An Efficient Protocol for Tag-Information Sampling in RFID Systems

Xiujun Wang[1(✉)], Yan Gao[1], Yangzhao Yang[2], Xiao Zheng[1], Xuangou Wu[1], and Wei Zhao[1]

[1] School of Computer Science and Technology, Anhui University of Technology, Maanshan 243032, China
wxj@mail.ustc.edu.cn

[2] Research Institute of Cyberspace Security of CETC, Beijing, China
forester@mail.ustc.edu.cn

Abstract. Given a population S of N tags in an RFID system, the tag-information sampling problem is to randomly choose K distinct tags from S to form a subset T, and then inform each tag in T of a unique integer from $\{1, 2, ..., K\}$. This is a fundamental problem in many real-time analysis applications in RFID systems. Because it enables rapidly selecting a random subset T and collecting the tag-information from T. However, existing protocols for this problem are far from satisfactory due to high communication costs. In this paper, our objective is to solve this problem by using a small communication cost. We first obtain a lower bound on communication cost, denoted by C_{lb}, for this problem. Then we design a protocol, denoted by P_{s}, to solve this problem, and prove that the communication cost of P_{s} stays within a factor of 2 of C_{lb}. Extensive simulations verifies the advantages of P_{s} comparing with other protocols.

Keywords: IoT networks · RFID systems · Tag-information sampling problem · Communication cost

1 Introduction

Over the past dozen years, Radio Frequency Identification (RFID) technology has been widely applied in tracking moving objects [8,21,26], managing supply chains [13,20,22], and controlling warehouse inventory [2,9,14,23,25]. Conceptually, an RFID system in these applications consists the following components:

- RFID tags, each of which carries a unique 96-bit or 128-bit ID stored in its chip, are attached to different physical objects and serve as the unique identifiers of these objects. A tag also carries the tag-information which can be either the attribute data of the tagged object or the sensor data [3,12,15].
- RFID readers, each of which is deployed to a location of interest, are used for collecting the IDs and other related information from those tags within their range.

S. W. Loke et al. (Eds.): MONAMI 2020, LNICST 338, pp. 198–212, 2020.
https://doi.org/10.1007/978-3-030-64002-6_13

- A backend server connects to each reader in an RFID system for offering them with needed information storage and computation.

A fundamental yet important functionality needed in RFID systems is called tag-information sampling which is to randomly pick K tags from a large population S to construct a subset T, and then inform each tag in T of a unique order from $\{1, 2, ..., K\}$. More concretely, this problem requires to design a protocol P that effectively puts all the tags in S into a hat, then continuously determines the next tag t by randomly drawing a tag from the hat until K tags have been chosen, and inform each of these K tags of their ordering.

This function has a wide range of applications in many tag-management problems, such as monitoring and gathering tags' information. Because, when users need to analyze the status or characteristics of a large population S, it is time-consuming and sometimes unnecessary to collect the tag-information from every tag in S. In contrast, the tag-information from a random and small subset T is good enough. For example, we consider the type of RFID tags that are augmented with sensors, e.g., WISP tags [1]. This kind of tags can feedback their IDs as well as real-time sensor data related to the status of the tagged objects or surrounding environmental conditions [3,12,15]. In such scenario, when we need to periodically collect sensor data from a large tag population, it is common that we randomly choose a small number of tags at a time and collect their sensor data, due to the redundancy in environmental data (for example tags within in a small area sense the same temperature data). Tag-information sampling also plays an important role in IoT networks where massive data captured by various sensor-augmented RFID tags usually contains a large amount of redundancy, and must be smartly managed and timely analyzed [6,7,17,24]. For other examples, please refer to [10,15,16,22,27].

There are a number of research works [3,12,19,27] for collecting tags' information. However, these protocols work for collecting the information from either a tag population S or a specific tag subset B which is predetermined by users. This limitation incurs a high communication cost when we apply these protocols to solve the tag-information sampling problem. There are also some research works [10,15] for collecting specific information, e.g., category information of the tagged objects, from a tag population which is pre-divided into multiple groups (subsets). However, these protocols still need to pre-set several groups (subsets), and only consider how to choose one tag randomly from a group each time. Therefore, these protocols are not suitable for solving the tag-information sampling problem.

For designing an efficient protocol for the tag-information sampling problem, we need to figure out the lower bound of communication cost when solving this problem and design a protocol that has a low communication cost. There are two technical challenges for achieving these two objectives. First, to obtain a lower bound, we need to analyze the essential information that must be transmitted between readers and tags such that the tag-information sampling problem is solved. This is not easy as we need to build a tricky process that transforms a protocol capable of solving the tag-information sampling problem into a coding process that represents any subset of a tag population. To the best of our

knowledge, none of the existing works has achieved this goal, and there is no other exiting lower bound we can rely on. Secondly, we need to design a protocol, denoted by P_s and prove its efficiency. This is hard because we can not let the user pre-set a subset B of K tags, instead, we need to guarantee that every subset of K tags from S has an equal probability of being chosen. This point is even harder due to the extra requirement that each tag in T needs to be quickly informed with a unique order from $\{1, 2, ..., K\}$.

The rest of this paper is organized as follows. Section 2 defines the tag-information sampling problem. Section 3 presents the analysis of the lower bound of communication cost. Section 4 proposes an efficient protocol for this problem. Section 5 evaluates the performances of our protocol. Lastly, we conclude this paper in Sect. 6.

2 Problem Statement

We consider an RFID system that includes a reader R and a population S of N tags. All of the tags in S are within the interrogating range of reader R. Each tag t carries an ID, denoted by t^{ID}, which uniquely identifies the tagged object, and t^{ID} has already been collected by reader R and stored into the backend server. Each tag t also contains some kind of information (such as the attribute data of the object that tag t is associated, or the environmental data from the sensor installed on tag t) which the reader wants to collect periodically. We call this kind of information of t as its tag-information, denoted by t^{Info}. Then the tag-information sampling problem can be defined as follows.

Definition 1. *In an RFID system, let K be the pre-determined size of sampled subsets, then the tag-information sampling problem requires to design a protocol P between the reader R and tags such that the following two statements are satisfied at the end of protocol P.*

C-I: A random subset T of K distinct tags is selected from the whole population S and every possible subset of K distinct tags has an equal probability of being chosen as T;

C-: Each tag in T is informed of a unique order ranging from 1 to K.

3 Analysis of the Lower Bound on Communication Cost

When exploring the lower bound for solving this problem, we omit the communication cost between the reader R and backend server.[1] Because they are connected over a high-speed connection where the data rate is usually above 10Mit/s. For example, the IMPINJ R220 UHF RFID READER and ZEBRA

[1] The reader R may transmit IDs and newly collected tag-information either to or from the backend server. or transmit newly collected tag-information to the backend server.

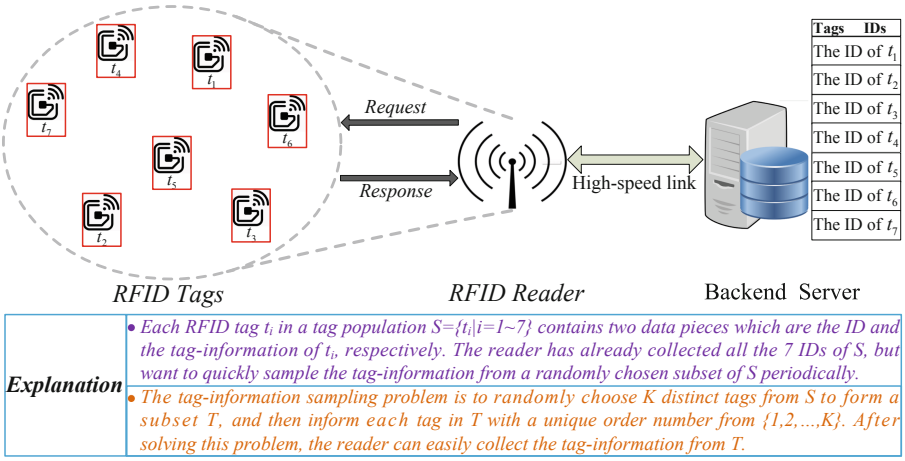

Fig. 1. An example of the tag-information sampling problem.

FX7500 RFID READER use Ethernet cable to join a network and connect to the backend server. In contrast, the reader R communicates with tags over a low-speed connection where the data rate ranges from 26.7 Kbit/s to 128 Kbit/s.

Before analyzing the lower bound, some symbols are defined as follows. We use T_{id} to denote a time frame for the reader R to broadcast a 96-bit array to all tags within its range. Thus, if reader R needs to send a total number of M bits to all tags during the execution of a protocol P, the communication cost and communication time of P are M and $\frac{MT_{\mathrm{id}}}{96}$, respectively. Let S denote a tag population of N tags, i.e., $S = \{t_i | i = 1 \sim N\}$, and K denote the size of the subset T. Without loss of generality, we assume that the ID of a tag is a 96-bit string, then set $\Gamma = 2^{96}$. Let U denote the universe of all the possible IDs, i.e., $U = \{0, 1, 2, ..., \Gamma - 1\}$. Lastly, t_i^{ID} is also used to represent a tag t, because t_i^{ID} is a unique identifier of tag t. Thus a tag population $S = \{t_i | i = 1 \sim N\}$ can also be represented by $S = \{t_i^{\mathrm{ID}} | i = 1 \sim N\}$.

The following theorem shows a nontrivial lower bound on the communication cost for any protocol P to solve the tag-information sampling problem. The road map of the proof is as follows. At first, based on the assumption that protocol P can randomly pick K tags from a tag population S, protocol P can be transformed into an encoding process that can represent S. Secondly, through analyzing the minimum number of bits required in the encoding process, we derive the lower bound on communication cost of P.

Theorem 1. *If a protocol P can solve the tag-information sampling problem by broadcasting $|P|$ bits from the reader R to all tags, then the following inequality is true:*

$$|P| \geq C_{lb} = \log_2(e)K, \tag{1}$$

where C_{lb} represents the lower bound of communication cost. Consequently, the communication time of protocol P can not be less than

$$T_{lb} = (C_{lb}T_{id})/96. \tag{2}$$

Proof. First, we analyze the general process when reader R solves the tag-information sapling problem by using a protocol P. Given a tag population S of N distinct IDs from the universe U ($S \subset U$), reader R takes S as the input to protocol P, and sends messages to tags. We can concatenate all of these messages into a single bit-string $\mathbb{M}$. Upon receiving $\mathbb{M}$, each tag $t \in S$ takes $\mathbb{M}$ and t^{ID} as the input to a decision function $Df()$ to decide whether t is chosen or not, and if so, which integer from $\{1, 2, ..., K\}$ is assigned to t. In general, the decision process of each tag $t \in S$ can be formulated as follows:

(1) If $\mathbf{Df}(t^{\mathrm{ID}}, \mathbb{M}) = K + 1$, tag t decides that it is not chosen;
(2) If $\mathbf{Df}(t^{\mathrm{ID}}, \mathbb{M}) = I \in \{1, 2, ..., K\}$, tag t decides that it is chosen and the unique integer assigned to t is I (when protocol P ends, tag t needs to send its tag-information in the I-th slot of the process that collects tag-information).

Based on the above, the transmitted message $\mathbb{M}$ actually is the parameter that decides how $\mathbf{Df}()$ maps every ID from U is mapped to the range $\{1, 2, ..., K\}$. A different value of $\mathbb{M}$ will lead to a different way that $\mathbf{Df}()$ maps IDs in U to $\{1, 2, ..., K\}$. Consider a specific value v of $\mathbb{M}$, let S_j represent the set of IDs in U that are mapped to $j \in \{1, 2, ..., K + 1\}$, i.e., $S_j = \{t^{\mathrm{ID}} \in U | \mathbf{Df}(t^{\mathrm{ID}}, \mathbb{M}) = j\}$, and let x_i represent the size of S_j ($x_i = |\{S_j\}|$). Then it is clear that protocol P can use a value v to solve the tag-information sampling problem, when the tag population S contains exactly one ID from S_j for $j = 1, 2, ..., K$ and $N - K$ IDs from S_{K+1}. Because there are

$$Enum(v) = (\prod_{j=1}^{K} \binom{x_j}{1}) \times \binom{x_{K+1}}{N-K} \tag{3}$$

different sets that can be handled by a specific value v of $\mathbb{M}$, a value of $\mathbb{M}$ encodes $(\prod_{j=1}^{K} \binom{x_j}{1}) \times \binom{x_{K+1}}{N-K}$ different sets of N IDs from U.

Next, we simplify $Enum(v)$ in (3) as follows:

$$\begin{aligned} Enum(v) &= (\prod_{j=1}^{K} \binom{x_j}{1}) \times \binom{x_{K+1}}{N-K} \\ &\leq (\prod_{j=1}^{K} x_j) \times (x_{K+1})^{N-K}/(N-K)!, \end{aligned} \tag{4}$$

where the inequality comes from the fact that $\binom{x_{K+1}}{N-K} = (x_{K+1})^{\underline{N-K}}/(N - K)! \leq (x_{K+1})^{N-K}/(N - K)!$. We then use the method of Lagrange multipliers to find the maximal value of (4), under the constraint that $\sum_{j=1}^{K+1} x_j \leq \Gamma$. More specifically, we set up a Lagrangian function:

$$\mathsf{L}(x_1, ..., x_{K+1}, \lambda) = (\prod_{j=1}^{K} x_j) \times (x_{K+1})^{N-K}/(N-K)! + \lambda(\sum_{j=1}^{K+1} x_j - \Gamma). \tag{5}$$

By taking derivative with respect to x_i $(i = 1, 2, ..., K+1)$ and setting it equal to 0, it is easy to get the following equalities

$$\begin{cases} \prod_{j=1, j\neq i}^{K} x_j \times (x_{K+1})^{N-K} = \lambda, & \text{for } i = 1, 2, ..., K, \\ \prod_{j=1}^{K} x_j \times (N-K)(x_{K+1})^{N-K-1} = \lambda, & \text{for } i = K+1. \end{cases}$$

Then based on the constraint $\sum_{j=1}^{K+1} x_j \leq \Gamma$, the maximum of (4) is achieved when $x_1 = x_2 = ..x_K = \Gamma/N$ and $x_{K+1} = \Gamma(N-K)/N$. Then, this indicates

$$\begin{aligned} Enum(v) &\leq (\prod_{j=1}^{K} x_j) \times (x_{K+1})^{N-K}/(N-K)! \\ &\leq [\Gamma/N]^K \times [\Gamma(N-K)/N]^{N-K}/(N-K)!. \end{aligned} \tag{6}$$

Since there are $\binom{\Gamma}{N}$ different sets of N IDs to encode, and every single value v of the transmitted message $\mathbb{M}$ can encode at most $[\Gamma/N]^K \times [\Gamma(N-K)/N]^{N-K}/(N-K)!$, $\mathbb{M}$ must have at least $\frac{\binom{\Gamma}{N}}{[\Gamma/N]^K \times [\Gamma(N-K)/N]^{N-K}/(N-K)!}$ different values. Let $|\mathbb{M}|$ denote the number of bits contained in $\mathbb{M}$, the followings are true:

$$\begin{aligned} |\mathbb{M}| &\approx -\log_2(N!) + K\log_2(N) + (N-K)\log_2(N/(N-K)) + \log_2((N-K)!) \\ &\geq -\log_2(2\sqrt{2\pi N}(N/e)^N) + \log_2(\sqrt{2\pi(N-K)}((N-K)/e)^{N-K}) + K\log_2(N) \\ &\quad + (N-K)\log_2(N/(N-K)) \\ &= K\log_2(e) + 0.5\log_2((N-K)/N) - 1. \end{aligned} \tag{7}$$

Note the first approximation in the above comes from the fact that $\Gamma \gg N$, and then $\Gamma^{\underline{N}} \approx \Gamma^N$. The inequality in the above is based on Stirling's Formula shown in Lemma 7.3 of [18]. We can obtain the lower bound shown in (1), because $0.5\log_2((N-K)/N) - 1$ is comparatively smaller than $K\log_2(e)$.

Lastly, since reader R needs to send at least $C_{\text{lb}} = K\log_2(e)$ bits to tags, and T_{id} is a time frame for the reader R to broadcast a 96-bit array to all tags within its range, protocol P must have a communication time at least $\frac{C_{\text{lb}} T_{\text{id}}}{96}$.

4 Design of an Efficient Protocol for the Tag-Information Sampling Problem

In this section, we propose an efficient protocol, represented by P_{s}, which can solve the tag-information sampling problem by using a small communication cost. The basic idea of P_{s} is to design two separate phases: one phase, denoted as P_{s}-1, is for achieving the first requirement of Definition 1 (see *C-I*); and the other phase, denoted as P_{s}-2, is to satisfy the second requirement of Definition 1 (see *C-II*). In the following, we define there states of a tag t during the execution of P_{s}.

- ⋆ *UNSELECTED STATE*: A tag t is in this state if t does not know about whether it needs to report its tag-information. All of the tags in a population S are initialized to be in *UNSELECTED STATE*.

- ⋆ *ACKNOWLEDGED STATE*: A tag t enters into this state if t is informed with a unique integer $I \in \{1, 2, ..., K\}$ and needs to report to R of its tag-information t^{Info} in the I-th slot of the subsequent process that collects tag-information.
- ⋆ *INACTIVE STATE*: A tag t enters into this state if t knows that it does not need to report to R of its t^{Info} in the subsequent process that collects tag-information.

Obviously, when protocol P_{s} ends, every tag $t \in S$ is in either *ACKNOWLEDGED STATE* or *INACTIVE STATE*.

4.1 The Design and Analysis of the First Phase

This phase includes one simple communication round where each tag in population S used the parameters sent from reader R to compute a random hash function to determine whether it should remain in the *UNSELECTED STATE* or move to *INACTIVE STATE*. Given N and K, we describe the steps of the first phase below.

Step-1: The reader sends out a request with a random seed r to all tags.
Step-2: Upon receiving this random seed, each tag t computes a random number $h(t) = H(t^{\text{ID}}, r) \mod N$.
Step-3: If $h(t) < K$, tag t shall remain in *UNSELECTED STATE*, otherwise, t will enter the *INACTIVE STATE*.

The function $h()$ in step-2 is a hash function that maps tag t uniformly at random to an integer $h(t) \in \{0, 1, ..., N-1\}$. Thus, on average, there K tags from population S that are mapped to integers ranging from 0 to $K-1$. For some random seeds, there may be more than K or less than K tags that are mapped to $\{0, 1, ..., K-1\}$. However, reader R can avoid using these seeds, because it can pre-test a random seed to see if this seed picks K tags from S or not (R knows all the IDs of the tags in S).

Next, we analyze the theoretical performance of P_{s}-1. We first prove that P_{s}-1 satisfies the first requirement of Definition 1 in the following theorem.

Theorem 2. *Given a population S of N tags, after P_{s}-1 finishes, the probability that any K distinct tags in S are chosen to remain in the UNSELECTED STATE is equal $K!/N^{\underline{K}}$.*

Proof. Let $SP(i_1, i_2, .., i_K)$ represent the probability that tag t_{i_1}, t_{i_2},..., t_{i_K} ($1 \leq i_1 < i_2 < ... < i_K \leq N$) are chosen to remain in the *UNSELECTED STATE* by P_{s}-1. Without loss of generality, we consider the first K tags: t_1, t_2,..., t_K, and analyze $SP(1, 2, .., K)$. Because each tag $t \in S$ is mapped to an integer $h(t)$ in $\{0, 1, ..., K-1\}$ independently and uniformly at random and reader R uses a random see that ensure that exactly K tags have their hashed integer less than K, the following is true:

$$SP(1,2,..,K) = \frac{\mathbf{Pr}\left(h(t_1) < K, h(t_2) < K, ..., h(t_K) < K, h(t_{K+1}) \geq K, ..., h(t_N) \geq K\right)}{\mathbf{Pr}\left(\text{Only } K \text{ tags in } S \text{ have hashed integers less than } K\right)}$$
$$= \frac{(K/N)^K (1-K/N)^{N-K}}{\binom{N}{K}(K/N)^K(1-K/N)^{N-K}} = K!/N^{\underline{K}}. \tag{8}$$

We have the conclusion.

Next, we analyze the communication cost of P_s-1.

Theorem 3. *Given a population S of N tags, during the execution of P_s-1, the reader R needs to send $\log_2(N)$ bits to all tags.*

Proof. The only transmission cost in P_s-1 is for sending out a random seed r. Based on [4], we know a random universal hash function, which requires about $2 \times \log_2(N)$ bits to describe, can perform nearly as a truly random hash function in practice.

4.2 The Design and Analysis of the Second Phase

The second phase P_s-2 targets for the K tags that remain in the *UNSELECTED STATE* by the end of P_s-1, with the aim that each of them is informed of a unique order from $\{1, 2, ...K\}$ (see the second requirement of Definition 1).

P_s-2 consists of a number of communication rounds, each of which will let some tags in *UNSELECTED STATE* enter the *ACKNOWLEDGED STATE* by informing each of them with a unique order from $\{1, 2, ..., K\}$, until no tag is in *UNSELECTED STATE*. Let B represent the set of the K tags that stay in *UNSELECTED STATE* at the end of P_s-1. Detailed steps of P_s-2 are given below.

(1): If B is not an empty set, reader R starts a new communication round by broadcasting a request with two numbers $< |B|, r >$ where $|B|$ is number of tags in B and r is new random seed r.

(2): Upon receiving $<|B|, r>$, each tag t in *UNSELECTED STATE* computes a random number $f(t) = H(t^{\mathrm{ID}}, r) \mod |B|$.

(3): Reader R knows all the IDs in set B, so reader R can also compute the random number $f(t)$ for each tag $t \in B$, and then construct a bit-array F with $|B|$ bits. Each bit $F[j], j \in \{0, 1, .., |B| - 1\}$ is set to $'1'$ if there exists exactly one tag $t \in B$ that has $f(t) = j$; $F[j]$ is set to $'0'$, otherwise.

(4): Reader R broadcasts the bit-array F out.

(5): Upon receiving F, each tag t in *UNSELECTED STATE* checks the $F(f(t))$. Then if $F(f(t)) = 1$ tag t shall take $Cnt(f(t)) + K - |B|$ as its order and enters the *ACKNOWLEDGED STATE*; Otherwise tag t remains in the *UNSELECTED STATE*. Note the $Cnt(f(t))$ represent the number of $'1's$ in the subarray $F[0], F[1], ..., F[f(t)]$.

(6) Reader R deletes those tags that enter the *ACKNOWLEDGED STATE* from B, and go to step (1).

We use an example to explain how P_s-2 informs each tag in B with a unique integer. Suppose B contains 3 tags: t_1, t_2, t_3. In the first round, we assume that the hash values of these 3 tags are: $f(t_1) = 1$, $f(t_2) = 1$ and $f(t_3) = 2$, then R shall build a 3-bit array $F =$ "001" which is broadcasted out to all tags. After receiving $F =$ "001", tag t_3 finds out that $F[f(t_3)] = F[2]$ is equal to $'1'$, thus t_3 will take $Cnt(f(t_3)) + K - |B| = Cnt(2) + 3 - 3 = 1 + 3 - 3$ as its integer and enters the *ACKNOWLEDGED STATE*. The other two tags: t_1 and

t_2 remain in the *UNSELECTED STATE*, as $F[f(t_1)] = F[(f(t_2)] = F[1] =' 0'$. equals $'0'$. At the end of this round, reader R deletes t_3 from B. In the second round, we assume that the hash values of the 2 tags in B are $f(t_1) = 1$ and $f(t_2) = 0$, then R shall build a 2-bit array $F =$"11" and send it out. After receiving $F =$"11", tag t_1 finds out that $F[f(t_1)] = F[1]$ is equal to $'1'$, then t_1 will take $Cnt(f(t_1)) + K - |B| = Cnt(2) + 3 - 2 = 2 + 3 - 2 = 3$ as its integer and enters the *ACKNOWLEDGED STATE*. Similar to tag t_1, t_2 takes $Cnt(f(t_2))+K-|B| = Cnt(1)+3-2 = 1+3-2 = 2$ as its integer and enters the *ACKNOWLEDGED STATE*. At the end of the second round, reader R deletes t_1 and t_2 from B, and then stops P_s-2 because B is empty.

Next, we analyze the theoretical performance of P_s-2. First, it is easy to observe from the above steps that all tags in B moves to the *ACKNOWLEDGED STATE*. Next, we analyze the integers that the K tags in B can get.

Theorem 4. *Each tag $t \in B$ can obtain a unique order from $\{1, 2, ..., K\}$ when t enters the ACKNOWLEDGED STATE.*

Proof. Without loss of generality, we assume that, in the first round, there are l tags $t_{i_1}, t_{i_2}, ..., t_{i_l}$ that enter the *ACKNOWLEDGED STATE*. So there will be l $'1's$ in the bit-array F sent from reader R. Then each of these tags shall have $Cnt(f(t_{i_1})) \neq Cnt(f(t_{i_2})) \neq .. \neq Cnt(f(t_{i_l}))$, and $\forall j \in \{1, 2, ..., l\}$, $Cnt(f(t_{i_j})) \in \{1, 2, ..., l\}$. Thus, we see that each t_{i_j}, $j \in \{1, 2, .., l\}$ gets a unique integer from $\{1, 2, .., K\}$.

Following a similar way, it is easy to prove that each of the l' tags that enter the *ACKNOWLEDGED STATE* in the second round can get a unique integer from $\{l + 1, l + 2, .., l + l'\}$.

Theorem 5. *During P_s-2, readers needs to send about $K \times e + \ln(K) \times 2\log_2(K)$ bits to all tags.*

Proof. In the first round, reader R needs to send two number $|B|$ and r to tags, each of which takes $\log_2(K)$ bits, and a bit-array of $|B|$ bits. Thus, R needs to send $2\log_2(K) + K$ bits.

A tag $t \in B$ is deleted from B in the first round if there does not exist another tag $t' \in B$ that has $f(t) = f(t')$. Then, the probability that t is deleted from B at the end of the first round is $(1-1/|B|)^{K-1} \approx e^{-1}$. Then, on average there are $(1-e^{-1})K$ tags left in B ($|B| = (1-e^{-1})K$) when the second round begins. In the second round, clearly, reader R still needs $2\log_2(K)$ bits for transmitting the new value of $|B|$ and a new random seed r, and then R sends out and a bit-array of $|B| = (1-e^{-1})K$ bits. Thus, R needs to send $2\log_2(K) + (1-e^{-1})K$.

Following a similar analysis, we know that the communication cost for the l-th round is $2\log_2(K) + (1-e^{-1})l$. Since each round reduces the size of B by a fixed proportion $(1-e^{-1})$, then after $\ln(K)$ rounds, B becomes empty. Therefore, the total number of bits that reader R sends to tags is $\ln(K) \times 2\log_2(K) + K + K(1-e^{-1}) + ... + K(1-e^{-1})^{\ln(K)} \leq K \times e + \ln(K) \times 2\log_2(K)$.

4.3 The Analysis of Protocol P_s

This section, we analyze the theoretical performance of protocol P_s.

Theorem 6. *The communication cost of protocol P_s is about 2 times of the lower bound C_{lb} shown in (1).*

Proof. Based on Theorem 3 and 4, we know that protocol P_s achieves the two requirements shown in Definition 1, and then solves the tag-information sampling problem. Next, based on Theorem 2 and 5, we see that protocol P_s has a communication cost of $\log_2(N) + K \times e + \ln(K) \times 2\log_2(K)$ bits. We compare this cost with the lower bound $C_{\text{lb}} = \log_2(e)K$ shown in Theorem 1 and obtain the following.

$$\frac{\log_2(N) + K \times e + \ln(K) \times 2\log_2(K)}{\log_2(e)K} \approx \frac{e}{\log_2(e)} \approx 2. \tag{9}$$

In the above formula, we can use $\log_2(N)/K \approx 0$ because $\log_2(N)$ is usually much less than K in RFID systems. Through (9), we see the communication cost of protocol P_s is within a factor of 2 of the lower bound C_{lb}, which shows the efficiency of P_s.

5 Evaluation

In this section, the performance of the proposed P_s is evaluated and compared with the existing protocols through simulations.

5.1 Simulation Setting

The simulation parameter is set according to the specification of the EPCglobal C1G2 standard [5], which has been used widely to test protocols' performance in RFID systems [11,14,22,27]. Any two consecutive communications between readers and tags are separated by a time frame of 302 µs. The data rate of the link from readers to tags is set to 26.5 kbps. Thus, it takes the reader about 3897 µs to broadcast out a 96-bit array to tags, i.e., $T_{\text{id}} = 3897\,\mu\text{s}$. As tags do not need to transmit any bits back to the reader, the data rate for the link from tags to the reader is not set.

We choose two state-of-the-art protocols which are the minimal Perfect hashing based Information Collection (PIC) in [27] and the Tag-Ordering Polling protocol (TOP) in [19], respectively. Because both of the two protocols require to preset a target T of a tag population S from which they collect the tag-information, we use them to solve the tag-information sampling problem in the following way:

(1) Let the reader R randomly generate a subset of K tags from S locally;
(2) Let R inform these K tags and then collect the tag-information from them by using either PIC or TOP.

Note that we set the length of the Bloom filter to be $24k$ in the following simulations when testing TOP [19]. This parameter setting is according to part A: Energy cost in section IV of [19]

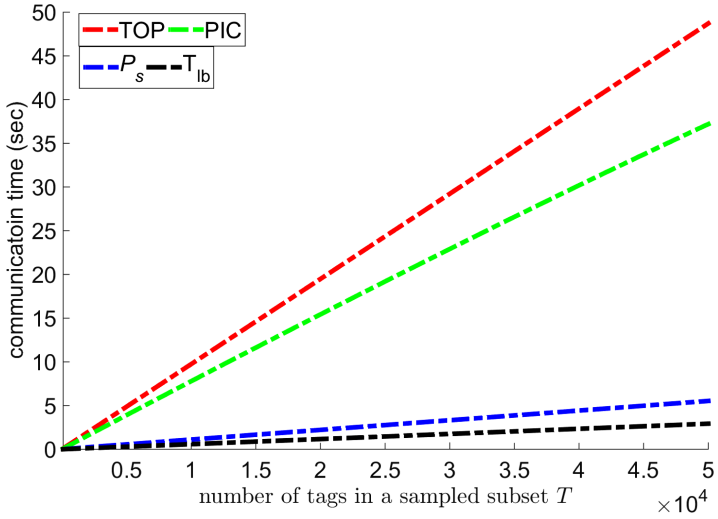

Fig. 2. Protocols' Performance in Scenario 1 where the number of tags in population S is fixed to 10^5 but the number of tags in a sampled subset T varies from 10^2 to 0.5×10^5 ($N = 10^5$ and $K \in [10^2, 0.5 \times 10^5]$).

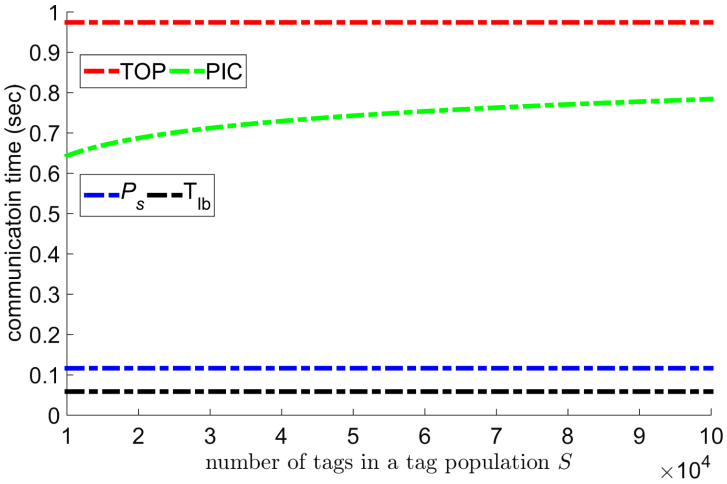

Fig. 3. Protocols' Performance in Scenario 2 where the number of tags in population S varies from 10^4 to 10^5 but the number of tags in a sampled subset is set fixed to 10^3 ($N \in [10^4, 10^5]$ and $K = 10^3$).

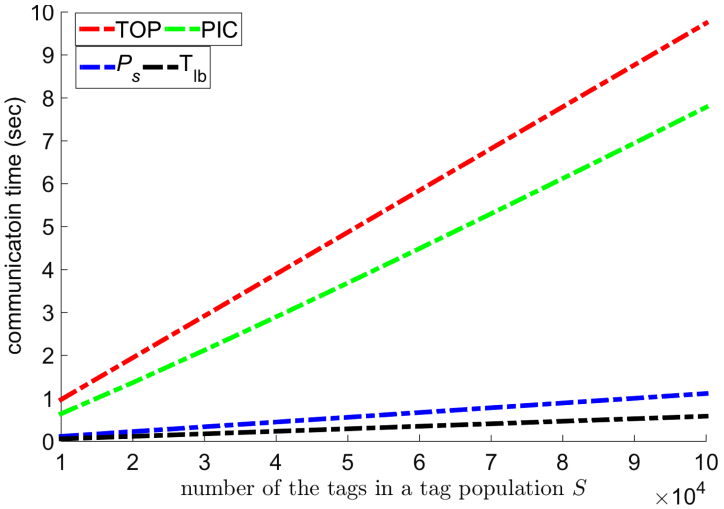

Fig. 4. Protocols' Performance in Scenario 3 where the number of tags in population S varies from 10^4 to 10^5 and the number of tags in a sampled subset is set equal to $0.1 \times N$ ($N \in [10^4, 10^5]$ and $K = 0.1 \times N$).

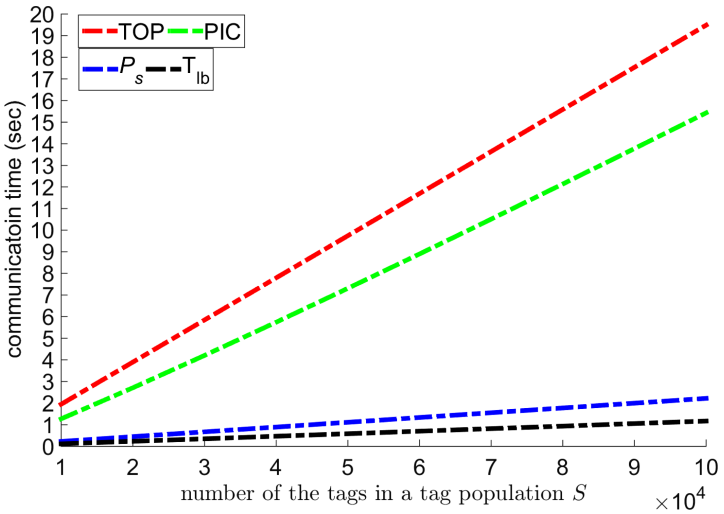

Fig. 5. Protocols' Performance in Scenario 4 where the number of tags in population S varies from 10^4 to 10^5 and the number of tags in a sampled subset is set equal to $0.1 \times N$ ($N \in [10^4, 10^5]$ and $K = 0.2 \times N$).

5.2 Simulation Results

We compare the performance of the three protocols in four different scenarios with different settings for K (the number of tags in a sampled subset T) and N (the number of tags in tag population S). All the simulation results in the following are the average outcome of 100 independent trials using MATLAB. In scenario 1, we set $N = 10^5$ but vary K from 10^2 to $5 * 10^5$. In scenario 2, we fix $K = 10^2$ but vary N from 10^4 to 10^5. For scenario 3, both N and K are varied in a way that $N \in \{10^3, 2 \times 10^3, .., 10^5\}$ and $K = 10^{-1} \times N$. Scenario 4 takes the same setting as scenario 4 except setting K to be equal $2 \times 10^{-1} \times N$. The communication time of the three protocols is presented in Fig. 2, 3, 4 and 5.

We can observe from the experiments that the communication time of P_{s} is the smallest among the three protocols. Indeed, it stays close the lower bound T_{lb}. Taking scenario 1 for example, when $K = 10^3$ and $N = 10^5$, the communication time of P_{s} is on 0.12 s, while the communication times of TOP and PIC are 0.98 s and 0.78s, respectively. This In scenario 3, when $K = 10^3$ and $N = 5 \times 10^4$, the communication time of P_{s} is only 0.56s, while the communication times of TOP and PIC are 4.87 s and 3.68 s, respectively. In summary, the proposed P_{s} has a communication time about 10%–20% of that of the other two protocols. The superiority of P_{s} as compared with other protocols can be explained in two angles. First, protocol P_{s} allows each tag in S to randomly decide to join in T or not (does not require to preset a subset T) locally, while the other protocols let the reader to globally pre-determine a subset of K tags and then send the information of this subset to tags. Secondly, protocol P_{s} smartly transmits a small number of bits by focusing on K tags out randomly chosen from S, while the other protocols need much more bits to take good care of every tag in S and inform them about which tags are in the pre-fixed subset and which are not.

Next, we see the ratio between the communication time of P_{s} and the lower bound T_{lb} in these four scenarios. Take scenario 1 for example, we see P_{s} uses a communication that is about 2 times of T_{lb}. In scenario 4, we also observe that P_{s} uses a communication that is about 2 times of T_{lb}.

6 Conclusion

This paper studies the problem of tag-information sampling problem in RFID systems, and obtains a lower bound of communication cost and proposes an efficient protocol P_{s} for this problem. The proposed protocol P_{s} has a communication cost that is not only much less than that of the state-of-art protocols but also proven to stay within a factor of 2 of the lower bound. Extensive simulations are conducted to evaluate the performance of the propose protocol P_{s}. The results show that P_{s} outperforms the state-of-art protocols.

Acknowledgements. This work was supported in part by the National Natural Sciences Foundation of China under Grants 61402008, 61702006 and 61672038, in part by the Provincial Key Research and Development Program of Anhui Province under Grants 202004a05020009 and 201904a05020071, in part by the Electronic Information

and Control of Fujian University Engineering Research Center, Minjiang University, under Grant MJXY-KF-EIC1803, and in part by the Open Fund of Key Laboratory of Anhui Higher Education Institutes under Grant CS2020-006.

References

1. WISP: Wireless Identification and Sensing Platform (2011). https://sensor.cs.washington.edu/WISP.html
2. Bu, K., Xu, M., Liu, X., Luo, J., Zhang, S., Weng, M.: Deterministic detection of cloning attacks for anonymous rfid systems. IEEE Trans. Industr. Inf. **11**(6), 1255–1266 (2015)
3. Chen, S., Zhang, M., Xiao, B.: Efficient information collection protocols for sensor-augmented rfid networks. In: 2011 Proceedings IEEE INFOCOM. pp. 3101–3109. IEEE (2011)
4. Chung, K.M., Mitzenmacher, M., Vadhan, S.: Why simple hash functions work: exploiting the entropy in a data stream. Theory of Comput. **9**(1), 897–945 (2013)
5. EPCGlobal: EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID Standard, Specification for RFID Air Interface Protocol for Communications at 860 MHz - 960 MHz. Technical Report (2018)
6. Gu, Y., Wang, Y., Liu, Z., Liu, J., Li, J.: Sleepguardian: An rf-basedhealthcare system guarding your sleep from afar. IEEE Network (2020)
7. Hu, H., Liu, Z., An, J.: Mining mobile intelligence for wireless systems: a deep neural network approach. IEEE Comput. Intell. Mag. **15**(1), 24–31 (2020)
8. Li, J., et al.: Psotrack: A rfid-based system for random moving objects tracking in unconstrained indoor environment. IEEE IoT J. **5**(6), 4632–4641 (2018)
9. Li, Q., et al.: Af-dcgan: Amplitude feature deep convolutional gan for fingerprint construction in indoor localization systems. IEEE Transactions on Emerging Topics in Computational Intelligence (2019)
10. Liu, J., Chen, S., Xiao, B., Wang, Y., Chen, L.: Category information collection in rfid systems. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). pp. 2220–2225. IEEE (2017)
11. Liu, J., Chen, S., Xiao, Q., Chen, M., Xiao, B., Chen, L.: Efficient information sampling in multi-category rfid systems. IEEE/ACM Trans. Netwk. **27**(1), 159–172 (2018)
12. Liu, X., et al.: Fast rfid sensory data collection: trade-off between computation and communication costs. IEEE/ACM Trans. Netwk. **27**(3), 1179–1191 (2019)
13. Liu, X., et al.: Efficient unknown tag identification protocols in large-scale rfid systems. IEEE Trans. Parallel Distrib. Syst. **25**(12), 3145–3155 (2014)
14. Liu, X., et al.: Top-k queries for multi-category rfid systems. In: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. pp. 1–9. IEEE (2016)
15. Liu, X., et al.: Efficient range queries for large-scale sensor-augmented rfid systems. IEEE/ACM Trans. Networking **27**(5), 1873–1886 (2019)
16. Liu, X., Yang, Q., Luo, J., Ding, B., Zhang, S.: An energy-aware offloading framework for edge-augmented mobile rfid systems. IEEE IoT J. **6**(3), 3994–4004 (2018)
17. Liu, Z., Ota, K.: Smart technologies for emergency response and disaster management. IGI Global (2017)
18. Mitzenmacher, M., Upfal, E.: Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis. Cambridge university press (2017)

19. Qiao, Y., Chen, S., Li, T., Chen, S.: Tag-ordering polling protocols in rfid systems. IEEE/ACM Trans. Netwk. **24**(3), 1548–1561 (2015)
20. Reyes, P.M., Worthington, W.J., Collins, J.D.: Knowledge management enterprise- and rfid systems. Management Research Review (2015)
21. Shangguan, L., Jamieson, K.: The design and implementation of a mobile rfid tag sorting robot. In: Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services. pp. 31–42 (2016)
22. Wang, X., Liu, Z., Gao, Y., Zheng, X., Dang, Z., Shen, X.: A near-optimal protocol for the grouping problem in rfid systems. IEEE Trans. Mobile Comput. pp. 1–1 (2019). https://doi.org/10.1109/TMC.2019.2962125
23. Wang, X., Liu, Z., Gao, Y., Zheng, X., Chen, X., Wu, C.: Near-optimal data structure for approximate range emptiness problem in information-centric internet of things. IEEE Access **7**, 21857–21869 (2019)
24. Wu, C., Liu, Z., Zhang, D., Yoshinaga, T., Ji, Y.: Spatial intelligence toward trustworthy vehicular iot. IEEE Commun. Mag. **56**(10), 22–27 (2018)
25. Xie, L., Han, H., Li, Q., Wu, J., Lu, S.: Efficiently collecting histograms over rfid tags. In: IEEE INFOCOM 2014-IEEE Conference on Computer Communications. pp. 145–153. IEEE (2014)
26. Xie, L., Sun, J., Cai, Q., Wang, C., Wu, J., Lu, S.: Tell me what i see: Recognize rfid tagged objects in augmented reality systems. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 916–927 (2016)
27. Xie, X., Liu, X., Li, K., Xiao, B., Qi, H.: Minimal perfect hashing-based information collection protocol for rfid systems. IEEE Trans. Mob. Comput. **16**(10), 2792–2805 (2017)