# Time-efficient RFID-based Stocktaking with a Coarse-grained Inventory List

Weiping Zhu*, Xing Meng†, Xiaolei Peng*, Jiannong Cao‡ and Michel Raynal§
*School of Computer Science, Wuhan University, P. R. China
†School of Mathematics and Statistics, Wuhan University, P. R. China
‡Department of Computing, The Hong Kong Polytechnic University, Hong Kong
§Institut Universitaire de France & IRISA-INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France
{wpzhu, xmeng.math, pengxiaolei}@whu.edu.cn, csjcao@comp.polyu.edu.hk, raynal@irisa.fr

***Abstract*—RFID-based stocktaking uses RFID technology to verify the presence of objects in a region e.g., a warehouse or a library. The existing approaches for this purpose assume that an inventory list of objects in the interrogation region of an RFID reader is known. This is not true in some cases. For example, for a handheld RFID reader, only the objects in a larger region (e.g., the warehouse) rather than in its interrogation region can be known. The additional objects significantly increase the time required for stocktaking. In this paper, we propose a time-efficient stocktaking algorithm called CLS (Coarse-grained inventory list based stocktaking) to solve this problem. We transform the problem to a missing tag identification problem with a large missing rate. CLS enables multiple missing objects to hash to a single time slot and thus verifies them together. CLS also improves the existing approaches by utilizing more kinds of RFID collisions and reducing approximately one-fourth of the amount of data sent by the reader. Extensive simulations are performed and the results show CLS outperforms the best existing algorithm.**

***Index Terms*—RFID; Stocktaking; Time Efficiency; Missing Rate; CLS**

## I. Introduction

Radio Frequency Identification (RFID) is a digital identification technology that employs radio frequency to collect identity information from RFID tags using RFID readers. During the identification process, a reader sends out a request to tags, and the tags reply with pre-stored IDs and the associated information. Compared with barcode technology, RFID has the advantages such as non-line-of-sight capability, long distance and fast identification, and high reliability. Stocktaking is one of the prominent applications of RFID [1], [2]. The purpose of stocktaking is to verify the presence of objects in a given region such as a warehouse, library, and shopping mall. In order to fulfill such work, we can use RFID readers to identify the tags (each tag is attached to an object) in their interrogation regions.

The existing algorithms for identifying tags in the interrogation region of an RFID reader include ID collection algorithms and missing tag identification algorithms. In an ID collection algorithm, a reader collects all the IDs of tags in its interrogation region. Typical algorithms include tree-based algorithms [3], [4], [5] and ALOHA-based algorithms [6], [7], [8]. These algorithms suffer from large execution time because the tag ID to be transmitted is long (e.g., 96 bits) and the collisions among tags are serious.

Missing tag identification algorithms [9], [10], [11] are proposed for accelerating the stocktaking process. This kind of algorithm requests a reply from tags to the reader in a slotted time frame. It assumes that an inventory list of tags in the interrogation region of the reader is known, and the time slot during which each tag replies can be computed in advance. Comparing pre-computed status of time slots with the actual results, missing tags can be determined. For example, if an expected tag reply is not received, this tag can be verified to be missing. A short message (e.g., 1-bit data) rather than the tag ID is used as the content of the tag reply to reduce time cost. SFMTI [11] achieves the best identification performance currently, which is approximately one tag per time slot. We use *missing tags* to denote the tags that are absent from the interrogation region but present in the inventory list, and *missing rate* to denote the ratio of missing tags to the tags in the inventory list. Existing missing tag identification algorithms function satisfactorily when the missing rate is small.

However, in many cases the inventory list is difficult to obtain. For example, in a library that uses a handheld RFID reader for stocktaking, the tags in its interrelation region cannot be known in advance owing to the mobility of the reader. The tags in the whole library are known but may be 20 times the number of tags to be identified, which means the missing rate is about 0.95. When using them as the inventory list, the execution time will increase significantly. Furthermore, even if we use a stationary reader to perform periodic stocktaking for the same region, an inventory list with high missing rate may only be possible if new objects enter the interrogation region between two consecutive stocktaking processes. The existing missing tag identification algorithms are not time-efficient in these scenarios and should be improved.

In this paper, we investigate the problem of RFID-based stocktaking without a fine-grained inventory list. We utilize the inventory list in a larger region (e.g., the whole library) and transform the problem into missing tag identification

978-1-5386-2542-2/18/$31.00 

with a high missing rate. An algorithm called Coarse-grained Inventory List based Stocktaking (CLS) is proposed to solve this problem. In the scenario with a high missing rate, CLS can identify multiple tags during a single time slot. Contrary to the existing works that consider only the time slots with $k$ ( $k \leq 3$) replies, CLS further utilizes the time slots with four or more replies in order to improve the time efficiency of identification. The number of states of time slots is also minimized and thereafter, Huffman code is used to shorten the request sent by the reader. Extensive simulations are carried out for validating the proposed approach. In summary, this paper offers the following contributions.

- We investigated the RFID-based stocktaking problem without a coarse-grained inventory list. To the best of our knowledge, this is the first work to investigate this practical problem.
- We proposed a new missing tag identification algorithm called CLS for solving this problem. CLS outperforms the existing algorithms when using an coarse-grained inventory list. In a typical handheld RFID reader based stocktaking scenario where the inventory list is 20 times the really existing ones, CLS requires 53.5% execution time compared with the existing approaches.

The rest of the paper is organized as follows: Section II describes the system models used in this study and formulates the problem. Our solution is illustrated in Section III. The simulation results are reported in Section IV. Section V reviews the related works and Section VI concludes the paper.

## II. System Model and Problem Formulation

In this paper, we assumed that there are many objects in a large region such as a warehouse and library. An RFID tag is attached to each object. The IDs of these RFID tags are recorded in an inventory list and known in advance. These RFID tags are called *candidate tags* and are denoted as $N^*$.

For the purpose of stocktaking, an RFID reader is used to identify the tags in its interrogation region. The interrogation region is much smaller than the total area, e.g., one-tenth of the total area. The tags residing in the interrogation region are called *present tags*, denoted as *N*. Accordingly, the tags in $N^* - N$ are called *missing tags*. *N* is not known and required to be identified. The only known information is $N \subseteq N^*$ and $|N| << |N^*|$ where $|x|$ denotes the cardinality of *x*. In this paper, we define the *missing rate* as $p = (|N^*| - |N|)/|N^*|$. The identification process is repeated multiple times to cover the region and obtain a list of all the present tags. The tag list can be compared with $N^*$ for audit proposes.

Following the Reader Talks First mode [12] used in the literature, the RFID reader queries the tags first, and the tags reply during a slotted time frame. An optional acknowledgement from the reader to the replies from the tags follows. The aforementioned processing is called a round of identification, and multiple rounds are required to finish the entire process. In each round, the identified tags remain silent and only the others participate in the identification. Each query sent by the reader contains a random number $r$ and a frame size *f*. On receiving the query, tag $t$ uses its hash function $H_t$ to determine the time slot of the reply, by computing $H_t(r) \mod f$. $H_t$ can be implemented by a pseudo-random method based on the pre-stored data in the tag [9]. The hash functions of all the candidate tags are known by the reader (e.g., provided by the manufacturer of the tags). Following the literature [9], [11], we assume reliable communications in the identification.

According to the replies of tags, time slots are classified into *empty slots*, *singleton slots*, or *k-collision slots*. Empty slot is a time slot that no tag replies to, singleton slot is a time slot that only one tag replies to, and *k*-collision slot is a time slot that $k$ tags reply to. In order to distinguish them, we refer to the empty slots computed based on the candidate tags as *expected empty slots*, and the empty slots in the real identification as *actual empty slots*. Similarly, we have *expected singleton slots*, *expected k-collision slots*, *actual singleton slots*, and *actual k-collision slots*.

We use $T_{tag}$ to denote the time required by a reader to transmit a 96-bit request to the tags. Requests of other lengths consume time proportional to $T_{tag}$ according to the amount of data. The time cost of a tag reply varies when satisfying different identification requirements. If the reader requires to distinguish empty slots from the others, the tags should transmit only 1-bit data, whereas if the reader requires to distinguish empty slots, singleton slots, and collision slots, the tags should transmit at least 10-bit data. The time duration required to transmit 1-bit data is denoted as $T_{short}$, and the time duration required to transmit 10-bit data is denoted as $T_{long}$ in this paper. We follow the parameter setting in [11], where $T_{tag}$ is 2.4 ms, $T_{short}$ is 0.4 ms, and $T_{long}$ is 0.8 ms.

Given the system models illustrated above, we need to design an algorithm to identify the present tags in the interrogation region of an RFID reader as quickly as possible.

## III. The Solution

We first illustrate our algorithm in detail. Subsequently, we discuss how to determine the key parameters in the algorithm.

### A. Basic Solution

We propose CLS to solve the formulated problem. CLS includes three phases: slot allocation phase, filter vector generation phase, and tag verifying phase.

In the slot allocation phase, the reader first generates a random number *r* and a frame size *f*, and then computes the expected time slots based on *r*, *f*, and $N^*$. According to our system model depicted in Section II, the time slot allocated to tag $t \in N^*$ can be determined by $H_t(r) \mod f$. Subsequently, an *allocation vector A* of length *f* is generated. The $i^{th}$ element of *A*, *A*(*i*), represents the number of tags corresponding to the $i^{th}$ time slot where *A*(*i*) = 0, 1, 2, 3,...... Further, we use *L* to store the detailed allocation where *L*(*i*) denotes all the tags corresponding to the $i^{th}$ time slot.

In the filter vector generation phase, we change all the elements of *A* greater than 1 into *m*. Subsequently, we construct a *filter vector V*. *V* is the Huffman coding of the changed *A*, where *m* is encoded into "1", 0 is encoded into"00", and 1 is encoded into "01". This technique cannot be used for SFMTI because they need to distinguish four states: empty slots, singleton slots, 2-collision slots, and 3-collision slots. CLS combines the processing of all *k*-collision slots ($k \geq 2$) and hence, only three states are required.

In the tag verifying phase, the reader broadcasts *r*, *f*, and $V$ to the tags in its interrogation region. Each present tag *t* computes an index *s* using $s = H_t(r) \mod f$. Subsequently, tag *t* decodes *V* and determines its time slot by counting the nonempty time slots before *s* in *V*. Each tag replies with a short message rather than its ID. According to our system model, the time duration of this message is $t_{short}$.

The operations of tags after replying differ according to their values in *V*. Suppose that tag *t* is allocated the $j^{th}$ time slot (i.e., $j = H_t(r) \mod f$). If $V(j) = 1$, tag *t* changes its status to "silent" and does not participate in the subsequent identification process. If $V(j) = m$, tag *t* does not change its status and is still required to participate in the subsequent identification process. CLS does not require the reader to acknowledge the replies of tags, and the tags can update their statuses immediately after replying.

At the reader side, the replies from tags are received in each time slot, and compared with *A*. If the number of received tag replies in time slot *i* is denoted as *k* , we have the following conditions.

If $A(i) = 1$ and $k = 1$, the tag corresponding to *i* is identified as present.

If $A(i) = 1$ and $k = 0$, the tag corresponding to *i* is identified as missing.

If $A(i) = m$ and $k = 0$, all the tags corresponding to *i* are identified as missing.

If $A(i) = m$ and $k > 0$, no tag is identified.

Notably, only expected singleton slots and expected *m*-collision slots exist in CLS.

All the tags identified as present or missing are removed from $N^*$. The process is repeated until $N^*$ is empty, by which time all the candidate tags are identified as missing or present. The candidate tags identified as present tags are the objective of the identification process of tags in the interrogation region.

### B. Optimal Frame Size of CLS

In each round of the aforementioned process, frame size $f$ affects the time efficiency; therefore, in this subsection we determine its optimal value. The analysis process is similar with [11] but with additional processing of missing rate. We first compute the number of expected *k*-collision slots, $N_k$, as follows:

$$N_k = f \times C_{N^*}^k (\frac{1}{f})^k (1 - \frac{1}{f})^{N^*-k}. \tag{1}$$

Considering that some candidate tags are missing, we compute the probability that no reply is detected in an expected *k*-collision slot in the tag verifying phase as follows:

$$P_k' = C_{N^*}^k (\frac{1}{f})^k (1 - \frac{1}{f})^{N^*-k} \times p^k. \tag{2}$$

The number of such time slots is

$$N_k' = f \times C_{N^*}^k (\frac{1}{f})^k (1 - \frac{1}{f})^{N^*-k} \times p^k. \tag{3}$$

In CLS, a tag can be identified present or missing in two cases. First, the tag corresponding to an expected singleton slot is identified as a present or missing tag, according to whether its reply is received or not. Second, the tags corresponding to an expected $k$-collision slot ($k > 2$) are identified as missing tags if no reply is received. Subsequently, the number of identified tags can be calculated as follows:

$$\Re = N_1 + \sum_{k=2}^{N^*} k N_k'. \tag{4}$$

CLS uses Huffman code to encode the expected empty slots, expected singleton slots and other slots. The former two kinds of time slots are encoded with 2 bits and the latter is encoded with 1 bit. Therefore, the time duration required for the reader to send a request is

$$T_r = (N_0 + N_1) \times \frac{t_{tag}}{96} \times 2 + (f - N_0 - N_1) \times \frac{t_{tag}}{96}. \tag{5}$$

The total time taken by the tags to reply in the tag verification phase is

$$T_v = \sum_{k=1}^{N^*} N_k \times t_{short}. \tag{6}$$

Thus, the average time required for identifying a tag is

$$\begin{aligned} \frac{T}{\Re} &= \frac{T_r + T_v}{N_1 + \sum_{k=2}^{N^*} k N_k'} \\ &= \frac{(1 + P_0 + P_1) \times t_{tag}/96 + (1 - P_0) \times t_{short}}{P_1 + \sum_{k=2}^{N^*} k P_k'}. \end{aligned} \tag{7}$$

Let $\rho = N^*/f$. When $N^*$ is large and $k$ is small, we obtain

$$P_k = C_{N^*}^k (\frac{1}{f})^k (1 - \frac{1}{f})^{N^*-k} \approx \frac{1}{k!} \rho^k e^{-\rho} (k = 0, 1....). \tag{8}$$

$$\sum_{k=2}^{N^*} k P_k' = \sum_{k=1}^{N^*} k P_k' - P_1' \approx p \rho e^{-\rho(1-p)} - p \rho e^{-\rho}. \tag{9}$$

Substitute Equation (7) with Equations (8) and (9), we obtain

$$\frac{T}{\Re} \approx \frac{(1 + e^{-\rho} + \rho e^{-\rho}) \times t_{tag}/96 + (1 - e^{-\rho}) \times t_{short}}{\rho p e^{-\rho(1-p)} + (1-p) \rho e^{-\rho}}. \tag{10}$$

We equate the derivation of $\frac{T}{\Re}$ to 0 to obtain its maximum and corresponding $\rho$. Although we cannot provide a closed-form expression of the result owing to the existence of $p$, it can be solved easily when $p$ is given. Subsequently, the optimal value of $f$ is determined by $N^*/\rho$.

In the above analysis, we assume that the missing rate $p$ is known. In practice, we should estimate it first. It can be approximately determined based on the area of interrelation region and the entire stocktaking region. The existing approaches for estimating the cardinality of tags [12], [13], [14] can also be used to estimate the missing rate based on the replies in the identification process. Moreover, we have observed that the missing rate changes in different rounds. Therefore, the optimal value of $f$ requires to change adaptively. However, we will show that even $f$ is set a sub-optimal value, CLS has desirable performance.

## IV. Performance Evaluation

Simulations are carried out to validate the effectiveness of the proposed approach. By default, $N^*$ is set to 10000. A hundred simulations are repeated to obtain each data point of the figures. The confidence level is 0.95.

### A. Impact of Missing Rate

We first compare the performances of CLS with those of SFMTI [11], IIP [9], and EDFSA [7] in terms of the execution time. IIP verifies the presence of a tag if a reply is received in an expected singleton slot, but does not verify the miss of tags. For the sake of fairness, we revise it slightly by allowing it to verify the missing of tags if no reply is received in a time slot. If such verification is performed at only the expected singleton slots, the algorithm is denoted as IIP-revised1, and if the verification is performed also at the expected collision slots, the algorithm is denoted as IIP-revised2. For EDFSA, the time required to estimate the number of tags is not considered assuming that this information can be calculated based on the area of the interrogation region. We set the frame size of SFMTI to $N^*/1.68$ [11], and the frame size of IIP-revised1 and IIP-revised2 to $N^*/1.516$ [9] to achieve the best performance. For the sake of fairness, we also set the frame size of CLS the same as that of SFMTI. We compare the performances of different approaches by varying the missing rate. The result is shown in Fig. 1.

It can be observed that the execution times of IIP-revised1 and SMFTI are stable. This is because they identify the candidate tags in the expected singleton slots. Although the missing rate changes, the number of candidate tags keeps the same and hence, the execution time remains the same. The execution time of IIP-revised2 is always less than that of IIP-revised1. However, its performance is still worse than that of SFMTI, which shows that many time slots in IIP-revised2 are still wasted. When the missing rate is small, the execution time of CLS is much greater than that of SFMTI, because most of the slots are collided by multiple tags. When the missing rate increases, the execution time of CLS decreases quickly. When using the same frame size, CLS outperforms SFMTI when the missing rate is more than 0.71. When the missing rate is 0.95, the execution time of CLS is approximately 53.5% that of SFMTI. EDFSA has the worst performance when the missing rate is small.

### B. Impact of Number of Candidate Tags

We change the number of candidate tags from 2000 to 10000 to verify the performances of different approaches. The missing rate is set to 0.9. The result is shown in Fig. 2.

It can be observed that the execution times of all the approaches increase with the increase in the number of candidate tags. This is because the number of tags to be identified increases. The execution time of IIP-revised1 is close to that of EDFSA. The other approaches have better performance compared with EDFSA. CLS always has the best performance in this simulation. The gap between its execution time and that of the other approaches becomes larger when the number of candidate tags increases.

### C. Impact of Different Hashing Strategies

We subsequently check the impact of different hashing strategies that can be used for solving our problem. As mentioned previously, in CLS, time slots are classified into $k$-collision slots ($k = 2, 3, 4......$). We can maintain such time slots for subsequent verification, or reconcile them using a second random number. We refer to the algorithm that maintains 2-collision slots but reconciles 3-collision slots as CLS-K2R3, and the algorithm that maintains both 2-collision slots and 3-collision slots as CLS-K2K3. Both these algorithms do not process $k$-collision slots ($k \geq 3$). If the algorithm maintains all $k$-collision slots ($k \geq 2$), we refer to it as CLS-K2+. CLS-K2R3, CLS-K2K3, and CLS-K2+ use the optimal frame size through a similar calculation as in Section III-B. We compare their execution times for different missing rates. The data of SFMTI is also plotted as the baseline. Notably, in all these algorithms, the Huffman coding is not used. We perform the first round of identification with these algorithms and the result is shown in Fig. 3.

According to the figure, when the missing rate is small, CLS-K2+ underperforms CLS-K2K3 and CLS-K2. This is because most of the actual time slots are collision slots and no tag can be identified. When the missing rate increases, the execution times of all these algorithms decrease and they outperform SFMTI gradually. CLS-K2+ exhibits the best performance among these algorithms when the missing rate is greater than 0.7. In this case, most of the expected collision slots are actually empty, and hence multiple candidate tags corresponding to such a slot are identified. CLS-K2R3 and CLS-K2K3 require more time to complete the identification process because the former considers only expected 2-collision slots and the latter considers only expected 2-collision slots and 3-collision slots. The expected $k$-collision slots ($k > 3$) are not considered in both the algorithms, but these slots can contribute to the time saving if properly used.

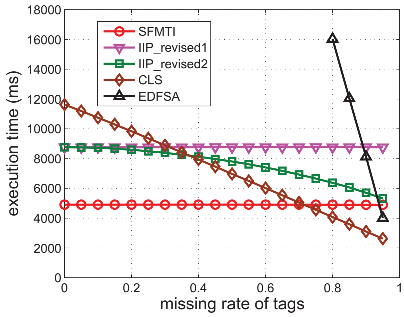


Fig. 1. Execution times of different approaches when varying the missing rate of tags

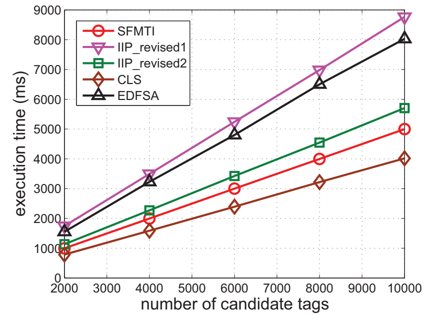


Fig. 2. Execution times of different approaches when varying the number of candidate tags

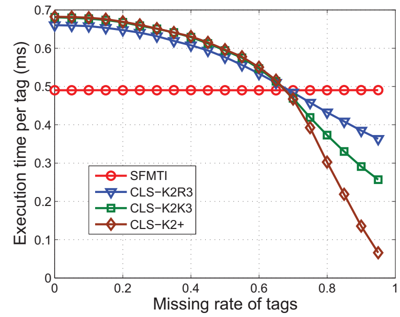


Fig. 3. Execution times of different hashing strategies used in CLS when varying the missing rate of tags

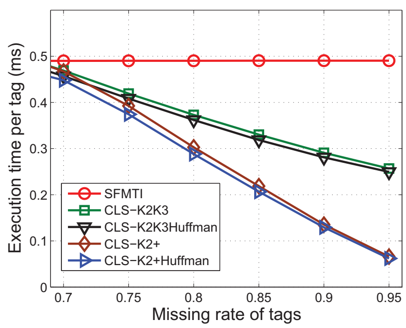


Fig. 4. Execution times of CLS variants with and without Huffman coding when varying the missing rate of tags

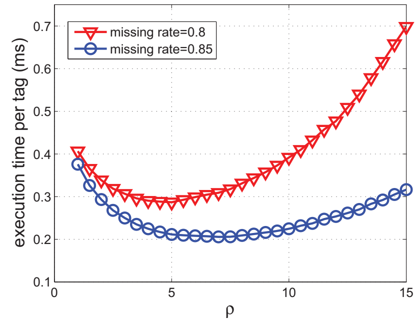


Fig. 5. Execution times of different algorithms when varying $\rho$

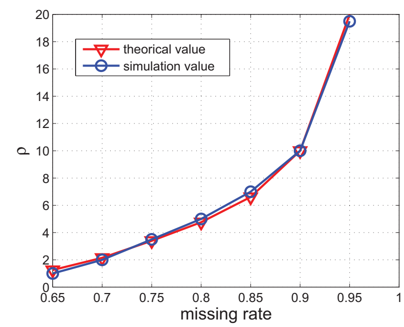


Fig. 6. The optimal $\rho$ computed and obtained in simulations

When the missing rate is small, the gap between CLS-K2K3 and CLS-K2+ is much less than that between CLS-K2K3 and CLS-K2R3 and that between CLS-K2R3 and SFMTI, which shows that the benefit of time saving is attributed mostly to the identification of the expected 2-collision slots and expected 3-collision slots, and marginally to the expected $k$-collision slots ($k > 3$). When the missing rate increases, the benefit of identifying the expected $k$-collision slots ($k > 3$) is more evident.

## D. Impact of Huffman Coding

We also check the effect of Huffman coding used in CLS. Huffman coding is used to reduce the amount of data sent from the reader. Using Huffman coding, a code of 1 bit is used to represent $k$-collision slots ($k \geq 2$), and the codes of 2 bits are used to represent empty slots and singleton slots. When adding Huffman coding to CLS-K2K3 and CLS-K2+, we refer to the algorithm as CLS-K2K3Huffman and CLS-K2+Huffman, respectively. The result is shown in Fig. 4.

It can be observed that the performance of CLS-K2+Huffman is always better than that of CLS-K2+. This is because Huffman coding is only used to reduce the amount of data sent from the reader but maintains the other processing the same. This result also can be verified by the difference between CLS-K2K3Huffman and CLS-K2K3. The benefit of Huffman coding is approximately 4.0%–8.1% for CLS-K2+Huffman over CLS-K2+, and 2.7%–4.6% for CLS-K2K3Huffman over CLS-K2K3, when the missing rate is between 0.7 and 0.95.

## E. Impact of Frame Size

The performance of CLS is affected by the frame size. According to our analysis in Section III-B, the optimal frame size should be set to $N^*/\rho$, where $\rho$ is computed based on the missing rate. We first check the execution time of CLS under different values of $\rho$ when the missing rates are 0.8 and 0.85. Similar to previous simulations, we enable all these algorithms to perform only the first round of identification and then analyze the average execution time to identify a candidate tag. The result is shown in Fig. 5.

It can be observed that there exists an optimal $\rho$ that minimizes the execution time. It is 5 when the missing rate is 0.8, and 7 when the missing rate is 0.85. We further compare the optimal value of $\rho$ in the simulation with the value computed in Section III-B. The result is shown in Fig. 6. It can be observed that the difference between them is small, which shows the correctness of our computation of the optimal frame size.

## V. Related Works

ID collection algorithms can be directly used for stocktaking. These algorithms are classified into tree-based algorithms [3], [4] and ALOHA-based algorithms [6], [7], [8]. They follow a tree traversal model or ALOHA communication model, respectively, to request tags to send their IDs to the reader. Since a tag ID is long and many collisions exist in the transmissions, these algorithms cost significant time.

Recently, some researchers have investigated the problems of missing tag detection and missing tag identification. Missing tag detection [15], [16], [17] aims to detect whether any tag is missing in a given region, but does not care about which tags are missing. It is different from the problem addressed in this paper. Missing tag identification requires obtaining all the IDs of missing tags, which can be used for stocktaking. In the solutions for this problem, the expected time slots computed based on the inventory list are compared with the actual time slots to verify the presence or missing of tags. A tag reply carries 1-bit data rather than the entire ID. IIP [9] verifies the presence of tags corresponding to the expected singleton slots. In order to increase the number of expected singleton slots, IIP requests the expected collided tags to reply with a 50% probability. The drawback of IIP is that the number of expected collision slots accounts for a large proportion, and the expected empty slots are entirely unused. According to [11], these unused time slots account for approximately 48% of all the time slots. In [10], Zhang et al. used multiple RFID readers to identify missing tags. The readers are coordinated to work concurrently and thus reduce the execution time. This approach does not improve the time efficiency of identification for a single reader. SFMTI [11] is proposed to improve IIP. SFMTI reconciles some 2-collision slots and 3-collision slots into singleton slots using a second hashing process. The tags corresponding to the expected empty slots and collision slots are requested not reply and their time slots are skipped. This algorithm is designed for a stationary environment with rare missing tags, and its performance deteriorates when the missing rate increases.

## VI. Conclusion

In this study, we investigated RFID-based stocktaking with a coarse-grained inventory list for the first time. We propose an algorithm called CLS to solve this problem. CLS enables multiple missing tags to hash to a single time slot and thus verifies them together. The processing for all the collision time slots is combined and hence, $k$-collision slots ($k > 3$) can contribute to the identification. CLS reduces the number of states of the time slots from 4 to 3 and subsequently, a Huffman coding technique can be used to reduce the amount of data to be sent from the reader. Extensive simulations are performed and the results show CLS outperforms the state-of-the-art solutions.


## Acknowledgment

This research is supported in part by National Natural Science Foundation of China No. 61502351, Luojia Young Scholar Funds of Wuhan University No. 1503/600400001, and Chutian Scholars Program of Hubei, China.


## References


[1] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.

[2] X. Liu, B. Xiao, S. Zhang, and K. Bu, "One more hash is enough: Efficient tag stocktaking in highly dynamic RFID systems," in *Proc. of IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[3] N. Bhandari, A. Sahoo, and S. Iyer, "Intelligent query tree (IQT) protocol to improve RFID tag read efficiency," in *Proc. of IEEE International Conference on Information Technology*, 2006, pp. 46–51.

[4] L. Pan and H. Wu, "Smart trend-traversal: A low delay and energy tag arbitration protocol for large RFID systems," in *Proc. of International Conference on Computer Communications (mini-conference)*, 2009, pp. 2571–2575.

[5] M. Shahzad and A. X. Liu, "Probabilistic optimal tree hopping for RFID identification," in *Proc. of ACM Sigmetrics/international Conference on Measurement and Modeling of Computer Systems*, 2013, pp. 293–304.

[6] C. Floerkemeier, "Bayesian transmission strategy for framed ALOHA based RFID protocols," in *Proc. of IEEE International Conference on RFID*, 2007, pp. 228–235.

[7] S. Lee, S. Joo, and C. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. of International Conference on Mobile and Ubiquitous Systems:Computing, Networking and Services (Mobiquitous)*, 2005, pp. 166–172.

[8] J.-R. Cha and J.-H. Kim, "Dynamic framed slotted aloha algorithms using fast tag estimation method for RFID system," in *Proc. of the 3rd IEEE Consumer Communications and Networking Conference*, vol. 2, Jan 2006, pp. 768–772.

[9] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2010, pp. 1–10.

[10] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast identification of the missing tags in a large RFID system," in *Proc. of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2011, pp. 278–286.

[11] 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 Transactions on Computers*, vol. 64, no. 1, pp. 87–96, Jan 2015.

[12] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID tags efficiently and anonymously," *Proc. of International Conference on Computer Communications (INFOCOM)*, pp. 1–9, 2010.

[13] W. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length aloha," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 9–15, 2009.

[14] C. Qian, H. Ngan, Y. Liu, and L. M. Ni, "Cardinality estimation for large-scale RFID systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1441–1454, 2011.

[15] C. C. Tan, B. Sheng, and Q. Li, "Efficient techniques for monitoring missing RFID tags," *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, pp. 1882–1889, June 2010.

[16] W. Luo, S. Chen, T. Li, and S. Chen, "Efficient missing tag detection in RFID systems," in *Proc. of International Conference on Computer Communications (INFOCOM)*, 2011, pp. 356–360.

[17] W. Luo, S. Chen, T. Li, and Y. Qiao, "Probabilistic missing-tag detection and energy-time tradeoff in large-scale RFID systems," in *Proc. of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2012, pp. 95–104.