Fast and Accurate Detection of Unknown Tags for RFID Systems – Hash Collisions are Desirable

Xiulong Liu, Sheng Chen, Jia Liu, *Member, IEEE, ACM*, Wenyu Qu, Fengjun Xiao, Alex X. Liu, *Fellow, IEEE*, Jiannong Cao, *Fellow, IEEE,* and Jiangchuan Liu, *Fellow, IEEE*

***Abstract*— Unknown RFID tags appear when tagged items are not scanned before being moved into a warehouse, which can even cause serious security issues. This paper studies the practically important problem of unknown tag detection. Existing solutions either require low-cost tags to perform complex operations or beget a long detection time. To this end, we propose the Collision-Seeking Detection (CSD) protocol, in which the server finds out a collision-seed to make massive known tags hash-collide in the last N slots of a time frame with size f. Thus, all the leading $f - N$ pre-empty slots become useful for detection of unknown tags. A challenging issue is that, computation cost for finding the collision-seed is very huge. Hence, we propose a supplementary protocol called Balanced Group Partition (BGP), which divides tag population into n small groups. The group number n is able to trade off between communication cost and computation cost. We also give theoretical analysis to investigate the parameters to ensure the required detection accuracy. The major advantages of our CSD+BGP are two-fold: (i) it only requires tags to perform lightweight operations, which are widely used in classical framed slotted Aloha algorithms. Thus, it is more suitable for low-cost tags; (ii) it is more time-efficient to detect the unknown tags. Simulation results reveal that CSD+BGP can ensure the required detection accuracy, meanwhile achieving $1.7\times$ speedup in the single-reader scenarios and $3.9\times$ speedup in the multi-reader scenarios than the state-of-the-art detection protocol.**

***Index Terms*— RFID, unknown tag, detection, hash collision.**

Manuscript received December 25, 2018; revised June 28, 2019; accepted November 1, 2019; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Li. Date of publication January 1, 2020; date of current version February 14, 2020. This work was supported in part by National Key R&D Program of China under Grant 2019YFB2102404, in part by NSFC under Grant 61772251, in part by Hong Kong RGC Research Impact Fund under Grant R5034-18, in part by Shenzhen Basic Research Funding Scheme under Grant JCYJ20170818104222072, in part by the National Science Foundation under Grant Number CNS-1837146, in part by a Canada NSERC Discovery Grant, in part by Technology Demonstration Program (TDP) Grant, in part by Key Research and Development Program for Guangdong Province under Grant No. 2019B010136001, and in part by the Science Innovation Foundation of Dalian under Grant 2019J12GX037. *(Corresponding authors: Wenyu Qu; Fengjun Xiao.)*

Xiulong Liu, Sheng Chen, and Wenyu Qu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China (e-mail: wenyu.qu@tju.edu.cn).

Jia Liu is with the Department of Computer Science and Technology, Nanjing University, Nanjing 210008, China.

Fengjun Xiao is with the Management School, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: bhxfj@126.com).

Alex X. Liu is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA.

Jiannong Cao is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

Jiangchuan Liu is with the School of Computing, Simon Fraser University, Burnaby, BC V5A 1S6, Canada.

Digital Object Identifier 10.1109/TNET.2019.2957239

I. INTRODUCTION

A. Motivation and Problem Statement

RADIO Frequency Identification (RFID) is expected to be one of the cornerstone technologies in future smart warehouse management because of its various promising advantages, *e.g.*, long scanning distance, simultaneous identification of multiple objects, and no requirement for line-of-sight [1]–[12]. Normally, the tags in a warehouse should be consistent with tag IDs kept on the server. However, such a desirable consistency cannot be ensured when unregistered tagged items are moved into a warehouse or tagged items are misplaced at incorrect zones. We refer to the tags whose IDs are not available on server as *unknown tags*. The existence of unknown tags may cause serious economic loss or even security issues, *e.g.*, chemical reagent that is misplaced in the food area will contaminate the food items and further threaten the human safety. Clearly, it is practically important to detect the existence of unknown tags in an accurate and time-efficiency way. Hence, this paper studies the problem of *unknown tag detection* [13], [14], which can be formally defined as follows. *Given a set $\mathcal{K}$ of k known tags whose IDs are available on the server in advance, if the number of unknown tags in system exceeds a predefined threshold u, an unknown tag detection protocol needs to discover the existence of unknown tags with a detection probability $\alpha \in (0, 1)$ specified by the user.*

B. Limitations of Prior Art

The existing schemes, which can be used to address the problem of unknown tag detection, are classified into three categories: tag identification protocols [15], [16], unknown tag identification protocols [17], [18], and dedicated unknown tag detection protocols [13], [14], [19], [20]. Next, we will discuss their limitations. **(i)** Tag identification protocols indiscriminately collect IDs of all tags in the RFID system, thus we naturally learn whether there are unknown tags by comparing the set of collected tag IDs with that in database. Although workable, they are seriously time-consuming because massive known tag IDs are re-collected. **(ii)** Unknown tag identification protocols aim at identifying only unknown tag IDs instead of all tag IDs. Although a large amount of time for re-collecting known tag IDs is cut off, they cannot ensure the detection accuracy described in the above problem definition. **(iii)** The state-of-the-art unknown tag detection protocol, White

1063-6692 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

Paper (WP) [14], can satisfy the required detection accuracy. However, it is not suitable for low-cost RFID tags because the operations performed on the tag side are much more complex than that used in the classical framed slotted Aloha algorithm [15]. Moreover, it is still time-consuming particularly in the multi-reader RFID systems.

C. Basic Approach

We propose the Collision-Seeking Detection (CSD) protocol, which follows the framed slotted Aloha algorithm [15]. Specifically, the reader broadcasts parameters $\langle \mathcal{S}, f \rangle$ to initialize a slotted time frame, where $\mathcal{S}$ is a hash seed and f indicates the number of time slots in the forthcoming slotted frame. Then, each tag uses its ID to calculate a hash function $c = \mathcal{H}(ID, \mathcal{S}) \bmod f$ and replies in the c-th slot of the time frame. Since known tag IDs and all hash parameters are available on the server, we can exactly predict which slot each known tag selects. The pre-empty slots (*i.e.*, none of known tags selects) can be used to detect the unknown tags, because any responses observed in them can indicate the existence of unknown tags. Hence, before actually initializing a time frame, the server tests a large number of hash seeds to find out a special collision-seed $\mathcal{S}$, which makes all known tags *hash-collide* in the last N time slots. All the leading $f - N$ slots are expected to be pre-empty, *i.e.*, the ratio of useful pre-empty slots is significantly increased.

D. Technical Challenges and Solutions

We need to solve three technical challenges before completing the design of our detection protocol.

The first challenge is to guarantee the required detection accuracy. As we know, the ratio of pre-empty slots in a time frame, *i.e.*, $\frac{f-N}{f}$, significantly affects the detection accuracy. Hence, a key parameter that we need to investigate is the frame size f. We propose sufficient theoretical analysis to prove that, if the frame size f satisfies $f \geq N(1-\alpha)^{-\frac{1}{u}}$, the proposed CSD protocol can guarantee the required detection accuracy.

The second challenge is to reduce the total detection time. There are two types of costs: *computation cost* for finding out a collision-seed, and *communication cost* for executing the slotted time frame. Although communication cost is very small, we find that the computation cost of CSD is extremely huge. To achieve balance between these two types of costs, we propose a supplementary protocol called Group Partition (GP), which partitions the tag population into n small groups. And the CSD protocol is executed on each small group. Simulation results reveal that the groups partitioned from GP may have different sizes due to probabilistic randomness. Such an unbalance issue makes the actual time-efficiency of CSD+GP far from its ideal case, and motivates us to further propose the enhanced supplementary protocol called Balanced Group Partition (BGP).

The third challenge is to make the proposed CSD+BGP protocol scalable to multi-reader scenarios. A straightforward solution is to use the whole set of known tags $\mathcal{K}$ as the input for each reader. However, it is not time-efficient because the detection time on each reader exponentially increases against the number of known tags it deals with. To improve time-efficiency, we propose to use the slotted time frame as a bloom filter to remove the irrelevant known tag IDs for each reader.

E. Novelty and Advantages Over Prior Art

Unlike previous works [2], [21]–[23], which desire to get rid of the hash collisions, the key novelty of this paper is to deliberately create hash collisions for improving the utilization of time frame when addressing the problem of unknown tag detection. The key technical depth is to guarantee the detection accuracy of CSD+BGP, and optimize the involved parameters to minimize its detection time. The key advantages of our CSD+BGP protocol over previous schemes are two-fold: (i) it only requires tags to perform some lightweight operations, which are widely used in classical framed slotted Aloha algorithms. Thus it is suitable for low-cost tags. The heavy computation tasks, *e.g.*, searching the collision-seed and optimizing the parameters, are performed on the server side. (ii) it is very time-efficient. Compared with the state-of-the-art detection protocol, simulation results reveal that CSD+BGP can achieve $1.7\times$ speedup in the single-reader scenarios and $3.9\times$ speedup in the multi-reader scenarios.

The remainder of this paper is organized as follows. Section II presents the detailed design of CSD. In Sections III and IV, we sequentially present the supplementary protocols GP and BGP. In Section V, we conduct extensive simulations to evaluate the performance of CSD+BGP. We discuss the related work in Section VI. Finally, Section VII concludes this paper.

II. The Basic Protocol: CSD

In this section, we will first present the detailed design of the proposed Collision-Seeking Detection (CSD) protocol. Then, theoretical analysis will be given to guarantee the required detection accuracy of the CSD protocol and minimize the involved time cost as well.

A. Detailed Design of The CSD Protocol

The proposed CSD protocol follows the classical framed slotted Aloha (FSA) algorithm, which is specified in the EPC Global C1G2 standard [24]. Specifically, the reader broadcasts initialization parameters $\langle \mathcal{S}, f \rangle$ to start a slotted time frame, where $\mathcal{S}$ is a hash seed and f indicates the number of slots in the forthcoming slotted time frame. Upon receiving these parameters, each tag resets its slot counter $c \in [0, f-1]$ by using its ID to calculate $c = \mathcal{H}(ID, \mathcal{S}) \bmod f$. Then, the reader broadcasts the `QueryRep` command at the end of each time slot to notify tags to decrement the slot counter c by one. Once the slot counter c of a tag becomes 0, it will reply to the reader with a 1-bit tag response, which is enough to announce its presence in this slot. It can be interpreted as that, the tag replies in the c-th slot, where $c = \mathcal{H}(ID, \mathcal{S}) \bmod f$. A natural assumption made in the unknown tag detection problem is that, we know the IDs of normal tags (called known tags in this paper) in advance. Since the hash parameters are also known, we can predict each slot status in advance. Generally,

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

time frame is expected to contain two types of time slots: *pre-empty slot*, in which no known tag will respond; *pre-busy slot*, in which at least one known tag will respond.

In the above process, the hash function embedded in tags could be MD5, SHA-1, or other lightweight hash functions [25]–[28]. The combination of tag ID and the hash seed $\mathcal{S}$ is used as input to the hash function. For example, if a tag ID is $101\cdots1$ and hash seed is $\underline{110101}$, the input to hash function will be the string $101\cdots1\underline{110101}$. The hash function typically has a property: *"a small modification to the hash input will change the hashing result so extensively that the new hash result appears uncorrelated with the old hash result"* [29]. Hence, the slot-selecting results of different tags can be treated as independence and randomness.

Next, we will explain how to detect the unknown tags in a system. If there is no unknown tag in the system, all pre-empty slots are necessary to be empty eventually. On the contrary, if there are some unknown tags in the system, some pre-empty slots may turn out to be busy slots, *i.e.*, the reader may receive the unexpected tag responses in pre-empty slots. This phenomenon can be used to assert the existence of unknown tags in a system. Clearly, the pre-empty slots are useful for detection of unknown tags, whereas, the pre-busy slots are useless. The existing FSA-based methods randomly use a hash seed $\mathcal{S}$ to initialize the slotted time frame, which results in a low frame utilization. For example, the ratio of pre-empty slots in a time frame is only 36.8%, if we exploit a normal setting that frame size f is equal to the number of known tags [15].

If there is a hash seed $\mathcal{S}$ that can make all known tags hash-collide in a small number of time slots, the frame utilization can be significantly improved. The used hash function normally has the property of *pre-image resistance* [29]. That is, if a hash function $\mathcal{H}(\cdot)$ produced a hash value z, it is difficult to find a value x that exactly hashes to z. Hence, given the target slot index and the hash function, we cannot directly derive the useable hash seed for all target tag IDs. And this paper uses a brute-force searching method to find out such a collision-seed. Specifically, we let the server test a large number of hash seeds until it finds out a special one $\mathcal{S}_c$, which makes all known tags hash-collide in the last N slots. Note that, the seed-searching process is performed on the server side before actually running the framed slotted Aloha protocol. Such a hash seed is called the *collision-seed*, and will be actually used to initialize the forthcoming time frame. As an interesting result, all the first $f-N$ time slots will be pre-empty slots, which are useful for detecting unknown tags in the system. By using the reader to observe the actual status of the first $f-N$ time slots, we can determine whether unknown tags are discovered. After monitoring the first $f-N$ slots, reader terminates the time frame without executing the remaining N slots, because they are definitely the useless pre-busy slots. Table I summarizes the main notations in this paper.

TABLE I
MAIN NOTATIONS USED IN THE PAPER

Notations	Descriptions
$\mathcal{K}$	Set of the known tag IDs in the system.
k	Number of the known tags in the system, *i.e.*, $k=\mathcal{K}$.
u	Tolerant threshold of unknown tags.
α	Required detection probability.
ν	Number of unknown tags in the system.
$\mathcal{S}$	Random seed used to initialize a slotted time frame.
$\mathcal{S}_c$	Collision-seed that makes known tags collide in last N slots.
f	Frame size, *i.e.*, number of slots in a time frame.
n	Number of groups partitioned by the GP protocol.
$\mathcal{H}(\cdot)$	Uniform hash function embedded in each RFID tag.
$\mathcal{G}_i$	The i-th group of tags.
τ_s	Slot duration for a reader to transmit a `SELECT` command.
τ_p	Slot duration for a reader to transmit frame parameters.
τ_r	Slot duration for a tag to transmit a 1-bit response.
t_c	Duration of the clock cycle of the server.
η	Number of clock cycles for calculating and checking a hash.
x	Number of readers in a multi-reader system.
$\mathcal{R}_i$	The i-th reader in a multi-reader system.
$P_{CSD}(\cdot)$	Probability that CSD detects the unknown tag event.
$\mathcal{T}_{CSD}$	Total time cost of the CSD protocol.
$P_{GP}(\cdot)$	Probability that CSD+GP detects the unknown tag event.
$\mathcal{T}_{GP}$	Total time cost of the CSD+GP protocol.

B. Parameter Configuration

In what follows, we will investigate the configuration of the most important parameter f involved in the CSD protocol, which significantly affects the detection accuracy of CSD and its time cost. As we know, the collision-seed $\mathcal{S}_c$ is *special* for the known tags in the system, which makes them hash-collide in the last N slots of a time frame. However, for the unknown tags, the collision-seed $\mathcal{S}_c$ painstakingly found by the server is just the same as a randomly picked hash seed. Hence, using the collision-seed to initialize a time frame, each unknown tag has the equal probability of $\frac{1}{f}$ to be hashed to an arbitrary slot of the time frame. Since there are $f-N$ pre-empty slots in the time frame with size of f, the probability that a certain unknown tag can be discovered by the reader is $\frac{f-N}{f}$. If there are ν unknown tags in the system, we can discover the existence of unknown tags when at least one of them is discovered. Hence, the corresponding detection probability, denoted as $P_{CSD}(\nu)$, can be calculated as follows.

$$P_{CSD}(\nu)=1-\left[1-\left(\frac{f-N}{f}\right)\right]^{\nu}=1-\left(\frac{N}{f}\right)^{\nu} \quad (1)$$

It is easy to observe from Eq. (1) that $P_{CSD}(\nu)$ is a monotonically increasing function with respect to ν. Hence, when $\nu\geq u$, we have $P_{CSD}(\nu)\geq P_{CSD}(u)$. To guarantee the required detection accuracy, *i.e.*, $P_{CSD}(\nu)\geq\alpha$, we only need to ensure the inequality $P_{CSD}(u)=1-(\frac{N}{f})^{u}\geq\alpha$. By solving this inequality, we have $f\geq N(1-\alpha)^{-\frac{1}{u}}$, which is a sufficient condition for ensuring the required detection accuracy. Although a larger frame size f can increase the detection probability, it means more time cost will be involved at the same time. Hence, for guaranteeing the required detection accuracy and meanwhile achieving the maximum time-efficiency of the CSD protocol, we set frame size f to its minimum value, *i.e.*, $f=N(1-\alpha)^{-\frac{1}{u}}$.

III. THE SUPPLEMENTARY PROTOCOL: GP

In this section, we first deeply analyze the performance of the basic CSD protocol and point out its disadvantages, which

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

motivate us to further propose the Group Partition (GP) protocol. Then, we describe the detailed design of the CSD+GP protocol, in which GP is a complementary to the basic CSD protocol. Finally, rigorous theoretical analysis is proposed to optimize the parameters involved in our CSD+GP protocol, thereby guaranteeing the required detection accuracy and minimizing the time cost.

A. Motivation of the GP Protocol

In Section II-B, we have only discussed how to guarantee the detection accuracy of CSD and minimize the frame size, *i.e.*, minimizing the communication cost. In fact, the basic CSD protocol involves two types of time cost: (1) *the computation cost*, denoted by $\mathcal{T}_{CSD}^{comp}$, is the time cost used for searching collision-seed on the server side; and (2) *the communication cost*, denoted by $\mathcal{T}_{CSD}^{comm}$, is the sum of all time slots used for exchanging data between reader and tags.

Next, we first calculate the computation cost of our CSD protocol. A hash seed is a collision-seed if and only if it makes all k known tags hash-collide in the last N time slots of a time frame. And each of the k known tag has the probability $\frac{N}{f}$ to be hashed to one of the last N slots of a time frame with size f. Hence, for an arbitrary hash seed, the probability that it is a collision-seed, denoted by p_c, is calculated as follows.

$$p_c = \left(\frac{N}{f}\right)^k = (1-\alpha)^{\frac{k}{u}} \tag{2}$$

We let the server test λ hash seeds. The probability that we can find out at least one collision-seed among them is $1-(1-p_c)^{\lambda} \approx 1-e^{-\lambda p_c}$, where e is the natural constant. Here, we set this probability as large as 99.9%, and calculate that $\lambda = \frac{7}{p_c}$. That is, we can find out at least one collision-seed with a quite high probability of 99.9% by testing $\frac{7}{p_c}$ hash seeds. Note that, if no collision-seed is found even after testing $\frac{7}{p_c}$ hash seeds, we will continue to test more hash seeds until a collision-seed is found. Fortunately, this is just a small-probability event with the probability as small as 0.1%. Hence, the computation cost $\mathcal{T}_{CSD}^{comp}$ for testing λ hash seeds is calculated as follows.

$$\mathcal{T}_{CSD}^{comp} = \lambda \times k \times \eta \times t_c = \frac{7k\eta t_c}{(1-\alpha)^{\frac{k}{u}}}, \tag{3}$$

where η represents the number of clock cycles required by the server to calculate the hash function $\mathcal{H}(ID, \mathcal{S}) \mod f$ and check whether the hashing result is not less than $f-N$; t_c represents the duration of the clock cycle, which depends on the CPU frequency of the server.

On the other hand, the communication cost of the CSD protocol can be calculated as follows.

$$\begin{aligned}\mathcal{T}_{CSD}^{comm} &= \tau_p + (f-N)\times\tau_r \\ &= \tau_p + N[(1-\alpha)^{-\frac{1}{u}} - 1]\times\tau_r,\end{aligned} \tag{4}$$

where τ_p represents the duration of the time slot for transmitting frame-initialization parameters $\langle \mathcal{S}, f\rangle$ from the reader to tags; and τ_r represents the duration of a time slot for transmitting 1-bit response from a tag to a reader. Then, the total time cost of CSD, denoted by $\mathcal{T}_{CSD}$, can be calculated as follows.

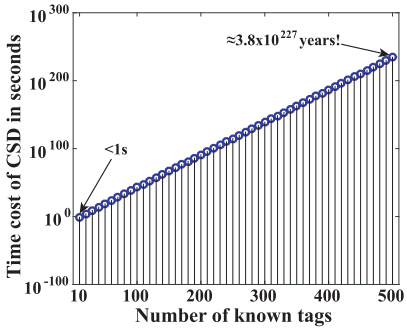


Fig. 1. Impact of the number of known tags on time cost of the basic CSD protocol. $u = 10$, $\alpha = 99\%$, k varies from 10 to 500.

$$\begin{aligned}\mathcal{T}_{CSD} &= \mathcal{T}_{CSD}^{comp} + \mathcal{T}_{CSD}^{comm} \\ &= \frac{7k\eta t_c}{(1-\alpha)^{\frac{k}{u}}} + \left\{\tau_p + N\left[(1-\alpha)^{-\frac{1}{u}} - 1\right]\times\tau_r\right\}\end{aligned} \tag{5}$$

We observe from Eq. (5) that the total time cost of the CSD protocol monotonically increases with respect to the value of N. Hence, we should set the N to its minimum value, *i.e.*, $N = 1$. Thus, we have $f = (1-\alpha)^{-\frac{1}{u}}$, and the total time cost of CSD can be transformed as follows.

$$\begin{aligned}\mathcal{T}_{CSD} &= \mathcal{T}_{CSD}^{comp} + \mathcal{T}_{CSD}^{comm} \\ &= \frac{7k\eta t_c}{(1-\alpha)^{\frac{k}{u}}} + \left\{\tau_p + \left[(1-\alpha)^{-\frac{1}{u}} - 1\right]\times\tau_r\right\} \\ &= 7f^k k\eta t_c + \tau_p + (f-1)\times\tau_r\end{aligned} \tag{6}$$

The above equation indicates that the time cost of CSD *exponentially* increases as the number k of known tags increases. The numerical results in Fig. 1 clearly show this point. For example, the total time cost of CSD is less than $1s$ when $k = 10$; however, it requires an incredibly huge time cost of around 3.8×10^{227} years when $k = 500$. Hence, the basic CSD protocol does not scale well for a large-scale RFID system, which usually contains thousands of known tags. Note that, the detailed settings of τ_p, τ_r, η, and t_c used when getting the results in Fig. 1 will be specified in Section V. The underlying reason to such a huge time cost lies in the high computation complexity for finding the collision-seed, which is $\mathcal{O}(f^k)$ according to Eq. (3). To reduce the computation complexity, we will propose the Group Partition (GP) operation to partition tags in the system into n small-size groups. On each small-size group, we can perform the CSD protocol with a low computation cost $\mathcal{O}(f^{\frac{k}{n}})$. And the total computation cost of all n small groups will be $\mathcal{O}(nf^{\frac{k}{n}})$, which is still significantly smaller than $\mathcal{O}(f^k)$. Liu *et al.* proposed a method in [2] to partition the tag population into n small groups. The basic idea is as follows. The server maps the known tags to a large one-dimension space and finds out $n-1$ appropriate boundary points in the space to generate n ranges. Their expectation is that the number of tags within each range is equal to the average value $\frac{k}{n}$. The tags mapped into the same range are treated as the ones in the same group. Then, the reader broadcasts range parameters to activate

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

the tags in corresponding group. Such a method requires the tags to perform some complex operations (*e.g.*, understanding range parameters), which can be applied on *high-performance* tags, *e.g.*, the sensor-augmented RFID tags considered in [2]. However, it cannot be borrowed in this paper, because we focus on *low-cost RFID tags*.

B. Detailed Design of The CSD+GP Protocol

In what follows, we will describe the detailed design of our CSD+GP protocol. First, we use the Group Partition (GP) protocol to logically partition the whole tag population into n small groups. Specifically, the reader broadcasts the number of groups n and a random seed s to all tags. Each tag calculates $i = \mathcal{H}(ID, s) \mod n$ to determine its group index i, and then stores the obtained group index in its memory. We use $\mathcal{G}_i$ to denote the group i, *i.e.*, the set of tags whose group index equal i, where $i \in [0, n-1]$. Note that, the group $\mathcal{G}_i$ may contain not only the known tags but also some unknown tags, because unknown tags (if there are) also participate in the group partition process. Since the hash parameter s and all known tag IDs are available in advance, we can know the specific IDs of known tags in each group $\mathcal{G}_i$, by virtually executing the group partition process on the server side. We use k_i to denote the number of known tags in group $\mathcal{G}_i$, which obviously satisfies $\sum_{i=0}^{n-1} k_i = k$.

The CSD protocol is executed on n tag groups separately to detect the existence of unknown tags in the system. Specifically, when executing the CSD protocol on a certain group, says $\mathcal{G}_i$, the reader first sends the SELECT command integrated with group index i to activate the tags in this group. On the contrary, the tags in other groups keep silent. Then, the reader initializes a time frame with size of f to detect whether any unknown tags fall in this group. The detailed unknown tag detection processes on each group are the same as that in Section II-A.

Next, we will analyze the computation cost and communication cost corresponding to each group, respectively. For a certain group, says $\mathcal{G}_i$, the server also needs to find a collision-seed to make the known tags in this group hash-collide in the last slot of the corresponding time frame with size f. Similar with Eq. (2), the probability that a randomly picked hash seed is the desired collision-seed for group $\mathcal{G}_i$, denoted by p_c^i, can be calculated as follows.

$$p_c^i = \left(\frac{1}{f}\right)^{k_i}, \tag{7}$$

where k_i is the number of known tags in group $\mathcal{G}_i$ and f is the size of slotted time frame. Using the analysis in Section III-A, we know that, we can find out a collision-seed for the group $\mathcal{G}_i$ with a very high probability 99.9% after testing $\frac{7}{p_c^i}$ seeds. The computation cost of CSD corresponding to group $\mathcal{G}_i$, denoted by $\mathcal{T}_{CSD,\mathcal{G}_i}^{comp}$, can be calculated as follows.

$$\mathcal{T}_{CSD,\mathcal{G}_i}^{comp} = \frac{7}{p_c^i} \times k_i \times \eta \times t_c = 7f^{\frac{k}{n}} k\eta t_c/n \tag{8}$$

In Eq. (8), the term of k_i is replaced by $\frac{k}{n}$ because each group is expected to contain $\frac{k}{n}$ known tags on average. Consequently, the term of p_c^i is replaced by $(\frac{1}{f})^{\frac{k}{n}}$.

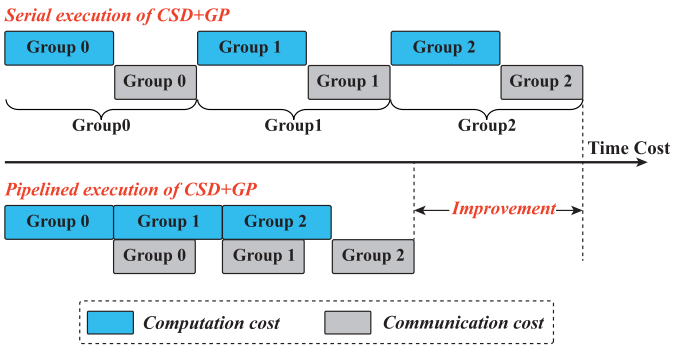


Fig. 2. Performing CSD+GP in the serial mode *vs.* that in the pipelined mode.

On the other hand, the communication cost for performing the CSD protocol on group $\mathcal{G}_i$ contains not only the time for transmitting the SELECT command and frame initialization parameters from reader to tags, but also the time for executing f-slot time frame. Hence, the communication cost for performing the CSD protocol on tag group $\mathcal{G}_i$, denoted by $\mathcal{T}_{CSD,\mathcal{G}_i}^{comm}$, can be calculated as follows.

$$\mathcal{T}_{CSD,\mathcal{G}_i}^{comm} = \tau_s + \tau_p + (f-1) \times \tau_r, \tag{9}$$

where τ_s is the length of a slot for transmitting a SELECT command from the reader to tags; τ_p is the length of a slot for transmitting frame initialization parameters from RFID reader to tags; τ_r is the duration of each slot in the time frame. By jointly considering Eq. (8) and (9), the total time cost of performing the CSD protocol on tag group $\mathcal{G}_i$, denoted as $\mathcal{T}_{CSD}^{\mathcal{G}_i}$, is calculated as follows.

$$\begin{aligned}\mathcal{T}_{CSD}^{\mathcal{G}_i} &= \mathcal{T}_{CSD,\mathcal{G}_i}^{comp} + \mathcal{T}_{CSD,\mathcal{G}_i}^{comm} \\ &= 7f^{\frac{k}{n}} k\eta t_c/n + \tau_s + \tau_p + (f-1) \times \tau_r \end{aligned} \tag{10}$$

A simple way of performing the CSD+GP protocol is to perform the CSD protocol on n tag groups one by one. The total time for performing CSD+GP in such a *serial* manner, denoted by $\mathcal{T}_{GP}^{s}$, can be calculated as follows.

$$\mathcal{T}_{GP}^{s} = \sum_{i=0}^{n-1} \mathcal{T}_{CSD}^{\mathcal{G}_i} = \sum_{i=0}^{n-1} \left(\mathcal{T}_{CSD,\mathcal{G}_i}^{comp} + \mathcal{T}_{CSD,\mathcal{G}_i}^{comm}\right) \tag{11}$$

Inspired by [2], we can perform the CSD+GP protocol in a *pipelined* manner. As illustrated in Fig. 2, when executing the time frame for tag group $\mathcal{G}_j$ to detect whether there are any unknown tags in this group, we can start to find the collision-seed for the next group $\mathcal{G}_{j+1}$ at the same time, where $j \in [0, n-2]$. We use $\mathcal{T}_{GP}^{p}$ to represent the total time cost for performing the CSD+GP protocol in such a pipelined manner, which can be calculated as follows.

$$\begin{aligned}\mathcal{T}_{GP}^{p} = \mathcal{T}_{CSD,\mathcal{G}_0}^{comp} + \sum_{j=1}^{n-1} \max\left\{\mathcal{T}_{CSD,\mathcal{G}_j}^{comp}, \mathcal{T}_{CSD,\mathcal{G}_{j-1}}^{comm}\right\} \\ + \mathcal{T}_{CSD,\mathcal{G}_{n-1}}^{comm}\end{aligned} \tag{12}$$

In what follows, we will compare the time cost of performing the CSD+GP protocol in the serial mode and that in the

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

pipelined mode. Hence, we use Eqs. (11)(12) to calculate the difference between $\mathcal{T}_{GP}^{s}$ and $\mathcal{T}_{GP}^{p}$ as follows.

$$\mathcal{T}_{GP}^{s}-\mathcal{T}_{GP}^{p}=\sum_{j=1}^{n-1}\Big(\mathcal{T}_{CSD,\mathcal{G}_j}^{comp}+\mathcal{T}_{CSD,\mathcal{G}_{j-1}}^{comm}-\max\left\{\mathcal{T}_{CSD,\mathcal{G}_j}^{comp},\mathcal{T}_{CSD,\mathcal{G}_{j-1}}^{comm}\right\}\Big)$$

We observe from the above equation that, the difference $\mathcal{T}_{GP}^{s}-\mathcal{T}_{GP}^{p}$ is always larger than 0, which means that it is more time-efficient to perform the CSD+GP protocol in the *pipelined* mode. Hence, in the remainder of this paper, the proposed CSD+GP protocol works in the pipelined manner by default.

C. Parameter Configuration

The number of groups n and the used frame size f significantly affect the performance of the CSD+GP protocol in terms of both detection accuracy and time-efficiency. Hence, we will propose rigorous theoretical analysis to optimize these parameters in the following.

1) Guarantee the Detection Accuracy: The most fundamental performance metric of an unknown tag detection protocol is its real detection accuracy. For an arbitrary unknown tag, it will be assigned into one of the n tag groups, says group $\mathcal{G}_i$, by the group partition operation. For group $\mathcal{G}_i$, the ratio of pre-empty slots in the corresponding time frame is $\frac{f-1}{f}$. Hence, this unknown tag has the probability of $\frac{f-1}{f}$ to be detected. We can report the existence of unknown tags when at least one of ν unknown tags is detected. Hence, the probability that our CSD+GP protocol can detect the existence of unknown tags in the system, denoted by $P_{GP}(\nu)$, can be calculated as follows.

$$P_{GP}(\nu)=1-\left(1-\frac{f-1}{f}\right)^{\nu}=1-\left(\frac{1}{f}\right)^{\nu} \tag{13}$$

We can observe from Eq. (13) that the detection probability $P_{GP}(\nu)$ is a monotonically increasing function with respect to ν. Hence, we have $P_{GP}(\nu)\geq P_{GP}(u)$ when $\nu\geq u$. To satisfy the detection accuracy that $P_{GP}(\nu)\geq\alpha$, we only need to guarantee $P_{GP}(u)\geq\alpha$. Solving the inequality, we still have $f\geq(1-\alpha)^{-\frac{1}{u}}$, which coincides with our previous analytics.

2) Minimize the Time Cost: It is easy to observe from Eqs. (8)(9) that the computation cost $\mathcal{T}_{CSD,\mathcal{G}_i}^{comp}$ and computation cost $\mathcal{T}_{CSD,\mathcal{G}_i}^{comm}$ are both monotonically increasing functions with respect to frame size f. Hence, we can assert that the total time cost of our CDS+GP protocol, *i.e.*, $\mathcal{T}_{GP}^{p}$ in Eq. (12), is also a monotonically increasing function against the frame size f. Therefore, we should set the frame size f to its minimum integer value, *i.e.*, $f=\lceil(1-\alpha)^{-\frac{1}{u}}\rceil$.

According to Eq. (8), we know that the computation complexity of the CSD+GP protocol for finding the collision-seeds for each group $\mathcal{G}_i$ is $\mathcal{O}(f^{\frac{k}{n}})$. To reduce the computation complexity, n should be set to a relatively large value. Then, the time cost of our CSD+GP protocol, *i.e.*, $\mathcal{T}_{GP}^{p}$ in Eq. (12), can be approximated as follows.

$$\mathcal{T}_{GP}^{p}\approx\widetilde{\mathcal{T}}_{GP}^{p}=\sum_{j=1}^{n-1}\max\left\{\mathcal{T}_{CSD,\mathcal{G}_j}^{comp},\mathcal{T}_{CSD,\mathcal{G}_{j-1}}^{comm}\right\} \tag{14}$$

Substituting Eqs. (8)(9) into the above expression, $\widetilde{\mathcal{T}}_{GP}^{p}$ can be transformed as follows.

$$\begin{aligned}\widetilde{\mathcal{T}}_{GP}^{p}&=(n-1)\times\max\left\{\tfrac{7f^{\frac{k}{n}}k\eta t_c}{n},\tau_s+\tau_p+(f-1)\times\tau_r\right\}\\&=\max\left\{\tfrac{7f^{\frac{k}{n}}k\eta t_c(n-1)}{n},\left[\tau_s+\tau_p+(f-1)\times\tau_r\right](n-1)\right\}\end{aligned} \tag{15}$$

In what follows, we propose Theorem 1 to prove that, we can obtain the optimal number n_o of groups that minimizes the time cost $\widetilde{\mathcal{T}}_{GP}^{p}$, by solving Eq. (16). Note that, if the value of n_o is not an integer, we will use its nearest integer.

Theorem 1: Given the number of known tags k, tolerance threshold u, and required detection probability α, the optimal number of groups n_o, which minimizes the approximate time cost $\widetilde{\mathcal{T}}_{GP}^{p}$, should satisfy the following equation.

$$\frac{7\lceil(1-\alpha)^{-\frac{1}{u}}\rceil^{\frac{k}{n_o}}k\eta t_c}{n_o}=\tau_s+\tau_p+\left(\left\lceil(1-\alpha)^{-\frac{1}{u}}\right\rceil-1\right)\times\tau_r \tag{16}$$

Proof: We treat the approximate time cost $\widetilde{\mathcal{T}}_{GP}^{p}$ in Eq. (15) as a function of n and transform it as $\widetilde{\mathcal{T}}_{GP}^{p}(n)=\max\{\mathcal{P}(n),\mathcal{C}(n)\}$, where $\mathcal{P}(n)=\frac{7f^{\frac{k}{n}}k\eta t_c(n-1)}{n}$ and $\mathcal{C}(n)=[\tau_s+\tau_p+(f-1)\times\tau_r]\times(n-1)$. Here, we first assume that, both $\mathcal{P}(n)$ and $\mathcal{C}(n)$ are continuous functions with respect to the number of groups n, which is within the range of $[2,+\infty]$. When $n=2$, we have $\mathcal{P}(2)=3.5f^{\frac{k}{2}}k\eta t_c$ and $\mathcal{C}(2)=\tau_s+\tau_p+(f-1)\times\tau_r$. In a large practical RFID system, we normally have $k>50$ and $f\geq2$. Substituting the values of τ_s, τ_p, τ_r, t_c, and η into $\mathcal{P}(2)$ and $\mathcal{C}(2)$, we have $\mathcal{P}(2)>420f$ and $\mathcal{C}(2)=6.028\times10^{-4}f+0.0012$. Clearly, we have $\mathcal{P}(n)>\mathcal{C}(n)$ when $n=2$. On the other hand, when $n\to+\infty$, we have $\lim\mathcal{P}(n)=7k\eta t_c$, and $\lim\mathcal{C}(n)=+\infty$. Hence, we have $\mathcal{P}(n)<\mathcal{C}(n)$ when $n\to+\infty$. Based on the above analysis, we assert that functions $\mathcal{P}(n)$ and $\mathcal{C}(n)$ definitely have an intersection point, *i.e.*, there exists $n_o\in[2,+\infty)$, which satisfies $\mathcal{P}(n_o)=\mathcal{C}(n_o)=\ell$. Then, we have $\widetilde{\mathcal{T}}_{GP}^{p}(n_o)=\max\{\mathcal{P}(n_o),\mathcal{C}(n_o)\}=\ell$.

We calculate the first-order derivatives of $\mathcal{P}(n)$ and $\mathcal{C}(n)$ with respect to n, respectively.

$$\begin{aligned}\frac{\partial\mathcal{P}(n)}{\partial n}&=\frac{7k\eta t_c f^{\frac{k}{n}}}{n^2}\left[1-k\ln f\times\frac{n-1}{n}\right]\\\frac{\partial\mathcal{C}(n)}{\partial n}&=\tau_s+\tau_p+(f-1)\times\tau_r\end{aligned} \tag{17}$$

Since k is normally a large number in practice, $k\ln f\times\frac{n-1}{n}$ should be larger than 1. Hence, we have $\frac{\partial\mathcal{P}(n)}{\partial n}<0$, which means that $\mathcal{P}(n)$ is a monotonously decreasing function with respect to n. On the contrary, since $\frac{\partial\mathcal{C}(n)}{\partial n}$ is always larger than 0, $\mathcal{C}(n)$ is a monotonously increasing function with respect to n. Then, for an arbitrary group number $n>n_o$, we can have $\mathcal{P}(n)<\mathcal{P}(n_o)=\ell$ and $\mathcal{C}(n)>\mathcal{C}(n_o)=\ell$. Thus, we have that $\widetilde{\mathcal{T}}_{GP}^{p}(n)=\mathcal{C}(n)>\ell$. On the other hand, for an arbitrary group number $n<n_o$, we can have $\mathcal{P}(n)>\mathcal{P}(n_o)=\ell$ and $\mathcal{C}(n)<\mathcal{C}(n_o)=\ell$. Thus, we have that $\widetilde{\mathcal{T}}_{GP}^{p}(n)=\mathcal{P}(n)>\ell$. Based on the above analysis, we can assert that $\widetilde{\mathcal{T}}_{GP}^{p}(n)$ achieves its minimum value ℓ when group number $n=n_o$. That is, n_o is the optimal

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

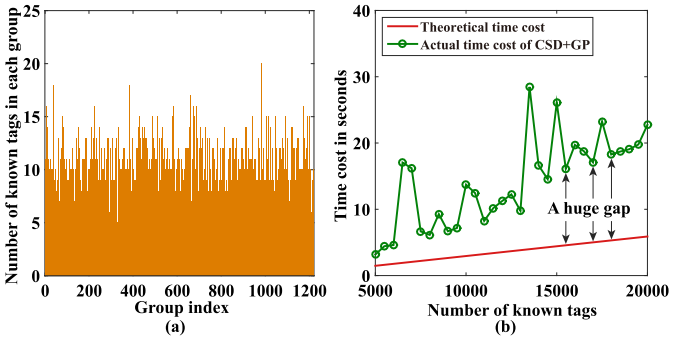


Fig. 3. Performance of CSD+GP. (a) Number of known tags in each group, $k = 10000$, $n = 1220$. (b) Actual time cost of CSD+GP *vs.* theoretical time cost, k varies from 5000 to 20000, $u = 10$, and $\alpha = 99\%$.

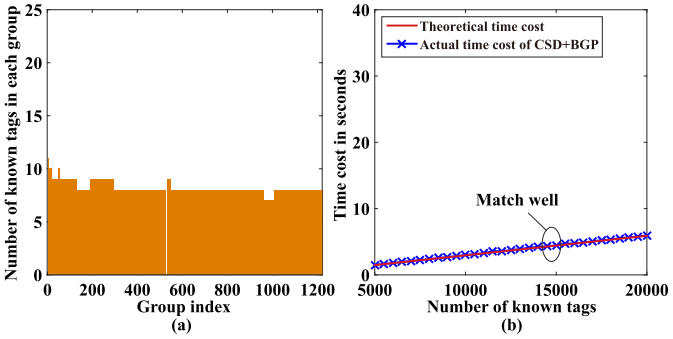


Fig. 4. Performance of CSD+BGP. (a) Number of known tags in each group, $k = 10000$, $n = 1220$. (b) Actual time cost of CSD+BGP *vs.* theoretical time cost, k varies from 5000 to 20000, $u = 10$, and $\alpha = 99\%$.

number of groups, which satisfies $\mathcal{P}(n_o) = \mathcal{C}(n_o)$. By solving this equation, we obtain Eq. (16) in the theorem statement. □

IV. The Enhanced Supplementary Protocol: BGP

In this section, we will first point out the unbalance issue inherent in the GP protocol, which motivates us to further propose the Balanced Group Partition (BGP) protocol. Then, we present the detailed design of CSD+BGP, and also give some numerical results to show its advantage over CSD+GP. Finally, we extend CSD+BGP to multi-reader RFID systems.

A. Motivation of the BGP Protocol

In the previous GP protocol, k known tags are randomly hashed into n groups. For easy understanding, we assume that, each tag group exactly contains $\frac{k}{n}$ known tags when analyzing and optimizing the performance of the CSD+GP protocol in Section III-B. However, we observe from the simulation results in Fig. 3(a) that, the number of known tags in each group differs greatly. This phenomenon is caused by the probabilistic nature of the group partition processes in the GP protocol. Such kind of unbalance issue leads to huge computation overhead for large-size groups because the computation cost *exponentially* increases with respect to the number of known tags in a group. Thus, the time-efficiency of CSD+GP will seriously deteriorate. The numerical results in Fig. 3(b) reveal that, the actual time cost of our CSD+GP protocol intensely fluctuates and consistently keeps much larger than the theoretical value. For better performance in detection of unknown tags, we propose the Balanced Group Partition (BGP) protocol to achieve relatively balanced tag distribution among groups.

B. Detailed Design of the CSD+BGP Protocol

We propose the BGP protocol by making some simple but very effective modifications to the GP protocol. Specifically, if we want to partition the tags into n groups eventually, we will first invoke the GP protocol to partition the tags into $2n$ groups: $\mathcal{V}_0, \mathcal{V}_1, \cdots, \mathcal{V}_{2n-1}$. As aforementioned, we are able to know which known tags are within each group. Hence, we are able to know the number of known tags in each group. We say group $\mathcal{V}_i$ is larger (or smaller) than group $\mathcal{V}_j$, if $\mathcal{V}_i$ contains more (or less) known tags than $\mathcal{V}_j$. For each $i \in [0, n-1]$, we choose the *i-th largest* group and the *i-th smallest* group from these $2n$ groups, and then *logically* merge them to obtain the group $\mathcal{G}_i$. Thus, we get n logical groups: $\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_n$. Intuitively, such n tag groups should be much more balanced than that obtained from the GP protocol, because BGP pairs a large group with a small group. For an arbitrary logical group $\mathcal{G}_i$, we assume it is obtained by merging groups $\mathcal{V}_x$ and $\mathcal{V}_y$. When performing CSD on the group $\mathcal{G}_i$, we let the reader send a `SELECT` command incorporated with the group indexes x and y to simultaneously activate the tags in both groups $\mathcal{V}_x$ and $\mathcal{V}_y$. Thus, the tags in virtual group $\mathcal{G}_i$ will participate in the CSD protocol, as what we expect.

We conduct a set of simulations to validate the effectiveness of the proposed BGP protocol. And the numerical results in Fig. 4(a) shows that, the tag distribution among n groups is much more balanced than that in Fig. 3(a). We calculate the following *Jain's fairness index* [30] to quantitively evaluate the balance of tag distribution corresponding to Fig. 3(a) and Fig. 4(a), respectively.

$$\mathcal{J} = \frac{(\sum_{i=0}^{n-1} |\mathcal{G}_i|)^2}{n \cdot \sum_{i=0}^{n-1} |\mathcal{G}_i|^2} \tag{18}$$

The value of $\mathcal{J}$ in Eq. (18) ranges from $\frac{1}{n}$ (worst case) to 1 (best case). If n groups are fully balanced (*i.e.*, $|\mathcal{G}_i| = \frac{k}{n}$ for each $i \in [0, n-1]$), $\mathcal{J}$ will achieve its maximum value 1. We find that, compared with GP, the proposed BGP protocol can significantly increase the value of $\mathcal{J}$ from 0.8936 to 0.9957. These two values of $\mathcal{J}$ correspond to tag distribution in Fig. 3(a) and Fig. 4(a), respectively. Moreover, the simulation results in Fig. 4(b) reveal that, the actual time cost of the proposed CSD+BGP protocol matches well with the theoretical time cost. That is, the BGP protocol performs much better than the previous GP protocol. Hence, we will use the BGP protocol as a complementary to our basic CSD, and use CSD+BGP as the final unknown tag detection protocol in rest of this paper.

C. Multi-Reader RFID Systems

A practical RFID application scenario (*e.g.*, a warehouse) usually has hundreds or even thousands of square meters. To seamlessly cover such a large area, we usually need to

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

deploy multiple readers $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_x$, because a single reader only has a limited interrogation range (normally less than 10 meters). In a multi-reader RFID system, if two or more adjacent RFID readers query tags simultaneously, the tags locating in their overlapping region cannot successfully receive any commands due to signal corruption. To avoid such reader-conflict, we need to effectively schedule the readers. Since many effective reader-scheduling methods [31], [32] were proposed, we will not make more efforts on this aspect. In this paper, we mainly focus on how to efficiently perform the CSD+BGP protocol on each individual reader. When executing CSD+BGP on an arbitrary reader, says $\mathcal{R}_i$, a straightforward solution is to use the whole set of known tags $\mathcal{K}$, parameters u and α as the protocol inputs. The parameters f and n corresponding to reader $\mathcal{R}_i$ are configured as described in Section III-C. In the following, we will first propose some theoretical analysis to point out the deficiency of this straightforward solution. Then, we will use the bloom filter technique to propose a much more efficiency solution for executing the CSD+BGP in the multi-reader RFID systems.

In Corollary 2, we prove that the time cost of our CSD+BGP protocol is a linearly increasing function with respect to the number of known tags it deal with. If we simply use the large universal set $\mathcal{K}$ as the protocol input for an arbitrary reader $\mathcal{R}_i$, it inevitably leads to a huge time cost on this reader. In fact, the reader $\mathcal{R}_i$ cannot cover all tags but just a small subset of tags in the system, which is represented as $\mathcal{K}_i$. We let the reader $\mathcal{R}_i$ execute a slotted time frame, and use the monitored time frame as a bloom filter to exclude most irrelevant known tag IDs in $\mathcal{K}-\mathcal{K}_i$, and thus obtaining a much smaller set of known tags $\mathcal{K}_i{}'$. It satisfies $\mathcal{K} \supseteq \mathcal{K}_i{}' \supseteq \mathcal{K}_i$. Specifically, we set $\mathcal{K}_i{}' = \mathcal{K}$ at the very beginning. Then, the reader $\mathcal{R}_i$ uses an arbitrarily picked hash seed δ and frame size ℓ to perform the *framed slotted Aloha* protocol on the tags within its interrogation range. Each tag will respond in the slot with index of $\mathcal{H}(ID, \delta) \mod \ell$. According to the slot status, we can obtain an ℓ-bit bloom filter, in which bit 0 represents an empty slot; and bit 1 represents a busy slot. Such a bloom filter can be used to determine whether a known tag is within the interrogation range of $\mathcal{R}_i$ or not. The details are as follows. We calculate the above hash function for each known tag ID in $\mathcal{K}_i{}'$, if a tag ID is hashed to bit 0, this is definitely not within the interrogation region of $\mathcal{R}_i$, because the reader does not receive its response in that slot. Then, the tags that are hashed to bits 0s will be removed from the set $\mathcal{K}_i{}'$.

We use $\aleph_i$ to denote the number of tags that are actually within the interrogation region of reader $\mathcal{R}_i$, which is unknown by us. We can use the existing tag cardinality estimation protocol [33] to accurately estimate the value of $\aleph_i$ with a very small overhead (*e.g.*, about 1 second). An arbitrary bit in the bloom filter is 0 if and only if none of the $\aleph_i$ tags chooses the corresponding time slot. Hence, the probability that a certain bit in the ℓ-bit bloom filter is 0 can be calculated as $(1-\frac{1}{\ell})^{\aleph_i}$, which is obviously equal to the ratio of bits 0s in the binary bloom filter. For each irrelevant known tag ID in $\mathcal{K}-\mathcal{K}_i$, it has the probability $(1-\frac{1}{\ell})^{\aleph}$ to be excluded from $\mathcal{K}_i{}'$. Hence, the number of known tag IDs remaining in the set $\mathcal{K}_i'$ is expected to be $|\mathcal{K}| - |\mathcal{K}-\mathcal{K}_i| \times (1-\frac{1}{\ell})^{\aleph_i}$, which can be much smaller than the universal set size $|\mathcal{K}|$. Intuitively, the detection time cost on reader $\mathcal{R}_i$ could be significantly reduced because we use a much smaller known tag set $\mathcal{K}_i{}'$ instead of the whole known tag set $\mathcal{K}$. However, the above *bloom filtering* process is not cost-free, and the corresponding time cost, denoted by $\mathcal{T}_{BF}^{\mathcal{R}_i}$, is calculated as follows.

$$\mathcal{T}_{BF}^{\mathcal{R}_i} = \tau_p + \ell \times \tau_r \tag{19}$$

On the other hand, according to Corollary 2, the time cost of performing CSD+BGP on reader $\mathcal{R}_i$, denoted as $\mathcal{T}_{BGP}^{\mathcal{R}_i}$, can be calculated as follows.

$$\mathcal{T}_{BGP}^{\mathcal{R}_i} = \frac{\Phi(u,\alpha)}{\phi(u,\alpha)} \times \left\{ |\mathcal{K}| - |\mathcal{K}-\mathcal{K}_i| \times \left(1-\frac{1}{\ell}\right)^{\aleph_i} \right\} + \Phi(u,\alpha), \tag{20}$$

where $\Phi(u,\alpha) > 0$ and $\phi(u,\alpha) > 0$ can be directly calculated by the values of u and α. Details about $\Phi(u,\alpha)$ and $\phi(u,\alpha)$ can be found in Corollary 2. According to Eq. (20), it is easy to find that, a larger bloom filter length ℓ can help reduce more time cost for performing CSD+BGP on the reader $\mathcal{R}_i$. However, according to Eq. (19), a larger bloom filter length ℓ also means a longer time frame should be executed to remove the irrelevant known tags. Fundamentally, the bloom filter length ℓ trades off between two types of time costs, *i.e.*, $\mathcal{T}_{BF}^{\mathcal{R}_i}$ and $\mathcal{T}_{BGP}^{\mathcal{R}_i}$. Due to the space limitation, we do not investigate sophisticated method to optimize the bloom filter length ℓ for each reader in this paper. A easy solution is to enumerate possible values of bloom filter length in a feasible space to find the optimal one that minimizes the total time cost on each reader $\mathcal{R}_i$, *i.e.*, $\mathcal{T}_{Total}^{\mathcal{R}_i} = \mathcal{T}_{BF}^{\mathcal{R}_i} + \mathcal{T}_{BGP}^{\mathcal{R}_i}$. Such a simple solution only takes linear computation cost.

Besides maximizing the time-efficiency of CSD+BGP, we also need to discuss whether its detection accuracy is still guaranteed in the multi-reader system. The detailed theoretical analysis about unknown tag detection accuracy of CSD+BGP in a multi-reader system is given as follows. As a matter of fact, an arbitrary unknown tag should locate in either the exclusive region of a reader or the overlapping region shared by multiple readers. Hence, this tag will participate in the detection process *at least once*. Given a frame size f in the CSD+BGP protocol, the probability that this unknown tag can be detected out is not less than $\frac{f-1}{f}$. If there are u unknown tags in the system, the probability that we can discover the existence of unknown tags will be $1-(1-\frac{f-1}{f})^u = 1-(\frac{1}{f})^u$ at least. In Section III-C.2, the frame size should satisfy $f = \lceil (1-\alpha)^{-\frac{1}{u}} \rceil$. Substituting the value of f into $1-(\frac{1}{f})^u$, we have that the detection probability is not less than α, *i.e.*, we can still ensure the required detection accuracy of CSD+BGP in the multi-reader RFID systems.

Corollary 2: Given the tolerance threshold u, detection probability α, the time cost of our CSD+BGP protocol denoted as $\mathcal{T}_{BGP}$ is a linearly increasing function with respect to the number k of known tags, i.e., $\mathcal{T}_{BGP} = \frac{\Phi(u,\alpha)}{\phi(u,\alpha)} \times k + \Phi(u,\alpha)$, *where $\Phi(u,\alpha)$ and $\phi(u,\alpha)$ can be directly calculated by values of u and α.*

Proof: We observe from Eq. (16) that the value of $\frac{k}{n_{op}}$ closely depends on the values of u and α. Hence, we treat $\frac{k}{n_{op}}$

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

as a function of u and α, *i.e.*, $\frac{k}{n_{op}} = \phi(u, \alpha)$, which makes Eq. (16) hold on. Jointly considering Eqs. (12)(16), the time cost of CSD+BGP can be further transformed as follows.

$$\mathcal{T}_{BGP} = [\tau_s + \tau_p + (f-1) \times \tau_r] \times (n_{op} + 1) \quad (21)$$

Replacing f by $\lceil (1-\alpha)^{-\frac{1}{u}} \rceil$ and n_{op} by $\frac{k}{\phi(u,\alpha)}$, we have:

$$\mathcal{T}_{BGP} = \left[\tau_s + \tau_p + \left(\left\lceil (1-\alpha)^{-\frac{1}{u}} \right\rceil - 1\right)\tau_r\right] \times \left[\frac{k}{\phi(u,\alpha)} + 1\right] \quad (22)$$

For achieving a clear presentation, we use $\Phi(u, \alpha)$ to represent the complex expression $[\tau_s + \tau_p + (\lceil (1-\alpha)^{-\frac{1}{u}} \rceil - 1) \times \tau_r]$, and substitute it into the above equation. Then, we can obtain the corollary statement that $\mathcal{T}_{BGP} = \frac{\Phi(u,\alpha)}{\phi(u,\alpha)} \times k + \Phi(u, \alpha)$. □

V. Performance Evaluation

In this section, we will first briefly describe the benchmark protocols that we will compare with the proposed protocol. Then, the simulation settings will be specified. After that, we will evaluate the time-efficiency and accuracy of our protocol in both single-reader and multi-reader scenarios.

A. Benchmark Schemes

Six representative protocols are briefly described as follows.

• Enhanced Dynamic Framed Slotted Aloha (EDFSA) [15]: It is a well known Aloha-based tag identification scheme. In EDFSA, tags reply IDs in the slots randomly selected from a time frame. The reader can successfully receive a tag ID in a slot if only one tag replies in this slot. Frames are repeated until all tags are identified.

• Tree Hopping (TH) [16]: It is an advanced tree-based tag identification scheme. In TH, the reader estimates the cardinality of the unidentified tags, and then uses the query string with an optimal length to identify tags. If a tag finds the queried string is the prefix of its ID, it will reply its ID to the reader. The reader can identify a tag ID if only one tag replies the ID. The reader tries different query strings to identify all tags. The key point in [16] is how to reduce the number of transmitted query strings as much as possible.

• Collect Unknown-tag (CU) [17]: It is a representative scheme for probabilistic unknown tag identification. In CU, the reader initializes a slotted time frame to query all tags. The tags reply in the expected empty slots should be definitely unknown tags. Then, the reader sends a special command at the end of such kind of slots to label these unknown tags. Multiple frames are repeated to make the ratio of labeled unknown tags meet a required level. Then, a tag identification protocol is invoked to identify the labeled unknown tags.

• Basic Unknown tag Identification Protocol (BUIP) [18]: It is a representative scheme for complete unknown tag identification, *i.e.*, identifying all unknown tags with a confidence of 100%. Different from CU, BUIP not only uses the expected empty slots to label unknown RFID tags, but also uses the expected singleton slots to deactivate the known RFID tags. After deactivating all known tags, the remaining active tags as well as the labeled tags are definitely unknown tags, which will be completely collected by a tag identification scheme.

• Single Echo based Batch Authentication Plus (SEBA+) [20]: It is a representative unknown tag detection protocol. The reader initializes a slotted time frame, and each tag pseudo-randomly selects a certain number of slots to reply responses. Since the server knows all hash parameters and known tag IDs, it can predict the status of each time slot. If the reader receives a tag response in an expected empty slot, the existence of unknown tags will be discovered.

• White Paper (WP) [14]: It is the-state-of-the-art unknown tag detection protocol. In WP, the reader broadcasts a long and complex seed vector to guide the tag-slot selecting process. In expectation, the seeds in vector can make the corresponding time slots empty. If the reader receives any responses from the time frame, it can detect the existence of unknown tags.

B. Simulation Settings

We mainly evaluate time-efficiency of the related protocols and validate the actual detection accuracy of our CSD+BGP protocol as well. For the fair comparison, we use the same wireless communication settings for each protocol as follows. The wireless transmission rate between a reader and a tag is $40Kb/s$, *i.e.*, it takes $25us$ to transmit 1-bit data from a tag to a reader and vice versa [20]. Any two consecutive data transmissions are separated by a waiting time $302us$ [33]. That is, the duration of a slot for exchanging m-bit data between a reader and a tag should be $(25m + 302)us$. On the other hand, when evaluating the computation cost of the CSD+BGP protocol, we set $t_c = 4.17 \times 10^{-10}s$ and $\eta = 344$ [27]. Here, t_c is the clock cycle of the server with 2.4 GHz CPU, and η is the number of required clock cycles for calculating a hash function and checking the result on server. Since CU and BUIP aim at exactly identifying the IDs of all unknown tags. For their sake, we only simulate their process of labeling unknown tags, which is enough to detect the unknown tags. And the time cost for collecting specific IDs of unknown tags is not counted. In [14], a default assumption is that all tags particularly the known tags have the same set of hash seeds before running WP. Then, the reader can simply send a vector of seed indexes (instead of the detailed hash seeds) to notify tags which seeds should be used. However, in practice, tags from different tenants may not have the same hash seeds in memory at all. And a straightforward countermeasure is to dynamically write the same set of hash seeds into all tags' memory before running WP, which, however, may not be allowed due to security concerns. Hence, only transmitting a vector of seed indexes to tags when performing WP may not work in practice. For practical reasons and fair comparison, when simulating WP protocol, we let the reader send the vector of seeds instead of seed indexes. Moreover, a long seed vector is segmented into 96-bit pieces and transmitted via multiple slots [22]. In the following, we conduct simulations to evaluate the performance of these protocols in single-reader and multi-reader scenarios, respectively. Each set of simulations are repeated for tens of times and we report the averaged results.

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

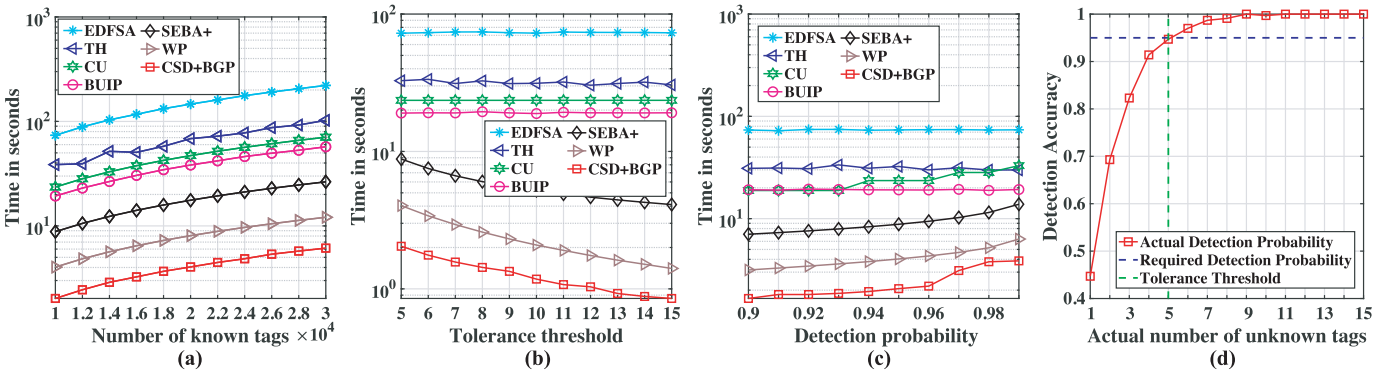


Fig. 5. Investigating time-efficiency and actual detection accuracy of our protocol in a single-reader system. (a) Time cost *vs.* number of known tags; (b) time cost *vs.* tolerance threshold of unknown tags; (c) time cost *vs.* required detection probability; (d) detection accuracy *vs.* actual number of unknown tags.

C. Single-Reader Scenarios

We first consider the single-reader scenarios, where a reader is able to cover all thousands of RFID tags. Such an assumption is reasonable because we can use the following methods [34] to significantly extend the reading range of a single reader. First, we can employ the powerful RFID antenna with a larger coverage area, *e.g.*, the Impinj LHCP Far Field Antenna is able to cover 139 square meters [35]. Second, a reader can be connected with multiple antennas to extend its tag interrogation range, *e.g.*, the Impinj R420 reader can support 32 RFID antennas at most [36]. Jointly using the above countermeasures, the monitoring area of a single reader can be theoretically extended to as large as $4,448$ square meters, which is enough to cover thousands of tagged items. Side-by-side comparison of protocols will be given in the following.

1) Time-Efficiency: The number of known tags k, tolerance threshold of unknown tags u, and required detection accuracy α may significantly affect the performance of the concerned protocols. Hence, we conduct simulations to investigate the impact of these parameters. Unless otherwise specified, we use the default settings $k = 10000$, $u = 5$, and $\alpha = 95\%$ when conducting simulations.

Impact of k: We vary the value of k from 10000 *to* 30000. *We observe from the simulation results in Fig. 5(a) that, our CSD+BGP protocol is always the fastest with varying values of k.* For example, when $k = 30000$, the execution time of EDFSA, TH, CU, BUIP, SEBA+, and WP is $221.2s$, $102.5s$, $70.6s$, $57.2s$, $26.4s$, and $12.1s$, respectively. And the time cost of our CSD+BGP protocol is just $6.1s$, which means it achieves $1.98\times$ speedup than the state-of-the-art WP protocol. Moreover, the execution time of each protocol increases as the number of known tags increases, because more tag IDs need to be tackled.

Impact of u: We vary the value of u from 5 *to* 15. *We observe from the simulation results in Fig. 5(b) that, the proposed CSD+BGP protocol keeps significantly outperforming the other protocols with varying values of u.* For example, when $k = 5$, the execution time of EDFSA, TH, CU, BUIP, SEBA+, and WP is $72.6s$, $32.7s$, $23.6s$, $19.0s$, $8.8s$, and $4.0s$, respectively. And the time cost of our CSD+BGP protocol is just $2.0s$, which still means $2\times$ speedup than the state-of-the-art WP protocol. Moreover, the execution time of EDFSA, TH, CU, and BUIP keeps stable with varying value of u, whereas, that of SEBA+, WP, and our CSD+BGP protocol decreases as the value of u increases. The underlying reason is that, a larger tolerant threshold u means imposing a looser requirement for the unknown tag detection protocols, and thus resulting in a smaller detection time.

Impact of α: We vary the value of α from 0.90 *to* 0.99. *We observe from the simulation results in Fig. 5(c) that, the proposed CSD+BGP protocol is continuously the fastest with varying values of α.* For example, when $\alpha = 0.99$, the execution time of EDFSA, TH, CU, BUIP, SEBA+, and WP is $73.8s$, $30.0s$, $33.0s$, $19.3s$, $13.8s$, and $6.3s$, respectively. And the time cost of our CSD+BGP protocol is just $3.7s$, which means $1.7\times$ speedup than the state-of-the-art WP protocol. Moreover, the execution time of EDFSA, TH, and BUIP keeps stable with varying value of α, whereas, that of CU, SEBA+, WP, and our CSD+BGP protocol increases as the value of α increases. The underlying reason is that, a larger value of α means a stricter requirement for the unknown tag detection protocols, and thus leading to a larger detection time.

2) Detection Accuracy: The authors in [14], [20] have proposed sufficient theoretical analyses to guarantee the detection accuracy of the dedicated unknown tag detection protocols, *i.e.*, SEBA+ and WP. And simulation results in these two literatures have demonstrated that their protocols can satisfy the required unknown tag detection accuracy indeed. Hence, we do not conduct simulations to evaluate their detection accuracy any more. In this set of simulations, we mainly aim at validating the actual detection probability of our CSD+BGP protocol with varying number of unknown tags that really appear in the system. Here, the actual detection probability is measured by the ratio of the number simulations in which existence of unknown tags is successfully detected to the total number of simulations. The number of unknown tags really appearing in the system, *i.e.*, ν, varies from 1 to 15. *We observe from the simulation results in Fig. 5(d) that, when the value of ν exceeds the given tolerance threshold $u = 5$, the actual detection probability of CSD+BGP is always larger than the required detection probability α.* This means our CSD+BGP protocol is able to satisfy the required detection accuracy in single-reader RFID systems.

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

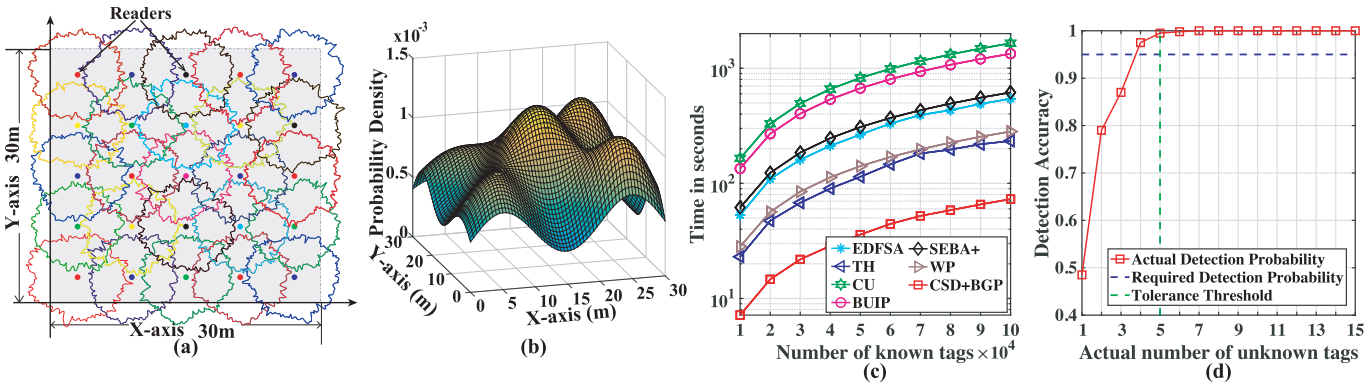


Fig. 6. Investigating time-efficiency and actual detection accuracy of our protocol in a multi-reader system. (a) Deployment of readers; (b) distribution likelihood of tags; (c) time cost *vs.* number of known tags; (d) detection accuracy *vs.* actual number of unknown tags.

D. Multi-Reader Scenarios

To seamlessly cover the a large monitoring region, we need to deploy multiple readers with overlaps. In this section, we will evaluate the performance of concerned protocols in a multi-reader RFID system. As illustrated in Fig. 6(a), we deploy 5×5 RFID readers in grid to cover a $30m\times30m$ region. Due to the impact of occlusion and multi-path, the reader's probing distance along different angles may not be consistent and is assumed to follow the normal distribution $\texttt{Nor}(5m, 0.25m)$. That is, we assume that the probing distance of each reader is with an average value of 5 meters, but with a standard variance of 0.25 meters. Deployment of multiple readers inevitably begets the reader-collision issue. That is, if two nearby readers simultaneously probe the tags locating in their overlapping area, these tags cannot correctly receive any commands from the readers. To avoid the reader-collision issue, several research works [31], [32], [37] were proposed to investigate the optimal reader scheduling strategies. Since reader scheduling strategy is not the key point of this paper, we just use a greed method similar with the Colorwave scheme [37]. Specifically, we give a distance threshold $\mathcal{D}$ to indicate whether two readers may conflict or not. For example, if the average probing distance of each reader is 5 meters, we can say two readers with a distance larger than $\mathcal{D} = 14$ meters will not conflict with each other. We use a color to mark the readers without conflict as much as possible. Then, we use another color to mark the readers without conflict in the remaining set of uncolored readers. This reader-coloring process is repeated until all readers are colored. Clearly, we can simultaneously activate the readers in the same color to execute the query protocols without reader-collision issue. After performing the detection protocol simultaneously on the readers in a certain color, we turn to activate the readers in another color and simultaneously perform the detection protocol. This process is repeated until unknown tag detection is performed on all readers. We use the probability density shown in Fig. 6(b) to randomly generate the location of each tag.

1) Time-Efficiency: In this set of simulations, we vary the number of known tags k from 10000 to 100000, thereby investigating its impact on the performance of each protocol in the multi-reader RFID systems. We can make two major observations from the simulation results in Fig. 6(c). First, *the proposed CSD+BGP protocol is always the fastest with varying values of* k. For example, when $k = 100000$, the execution time of EDFSA, TH, CU, BUIP, SEBA+, and WP is $543.0s$, $233.7s$, $1648.2s$, $1337.9s$, $616.6s$, and $282.2s$, respectively. And the time cost of our CSD+BGP protocol is just $72.9s$, which means $3.9\times$ speedup than the state-of-the-art WP protocol, and $3.2\times$ speedup than the TH protocol. Second, different from the simulation results corresponding to single-reader scenarios, the execution time of CU, BUIP, SEBA+, and WP is very huge, while the TH protocol becomes the second fastest one. The underlying reasons are that, CU, BUIP, SEBA+, and WP simply treat all readers as one logical reader, thus they cannot take the advantage of multiple readers. On the contrary, in EDFSA, TH, and our CSD+BGP, each reader only needs to tackle with the tags within its coverage. It can be interpreted as that a heavy tag interrogation task is divided into multiple small pieces, and each piece is taken by a reader. As a result, execution time of EDFSA, TH, and CSD+BGP can be significantly reduced.

2) Detection Accuracy: In this set of simulations, we will investigate the actual detection probability of our CSD+BGP protocol in the multi-reader RFID systems. The number of unknown tags ν varies from 1 to 15. *We observe from the simulation results in Fig. 6(d) that, when the value of* ν *exceeds the given tolerance threshold* $u = 5$*, the actual detection probability of CSD+BGP is much larger than the required detection probability* α. Comparing the simulation results in Fig. 5(d) and that in Fig. 6(d), we find that when ν exactly equals u, the actual detection probability of CSD+BGP in the multi-reader RFID system is much higher than that in the single-reader RFID system. The underlying reasons are that, some unknown tags may locate in the overlapping region of two adjacent readers. Thus, such an unknown tag has a larger chance to be detected than the unknown tag that is covered by only one reader. Hence, it is relatively easier to discover the existence of unknown tags in a multi-reader RFID system. In summary, the proposed CSD+BGP protocol can also satisfy the required unknown tag detection accuracy in the multi-reader RFID systems.

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

VI. Related Work

The existence of unknown tags in an RFID system may cause serious risks to economic profit or even human safety. Hence, the academical communities have made a great deal of efforts to address the unknown tag issues. We classify the existing unknown tag-related works into three categories: *Unknown tag identification* aims at identifying the exact IDs of unknown tags; *Unknown tag estimation* is to estimate the cardinality of unknown tags in an RFID system; *Unknown tag detection* targets at detecting whether there are any unknown tags in a system with a predefined probability.

Unknown Tag Identification

In some cases, we need to exactly identify the IDs of unknown tags in the RFID system. Then, we can take the proper countermeasures to deal with these unknown tags, *e.g.*, pinpointing the locations of unknown tags and moving the corresponding tagged items out of the system. The Collect Unknown Tags (CU) protocol [17] is a variant of the classical framed slotted aloha mechanism. The RFID reader sends a special command in pre-empty slots to label unknown tags. Then, labeled unknown tags will be collected by the tag identification protocol. In CU, all known tags keep contending for each round of time frame, and thus seriously affecting the unknown tag labeling process. To overcome this drawback, the Basic Unknown Tag Identification Protocol (BUIP) [18] not only uses the pre-empty slots to label the unknown RFID tags but also uses the pre-singleton slots to deactivate the known RFID tags. Specifically, if only one tag replies in a pre-singleton slot, this tag should be a known tag and will be deactivated. Thus, the number of known tags that contend for time frame will quickly decrease after several time frames. In [38], Liu *et al.* first proposed the Filtering-based Unknown Tag Identification (FUTI) protocol to label the unknown tags at the bit-level instead of slot-level. Thus, it is expected to achieve better time-efficiency than the CU and BUIP protocols. Then, they further proposed an enhanced unknown tag identification protocol called Interactive Unknown Tag Identification (IFUTI), which leverages the interactive filters to not only label the unknown tags but also accelerate the process of identifying the labeled unknown tags. From the perspective of time-efficiency, the IFUTI protocol performs better than the other unknown tag identification protocols. In terms of identification accuracy, the unknown tag identification protocols in [17], [38] can only identify a given ratio of the unknown tags, while the protocols in [18] can identify all unknown tags in the system with a confidence of 100%. In terms of deployability, CU and BUIP, which only require the C1G2-complaint commands, are more easier to apply on the COTS RFID tags than IFUTI.

Unknown Tag Estimation

Sometimes, knowing the approximate cardinality of unknown tags in an RFID system is enough for the users. For example, the unknown tags in a system may mean the new products that are just moved into a logistics warehouse. The manager needs to assign a proper number of workers to this area with consideration of the number of new products. A batch of efficient tag cardinality estimation protocols [33], [39], [40] were proposed to accurately estimate the number of tags present in an RFID system. However, they cannot tell us how many tags are newly moved into the system compared with the last round of inventory. To this end, Xiao *et al.* proposed the Zero Differential Estimator (ZDE) protocol [41], in which the slotted time frame observed by the reader in a tag inventory process is transformed into a binary vector (the bit 0 means an empty slot and the bit 1 means a non-empty slot). If unknown tags appear in the system, some bits 0s in the vector will turn out to be 1s. They quantitively established the functional relationship between the number of unknown tags and the number of bits that change from 0s to 1s. Then, the number of unknown tags can be estimated by using the number of bits whose status changes. Unlike ZDE that uses the uniform hash, Gong *et al.* proposed INformative Counting (INC) [42], which uses a geometric hash function on tag side. Benefiting from the geometric distribution characteristic, the frame size in INC can be significantly reduced compared with ZDE.

Unknown Tag Detection

Frequently executing the unknown tag identification or estimation protocols in an RFID system usually wastes a lot of time, because a system may not contain any unknown tags at all. An effective solution is to first perform a lightweight unknown tag detection protocol to detect whether there are unknown tags in the system. The heavy identification or estimation protocols will be invoked only if the detection result is positive. In what follows, we will discuss the existing unknown tag detection protocols. The Single Echo based Batch Authentication (SEBA) protocol proposed in [19] discovers the existence of unknown tags if the reader finds that a pre-empty slot becomes a non-empty time slot or a pre-singleton slot turns out to be a collision time slot. The Single Echo based Batch Authentication Plus (SEBA+) protocol proposed in [20] exploits the Bloom Filter (BF) technique to extend the previous SEBA protocol. In the SEBA+ protocol, each tag pseudo-randomly selects $\hbar \geq 1$ slots (instead of a single slot) within a time frame to reply responses. In SBF-UDP [13], a sampling bloom filter $\mathtt{BF}$ is constructed on the sever side by using multiple hash functions to hash known tags to the filter. Then, the reader broadcasts the hash parameters and the sampling bloom filter $\mathtt{BF}$ to all tags. After receiving the filter, each tag also uses the same hash parameters to calculate hash functions and checks whether all its corresponding bits in $\mathtt{BF}$ are 1s. If any corresponding bits are sampled, but turn out to be 0s, the tag will label itself as an unknown tag and report this event to the reader. SBF-UDP uses random hashing seeds and the known tags are *uniformly distributed* along the whole bloom filter, which incurs a large ratio of bit 1s and results in low-efficiency of the sampling bloom filter. In the White Paper (WP) protocol [14], the reader broadcasts a seed vector $\mathtt{V}$ to guide the communication of tags. Specifically, the number of seeds specified in the vector is equal to size f of the follow-up time frame. According to the rule of constructing seed vector, no known tag will be hashed to the i-th slot if using the seed $\mathtt{V}[i]$. When actually executing the time frame to query tags, the WP protocol requires that a tag can respond in

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

the i-th slot if and only if $\mathcal{H}(ID, \mathrm{v}[i]) \mod f = i$. Clearly, the known tags will not respond in the time frame at all. If the reader receives any responses, the existence of unknown tags will be discovered. Thus, all slots are pre-empty slots and can be used to detect unknown tags. A default assumption inherent in WP [14] is that, all tags in the system (even including the unknown tags) have the same set of hash seeds. Thus, for saving time, the reader only needs to send a vector of seed indexes. However, it not reasonable to assume all tags particularly the unknown tags have the same set of hash seeds, because the unknown tags usually belong to other users in a multi-tenant warehouse.

VII. Conclusion

This paper studied the practically important problem of unknown tag detection, and made the following major contributions. First, we proposed the Collision-Seeking Detection (CSD) protocol. Unlike previous works that try to avoid hash collision, the proposed CSD protocol deliberately creates hash collision to increase frame utilization. Second, we proposed the supplementary protocol called Group Partition (GP) protocol to effectively reduce computation cost of CSD, and another enhanced supplementary protocol called Balanced Group Partition (BGP) protocol to further address the unbalance issue in GP. Third, we used the bloom filter technique to make CSD+BGP scalable for multi-reader RFID systems. Finally, we proposed sufficient theoretical analysis to optimize the involved parameters for guaranteeing the required detection accuracy and minimizing the detection time. Extensive simulation results reveal that our CSD+BGP protocol can guarantee the required detection accuracy, meanwhile significantly reducing the detection time compared with the state-of-the-art unknown tag detection protocol.

Acknowledgment

Partial work of Xiulong Liu was done at SFU as a postdoctoral fellow under supervision of Prof. Jiangchuan Liu.

References

[1] S. Qi, Y. Zheng, M. Li, L. Lu, and Y. Liu, "COLLECTOR: A secure RFID-enabled batch recall protocol," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2014, pp. 1510–1518.

[2] X. Liu *et al.*, "Fast RFID sensory data collection: Trade-off between computation and communication costs," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1179–1191, Jun. 2019.

[3] J. Yu, W. Gong, J. Liu, L. Chen, K. Wang, and R. Zhang, "Missing tag identification in COTS RFID systems: Bridging the gap between theory and practice," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2018.2889068.

[4] J. Yu, W. Gong, J. Liu, L. Chen, and K. Wang, "On efficient tree-based tag search in large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 42–55, Feb. 2018.

[5] X. Liu *et al.*, "Efficient range queries for large-scale sensor-augmented RFID systems," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 1873–1886, Oct. 2019.

[6] Y. Hou, Y. Wang, and Y. Zheng, "TagBreathe: Monitor breathing with commodity RFID systems," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, Jun. 2017, pp. 404–413.

[7] X. Liu, S. Zhang, B. Xiao, and K. Bu, "Flexible and time-efficient tag scanning with handheld readers," *IEEE Trans. Mobile Comput.*, vol. 15, no. 4, pp. 840–852, Apr. 2016.

[8] S. Qi, Y. Zheng, M. Li, Y. Liu, and J. Qiu, "Scalable industry data access control in RFID-enabled supply chain," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3551–3564, Dec. 2016.

[9] X. Liu *et al.*, "Fast identification of blocked RFID tags," *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2041–2054, Sep. 2018.

[10] K. Xie, L. Wang, X. Wang, G. Xie, and J. Wen, "Low cost and high accuracy data gathering in WSNs with matrix completion," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1595–1608, Jul. 2018.

[11] K. Xie *et al.*, "Recover corrupted data in sensor networks: A matrix completion solution," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1434–1448, May 2017.

[12] X. Liu, Q. Yang, J. Luo, B. Ding, and S. Zhang, "An energy-aware offloading framework for edge-augmented mobile RFID systems," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 3994–4004, Jun. 2019.

[13] X. Liu *et al.*, "Sampling Bloom filter-based detection of unknown RFID tags," *IEEE Trans. Commun.*, vol. 63, no. 4, pp. 1432–1442, Apr. 2015.

[14] W. Gong, J. Liu, and Z. Yang, "Fast and reliable unknown tag detection in large-scale RFID systems," in *Proc. ACM Mobihoc*, 2016, pp. 141–150.

[15] S. R. Lee, S. D. Joo, and C. W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. IEEE MobiQuitous*, Jul. 2005, pp. 166–172.

[16] M. Shahzad and A. X. Liu, "Probabilistic optimal tree hopping for RFID identification," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 796–809, Jun. 2015.

[17] B. Sheng, Q. Li, and W. Mao, "Efficient continuous scanning in RFID systems," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[18] X. Liu, B. Xiao, S. Zhang, and K. Bu, "Unknown tag identification in large RFID systems: An efficient and complete solution," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1775–1788, Jun. 2015.

[19] L. Yang, J. Han, Y. Qi, and Y. Liu, "Identification-free batch authentication for RFID tags," in *Proc. IEEE ICNP*, Oct. 2010, pp. 154–163.

[20] G. Bianchi, "Revisiting an RFID Identification-free batch authentication approach," *IEEE Commun. Lett.*, vol. 15, no. 6, pp. 632–634, Jun. 2011.

[21] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, "A time-efficient information collection protocol for large-scale RFID systems," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2158–2166.

[22] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu, "Completely pinpointing the missing RFID tags in a time-efficient way," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 87–96, Jan. 2015.

[23] S. Chen, M. Zhang, and B. Xiao, "Efficient information collection protocols for sensor-augmented RFID networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 3101–3109.

[24] "Radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz–960 MHz version 1.0.9," *K. Chiew/On False Authenticationsfor C1G2 Passive RFID Tags, EPCglobal*, vol. 65, pp. 1–94, Apr. 2004.

[25] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, and Y. Seurin, "Hash functions and RFID tags: Mind the gap," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2008, pp. 283–299.

[26] C. Qian, Y. Liu, R. H. Ngan, and L. Ni, "ASAP: Scalable collision arbitration for large RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1277–1288, Jul. 2013.

[27] O. N. Maire, "Low-cost SHA-1 hash function architecture for RFID tags," *RFIDSec*, vol. 8, pp. 41–51, Jul. 2008.

[28] L. Yang, Q. Lin, C. Duan, and Z. An, "Analog on-tag hashing: Towards selective reading as hash primitives in Gen2 RFID systems," in *Proc. ACM MobiCom*, Oct. 2017, pp. 301–314.

[29] *HashFuncProperty*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Cryptographic_hash_function

[30] *Jain's Fairness Index*. Accessed: Aug. 23, 2019. [Online]. Available: https://en.wikipedia.org/wiki/fairness_measure

[31] L. Yang, J. Han, Y. Qi, C. Wang, T. Gu, and Y. Liu, "Season: Shelving interference and joint identification in large-scale RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 3092–3100.

[32] S. Tang, J. Yuan, X. Li, G. Chen, Y. Liu, and J. Zhao, "RASPberry: A stable reader activation scheduling protocol in multi-reader RFID systems," in *Proc. IEEE ICNP*, Oct. 2009, pp. 304–313.

[33] M. Shahzad and A. X. Liu, "Fast and accurate estimation of RFID tags," *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 241–254, Feb. 2015.

[34] X. Xie *et al.*, "Implementation of differential tag sampling for COTS RFID systems," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2019.2917444.

[35] *Impinj Xarray Solution*. Accessed: Aug. 23, 2019. [Online]. Available: https://support.impinj.com/hc/article_attachments/115001459570/Impinj_xPortalProductBrief_7.23.17_FINAL.pdf

[36] *R420 RFID Reader*. Accessed: Jul. 25, 2019. [Online]. Available: https://support.impinj.com/

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.

[37] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An anticollision algorithm for the reader collision problem," in *Proc. IEEE Int. Conf. Commun.*, May 2003, pp. 1206–1210.
[38] X. Liu *et al.*, "Efficient unknown tag identification protocols in large-scale RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3145–3155, Dec. 2014.
[39] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MobiCom*, Sep. 2006, pp. 322–333.
[40] C. Qian, H. Ngan, Y. Liu, and L. M. Ni, "Cardinality estimation for large-scale RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1441–1454, Sep. 2011.
[41] Q. Xiao, B. Xiao, and S. Chen, "Differential estimation in dynamic RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 295–299.
[42] W. Gong, K. Liu, X. Miao, Q. Ma, Z. Yang, and Y. Liu, "Informative counting: Fine-grained batch authentication for large-scale RFID systems," in *Proc. ACM MobiHoc*, Aug. 2013, pp. 21–30.

Xiulong Liu received the B.E. and Ph.D. degrees from the Dalian University of Technology, China, in 2010 and 2016, respectively. He worked as a Visiting Researcher at Aizu University, Japan, a Post-Doctoral Fellow at The Hong Kong Polytechnic University, Hong Kong, and a Post-Doctoral Fellow at the School of Computing Science, Simon Fraser University, Canada. He is currently a Professor with the College of Intelligence and Computing, Tianjin University, China. His research articles were published in many prestigious journals and conferences, including TON, TMC, TC, TPDS, TCOM, INFOCOM, and ICNP. His research interests include wireless sensing and communication, indoor localization, and networking. He received the Best Paper Award from ICA3PP 2014 and the IEEE SYSTEMS JOURNAL 2017. He was also a recipient of the CCF Outstanding Doctoral Dissertation Award 2017.

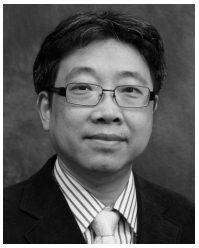
Sheng Chen received the bachelor's and master's degrees from Dalian Maritime University and Dalian University of Technology in 2011 and 2017, respectively. He is currently pursuing the Ph.D. degree with the College of Intelligence and Computing, Tianjin University, China. His research interests include data center networks, edge computing, wireless sensing, and indoor localization.

Jia Liu (M'13) 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 RFID systems. He is a member of ACM.

Wenyu Qu received the bachelor's and master's degrees from the Dalian University of Technology, China, in 1994 and 1997, respectively, and the Ph.D. degree from the Japan Advanced Institute of Science and Technology, Japan, in 2006. She was a Professor with Dalian Maritime University, China, from 2007 to 2015. She was an Assistant Professor with the Dalian University of Technology, China, from 1997 to 2003. She is currently a Professor with the College of Intelligence and Computing, Tianjin University. She has authored over 80 technical articles in international journals and conferences. Her research interests include cloud computing, computer networks, and information retrieval. She is on the committee board for a couple of international conferences.

Fengjun Xiao received the B.S. degree in economics from Beihang University in 2009 and the master's degree in technology policy in 2014, under the supervision of Prof. Shi Li. He is currently pursuing the Ph.D. degree, under the supervision of Prof. Chengzhi Li, and he has been researching on the network security and emergency management since 2015.

Alex X. Liu (F'19) received the Ph.D. degree in computer science from The University of Texas at Austin in 2006. He is currently a Professor with the Department of Computer Science and Engineering, Michigan State University. His research interests focus on networking and security. He received the IEEE and IFIP William C. Carter Award in 2004, the National Science Foundation CAREER Award in 2009, and the Michigan State University Withrow Distinguished Scholar Award in 2011. He also received the Best Paper Award from ICNP-2012, SRDS-2012, and LISA-2010. He has served as a TPC Co-Chair for ICNP 2014 and IFIP Networking 2019. He has served as an Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING. He is currently an Associate Editor for the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING and the IEEE TRANSACTIONS ON MOBILE COMPUTING, and an Area Editor for *Computer Communications*.



Jiannong Cao (F'14) is currently the Chair Professor of the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He is also the Director of the Internet and Mobile Computing Laboratory, Department of Computing, and the Director of the University's Research Facility in Big Data Analytics. He has coauthored five books, co-edited nine books, and published over 500 articles in major international journals and conference proceedings. His research interests include parallel and distributed computing, wireless sensing and networks, pervasive and mobile computing, and big data and cloud computing. He received the Best Paper Award from conferences, including DSAA'2017, the IEEE SMARTCOMP 2016, and ISPA 2013.

Jiangchuan Liu (F'17) received the B.Eng. degree *(cum laude)* from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from The Hong Kong University of Science and Technology in 2003. He is currently a Full Professor (with University Professorship) with the School of Computing Science, Simon Fraser University, British Columbia, Canada. He is a Canadian Academy of Engineering Fellow and an NSERC E.W.R. Steacie Memorial Fellow. He is a Steering Committee Member of the IEEE TRANSACTIONS ON MOBILE COMPUTING. He was a co-recipient of the Test of Time Paper Award of IEEE INFOCOM in 2015, the ACM TOMCCAP Nicolas D. Georganas Best Paper Award in 2013, and the ACM Multimedia Best Paper Award in 2012. He is an Associate Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON BIG DATA, and the IEEE TRANSACTIONS ON MULTIMEDIA.

Authorized licensed use limited to: SIMON FRASER UNIVERSITY. Downloaded on March 10,2020 at 19:18:18 UTC from IEEE Xplore. Restrictions apply.