds, for example, using one protocol for entropy estimation, two different protocols for estimating L_1/L_2 moments, and another two protocols for identifying L_1/L_2 popular groups. It would be the best if we could address all these problems simultaneously, by a generic protocol that can run once and later answer an arbitrary query for moments or popular groups with any defined function g.

IV. Moment Estimation Protocol

This section presents a protocol named GME (generic moment estimator) to estimate the moment F_g for any monotonic function g bounded by $O({n_i}^2)$. This protocol relies on the assumption that there is a popular group identification protocol to report the groups whose sizes are above a threshold, and its estimation accuracy must satisfy the constraints (5) and (6). In the next section, we will describe such a protocol named TBC (threshold-based classification) for our paper's completeness.

A. Basic Idea

Clearly, when the number of groups in a warehouse is small, their moment can be estimated at low time cost, by running a cardinality estimation protocol [4], [5], [7], [16], [25] to determine the size of each group. In this paper, we use the SRC protocol [4] by default for estimating the size of a single group.

However, this method will become time-consuming when the number of groups in a product inventory is very large. Especially, according to EPC C1G2 standard [1], there is a considerable inter-cycle overhead between any two cycles that query different tag groups. The inter-cycle overhead consists of the time between cycles when the reader is powered down, and the continuous RF wave transmission time used to power up the tags before beginning real data communication. These times are typically 40ms and 3ms, by the empirical results in [3], while the average time interval per slot is about 1$\sim$2ms. So after the transmission of each query cycle, there is a 40ms reader power-down interval. If the powered-down interval is not long enough, it is possible that some tags will maintain their former state with local residual power, which may cause them to behave unpredictably in the upcoming query cycle. In a word, a protocol that estimates the size of each group will be very time-consuming, as each tag group needs a separate query cycle and 43ms inter-cycle overhead.

To reduce the high time cost of determining the size of each group, a straightforward optimization is group-level sampling in order to significantly reduce the number of groups that need to be scanned. However, this random sampling method will have poor accuracy when the sampling probability is configured too small (which regretfully has to be small since there are often a large number of product groups in a warehouse). The situation will grow even worse if the popular groups participate the sampling process. Since the sizes of popular groups occupy a significant fraction of moment, whether they are sampled will strongly affect the moment estimation result, making it fluctuate a lot. Later, in Fig. 5 of Sec. VI-C, we will use simulation results to better illustrate such a phenomenon.

We will mitigate the random sampling error by two methods: configure sampling probability p larger than a threshold (e.g., 30%), and intentionally keep popular groups away from random sampling process. In particular, we divide all groups into popular groups and non-popular ones, by running a TBC protocol, which will be introduced in Section V. The TBC protocol is very time-efficient for two reasons: it uses only one query cycle to scan tags in all groups, which can amortize the 43ms inter-cycle overhead, and it sacrifices the estimation accuracy of non-popular groups to preserve the accuracy of popular groups. Hence, the popular groups, whose sizes have been estimated at low cost, can directly participate the moment calculation, while the non-popular groups will be randomly sampled at probability p to take part in moment estimation.

However, there is another problem since the time cost of TBC protocol increases rapidly as the pre-defined threshold of popular groups decreases (later check Figure 11(b) for experimental verification). Thus, the threshold can not be too small, causing the problem that it is impossible to locate a very large number of popular groups and keep them away from the sampling process. The remaining non-popular groups will be numerous, even after the group sampling. We address this

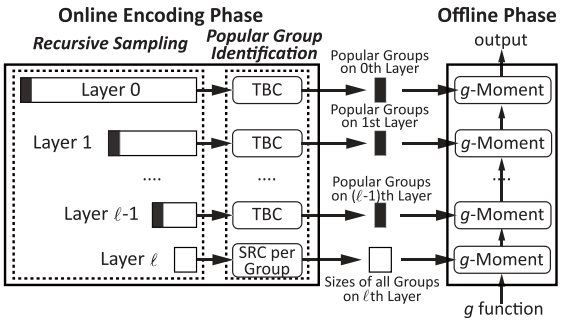

Fig. 1. Architecture of moment estimation protocol named GME.

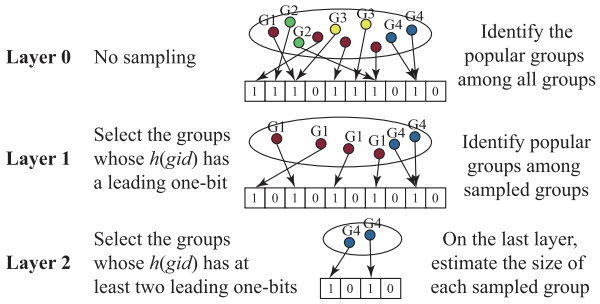

Fig. 2. An example of GME's online phase when $\ell = 2$ and $p = 0.5$.

problem by a recursive sampling technique: We construct multiple layers labeled by $0, 1, 2, \ldots, \ell$, as shown in Figure 1. The sampling probabilities of these layers reduce exponentially as $1, p, p^2, \ldots, p^\ell$, such that the sampled groups on each layer is a subset of the sampled groups on its immediate higher layer.

On the lowest (i.e., ℓth) layer, the number of sampled groups becomes small enough that it is time-permitting to estimate the size of each group by a tag cardinality estimation protocol, e.g., SRC [4]. Hence, the moment of sampled groups on the ℓth layer can be calculated directly. With such information, we can further derive the moment of $(\ell-1)$th layer, and its estimation accuracy can be greatly improved if we also leverage the popular groups on the $(\ell-1)$th layer. By applying the similar technique recursively, we can estimate the moment of each layer from the ℓth layer up to the 0th layer.

B. Detailed Protocol Design

Our moment estimation protocol is composed of two phases, as shown in Figure 1. In the online phase, our protocol scans the sampled groups on each layer and encodes them into a set of popular groups whose IDs and sizes are known. In the offline phase, we use the encoded information to estimate the g-moment of each layer, from the lowest to the highest layers.

Online Phase: Suppose there are $\ell+1$ layers whose indexes range from 0 to ℓ. For these layers, their group-level sampling probabilities reduce exponentially: on the 0th layer, sampling probability is 1; on the jth layer, sampling probability is p^j.

To attain this effect, on each jth layer (except the 0th layer), the RFID reader broadcasts a SELECT command (see the EPC C1G2 standard [1]) to let each tag invoke a boolean hash function $\rho_j(gid)$, which maps its group ID to one/zero with probability p and $1-p$, respectively. A tag group is sampled on the jth layer, when its boolean hash function outputs one not only on the current jth layer but also on all the previous layers. It can be expressed formally as $\prod_{1\le i\le j}\rho_i(gid) = 1$, and may be implemented by issuing j successive SELECT commands, each of which triggers a hash function call $\rho_i(gid)$.

A more simplified implementation of group sampling exists if the sampling probability p equals 0.5. When the protocol starts on each jth layer with $0 < j \le \ell$, the reader broadcasts only one SELECT command. When receiving the command, each RFID tag uses its group ID as a parameter to generate a hash value $\rho(gid)$. A tag is sampled on the jth layer, $0 \le j \le \ell$, if the initial j bits of its generated hash value are all ones.

An example of running our protocol is given in Fig. 2 assuming $\ell = 2$. On the 0th layer, there is no sampling. Thus, all the four groups $G1, G2, G3, G4$ respond on this layer. On the 1st layer, two groups are sampled, i.e., $G1, G4$, whose generated hash value $h(gid)$ has at least a leading one-bit. On the 2nd layer, one group is sampled, i.e., $G4$, whose generated hash value $h(gid)$ has at least two leading one-bits. On each jth layer, $0 \le j < \ell$, after the group-level sampling, we divide the sampled groups into popular groups and non-popular ones, by a TBC protocol. As shown in Fig. 2, TBC attains this goal using just one ALOHA frame, which will be elaborated in the next section. On the last layer, the number of sampled groups becomes sufficiently small that it is time-permitting to determine the size of each group. Note that, since the set of group IDs is prior knowledge and the group-sampling hash function is pre-installed, we know in advance which groups are sampled on the ℓth layer. We can use the SELECT command to activate them one by one, and then estimate the size of each group by running the SRC protocol [4] per group.

We introduce the notations used by this online phase. Let S_j be the sampled groups on the jth layer. Then, we have

$$S_0 = \{1, 2, \ldots, m\}, \quad S_j = \{i | i \in S_{j-1} \wedge \rho_j(i) = 1\}. \tag{8}$$

On each jth layer, we run the TBC protocol to identify the popular groups among the sampled groups S_j. Let H_j be the set of actual popular groups on the jth layer:

$$H_j = \{i | i \in S_j \wedge g(n_i) \ge \alpha F_j\}, \tag{9}$$

where F_j is the moment of sampled groups on the jth layer:

$$F_j = \sum\nolimits_{i\in S_j} g(n_i). \tag{10}$$

Let $\hat{H}_j$ be the set of identified popular groups on the jth layer.

When the online phase completes, on a jth layer with $0 \le j < \ell$, by running the TBC protocol, we obtain a set of popular groups $\hat{H}_j$. Let $\hat{n_i}^j$ be the estimated size of the ith popular group with $i \in \hat{H}_j$. On the ℓth layer, the set of sampled groups is S_ℓ, and we know the size of each group by running SRC per group. Let $\hat{n_i}^\ell$ be the estimated size of ith group, $i \in S_\ell$.

Offline Phase: In this phase, we will estimate the moment F_j of sampled groups on each jth layer with $0 \le j \le \ell$: The moment F_ℓ of the ℓth layer can be directly calculated as

$$\hat{F}_\ell = \sum\nolimits_{i\in S_\ell} g(\hat{n_i}^\ell). \tag{11}$$

Assume the moment F_{j+1} of $(j+1)$th layer has been estimated as $\hat{F}_{j+1}$. Combining it with the identified popular groups $\hat{H}_j$ on the jth layer, we can estimate the moment of jth layer as

$$\begin{aligned}\hat{F}_j &= \sum\nolimits_{i\in\hat{H}_j} g(\hat{n_i}^j) + \frac{1}{p}\big(\hat{F}_{j+1} - \sum\nolimits_{i\in\hat{H}_j}\rho_{j+1}(i)g(\hat{n_i}^j)\big)\\ &= \frac{1}{p}\hat{F}_{j+1} + \sum\nolimits_{i\in\hat{H}_j}\big(1 - \frac{1}{p}\rho_{j+1}(i)\big)g(\hat{n_i}^j),\end{aligned} \tag{12}$$

where $\hat{n_i}^j$ is the estimated size of ith popular group by running TBC protocol on jth layer. By applying (12) recursively, we can obtain $\hat{F}_0$, as an estimation of the moment of the 0th layer. Since the sampling probability of this layer is one, we use $\hat{F}_0$ as an estimation of the moment of all groups.

In following, we briefly explain the basic idea of (12). The actual moment of jth layer defined in (10) can be rewritten as

$$F_j = \sum\nolimits_{i\in\hat{H}_j} g(n_i) + \sum\nolimits_{i\in S_j\setminus\hat{H}_j} g(n_i).$$

Hence, the moment F_j is out of the contributions of both popular groups $\hat{H}_j$ and non-popular groups $S_j \setminus \hat{H}_j$. Clearly, the contribution of popular groups can be estimated by the first term $\sum_{i\in\hat{H}_j} g(\hat{n_i}^j)$ in (12). To estimate the contribution of non-popular groups $S_j \setminus \hat{H}_j$, we must leverage the moment estimation $\hat{F}_{j+1}$ of $(j+1)$th layer, where the sampled groups S_j are recursively sampled with probability p. We may use $\frac{1}{p}\hat{F}_{j+1}$ to estimate the contribution of non-popular groups. However, such an estimation is biased, since $\hat{F}_{j+1}$ not only include the contribution of non-popular groups $S_j\setminus\hat{H}_j$ (further sampled on $(j+1)$th layer with probability p), but may also include the contribution of popular groups $\hat{H}_j$. Equation (12) can compensate the effect of these sampled popular groups by deducting $\sum_{i\in\hat{H}_j}\rho_{j+1}(i)g(\hat{n_i}^j)$ from $\hat{F}_{j+1}$, since a popular group $i\in\hat{H}_j$ is sampled on $(j+1)$th layer when $\rho_{j+1}(i)=1$.[1]

C. Protocol Analysis and Parameter Setting

To satisfy the constraint in (7) for moment estimation accuracy, we must properly configure the protocol parameters, including the sampling probability p, the number of layers ℓ, and the (ϵ,δ)-accuracy constraint of TBC protocol, which is used to identify the α-fraction popular groups on each layer. Hence, GME protocol has five parameters p, ℓ, ϵ, δ and α.

There is a large design space for optimizing the parameter settings. Firstly, we present the following theorem.

Theorem 1 (Moment Estimation Accuracy): For any given threshold of moment estimation error $\gamma = 2\theta\ell\epsilon$, GME protocol ensures that the probability for relative estimation error $\frac{\hat{F}_0-F_0}{F_0}$ of moment F_0 to exceed the threshold $\pm\gamma$ is upper bounded:

$$Pr\{|\hat{F}_0 - F_0| \geq 2\theta\ell\epsilon F_0\} \leq \frac{1-p}{p}\ell\alpha/\epsilon^2 + (2\ell+1)\delta + \frac{1}{\theta}, \tag{13}$$

where θ is a tunable constant, ℓ is the number of recursive sampling layers configured for GME protocol, α is the popular group threshold in (4) configured for underlying popular group identification protocol, and (ϵ,δ) is the accuracy provided by popular group identification protocol as shown in (5) and (6).

Proof: See Appendix A for detailed proof. ∎

Secondly, we discuss the settings of protocol parameters. The parameter α is an important parameter for a popular group identification protocol named TBC: If the popular group threshold α has been configured to a too small value, then the time cost of TBC will skyrocket. We assume a proper value of α is known a priori for a particular group size distribution, so that the time cost of TBC is smaller than a preset threshold. Later in Section VI-E, we will evaluate the impact of α.

Choosing the number of layers ℓ is not difficult. The time cost of GME is the sum of the cost of running SRC protocol per group on ℓth layer and the cost of running TBC protocol for each jth layer, $0\leq j<\ell$. As ℓ grows, the time cost of running TBC protocol on extra layers will gradually neutralize the accuracy gain by increasing ℓ. Later in Section VI-E, we will provide simulation result on the impact of ℓ. Here, we assume that an appropriate value of ℓ is already known.

Next, we study the setting of the accuracy parameters ϵ and δ of TBC protocol (and also SRC protocol) used by GME, which strongly affect the accuracy of moment estimation. Note that ϵ and δ jointly controls one parameter, i.e., expected standard deviation $\epsilon/\Phi^{-1}(1-\frac{\delta}{2})$ of TBC, where Φ is cumulative distribution function (CDF) of standard normal distribution. By default, we set δ to 0.05, which makes $\Phi^{-1}(1-\frac{\delta}{2})=2$.

We discover that error bound ϵ and sampling probability p are the two most important parameters for our GME protocol. According to Theorem 1, we know that, for any small bound γ of moment estimation error, we can theoretically guarantee

$$Pr\{|\hat{F}_0 - F_0| \geq \gamma F_0\} \leq (2\ell+1)\delta + \frac{2}{\theta},$$

by choosing the following parameters:

$$\epsilon = \frac{\gamma}{2\theta\ell}, \quad p = 1/(1+\frac{\gamma^2}{\alpha 4\theta^3\ell^3}),$$

where θ is a predefined constant. However, since our analysis result of moment estimation accuracy in Theorem 1 is quite conservative, the above configurations of ϵ should be treated as a lower bound of ϵ, and the above configuration of p should be treated as an upper bound of p. For a particular group size distribution in practice, ϵ and p could be configured with more economic values to satisfy the (γ,η) accuracy constraint in (7). We provide simulation result of their impact in Section VI-E.

V. Popular Group Identification

In this section, we introduce TBC (Threshold-Based Classification) protocol to identify popular groups in a multigroup RFID system, and meanwhile it can satisfy the accuracy constraints in (5) and (6) at low time cost.

There exist several previous papers that have solved the popular group identification problem, such as TCS [18], TBC [15] and TKQ [13]. The reason for us to choose TBC is as follows. TBC outperforms TCS, particularly when the number of popular groups increases [15]. TBC is more scalable to a large number of popular groups than TCS, because TCS measures the size of one group at a time, while TBC adopts a different protocol design that measures the sizes of all groups together in one common ALOHA frame. The problem solved by TKQ [13] is different from the previous two protocols. It is to identify both the top-k largest groups and the top-k smallest groups. It has to add extra complexity and overhead to protocol design for identifying top-k smallest groups, which is not needed by us.

Although the TBC protocol has been proposed by Luo *et al.* [15], it is designed for identifying popular groups whose sizes pass a fixed threshold v, i.e., $n_i\geq v$. We need to adapt this protocol to fit our definition of popular groups in (4), i.e., $g(n_i)\geq\alpha F_g$, so that it can be used by our GME protocol to identify the needed popular groups on each layer.

[1]The probability for an identified popular group $i\in\hat{H}_j$ to be sampled on the next layer (i.e., $\rho_{j+1}(i)=1$) can be minimized by carefully choosing the random seed used by hash function ρ_{j+1}. By our simulation results, such an optimization can reduce L_2-moment estimation error by about 20%.

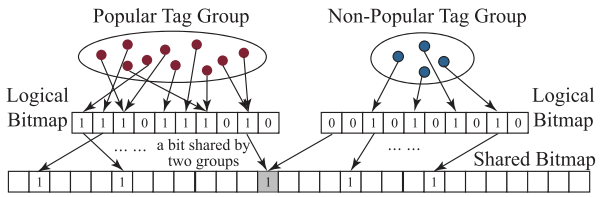

Fig. 3. Popular group identification protocol named TBC.

A. Basic Idea

We introduce the design rationale behind TBC protocol. Let us consider a simple RFID system with only one tag group. The state-of-the-art work to efficiently estimate its group size is SRC protocol [4], which has two phases. In the first phase, it generates a rough estimation of group size by LoF [16] or PET [25]. Thanks to such coarse knowledge, when RFID reader rescans the tag group by slotted ALOHA [7] in the second phase, it can control the length of the ALOHA frame to be proportional to the group size. Then, by listening to RF channel, the reader converts the slotted frame into *a bit array (or called a bitmap)* [7]. From the fraction of zero bits in the bitmap, the reader can accurately estimate the group size.

As the RFID system evolves from a single group of tags to multiple groups, there arises a need to identify popular groups. As stated before, when there are a large number of groups, it is time-consuming to determine the size of each group, even using SRC protocol. So instead of encoding each group into a separate bitmap, Luo *et al.* [15] proposed a TBC protocol to encode all groups into *a shared bitmap*. To greatly reduce the length of this shared bitmap (which is the time cost of TBC protocol), the estimation accuracy of the sizes of small tag groups is sacrificed. Meanwhile, the estimation accuracy of the popular groups is still satisfactory.

To implement the above effect, the TBC protocol [15] maps each group ID pseudo-randomly to s time slots in the shared time frame. For a particular group ID, as shown in Fig. 3, the bits interpreted from its mapped slots form a *logical bitmap*. Each tag in the group will randomly pick one of the slots (or bits) in the logical bitmap to transmit, which sets the bit to one. In Fig. 3, since a popular group has a larger number of tags, its logical bitmap has a greater proportion of bits assigned to ones than the logical bitmap of a non-popular group. Hence, intuitively from the number of bits in a logical bitmap that remain zeros, we may estimate the number of tags in a group.

However, such an estimation will be positively biased, since the logical bitmap of a group is not exclusively owned by the group. As in Fig. 3, there is a lower layer of hash mapping which projects the bits in the logical bitmap of a group to the shared bitmap. Thus, a bit in the shared bitmap (e.g., the gray-colored bit in Fig. 3) may be used by more than one groups. This sharing introduce *noise*: The logical bitmap of a group may carry some bits that are set to '1' not by transmission of tags in this group, but by transmission of tags from other groups that happen to be hashed to the same time slots. Fortunately, from a bird's eye view, all slots are shared by all groups uniformly at random, which means the noise uniformly distributes in the entire time frame. Therefore, we can easily measure the noise, and then subtract the noise from a group's size estimation.

B. Detailed Protocol Design and Notations

The TBC protocol is composed of two phases: an online phase for encoding all tag groups into a shared bitmap, and an offline estimation phase for recovering the size information of a group from its logical bitmap and reporting the group as a popular one if its estimated size exceeds the threshold in (18). For simplicity, our description of TBC protocol below is in the context that all groups participate for the identification of popular groups. It can be easily modified to fit the scenario that only a subset of groups are sampled to participate.

Online Encoding Phase: Similar to EPC C1G2 standard [1], the reader broadcasts a QUERY command to start an ALOHA frame that is shared by all the tag groups. This command has three parameters: the number of time slots f in the frame, the number of time slots s in logical frame, and the random seed r.

Consider an arbitrary tag iid in an arbitrary group gid. When receiving the reader's QUERY command, the tag computes a hash value $h(gid \oplus F(r, h(iid) \bmod s)) \bmod f$, as the index of the time slot it chooses for giving reply, where h is a hash function, $\oplus$ is concatenation operator, $F(x, y)$ is a pseudo-random number function taking two input parameters x and y. The transmission from all groups of tags forms an ALOHA frame. From the reader's perspective, this frame can be encoded as a bitmap B.

Clearly, for the tags in a group gid, the indices of their selected slots in the shared ALOHA frame can only be $h(gid \oplus F(r, 0)), h(gid \oplus F(r, 1)), \ldots, h(gid \oplus F(r, s-1))$, where $\bmod f$ has been omitted for simplicity. These slots or more precisely, the bits converted from these slots, form the logical bitmap of group gid, which is denoted by $LB(gid)$.

For the above ALOHA protocol, a single execution round may not attain the predefined estimation accuracy. So it can be executed for w rounds for accuracy boosting. All these rounds have the same frame length f and the same logical frame length s. But each ith round is given a different random seed r_i, $1 \le i \le w$. Let B_i be the bitmap collected in ith round. Let $LB_i(gid)$ be the logical bitmap of group gid in ith round.

Offline Estimation Phase: After transmitting the w bitmaps $B_1, B_2, \ldots, B_w$, the reader can estimate the popular groups H_g. Firstly, the total number of tags n is estimated as

$$\hat{n} = \frac{1}{w} \sum_{1 \le i \le w} \big(- f \log(\frac{z_i}{f}) \big), \tag{14}$$

where z_i is the number of zero bits in the shared bitmap B_i.

Secondly, for an arbitrary group gid, the number of tags in its logical bitmap is estimated as

$$\hat{n_s} = \frac{1}{w} \sum_{1 \le i \le w} \big(- s \log(\frac{z_i(gid)}{s}) \big), \tag{15}$$

where $z_i(gid)$ is the number of zero bits in the logical bitmap $LB_i(gid)$ of ith round. Of course, $\hat{n_s}$, which contains noise, cannot be used as an estimation of the size of group gid.

Thirdly, removing the noise by the following equation, we generate an unbiased estimation $\hat{n}_{gid}$ of the size of group gid.

$$\hat{n}_{gid} = \frac{f}{f - s} \big(\hat{n_s} - \frac{s}{f} \hat{n} \big) \tag{16}$$

For this group size estimation $\hat{n}_{gid}$, we can prove, if f and s are large enough, its probabilistic distribution approximates a Gaussian distribution, whose expected value and variance are

$$\begin{aligned} E(\hat{n}_{gid}) &\approx \frac{f}{f-s}\left(\frac{f-s}{f}n_{gid} + \frac{s}{f}n - \frac{s}{f}n\right) = n_{gid}, \\ Var(\hat{n}_{gid}) &\approx \frac{1}{w}s\left(e^{\frac{n_{gid}}{s}+\frac{n}{f}} - \frac{n_{gid}}{s} - 1\right) \\ &\quad + \frac{1}{w}\frac{s^2}{f}\left(e^{\frac{n}{f}} - \frac{n}{f} - 1\right). \end{aligned} \tag{17}$$

Please check [23, Appendix B] for detailed proof.

Finally, with the estimation $\hat{n}_{gid}$ of the size of each group gid in hand, we can report the set of popular groups as

$$\hat{H}_g = \{gid | 1 \le gid \le m \wedge g(\hat{n}_{gid}) \ge \alpha^* \hat{F}_g\}, \tag{18}$$

where $\hat{F}_g$ is an estimation of g-moment, which is generated by GME protocol, and α^* is the reporting threshold of popular groups. The value of α^* is given in (23), which is smaller than the actual threshold α in (4) to satisfy the constraint in (5).

C. Protocol Analysis and Parameter Setting

We analyze how to configure protocol parameters to satisfy the accuracy constraints in (5) and (6). A major difficult is the offline-vs-online problem: Only in the offline phase, can we know which function g is queried. Thus, our settings of online-phase parameters must make $Var(\hat{n}_{gid})$ sufficiently small that later in offline phase we can guarantee the accurate estimation of F_g-popular groups for an arbitrary function g. Fortunately, it can be shown that any F_g-popular group with $g(n_{gid}) \ge \alpha F_g$ is definitely a L_2-popular group satisfying $n_{gid}{}^2 \ge \alpha L_2$, since g is a monotonic function bounded by $O(n_i^2)$ [14].

Based on the above reasoning, we will focus on the accurate identification of L_2-popular groups. The accuracy constraint in (6) can be interpreted as $\forall i \in \hat{H}_g,\ Pr\{|\hat{n}_i{}^2 - n_i{}^2| \le \epsilon n_i{}^2\} \ge 1-\delta$. Since ϵ is small, $\sqrt{1 \pm \epsilon} \approx 1 \pm \frac{1}{2}\epsilon$, and the constraint is approximately $\forall i \in \hat{H}_g,\ Pr\{|\hat{n}_i - n_i| \le \frac{1}{2}\epsilon n_i\} \ge 1-\delta$. Since $\hat{n}_{gid}$ follows a Gaussian distribution, it can be translated to

$$\forall gid \in \hat{H}_g, \quad Var(\hat{n}_{gid})/n_{gid}{}^2 \le \frac{1}{4}\epsilon^2 / \left(\Phi^{-1}(1-\frac{\delta}{2})\right)^2, \tag{19}$$

where Φ is the CDF of standard normal distribution $\mathcal{N}(0,1)$. Next, we describe how to properly configure the protocol parameters s, w, f and α^* to make the constraint (19) satisfied.

Logical Bitmap Length s: According to (17), there are two methods to reduce $Var(\hat{n}_{gid})$, when the number of execution rounds w is fixed. One is to increase the length f of shared frame. Even if f tends to infinity, $Var(\hat{n}_{gid})$ in (17) is still greater than $\frac{1}{w}s(e^{\frac{n_{gid}}{s}} - \frac{n_{gid}}{s} - 1)$. Thus, we need the other method — increase the length s of logical frame. We often set

$$s = \frac{1}{c}\frac{n_{gid}^{\max}}{n}f, \tag{20}$$

where c is a constant typically assigned between 1 and 4, and $n_{gid}^{\max}$ is the size of a typical largest popular group. We will explain the equation (20) by details in [23, Appendix B].

Bitmap Number w and Bitmap Length f: The configuration of parameters w and f is important, since they determine the protocol execution time as $w \times f$. For L_2-popular groups, the lower bound of popular group size is $\sqrt{\alpha L_2}$. Even in this pessimistic situation, we will keep the inequality (19) satisfied.

$$Var(\hat{n}_{gid} | n_{gid} = \sqrt{\alpha L_2})/(\alpha L_2) \le \frac{1}{4}\epsilon^2 / \left(\Phi^{-1}(1-\frac{\delta}{2})\right)^2 \tag{21}$$

Applying the expression $Var(\hat{n}_{gid})$ in (17) to (21), we have

$$\begin{aligned} &\text{minimize } w \times f, \\ &\text{subject to } s\left(e^{\frac{\sqrt{\alpha L_2}}{s}+\frac{n}{f}} - \frac{\sqrt{\alpha L_2}}{s} - 1\right) + \frac{s^2}{f}\left(e^{\frac{n}{f}} - \frac{n}{f} - 1\right) \\ &\qquad \le \frac{w\alpha L_2 \epsilon^2/4}{(\Phi^{-1}(1-\frac{\delta}{2}))^2}, \end{aligned} \tag{22}$$

where the total number of tags n and the second-order moment L_2 can be substituted by their coarse estimations generated by GME. Since $s = \frac{1}{c}\frac{n_{gid}^{\max}}{n}f$ as in (20), the above constraint only has two unknown variables w and f. We will find an optimal combination of w and f that minimize the protocol execution time $w \times f$, subjective to the accuracy constraint in (22).

Reporting Threshold α^*: We must properly configure the popular group reporting threshold α^* in (18), so that $Pr\{H_g \in \hat{H}_g\}$, the probability for all the popular groups H_g to be identified, is at least $1-\delta$. Based on the analysis in [23, Appendix B],

$$\alpha^* = \frac{1}{1+\gamma^*}\frac{\sigma^{*2}}{L_2} f^{-1}_{\chi_1^2(\frac{\alpha L_2}{\sigma^{*2}})}(1 - \sqrt[k]{1-\delta}), \tag{23}$$

where σ^{*2} denotes $Var(\hat{n}_{gid})$ in (17) when $n_{gid} = \sqrt{\alpha L_2}$, $\chi_1^2(\frac{\alpha L_2}{\sigma^{*2}})$ is a noncentral chi-squared distribution with only one degree of freedom and noncentrality parameter $\frac{\alpha L_2}{\sigma^{*2}}$, $f_{\chi_1^2(\frac{\alpha L_2}{\sigma^{*2}})}$ is its CDF, k is the number of popular groups with sizes close to $\sqrt{\alpha L_2}$, and k is often set between 1 and 4, depending on the size distribution of groups. Eq. (23) needs coarse knowledge of L_2, and γ^* is the accuracy bound of the prior knowledge.

VI. Simulation

In this section, we evaluate the performance of our proposed GME protocol by simulations. To our best knowledge, there is no prior work for estimating generic moments in a multigroup RFID system. Hence, we have to compare the performance of GME with a straightforward solution that estimates the size of each sampled group by SRC [4]. In this section, we also evaluate the performance of TBC protocol [15] to validate our parameter settings in Section V-C, so that this protocol can be used by GME to identify L_2-popular groups on each layer.

A. Simulation Settings

Accuracy Model: We evaluate the performance of a protocol, by measuring its average estimation error when given a certain amount of running time. As in Section III-C, the estimation accuracy is specified by (ϵ, δ) for group size estimation, and by (γ, η) for moment estimation. We obtain each accuracy result by calculating the averaged value of 1000 independent trials.

Models for Group Size Distribution: The performance of our protocols heavily depend on the distribution of the sizes of all groups. The following distribution models are considered in our simulation studies. By default, we set the number of tags n to 120,000, and the number of groups m to 10,000.

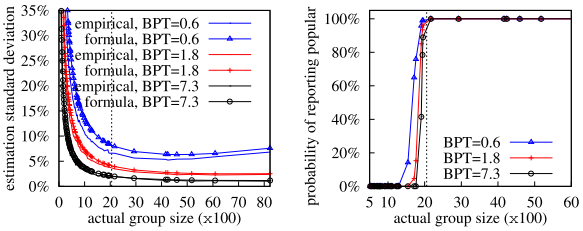

Fig. 4. Estimation accuracy of TBC protocol when $\alpha = 2.7\%$ and $\delta = 5\%$. (a) Group size estimation. (b) Popular group reporting.

We will also investigate other settings for a smaller-scale tag population (e.g., 1,000 groups and thousands of tags). It is impossible to take a snapshot of such a large number of tags using one RFID reader in a single scan. In practice, multiple readers are needed to cover a large place, like a warehouse.

We mainly consider the *Zipf distribution*, which is commonly found in many real-world random processes. This distribution is denoted by $ZD(m, \beta)$, where m is the number of groups, and β is the exponent characterizing the Zipf distribution. We configure the parameter β between 1 and 2 (by default, 1.8) to produce a long and heavy-tailed distribution, and adjust the parameter m to fit the settings of the number of tags n. We may also consider the *Log-normal distribution*, or consider the *Gaussian distribution* of non-popular groups mixed with tens of popular groups. However, due to page limit, we mainly show the simulation result of Zipf distribution.

B. Popular Group Identification Accuracy

In this subsection, we evaluate the accuracy of TBC protocol for identifying popular groups, when given a particular amount of execution time. We consider the normalized protocol running time quantified by BPT (Bits Per Tag), which is calculated as the number of time slots (or bits) $w \times f$ divided by the total number of tags n in all groups. We plot the simulation result in Figure 4a. It shows that, as BPT grows, relative estimation error $\frac{|\hat{n}_i - n_i|}{n_i}$ will reduce. In particular, when BPT is set to 0.6, 1.8 or 7.3, average estimation error is about 8%, 4% or 2%, respectively. However, such good accuracy only works for popular groups whose sizes are above 2000 (that is because we set $\alpha = 2.7\%$ and thus $\sqrt{\alpha L_2} \approx \sqrt{2.7\% \times 15 \cdot 10^7} \approx 2000$). For the groups below the threshold, the estimation error degrades dramatically as the group size reduces. In Figure 4a, we have plotted both the simulation result and the theoretical result calculated by the formula $\sqrt{Var(\hat{n}_{gid})}/n_{gid}$ in (17). The plot shows that the two results are quite consistent.

By contrast, to implement the same 8%, 4% or 2% estimation error for each group size, the SRC protocol [4] needs the time cost of 25, 27 or 38 bits per tag (BPT), respectively, as shown in Table I. This is because our simulated RFID system has a large number of small groups. SRC protocol has a significant overhead which is hard to amortize when scanning a small tag group, including both inter-cycle overhead (roughly equivalent to the time cost of transmitting 30 bits information as stated in Section IV-A) and the cost of running LoF protocol [16] to obtain coarse knowledge of the group size. Moreover, TBC has the optimization that sacrifices the accuracy of non-popular groups to preserve the accuracy of popular groups.

TABLE I

COMPARE THE PROTOCOL RUNNING TIME TO ATTAIN THE SAME AVERAGE ESTIMATION ERROR OF SIZES OF POPULAR GROUPS

Protocol	Mean Estimation Error of Popular Groups		
	8%	4%	2%
TBC	7.3 BPT	1.8 BPT	0.6 BPT
SRC for each group	25 BPT	27 BPT	38 BPT

Next, in Figure 4b, we illustrate the probability for TBC to report a group as popular. The plot shows that, for real popular groups whose sizes exceed $\sqrt{\alpha L_2}$ =2000, their probability of being reported is nearly 100%. Meanwhile, for non-popular groups below 2000, their reporting probability reduces rapidly as group size decreases. Thus, the probability for TBC to report all the L_2 popular groups is close to 100% in simulation. This is because our parameter setting for TBC is conservative. When the failure probability $\delta = 5\%$, using (19), we have $\epsilon = 32\%, 16\%$ or 8% corresponding to the expected relative error $8\%, 4\%$ or 2%. With the known ϵ and δ, we can compute the parameter settings: s by (20), w and f by (22), α^* by (23).

C. Compare Moment Estimation Accuracy

In this subsection, we compare the moment estimation accuracy of GME protocol with a *baseline protocol* that estimates the size of each sampled group by SRC [4]. To demonstrate the power of performing popular group identification before group-level sampling, we configure our GME protocol with only two layers (i.e., $\ell = 1$). As shown in Fig. 1, the layer 0 runs TBC protocol to identify L_2-popular groups, and the layer 1 runs SRC protocol per sampled group. We configure the TBC protocol with parameters $\alpha = 2.7\%$, $\delta = 5\%$ and $\epsilon = 16\%$ as the curve BPT $= 1.8$ in Fig. 4a. This setting needs TBC to run for 260s under simulated group distribution.

We compare the estimation accuracy of GME and SRC in Figure 5, when they are under the same time constraint, which is adjusted between 300s to 5000s. Figure 5a compares the entropy estimation accuracy, and Figure 5b compares L_2-moment estimation accuracy. Generally, the plots show that the accuracy of GME is much better than baseline. The accuracy advantage is much more prominent in Figure 5b for estimating L_2-moment than in Figure 5a for estimating entropy.

We use an example to better illustrate the advantage of GME protocol. Assume a warehouse manager prefers the time cost of performing moment estimation function to be within 600s, while scanning all groups by SRC needs as long as 4860s. With the constraint, the baseline protocol can only estimate the sizes of $\frac{600s}{4860s} \approx 12\%$ groups, and use the partial information to estimate moments. Due to low sampling probability, its L_2 estimation error is 130% in Fig. 5b. By contrast, GME's error for estimating L_2 drops to 12%, only one tenth of baseline protocol's error. This is because our GME can effectively mitigate the random sampling error of popular groups.

The comparison result shows that popular groups do have a significant impact on moment estimation accuracy. If we can identify them before sampling process, moment estimation accuracy will be dramatically improved. Otherwise, the baseline protocol, without such a step, needs 4700s (i.e., $\frac{4700s}{4860s} \approx 97\%$ sampling probability) to attain 10% estimation error as shown in Fig. 5b, which is seven times more expensive than GME.

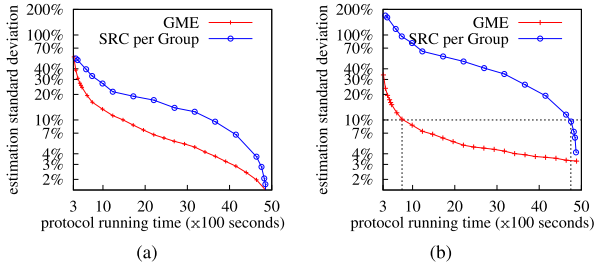

Fig. 5. Compare moment estimation accuracy of SRC and GME protocols (with $\ell = 1$, $\alpha = 2.7\%$, $\epsilon = 16\%$, $\delta = 5\%$), by giving the same time cost. (a) Entropy estimation. (b) L_2-moment estimation.

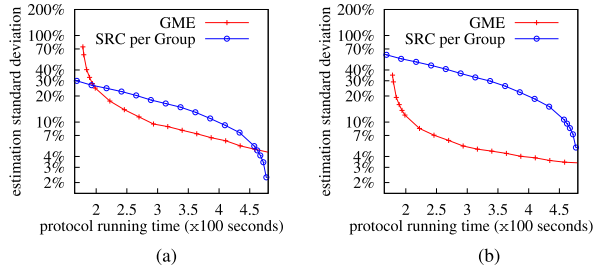

Fig. 6. Impact of the number of group m on the moment estimation accuracy of GME protocols (with $\ell = 1$, $\alpha = 2.7\%$, $\epsilon = 16\%$, $\delta = 5\%$). (a) Entropy estimation. (b) L_2-moment estimation.

D. Impact of Group Size Distribution

In this subsection, we evaluate how the group size distribution influences the moment estimation accuracy. In simulation, a Zipf distribution $ZD(m, \beta)$ is used to generate m groups with random sizes. The exponent β determines the decay speed of the long tail of Zipf distribution. When β is fixed, by varying the number of groups m, we can control the number of tags n in the system. We will evaluate the impact of m and β.

We reduce the number of groups m from its default value 10,000 to 1,000, and re-conduct our previous experiment in Fig. 5. Our purpose is to evaluate whether the advantage of our GME protocol over the baseline (i.e., running SRC [4] per group) will disappear for a smaller tag population. Our evaluation result is shown in Fig. 6. By comparing it with Fig. 5, the advantage of our GME over the baseline still exists if setting m ten times smaller. The reason is that, as long as the exponent β of the Zipf distribution is unchanged, the probability distribution of its long tail remains the same. Then, we can use the TBC protocol to capture the popular groups in the long tail, whose sizes exceed the threshold αF_g. Knowledge about the popular groups can greatly benefit the estimation of moment statistics, as we mentioned before.

Next, we configure the exponent β of the Zipf distribution to either of the two values 1.6 or 2, and evaluate its impact. Fig. 7 shows that the advantage of our GME over the baseline is more prominent when $\beta = 2$ than $\beta = 1.6$. This is because when β is larger, the tail decay rate of the Zipf distribution is faster, and therefore the popular groups become more outstanding from other non-popular groups, making them easier to recognize by a popular group identification protocol. This can help the moment estimation protocol to achieve better performance.

Finally, we change the group size distribution from Zipf to a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ of non-popular groups mixed with k popular groups whose sizes are more than three thousands. We configure $\mu = 30$ and $\sigma = 60$ for the Gaussian distribution to randomize the size of each non-popular group.

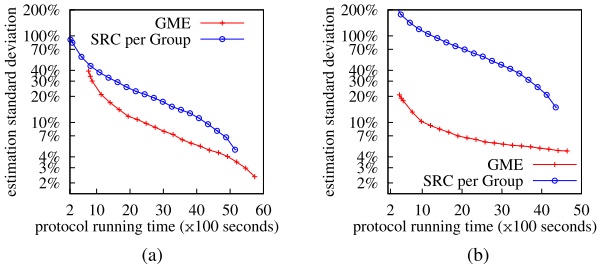

Fig. 7. Impact of the exponent β on the moment estimation accuracy of GME protocols (with $m = 10,000$, $\ell = 1$, $\alpha = 2.7\%$, $\epsilon = 16\%$, $\delta = 5\%$). (a) L_2-moment estimation, $\beta = 1.6$. (b) L_2-moment estimation, $\beta = 2$.

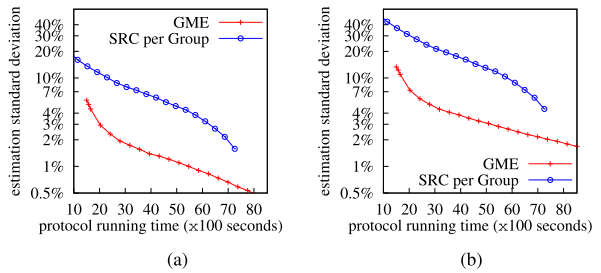

Fig. 8. Moment estimation accuracy of GME protocols (with $\ell = 1$, $\alpha = 3.2\%$, $\epsilon = 16\%$, $\delta = 5\%$) for Gaussian distribution of small groups mixed with popular groups. (a) Entropy estimation. (b) L_2-moment estimation.

We evaluate GME's moment estimation accuracy in Fig. 8. It shows that GME can accurately and efficiently estimate moments, as long as a certain number of popular groups H_g exist, each of which occupies an at least α share of the moment F_g as in (4). In this experiment, $\alpha = 3.2\%$, and $g(n_i) = {n_i}^2$.

Of course, in a warehouse inventory, it is possible for all groups to have similar quantities of tags. There may not exist any popular groups occupying an above-threshold share of the moment. In this case, we do not need to run any protocol to extract the information about popular groups. The good design of our moment estimation protocol named GME is that it will naturally degrade to a protocol that determines the size of each group. We can achieve it by configuring our GME protocol with only one layer by setting $\ell = 0$. Then, as shown in Fig. 1, the bottommost layer that runs the SRC protocol [4] per group will become the only layer.

E. Parameter Settings of GME Protocol

In Subsection VI-C, we have already verified the accuracy advantage of GME protocol over baseline protocol, when it is configured with two layers (by $\ell = 1$). Meanwhile, the popular group threshold α is fixed to 2.7%, the L_2-popular group error bound ϵ is set to 16%, and the failure probability δ is set to 5%. We will evaluate the performance of our GME protocol when it is configured with different parameter settings.

Popular Group Threshold α: The lower bound ratio α for popular groups is defined in (3), and we evaluate its impact as follows. We reconfigure α from 2.7% to 1.0% (or 0.7%), which moves the threshold of popular group size $\sqrt{\alpha L_2}$ from 2000 to 1250 (or 1000). In Figure 9b, we compare the L_2-moment estimation accuracy of GME protocol when α is set to 2.7%, 1.0% or 0.7%. The plot shows that smaller α ratio, which means the identification of more popular groups, can bring better accuracy for L_2 estimation. However, the price of smaller α is the higher time cost on 0th layer for popular group identification and therefore less remaining time on the 1st layer to estimate the sizes of sampled groups. More specifically,

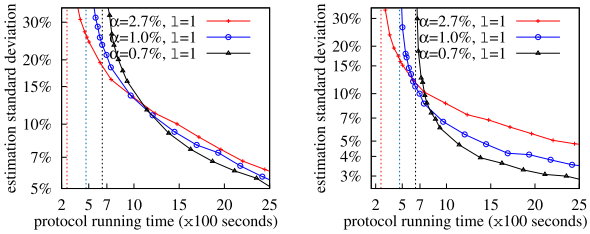

Fig. 9. Impact of the popular group threshold α on the moment estimation accuracy of GME protocol with $\ell = 1$, $\epsilon = 16\%$ and $\delta = 5\%$. (a) Entropy estimation. (b) L_2-moment estimation.

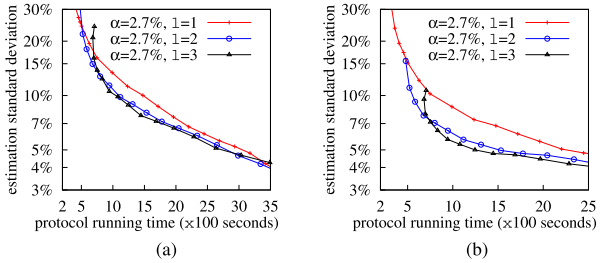

Fig. 10. Impact of the number of TBC layers ℓ on the moment estimation accuracy of GME protocol with $\epsilon = 16\%$ and $\delta = 5\%$. (a) Entropy estimation. (b) L_2-moment estimation.

the time cost of running TBC protocol on 0th layer grow to about 260s, 470s and 650s, when α is assigned to 2.7%, 1.0% and 0.7%, respectively. Hence, in Figure 9b, when the protocol running time approaches the needed time cost of running TBC, the moment estimation accuracy degrades rapidly. Choosing a proper value for α depends on the predefined time constraint.

Similar phenomenon can be witnessed in Figure 9a, which illustrates the accuracy of entropy estimation. But the accuracy gain by reducing α from 2.7% to 1.0% (or 0.7%) is much more modest for entropy estimation than for L_2 moment estimation.

Number of TBC Layers ℓ: We evaluate the impact of parameter ℓ, which controls the number of layers running TBC protocol for popular group identification. We illustrate the simulation result in Figure 10, which varies the ℓ value while keeping the popular group ratio α fixed. It shows that there exists obvious accuracy gain by increasing ℓ from 1 to 2, but the accuracy gain is no longer obvious if further increasing ℓ from 2 to 3. Our explanation is that, as the number of TBC layers ℓ grows, more popular groups will be identified, which can better mitigate the sampling error. However, larger ℓ value also means higher time cost of running TBC protocol on extra layers, which will neutralize the accuracy gain.

We can identify more popular groups either by increasing the number of sampling layers ℓ or by decreasing the popular group threshold α. Then, people may have the concerns — how to compare these two methods. In Figure 11a, we configure GME with different combinations of ℓ and α parameters, and evaluate its L_2-moment estimation accuracy. The figure shows that, as compared with the combination of $\alpha = 2.7\%$ and $\ell = 1$, the moment estimation accuracy can be improved either by increasing ℓ from 1 to 2 (see the second curve in the chart legend) or by decreasing α from 2.7% to 1.0% (see the third curve). The two methods of increasing ℓ and decreasing α can be combined to render even better result, as shown by the fourth curve $\alpha = 2.2\%$ and $\ell = 2$, as compared with the third curve that only reduces α.

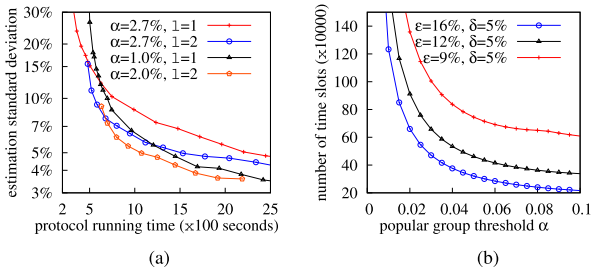

Fig. 11. Combination of the two methods of increasing ℓ and reducing α to improve the moment estimation accuracy of GME protocol. (a) L_2 estimation with $\epsilon = 16\%$, $\delta = 5\%$. (b) Impact of α on TBC time cost.

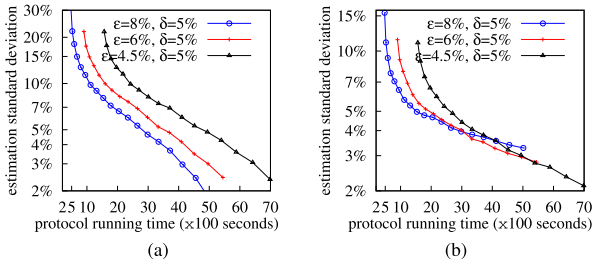

Fig. 12. Impact of the error bound ϵ of underlying TBC and SRC on the moment estimation of GME protocol with $\alpha = 2.7\%$, $\ell = 2$ and $\delta = 5\%$. (a) Entropy estimation. (b) L_2-moment estimation.

Our explanation is that, when increasing the number of TBC layers ℓ, the time cost of GME will increase only linearly. By contrast, when the threshold α of popular groups reduces, the time cost for TBC to identify them will skyrocket at nearly exponentially speed, as shown by Figure 11b. The plot is drawn using equation (22) to theoretically evaluate the relation between α and TBC's time cost $w \times f$. Thus, when reducing α becomes no longer cost-effective (e.g., when α is smaller than 3% as in Figure 11b), increasing ℓ will be a good choice.

Accuracy Parameters ϵ and δ of Underlying Protocols: We fix the number of TBC layers ℓ to 2, and the popular group threshold α to 2.7%. Then, we evaluate the impact of accuracy parameters ϵ and δ of the underlying TBC and SRC protocols. Essentially, ϵ and δ jointly controls one parameter, i.e., the expected standard deviation $\epsilon/\Phi^{-1}(1-\frac{\delta}{2})$. Hence, we fix the failure probability δ to 5%, and vary ϵ to evaluate its impact. We show the simulation result in Figure 12.

Figure 12a presents a surprising result: The time efficiency of GME protocol for entropy estimation improves, when the error ϵ of underlying protocols grows. This is because, when estimating L_1-moment or entropy, the size estimation errors of different groups have a good chance to counteract each other, no matter whether ϵ is configured with a small or large value. Thus, the key factor that decides the accuracy of L_1 or entropy estimation is not ϵ, but the number of groups (either popular or non-popular) whose sizes have been estimated. As ϵ grows, the constraint over estimation error gets loosened, and the time cost of running TBC and SRC protocols will decrease (see Figure 11b). Therefore, when ϵ grows, using the same time cost, a higher percentage of groups can have known sizes, which can improve the accuracy of entropy estimation.

For L_2-moment estimation, the situation is totally different. It becomes difficult for the size estimation errors of

different groups to counteract each other. A higher percentage of groups with known sizes does not necessarily mean the better accuracy of moment estimation. Figure 12b shows that reducing error bound ϵ can help improve accuracy of L_2-moment estimation if protocol time is sufficient, and we must carefully adjust the combination of error bound ϵ and sampling probability p, in order to achieve the best time efficiency.

VII. Conclusion

For an RFID system with multiple groups of tags, this paper investigates a new problem that simultaneously identifies popular groups and estimates moments. A naive solution is to take a random sample of groups, and sequentially determine the size of each sampled group. We demonstrate that this solution is quite inefficient, and we propose a GME protocol, to render much better moment estimation accuracy under the same time constraint. The accuracy advantage of GME is due to our optimization that, each time before performing group-level sampling, we execute a popular group identification protocol to avoid the sampling error of popular groups. We also discover that we can further improve moment estimation accuracy, if we perform the group-level sampling recursively with multiple layers. The accuracy gain of our GME protocol has been verified by both theoretical analysis and simulation studies. We have also presented the formal analysis of our protocol's estimation accuracy, and studied the method for computing the optimized protocol parameters.

Appendix A
Analysis of Moment Estimation Accuracy

We have presented the GME protocol in Section IV-B for generic moment estimation, and have presented the analysis result about its estimation accuracy in Theorem 1. In following, we formally prove the analysis result in Theorem 1.

Assuming Perfect Popular Group Identification: Firstly, we focus on the impact of recursive sampling on the moment estimation accuracy. Thus, we assume that a perfect protocol is available for identifying the popular groups H_j on the jth layer without mistakes, i.e., $\hat{H}_j \equiv H_j$, and also assume the size of each popular group group is precisely known, i.e., $\forall i \in \hat{H}_j,\ \hat{n_i}^j \equiv n_i$. We will relax these two assumptions later.

Since $\hat{n_i}^j \equiv n_i$, by (11), we have $\hat{F}_\ell \equiv F_\ell$. Consider the following recursive definition similar to (12), but with $\hat{n_i}^j$ replaced by its true value n_i and with $\hat{H}_j$ replaced by H_j:

$$\hat{X}_\ell = F_\ell, \quad \hat{X}_j = \frac{1}{p}\hat{X}_{j+1} + \sum\nolimits_{i\in H_j}(1-\frac{1}{p}\hbar_{j+1}(i))v_i, \quad (24)$$

where the symbol v_i is used to abbreviate $g(n_i)$, throughout this section. Suppose on the $(j+1)$th layer, $\hat{X}_{j+1}$ is an unbiased estimation of the moment F_{j+1} of this layer, i.e., $E(\hat{X}_{j+1}) = F_{j+1}$. Then, based on this assumption, we further analyze the attribute of the moment estimation $\hat{X}_j$ on jth layer. The relation between the neighboring jth and $(j+1)$th layers is that a group i on the jth layer is sampled on the $(j+1)$th layer exactly when the boolean hash function $\hbar_{j+1}(i)$ is equal to one, and therefore $F_{j+1} = \sum_{i\in S_{j+1}} v_i = \sum_{i\in S_j} \hbar_{j+1}(i)v_i$. Then, applying this equation to (24) and using $E(\hbar_{j+1}(i)) = p$,

$$\begin{aligned} E(\hat{X}_j) &= E\big(\frac{1}{p}\sum_{i\in S_j}\hbar_{j+1}(i)v_i - \frac{1}{p}\sum_{i\in H_j}\hbar_{j+1}(i)v_i + \sum_{i\in H_j} v_i\big) \\ &= E\big(\frac{1}{p}\sum\nolimits_{i\in S_j\setminus H_j}\hbar_{j+1}(i)v_i + \sum\nolimits_{i\in H_j} v_i\big) = F_j, \end{aligned} \quad (25)$$

Thus, $\hat{X}_j$ is also an unbiased estimation of the moment F_j.

We further analyze the estimation error of $\hat{X}_j$. The difficulty is that Eq. (24) has a recursive form with $\hat{X}_j$ depending on $\hat{X}_{j+1}$. So we define the symbol $\hat{Y}_j$, similar to $\hat{X}_j$ in (24), but with $\hat{X}_{j+1}$ replaced by its expected value F_{j+1}.

$$\hat{Y}_\ell = F_\ell, \quad \hat{Y}_j = \frac{1}{p}F_{j+1} + \sum\nolimits_{i\in H_j}(1-\frac{1}{p}\hbar_{j+1}(i))v_i \quad (26)$$

Thus, $\hat{Y}_j$ is an estimation of moment F_j, in an ideal situation that the moment F_{j+1} of immediately higher layer is known. Equation (26) has no recursive form and is easier to analyze. Similar to (25), we can prove that $\hat{Y}_j$ is an unbiased estimation of F_j, i.e., $E(\hat{Y}_j) = F_j$. Equation (26) can be rewritten as

$$\begin{aligned} \hat{Y}_j &= \sum\nolimits_{i\in S_j}\frac{1}{p}\hbar_{j+1}(i)v_i + \sum\nolimits_{i\in H_j}(1-\frac{1}{p}\hbar_{j+1}(i))v_i \\ &= \sum\nolimits_{i\in H_j} v_i + \frac{1}{p}\sum\nolimits_{i\in S_j\setminus H_j}\hbar_{j+1}(i)v_i. \end{aligned} \quad (27)$$

Clearly, when generating moment estimating $\hat{Y}_j$, the popular groups in H_j always participate, while the non-popular groups in $S_j\setminus H_j$ participate only when the boolean hash function $\hbar_{j+1}(i)$ for sampling is equal to one. So the popular groups H_j are kept away from the group-level random sampling, which can improve moment estimation accuracy.

From (27), by the properties of variance, and by the pairwise independence of $\hbar$, we have

$$Var(\hat{Y}_j) = \frac{1}{p^2}\sum\nolimits_{i\in S_j\setminus H_j} Var(\hbar_{j+1}(i))v_i^2$$

Using $Var(\hbar_{j+1}(i)) = p(1-p)$, and by the definition of popular groups,

$$Var(\hat{Y}_j) = \frac{1-p}{p}\sum\nolimits_{i\in S_j\setminus H_j} v_i^2 \le \frac{1-p}{p}\alpha {F_j}^2. \quad (28)$$

By ChebyShev inequality $Pr\{|\hat{Y} - E(\hat{Y})| \le k\sqrt{Var(\hat{Y})}\} \ge 1 - \frac{1}{k^2}$, we have

$$Pr\{|\hat{Y}_j - F_j| \le \epsilon F_j\} \ge 1 - \frac{1-p}{p}\alpha/\epsilon^2,$$

where ϵ is used to bound the relative error of $\hat{Y}_j$ for estimating moment F_j due to group sampling error, and the probability of successful bounding is at least $1 - \frac{1-p}{p}\alpha/\epsilon^2$, which increases as the sampling probability p grows or as the popular group threshold α grows. Note that the symbol ϵ is also used to bound the size estimation errors of popular groups in (6).

By (27), the moment estimations $\hat{Y}_j$ depends on the boolean hash functions $\hbar_{j+1}$. Due to the independence of $\hbar_{j+1}$ on different layers, the moment estimations $\hat{Y}_j$ with $0 \le j < \ell$ are mutually independent. Thus, the probability for any jth layer's moment estimation error $|\hat{Y}_j - F_j|$ to exceed the bound ϵF_j is

$$Pr\{\bigvee\nolimits_{0\le j<\ell}|\hat{Y}_j - F_j| \ge \epsilon F_j\} \le 1 - \big(1 - \frac{1-p}{p}\alpha/\epsilon^2\big)^\ell.$$

By the Taylor expansion $(1-x)^\ell \approx 1-\ell x+O(x^2)$,

$$Pr\{\bigvee\nolimits_{0\le j<\ell}|\hat{Y}_j-F_j|\ge \epsilon F_j\}\le \frac{1-p}{p}\ell\alpha/\epsilon^2. \quad (29)$$

Combining (26) and (24), $\hat{X}_j-\hat{Y}_j=\frac{1}{p}(\hat{X}_{j+1}-F_{j+1})$. Then, $\hat{X}_j-F_j=\frac{1}{p}(\hat{X}_{j+1}-F_{j+1})+(\hat{Y}_j-F_j)$. Hence,

$$\begin{aligned}\hat{X}_0-F_0&=\frac{1}{p}(\hat{X}_1-F_1)+(\hat{Y}_0-F_0)=\ldots=\frac{1}{p^\ell}(\hat{X}_\ell-F_\ell)\\&\quad+\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}(\hat{Y}_j-F_j)\\&=\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}(\hat{Y}_j-F_j). \quad (30)\end{aligned}$$

It implies that the moment estimation error of $\hat{X}_0$ is a linear combination of errors of $\hat{Y}_j$ on all layers with $0\le j<\ell$. Then,

$$|\hat{X}_0-F_0|\le\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}|\hat{Y}_j-F_j|,$$

and

$$Pr\{|\hat{X}_0-F_0|\ge\gamma F_0\}\le Pr\{\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}|\hat{Y}_j-F_j|\ge\gamma F_0\}.$$

Further, using $Pr\{A\}\le Pr\{A\wedge B\}+P\{\neg B\}$, we have

$$\begin{aligned}&Pr\{|\hat{X}_0-F_0|\ge\gamma F_0\}\\&\le Pr\Big\{\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}|\hat{Y}_j-F_j|\ge\gamma F_0\\&\quad\wedge\bigwedge\nolimits_{0\le j<\ell}|\hat{Y}_j-F_j|\le\epsilon F_j\Big\}\\&\quad+Pr\Big\{\bigvee\nolimits_{0\le j<\ell}|\hat{Y}_j-F_j|\ge\epsilon F_j\Big\}.\end{aligned}$$

Applying (29) to the second term of the above equation,

$$\begin{aligned}&Pr\{|\hat{X}_0-F_0|\ge\gamma F_0\}\\&\le Pr\Big\{\sum_{0\le j<\ell}\frac{1}{p^j}\epsilon F_j\ge\gamma F_0\Big\}+\frac{1-p}{p}\frac{\ell\alpha}{\epsilon^2}. \quad (31)\end{aligned}$$

Note that $F_j=\sum_{i\in S_j}v_i=\sum_{i\in S_0}v_i\prod_{1\le t\le j}\hbar_t(i)$, since a category i is sampled on the jth layer exactly when its boolean hash functions $\hbar_t(i)$ outputs one for all t-th layers, $1\le t\le j$. Considering that hash functions $\hbar_j$ for group sampling on different layers are mutually independent,

$$\begin{aligned}E(\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}F_j)&=E\big(\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}\sum_{i\in S_0}v_i\prod\nolimits_{1\le t\le j}\hbar_t(i)\big)\\&=\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}\sum_{i\in S_0}v_i\prod\nolimits_{1\le t\le j}E(\hbar_t(i))\\&=\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}\sum_{i\in S_0}v_ip^j=\ell F_0.\end{aligned}$$

Thus, and by Markov's inequality $Pr\{X\ge a\}\le E(X)/a$,

$$Pr\{\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}F_j\ge\theta\ell F_0\}\le\ell F_0/(\theta\ell F_0)=1/\theta, \quad (32)$$

where θ is a tunable ratio. By choosing $\gamma=\theta\ell\epsilon$ in (31),

$$\begin{aligned}&Pr\{|\hat{X}_0-F_0|\ge\theta\ell\epsilon F_0\}\\&\le Pr\Big\{\sum_{0\le j<\ell}\frac{1}{p^j}F_j\ge\theta\ell F_0\Big\}+\frac{1-p}{p}\frac{\ell\alpha}{\epsilon^2}\\&\le\frac{1}{\theta}+\frac{1-p}{p}\ell\alpha/\epsilon^2.\end{aligned}$$

It characterizes the moment estimation accuracy of $\hat{X}_0$, with the presence of ℓ-layer recursive sampling, and with the precise knowledge of α-fraction popular groups on each layer. By carefully choosing values for protocol parameters ℓ, p and α, we can attain arbitrary moment estimation accuracy. For example, if choosing $\ell=2$ and $p=0.5$, then we have

$$Pr\{|\hat{X}_0-F_0|\ge 2\theta\epsilon F_0\}\le 2\alpha/\epsilon^2+\frac{1}{\theta}.$$

Assuming Probabilistic Identification of Popular Groups: We analyze the accuracy of our moment estimation protocol, while relaxing the two assumptions about perfect popular group identification (i.e., $\hat{H}_j\equiv H_j$ and $\hat{n_i}^j\equiv n_i$). For the ease of understanding, we will divide our analysis into two parts, in which we relax these two assumptions one by one.

We relax the assumption $\hat{H}_j\equiv H_j$, and assume probabilistic identification with $Pr\{H_j\subseteq\hat{H}_j\}\ge 1-\delta$ on each jth layer. Since we firstly narrow our focus on the impact of H_j's identification error, we still assume that the estimation of popular group size $\hat{n_i}^j$ is equal to its actual value n_i.

We define the following variable $\hat{Y}'_j$, which is similar to (26) but has H replaced by its estimation $\hat{H}$:

$$\begin{aligned}\hat{Y}'_j&=\frac{1}{p}F_{j+1}+\sum\nolimits_{i\in\hat{H}_j}(1-\frac{1}{p}\hbar_{j+1}(i))v_i\\&=\sum\nolimits_{i\in S_j}\frac{1}{p}\hbar_{j+1}(i)v_i+\sum\nolimits_{i\in\hat{H}_j}(1-\frac{1}{p}\hbar_{j+1}(i))v_i\\&=\sum\nolimits_{i\in\hat{H}_j}v_i+\frac{1}{p}\sum\nolimits_{i\in S_j\setminus\hat{H}_j}\hbar_{j+1}(i)v_i, \quad (33)\end{aligned}$$

where v_i is an abbreviation of $g(n_i)$. Clearly, $\hat{Y}'_j$ is an unbiased estimation of F_j, since $E(\hbar_{j+1}(i))=p$. By the properties of variance, by the pairwise independence of $\hbar$, by the definition of popular groups, and using $Var(\hbar_{j+1}(i))=p(1-p)$,

$$\begin{aligned}Var(\hat{Y}'_j|H_j\subseteq\hat{H}_j)&=\frac{1}{p^2}\sum\nolimits_{i\in S_j\setminus\hat{H}_j}Var(\hbar_{j+1}(i))v_i^2\\&=\frac{1-p}{p}\sum\nolimits_{i\in S_j\setminus\hat{H}_j}v_i^2\le\frac{1-p}{p}\alpha{F_j}^2.\end{aligned}$$

According to the Chebyshev's inequality,

$$Pr\{|\hat{Y}'_j-F_j|\ge\epsilon F_j|H_j\subseteq\hat{H}_j\}\le\frac{1-p}{p}\alpha/\epsilon^2.$$

Applying the property $Pr\{A|B\}=\frac{Pr\{A\wedge B\}}{Pr\{B\}}\le Pr\{A\wedge B\}$,

$$Pr\{|\hat{Y}'_j-F_j|\ge\epsilon F_j\wedge H_j\subseteq\hat{H}_j\}\le\frac{1-p}{p}\alpha/\epsilon^2$$

Using the property $Pr\{A\}=Pr\{A\wedge B\}+Pr\{A\wedge\neg B\}\le Pr\{A\wedge B\}+Pr\{\neg B\}$, and due to $Pr\{H_j\not\subseteq\hat{H}_j\}\le\delta$,

$$\begin{aligned}&Pr\{|\hat{Y}'_j-F_j|\ge\epsilon F_j\}\\&\le Pr\{|\hat{Y}'_j-F_j|\ge\epsilon F_j\wedge H_j\subseteq\hat{H}_j\}\\&\quad+Pr\{H_j\not\subseteq\hat{H}_j\}\le\frac{1-p}{p}\alpha/\epsilon^2+\delta. \quad (34)\end{aligned}$$

Due to the mutual independence of boolean hash function $\hbar_j$ and error of identifying popular groups H_j on all layers, the probability that the error $|\hat{Y}'_j-F_j|$ exceeds the threshold

ϵF_j on any jth ($0 \le j < \ell$) layer is

$$Pr\{\bigvee\nolimits_{0\le j<\ell} |\hat{Y}'_j - F_j| \ge \epsilon F_j\} \le 1 - (1 - \frac{1-p}{p}\alpha/\epsilon^2 - \delta)^\ell \approx \ell(\frac{1-p}{p}\alpha/\epsilon^2 + \delta). \quad (35)$$

We further define the following symbol $\hat{X}'_j$ with a recursive form, which is similar to (24) but has H replaced by $\hat{H}$:

$$\hat{X}'_\ell = F_\ell, \quad \hat{X}'_j = \frac{1}{p}\hat{X}'_{j+1} + \sum\nolimits_{i\in\hat{H}_j}(1 - \frac{1}{p}\hbar_{j+1}(i))v_i. \quad (36)$$

By combining (36) and (33), we have $\hat{X}'_j - \hat{Y}'_j = \frac{1}{p}(\hat{X}'_{j+1} - F_{j+1})$, which can be rewritten as follows.

$$\hat{X}'_j - F_j = \frac{1}{p}(\hat{X}'_{j+1} - F_{j+1}) + (\hat{Y}'_j - F_j)$$

Applying this equation recursively and using $\hat{X}'_\ell = F_\ell$,

$$\hat{X}'_0 - F_0 = \frac{1}{p}(\hat{X}'_1 - F_1) + (\hat{Y}'_0 - F_0) = \frac{1}{p^2}(\hat{X}'_2 - F_2) + \frac{1}{p}(\hat{Y}'_1 - F_1) + (\hat{Y}'_0 - F_0) = \ldots = \sum\nolimits_{0\le j<\ell}\frac{1}{p^j}(\hat{Y}'_j - F_j).$$

Hence, the moment estimation error $\hat{X}'_0 - F_0$ is a linear combination of the errors $\hat{Y}'_j - F_j$ on different layers with $0 \le j < \ell$. Clearly, we have $|\hat{X}'_0 - F_0| \le \sum_{0\le j<\ell}\frac{1}{p^j}|\hat{Y}'_j - F_j|$. Further using (34), (35), and by a similar procedure as before,

$$\begin{aligned} &Pr\{|\hat{X}'_0 - F_0| \ge \gamma F_0\} \\ &\le Pr\{\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}|\hat{Y}'_j - F_j| \ge \gamma F_0 \wedge \bigwedge\nolimits_{0\le j<\ell}|\hat{Y}'_j - F_j| \le \epsilon F_j\} + Pr\{\bigvee\nolimits_{0\le j<\ell}|\hat{Y}'_j - F_j| \ge \epsilon F_j\} \\ &\le Pr\{\sum\nolimits_{0\le j<\ell}\frac{1}{p^j}\epsilon F_j \ge \gamma F_0\} + \ell(\frac{1-p}{p}\alpha/\epsilon^2 + \delta). \end{aligned}$$

By choosing $\gamma = \theta\ell\epsilon$, where θ is a tunable ratio, and using $Pr\{\sum_{0\le j<\ell}\frac{1}{p^j}F_j \ge \theta\ell F_0\} \le \frac{1}{\theta}$ in (32), we have

$$Pr\{|\hat{X}'_0 - F_0| \ge \theta\ell\epsilon F_0\} \le \ell(\frac{1-p}{p}\alpha/\epsilon^2 + \delta) + \frac{1}{\theta}.$$

Assuming Noisy Group Size Knowledge: Instead of $\hat{n_i}^j \equiv n_i$, we adopt a more realistic assumption that only noisy knowledge of popular group sizes is available. In this situation, we will analyze the moment estimation accuracy of $\hat{F}_0$, which is generated by the recursive formula in (12). More specifically, we suppose TBC protocol has been run on each jth sampling layer, $0 \le j < \ell$, to estimate the sizes of popular groups. Let $\hat{n_i}^j$ be the estimated size of popular group i on the jth layer. Let $\hat{v_i}^j = g(\hat{n_i}^j)$. From (5), we can suppose

$$Pr\{|\sum\nolimits_{i\in\hat{H}_j}\hat{v_i}^j - \sum\nolimits_{i\in\hat{H}_j}v_i| \le \sum\nolimits_{i\in\hat{H}_j}\epsilon v_i\} \ge 1 - \delta. \quad (37)$$

We also assume the size of each sampled group on ℓth layer has been estimated by some protocol, e.g., SRC [4], so that

$$Pr\{|\hat{F}_\ell - F_\ell| \le \epsilon F_\ell\} = Pr\{|\sum\nolimits_{i\in C_\ell}\hat{v_i}^\ell - \sum\nolimits_{i\in C_\ell}v_i| \le \sum\nolimits_{i\in C_\ell}\epsilon v_i\} \ge 1 - \delta, \quad (38)$$

where $\hat{F}_\ell = \sum_{i\in S_\ell}\hat{v_i}^\ell$. Based on the knowledge in (37) and (38), we analyze the accuracy of our moment estimator in (12).

By subtracting (33) from (12), we obtain

$$\hat{F}_j - \hat{Y}'_j = \frac{1}{p}(\hat{F}_{j+1} - F_{j+1}) + Err_j,$$

where

$$Err_j = \sum_{i\in\hat{H}_j}(1 - \frac{1}{p}\hbar_{j+1}(i))(\hat{v_i}^j - v_i). \quad (39)$$

From (39), $\hat{F}_j - F_j = \frac{1}{p}(\hat{F}_{j+1} - F_{j+1}) + (\hat{Y}'_j - F_j) + Err_j$. By applying this equation recursively, we have

$$\hat{F}_0 - F_0 = \frac{1}{p^\ell}(\hat{F}_\ell - F_\ell) + \sum\nolimits_{0\le j<\ell}\frac{1}{p^j}\big((\hat{Y}'_j - F_j) + Err_j\big).$$

Clearly, the following inequality holds.

$$|\hat{F}_0 - F_0| \le \frac{1}{p^\ell}|\hat{F}_\ell - F_\ell| + \sum\nolimits_{0\le j<\ell}\frac{1}{p^j}(|\hat{Y}'_j - F_j| + |Err_j|)$$

Let Z_0 be the right side of the above inequality. Then,

$$\begin{aligned} &Pr\big\{|\hat{F}_0 - F_0| \ge \gamma F_0\big\} \\ &\le Pr\{Z_0 \ge \gamma F_0\} \le Pr\big\{Z_0 \ge \gamma F_0 \\ &\quad \wedge(\bigwedge\nolimits_j|\hat{Y}'_j - F_j| < \epsilon F_j) \wedge (\bigwedge\nolimits_j|Err_j| < \epsilon F_j) \\ &\quad \wedge(|\hat{F}_\ell - F_\ell| < \epsilon F_\ell)\big\} \\ &\quad + Pr\{\bigvee\nolimits_j|\hat{Y}'_j - F_j| \ge \epsilon F_j\} + Pr\{\bigvee\nolimits_j|Err_j| \ge \epsilon F_j\} \\ &\quad + Pr\{|\hat{F}_\ell - F_\ell| \ge \epsilon F_\ell\}, \end{aligned}$$

where $\bigvee_j$ and $\bigwedge_j$ stand for $\bigvee_{0\le j<\ell}$ and $\bigwedge_{0\le j<\ell}$, respectively. Further applying (35) and (38), we have

$$\begin{aligned} &Pr\big\{|\hat{F}_0 - F_0| \ge \gamma F_0\big\} \\ &\le Pr\{\frac{1}{p^\ell}\epsilon F_\ell + \sum\nolimits_{0\le j<\ell}\frac{1}{p^j}(\epsilon F_j + \epsilon F_j) \\ &\ge \gamma F_0\} + \ell(\frac{1-p}{p}\alpha/\epsilon^2 + \delta) + Pr\{\bigvee\nolimits_j|Err_j| \ge \epsilon F_j\} + \delta \\ &\le Pr\{\sum\nolimits_{0\le j\le\ell}\frac{2}{p^j}\epsilon F_j \ge \gamma F_0\} + (\ell+1)\delta \\ &\quad + \frac{1-p}{p}\ell\alpha/\epsilon^2 + Pr\{\bigvee\nolimits_j|Err_j| \ge \epsilon F_j\}. \end{aligned}$$

By choosing $\gamma = 2\theta\ell\epsilon$, and using (32),

$$Pr\big\{|\hat{F}_0 - F_0| \ge 2\theta\ell\epsilon F_0\big\} \le \frac{1}{\theta} + (\ell+1)\delta + \frac{1-p}{p}\ell\alpha/\epsilon^2 + Pr\{\bigvee\nolimits_j|Err_j| \ge \epsilon F_j\}. \quad (40)$$

We study the property of popular group estimation error Err_j. Due to mutual independence of Err_j on all layers, and by (37),

$$Pr\{\bigvee\nolimits_{0\le j<\ell}|Err_j| \ge \sum_{i\in\hat{H}_j}\epsilon\hat{v_i}^j\} \le Pr\{\bigvee\nolimits_{0\le j<\ell}|\sum_{i\in\hat{H}_j}(\hat{v_i}^j - v_i)| \ge \sum_{i\in\hat{H}_j}\epsilon\hat{v_i}^j\} \le \ell\delta.$$

Clearly, $\sum_{i\in\hat{H}_j} v_i{}^j \leq F_j$, for any group size distribution. Then, $Pr\{\bigvee_{0\leq j<\ell} |Err_j| \geq \epsilon F_j\} \leq \ell\delta$. Applying it to (40),

$$Pr\{|\hat{F}_0 - F_0| \geq 2\theta\ell\epsilon F_0\} \leq \frac{1-p}{p}\ell\alpha/\epsilon^2 + \frac{1}{\theta} + (2\ell+1)\delta.$$

REFERENCES

[1] *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz-960MHz v2.0.0*, EPCglobal, 2014.

[2] G. W. Brindle and C. J. Gibson, "Entropy as a measure of diversity in an inventory of medical devices," *Med. Eng. Phys.*, vol. 30, no. 3, pp. 399–401, 2008.

[3] M. Buettner and D. Wetherall, "An empirical study of UHF RFID performance," in *Proc. ACM MOBICOM*, 2008, pp. 223–234.

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

[5] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID tags efficiently and anonymously," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[6] F. Isik, "Complexity in supply chains: A new approach to quantitative measurement of the supply-chain-complexity," in *Supply Chain Management*. Rijeka: Croatia: InTech, 2011.

[7] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MOBICOM*, 2006, pp. 322–333.

[8] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. ACM MOBIHOC*, 2010, pp. 1–10.

[9] H. Liu, W. Gong, L. Chen, W. He, K. Liu, and Y. Li, "Generic composite counting in RFID systems," in *Proc. IEEE ICDCS*, Jun./Jul. 2014, pp. 597–606.

[10] J. Liu, S. Chen, B. Xiao, Y. Wang, and L. Chen, "Category information collection in RFID systems," in *Proc. IEEE ICDCS*, Jun. 2017, pp. 2220–2225.

[11] J. Liu, B. Xiao, S. Chen, F. Zhu, and L. Chen, "Fast RFID grouping protocols," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 1948–1956.

[12] J. Liu, F. Zhu, Y. Wang, X. Wang, Q. Pan, and L. Chen, "Rf-scanner: Shelf scanning with robot-assisted RFID systems," in *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.

[13] X. Liu *et al.*, "Top-k queries for multi-category RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[14] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman, "One sketch to rule them all: Rethinking network flow monitoring with UnivMon," in *Proc. ACM SIGCOMM*, 2016, pp. 101–114.

[15] W. Luo, Y. Qiao, and S. Chen, "An efficient protocol for RFID multigroup threshold-based classification," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 890–898.

[16] C. Qian, H. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," in *Proc. IEEE PERCOM*, Mar. 2008, pp. 30–39.

[17] M. Shahzad and A. X. Liu, "Probabilistic optimal tree hopping for RFID identification," in *Proc. ACM SIGMETRICS*, 2013, pp. 293–304.

[18] B. Sheng, C. C. Tan, Q. Li, and W. Mao, "Finding popular categories for RFID tags," in *Proc. ACM MOBIHOC*, 2008, pp. 159–168.

[19] C. C. Tan, B. Sheng, and Q. Li, "How to monitor for missing RFID tags," in *Proc. IEEE ICDCS*, Jun. 2008, pp. 295–302.

[20] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An anticollision algorithm for the reader collision problem," in *Proc. IEEE ICC*, May 2003, pp. 1206–1210.

[21] Q. Xiao, S. Chen, and M. Chen, "Joint property estimation for multiple RFID tag sets using snapshots of variable lengths," in *Proc. MOBIHOC*, 2016, pp. 151–160.

[22] Q. Xiao, M. Chen, S. Chen, and Y. Zhou, "Temporally or spatially dispersed joint RFID estimation using snapshots of variable lengths," in *Proc. ACM MOBIHOC*, 2015, pp. 247–256.

[23] Q. Xiao, S. Chen, J. Liu, G. Cheng, and J. Luo. (2018). *Extended Online Version of A Protocol for Simultaneously Estimating Moments and Popular Groups in a Multigroup RFID System*. [Online]. Available: https://pan.baidu.com/s/1JM9Um8yaSnXvsQ5Q1pzFVg

[24] L. Xie, H. Han, Q. Li, J. Wu, and S. Lu, "Efficiently collecting histograms over RFID tags," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 145–153.

[25] 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.

Qingjun Xiao (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 Jiao Tong University, China, in 2007, and the Ph.D. degree from the Computer Science Department, The Hong Kong Polytechnic University, in 2011. He was a Post-Doctoral Researcher with Georgia State University and the University of Florida for three years. He is currently an Assistant Professor with Southeast University, China. His research interests include protocol and algorithm design in network traffic measurement, wireless sensor networks, and radio frequency identification systems. He is a member of the ACM and CCF.

Shigang Chen (M'02–SM'12–F'16) 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. In 2002, he joined the University of Florida, Gainesville, FL, USA. From 2002 to 2003, he served for the Technical Advisory Board with Protego Networks. He is currently a Professor with the Department of Computer and Information Science and Engineering, University of Florida. He has 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.

Jia Liu received the B.E. degree in software engineering from Xidian University, Xi'an, China, in 2010, and the Ph.D. degree in computer science and technology from Nanjing University, Nanjing, China, in 2016. He is currently a Research Assistant Professor with the Department of Computer Science and Technology, Nanjing University. His research mainly focuses on radio frequency identification technology. He is a member of the IEEE and CCF.

Guang Cheng received the B.S. degree in traffic engineering from Southeast University in 1994, the M.S. degree in computer application from the Heifei University of Technology in 2000, and the Ph.D. degree in computer network from Southeast University in 2003. He is currently a Full Professor with the School of Cyber Science and Engineering, Southeast University, Nanjing, China. His research interests are network security, network measurement, and traffic behavior analysis. He is a Senior Member of the IEEE.

Junzhou Luo received the B.S. degree in applied mathematics and the M.S. and Ph.D. degrees in computer network from Southeast University, Nanjing, China, in 1982, 1992, and 2000, respectively. He is currently a Full Professor with the School of Computer Science and Engineering, Southeast University. His research interests are future network architecture, network security, cloud computing, and wireless LAN. He is a member of the IEEE Computer Society and a Co-Chair of the IEEE SMC Technical Committee on Computer Supported Cooperative Work in Design. He is a member of the ACM and the Chair of ACM SIGCOMM, China.