The following publication W. Zhu, X. Meng, X. Peng, J. Cao and M. Raynal, "Collisions Are Preferred: RFID-Based Stocktaking with a High Missing Rate," in IEEE Transactions on Mobile Computing, vol. 19, no. 7, pp. 1544-1554, 1 July 2020 is available at https://doi.org/10.1109/TMC.2019.2911586.

# Collisions are Preferred: RFID-based Stocktaking with a High Missing Rate

Weiping Zhu, *Member, IEEE*; Xing Meng; Xiaolei Peng;
Jiannong Cao, *Fellow, IEEE;* and Michel Raynal

**Abstract**—RFID-based stocktaking uses RFID technology to verify the presence of objects in a region e.g., a warehouse or a library, compared with an inventory list. The existing approaches for this purpose assume that the number of missing tags is small. 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 as the inventory list, and hence many tags in the list are regarded as missing. The missing objects significantly increase the time required for stocktaking. In this paper, we propose an algorithm called CLS (Coarse-grained inventory list based stocktaking) to solve this problem. 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. Moreover, we observe that the missing rate constantly changes during the identification because some of tags are verified present or absent, which affects time efficiency; accordingly, we propose a hybrid stocktaking algorithm called DLS (Dynamic inventory list based stocktaking) to adapt to such changes for the first time. According to the results of extensive simulations, when the inventory list is twenty times that of actually present tags, the execution time of our approach is 36.3% that of the best existing algorithm.

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

## 1 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], [3]. The purpose of stocktaking is to verify the presence of tags (each tag is attached to an object) in a given region such as a warehouse, library, and shopping mall, compared with a given inventory list. The existing approaches for stocktaking mainly include ID collection approaches and missing tag identification approaches. In an ID collection approach, a reader collects all the IDs of tags in its interrogation region, and then compared them with the inventory list. Typical such approaches include tree-based approaches [4], [5], [6] and ALOHA-based approaches [7], [8], [9], [10], [11]. These approaches 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 approaches [12], [13], [14] are proposed for accelerating the stocktaking process. This kind of algorithm requests a reply from tags to the reader in a slotted time frame. Based on the given inventory list of tags, 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 [14] 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 missing rate is high in the stocktaking. One typical example is using handheld RFID reader for stocktaking in a warehouse. The inventory list in its interrelation region cannot be known in advance owing to the mobility of the reader. An approach to use existing stocktaking technique in this scenario is to use the tags in a larger region (e.g., the warehouse) rather than in the interrogation region as its inventory list. However, the tags in the inventory list are much more than the really existing ones, i.e., the missing rate is quite high, which makes the execution time of stocktaking increase significantly. Furthermore, even if we use a stationary reader to perform periodic

- *Weiping Zhu and Xiaolei Peng are with the School of Computer Science, Wuhan University, China.*
  *E-mail: wpzhu@whu.edu.cn, pengxiaolei@whu.edu.cn.*
- *Xing Meng is with the Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong.*
  *E-mail: xmeng.math@whu.edu.cn.*
- *Jiannong Cao is with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.*
  *E-mail: csjcao@comp.polyu.edu.hk.*
- *Michel Raynal is with the Institut Universitaire de France & IRISA-INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France.*
  *E-mail: raynal@irisa.fr.*

stocktaking for the same region (e.g., the inventory of books in a room of a library), an inventory list which is much more than the really existing tags (e.g., the books of the whole library) are probably required, if new objects enter the interrogation region between two consecutive stocktaking processes. Other examples include equipment management in a hospital and human surveillance in a shopping mall. The stocktaking with high missing rate also applies to the scenario that many tags in the interrogation region are moved out (e.g., stolen by some people in a shop). 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 with a high missing rate. We first analyze the impact of missing rate on the execution time of stocktaking. Subsequently we propose an algorithm called Coarse-grained Inventory List based Stocktaking (CLS) to solve this problem where coarse-grained inventory list denotes that the inventory list is much more than the existing tags. 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. Moreover, we observe that the missing rate constantly changes during the identification process, which has not been reported in the literature. This is because an increasing number of tags are verified present or absent with the progress of identification. We design a hybrid algorithm called Dynamic Inventory List based Stocktaking (DLS) adapted to the changing missing rate to achieve the best performance. In summary, this paper offers the following contributions.

- We investigate the RFID-based stocktaking problem with a high missing rate. In many scenarios of s-tocktaking, the inventory list is much more than the really existing objects, and hence required to handle the performance under a high missing rate.
- We proposed a new missing tag identification algorithm called CLS for solving this problem. CLS outperforms the existing algorithms when the missing rate is high.
- We observed that constantly changing missing rate during the identification affects the execution time. Accordingly, we propose an adaptive and hybrid identification algorithm called DLS for solving this problem for the first time.
- We conduct extensive simulations to compare the time efficiency of our approach with those of the existing algorithms. The results show that our approach outperforms the existing approaches especially when the missing rate is high. In a typical handheld RFID reader based stocktaking scenario where the inventory list is 20 times the really existing ones, DLS requires 36.3% execution time compared with the existing approaches.

The rest of the paper is organized as follows: Section 2 describes the system models used in this study and formulates the problem. Our solution is illustrated in Section 3. The simulation results are reported in Section 4. Section 5 reviews the related works and Section 6 concludes the paper.

This paper is based on our conference paper [15]. In this version, we extend the approach to automatically adapt to the change of missing rate during the identification. Additional discussions and evaluation results are also included in this paper.

## 2 System Model and Problem Formulation

In this section, we introduce the system models used in this paper and subsequently formulate the problem.

### 2.1 Application Model

There are many objects in a large region such as a warehouse, library, and shopping mall. An RFID tag is attached to each object. The IDs of these RFID tags are recorded in an inventory list and known in advance. For example, the inventory list of a warehouse can be obtained by monitoring the shipping in and out of objects at the entrances and exits. 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 follows:

$$p = \frac{|N^*| - |N|}{|N^*|}. \tag{1}$$

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.

### 2.2 Communication Model

Following the literature, we assume that the RFID reader communicates with the RFID tags using the Reader Talks First (RTF) mode [16]. In this mode, 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 [12]. The hash functions of all the candidate tags are known by the reader (e.g., provided by the manufacturer of the tags). We assume reliable communications in the identification in this study.

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*.

The time cost of reader requests and tag replies is considered in the identification process. 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 [14], where $T_{tag}$ is 2.4 ms (including the wait time between any two consecutive transmissions), $T_{short}$ is 0.4 ms, and $T_{long}$ is 0.8 ms.

In this study, we assumed that the communication is reliable, i.e., the requests from the readers and responses from the tags are intact. Although this assumption is commonly used in the existing missing tag identification algorithms [3], [12], [14], [17], it may not be true for real applications. To solve this problem, we use a probability model similar to those in [18], [19], [20] for the communication. The readers and tags will be required to communicate multiple times according to a required readability. The detailed design is out of the scope of this paper and will be addressed in future work.

### 2.3 The Problem

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.

Our problem is different from the existing missing tag identification problem, wherein the inventory list and present tags are almost the same, i.e., the missing rate is small. This is true if the RFID system performs high-frequency periodical stocktaking for the same region (therefore, the inventory list can be obtained from the previous stocktaking), and there is no object entering the interrogation region between two consecutive stocktaking processes. However, such an inventory list is often difficult to determine. For example, stocktaking is performed for the first time, tags move freely in the region, a handheld RFID reader is used, etc. In such scenarios, our algorithm can be used to improve the time efficiency of identification process.

## 3 The Solution

We first analyze some existing works for stocktaking, and thereafter illustrate our algorithm in detail. Subsequently, we discuss how to determine the key parameters in the algorithm and other technical considerations.

### 3.1 Existing Approaches

A straightforward approach for stocktaking is to collect all the IDs of present tags [10], [11]. Owing to the collisions among tags, each tag requires to transmit its ID 2.72 times on average before being identified [10], [11], and a short message from the reader is required to acknowledge the success or failure of each transmission. Therefore, the execution time of this approach is approximately 2.72$|N|(t_{tag}$ + $t_{short})$. When $|N|$ is large, it requires a significant amount of time.

Another approach we can use is the polling approach, wherein the reader sends the IDs of the candidate tags one by one, and a tag replies with a short message on receiving its ID. Its execution time is approximately $|N^*|(t_{tag}+t_{short})$. This approach requires even more time than the ID collection approach since $|N^*| >> |N|$ in our problem.

In order to avoid transmitting tag IDs, missing tag identification algorithms are proposed. The best performance among such algorithms is achieved by SFMTI [14], i.e., the execution time is approximately $|N^*|t_{short}$†. It has desirable performance when $|N^*|$ is approximately $|N|$, but when $|N^*| >> |N|$, its performance deteriorates quickly. However, we believe its basic idea is useful for solving our problem, and thus we describe it as follows.

SFMTI first generates random numbers $r_1$ and $r_2$, and frame size $f$. The expected time slots are computed based on $r_1$ and $f$ for all the tags in $N^*$. Since the hash functions of all the tags in $N^*$ are known, this work can be performed at the reader side. The tags corresponding to an expected 2-collision slot or expected 3-collision slot perform a second hashing using $r_2$, and the frame sizes of 2 and 3, respectively. If the tags are hashed to entirely different values (e.g., tag $t_1$, $t_2$, $t_3$ are hashed to 0, 1, and 2, respectively), each of them is allocated a different time slot; therefore, the expected collision slot is transformed into an expected singleton slot. This process is called reconcilement. Subsequently, the reader sends the corresponding information to the tags and requests them to reply. Only the tags corresponding to the expected singleton slots reply with a short message, and all the other tags remain silent. For each expected singleton slot, the corresponding tag can be identified as present or missing, and is not required to participate in subsequent identification.

In our problem, the number of missing tags is significantly large and SFMTI spends much time to identify them. Therefore, we believe that the identification priority should be given to the missing tags rather than the present tags. If possible, we can enable multiple missing tags to hash to a single slot and thus verifies them together to improve the time efficiency. Moreover, in SFMTI, the expected $k$-collision slots ($k > 3$) are not used, which not only misses the identification opportunities in these slots, but also perplexes the processing because tags require to distinguish their processing from that of the expected 2-collision slots. We will attempt to combine the processing of all the expected collision slots.

†. The average time for SFMTI to identify a tag is 0.474 ms, and $t_{short}$ is 0.4 ms [14]

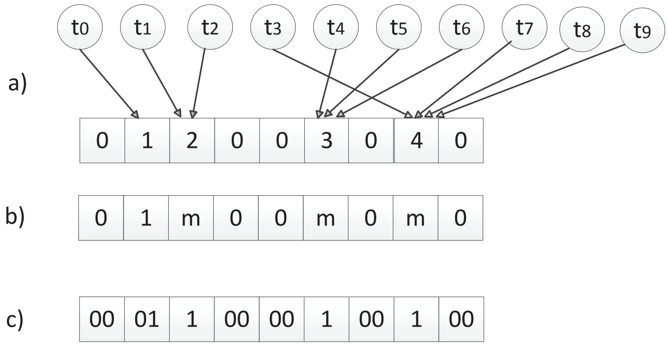


Fig. 1: Identification process of CLS. a) The candidate tags are allocated to time slots. The number notated in a time slot denotes the number of tags allocated to it. b) The changed allocation vector. c) The filter vector to be sent to the tags.

### 3.2 Basic Solution

We propose an algorithm called Coarse-grained Inventory List based Stocktaking (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, each candidate tag is allocated a time slot. In the filter vector generation phase, a filter vector is generated based on the expected time slots. In the tag verifying phase, the reader broadcasts the filter vector to the tags, and the tags reply to the reader. Subsequently, the reader computes the identified tags and simultaneously the tags change their status. We describe the details as follows.

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 2, 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. An example of the expected time slots and the corresponding allocation vector are shown in Fig. 1(a).

In the filter vector generation phase, we change all the elements of $A$ greater than 1 into $m$. For example, Fig. 1(b) illustrates the changed $A$ corresponding to that in Fig. 1(a). 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". We use Huffman coding because its decoding can be easily implemented in the tag side. Moreover, we observed that when $p$ is large, the number of $k$-collision slots ($k \geq 2$) is almost half of all the time slots; hence, $k$-collision slots are represented using a code of 1-bit length and the other two kinds of slots are represented using a code of 2-bit length. Such encoding can save about one fourth data amount to be sent. An example of the filter vector is shown in Fig. 1(c).

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 $V$. As shown in Fig. 1(b), tag $t_0$ will reply in the first time slot because there is only an empty time slot before its index, and tags $t_4$, $t_5$, and $t_6$ will reply in the third time slot because there is a singleton slot and an $m$-collision slot before their indexes. 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 and hence, $A(i)$ can only be 1 or $m$.

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.

There are several differences between CLS and SFMTI. First, CLS removes the reconcilement process in the slot allocation phase. To avoid collisions among the candidate tags, SFMTI reconciles a few of the expected 2-collision slots and expected 3-collision slots into the expected singleton slots. In CLS, we prefer such collisions because they are likely to lead to the successful identification of multiple missing tags. Second, CLS utilizes $k$-collision slots ($k > 3$) to improve its time efficiency. SFMTI does not use such slots because the processing of $k$-collision slots ($k > 3$) is different from that of the other types of slots at the tag side. To support these slots, SFMTI requires 3 bits (to distinguish empty slots, singleton slots, 2-collision slots, 3-collision slots, and $k$-collision slots ($k > 3$)) rather than 2 bits of data (to distinguish empty slots, singleton slots, 2-collision slots, and 3-collision slots) to represent an element of the filter vector. This incurs more time cost compared with benefits. CLS supports $k$-collision slots ($k > 3$), however, does not introduce an additional cost, because the processing of all the collision slots is the same. Third, CLS uses the Huffman code to further reduce approximately one-fourth of the amount of data sent by the reader. Fourth, in CLS, some tags that replied to the reader are required to reply again in subsequent rounds; whereas, in SFMTI, none of the replying tags are required to participate in subsequent rounds. This is because only the expected singleton slots exist in the tag verifying phase of SFMTI, and therefore, all the replies lead to the successful identification of a candidate tag. In contrast, CLS allows $k$-collision slots ($k \geq 2$) in the identification. Owing to the use of a short reply, we cannot distinguish between a reply from one tag and that from multiple tags. Therefore, these tags are required to reply again in subsequent rounds. However, such a cost is negligible compared with the benefits that can be incurred.

### 3.3 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 [14] but with additional processing of missing rate. As previously mentioned, $N^*$ denotes the candidate tags and $p$ denotes the missing rate. We first compute the probability that a time slot is a $k$-collision slot ($k = 0, 1, 2....$), $P_k$, as follows:

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

The number of expected $k$-collision slots, $N_k$, can be computed as follows:

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

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:

$$P'_k = C_{N^*}^k (\frac{1}{f})^k (1 - \frac{1}{f})^{N^* - k} \times p^k. \tag{4}$$

The number of such time slots is

$$N'_k = f \times C_{N^*}^k (\frac{1}{f})^k (1 - \frac{1}{f})^{N^* - k} \times p^k. \tag{5}$$

Therefore,

$$P'_k = P_k \times p^k. \tag{6}$$

In CLS, a tag can be identified present or missing in two cases. First, the tag corresponding to an expected singleton slot can be 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 ($k > 2$) can be identified as missing tags if no reply is received. Subsequently, the number of identified tags can be calculated as follows:

$$\Re = N_1 + \sum_{k=2}^{N^*} k N'_k. \tag{7}$$

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

$$T_r = (N_0 + N_1) \times \frac{t_{tag}}{96} \times 2 + (f - N_0 - N_1) \times \frac{t_{tag}}{96}. \tag{8}$$

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

$$T_v = \sum_{k=1}^{N^*} N_k \times t_{short}. \tag{9}$$

Thus, the average time required for identifying a tag is

$$\begin{aligned} \frac{T}{\Re} &= \frac{T_r + T_v}{N_1 + \sum_{k=2}^{N^*} k N'_k} \\ &= \frac{(f + N_0 + N_1) \times t_{tag}/96 + \sum_{k=1}^{N^*} N_k \times t_{short}}{N_1 + \sum_{k=2}^{N^*} k N'_k} \\ &= \frac{(1 + P_0 + P_1) \times t_{tag}/96 + (1 - P_0) \times t_{short}}{P_1 + \sum_{k=2}^{N^*} k P'_k}. \end{aligned} \tag{10}$$

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

$$\begin{aligned} P_k &= C_{N^*}^k (\frac{1}{f})^k (1 - \frac{1}{f})^{N^* - k} \\ &\approx \frac{1}{k!} \rho^k e^{-\rho} (k = 0, 1....). \end{aligned} \tag{11}$$

$$\begin{aligned} \sum_{k=2}^{N^*} k P'_k &= \sum_{k=1}^{N^*} k P'_k - P'_1 \\ &= N^* \frac{p}{f} \sum_{k=1}^{N^*} C_{N^*-1}^{k-1} (\frac{p}{f})^{k-1} (1 - \frac{1}{f})^{(N^* - k)} - P'_1 \\ &= N^* \frac{p}{f} (1 - \frac{1}{f} + \frac{p}{f})^{N^* - 1} - P'_1 \\ &\approx p\rho e^{-\rho(1-p)} - p\rho e^{-\rho}. \end{aligned} \tag{12}$$

Substitute Equation (10) with Equations (11) and (12), we obtain

$$\begin{aligned} \frac{T}{\Re} &\approx \frac{(1 + e^{-\rho} + \rho e^{-\rho}) \times t_{tag}/96 + (1 - e^{-\rho}) \times t_{short}}{\rho e^{-\rho} + p\rho e^{-\rho(1-p)} - p\rho e^{-\rho}} \\ &= \frac{(1 + e^{-\rho} + \rho e^{-\rho}) \times t_{tag}/96 + (1 - e^{-\rho}) \times t_{short}}{\rho p e^{-\rho(1-p)} + (1 - p)\rho e^{-\rho}}. \end{aligned} \tag{13}$$

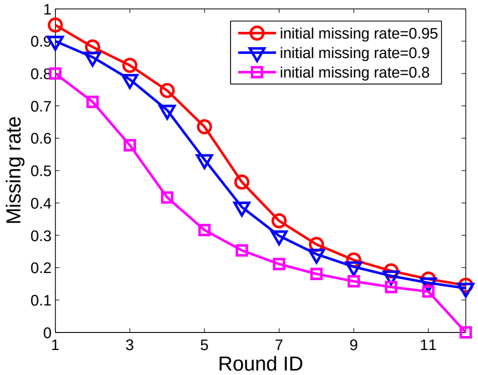


Fig. 2: Change of missing rate in the identification

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 [16], [21], [22], [23] can also be used to estimate the missing rate based on the replies in the identification process.

### 3.4 Adaptive Stocktaking

After verifying the identification process in detail, we observe that the missing rate changes in different rounds. This is because an increasing number of tags are verified present or absent with the progress of identification. As shown in Fig. 2, the missing rate gradually decreases with the execution when its initial value is 0.8, 0.9, or 0.95. This phenomenon has not been investigated previously. If a fixed frame size is used as in the existing works [12], [14], the time efficiency of identification will be affected.

We propose an algorithm called Dynamic Inventory List based Stocktaking (DLS) to solve this problem. It is a hybrid algorithm based on CLS and existing algorithms. First, it estimates the missing rate at the beginning of each round and adjusts the frame size accordingly.

We assume that before round $i$, the candidate tags are $N_i$ and the missing rate is $p_i$. During the identification of round $i$, the tags verified as missing are $M_i$ and the tags verified as present are $H_i$, which are recorded by the reader. Thus, the missing rate after this round can be computed as follows:

$$p_{i+1} = \frac{|N_i| \times p_i - |M_i|}{|N_i| - (|M_i| + |H_i|)}. \quad (14)$$

According to the updated missing rate, a new frame size can be computed. We can also compute the values of frame size in advanced based on a set of missing rates (e.g., from 0.6 to 1 with a step length of 0.5), and subsequently, in the identification process, the frame size is adjusted to that with the closest missing rate.

After that, DLS uses a hybrid and adaptive strategy for identification. CLS is designed for the scenarios with a high missing rate, and SFMTI is designed for the scenarios with a low missing rate. Therefore, when the missing rate is sufficiently small, the performance of CLS is worse than that of SFMTI. And when the missing rate is extremely large (i.e., the number of present tags is quite small), ID collection approaches can be directly used rather than using CLS. The strategy of DLS is as follows: when the missing rate is between $t_1$ and $t_2$, CLS is used and the frame size is adjusted as illustrated in the previous subsection; when the missing rate is less than $t_1$, SFMTI is used; when the missing rate is greater than $t_2$, ID collection approach is used. Using this method, CLS cooperates with other approaches to achieve the optimal performance in all scenarios.

We then compute the values of $t_1$ and $t_2$ as follows. Similar to the analysis of CLS, we have the following equation for SFMTI [14]:

$$\frac{T_{\text{SFMTI}}}{\Re_{\text{SFMTI}}} \approx \frac{t_{tag}/48 + (\rho e^{-\rho} + \frac{1}{2}\rho^2 e^{-\rho} + \frac{1}{9}\rho^3 e^{-\rho}) \times t_{short}}{\rho e^{-\rho} + \frac{1}{2}\rho^2 e^{-\rho} + \frac{1}{9}\rho^3 e^{-\rho}}. \quad (15)$$

By equating this equation to Equation 13, we obtain the value of $p$ as $t_1$ i.e., 0.679.

An ID collection approach requires 2.72($t_{tag}$ + $t_{short}$) to identify a tag on average. Combining Equation 13, we solve $p$ in the following equation and obtain $t_2$ of 0.994.

$$\frac{T}{\Re} \times |N^*| = 2.72(t_{tag} + t_{short}) \times (1-p)|N^*|. \quad (16)$$

Notably, while the difference between using SFMTI and CLS is large when the missing rate is small, the difference between using ID collection approaches and CLS is negligible when the missing rate is extremely large. If we intend to simplify the design, we can use CLS when the missing rate is greater than $t_1$, and use SFMTI otherwise.

### 3.5 Discussion

Finally, we analyze the compliance of DLS to the EPC C1G2 RFID Protocol and its implementations. A few modifications are required in the protocol to implement DLS. In DLS, apart from frame size, a filter vector should be sent to the tags. The Query command in the protocol should be modified to include the filter vector. Correspondingly, the process logic of tags should be modified. A tag in the ready state supports Huffman decoding and computes the replying time slot based on frame size, index $s$, and nonempty time slots before $s$. A tag in the reply state responds with a short message rather than an RN16, and then, updates its status. If the filter vector is too long to be transmitted in one time slot, similar to existing works [12], [13], [14], [24], the filter vector can be divided into multiple segments of 96 bits and sequentially transmitted. In this case, the Query command and its process logic should be further modified to support the multiple segments.

In the implementation of DLS, the method to compute $s$ is consistent with the existing ALOHA-based approaches. For the Huffman decoding, a similar function is already required for tags in the EPC C1G2 RFID Protocol, and hence, can be reused or revised. For example, the commands sent from the reader are distinguished by their codes with different lengths, including "00" for QueryRep, "01" for

ACK, "1000" for Query, "1001" for QueryAdjust, and others. Detailed implementations of this differ for different types of tags. For example, bit-by-bit matching is possible to implement DLS, considering that there are only three symbols: "*1*" for $m$, "*00*" for *0*, and "*01*" for *1*. $m$ can be represented by *2* in the implementation. 96 bits temporal storage for storing a segment of filter vector and a counter of non-zero time slots before $s$ are required in tags. 16 bits are sufficient for the counter if the maximum value of $N^*$ is 65536. The time complexity of this processing is proportional to the length of the filter vector (and $N^*$).

## 4 Performance Evaluation

Simulations are carried out to validate the effectiveness of the proposed approach. We first compare the execution time of our approach with that of the other approaches, and subsequently analyze the impact of different parameters on the execution time in the algorithm design. A hundred simulations are repeated to obtain each data point of the figures. The confidence level is 0.95.

### 4.1 Impact of Missing Rate

We first compare the performances of CLS and DLS with those of SFMTI [14], IIP [12], and EDFSA [10] 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 $N^*/1.68$ [14], and the frame size of IIP-revised1 and IIP-revised2 to $N^*/1.516$ [12] to achieve the best performance. For the sake of fairness, we also set the frame size of CLS the same as that of SFMTI. DLS uses the simplified design, by using CLS when the missing rate is greater than 0.679 and by using SFMTI otherwise. The frame size of DLS is dynamically adjusted. We compare the performances of these approaches by varying the missing rate. $N$* is set to 10000. The result is shown in Fig. 3.

It can be observed that the execution times of IIP-revised1 and SMFTI are stable, i.e., approximately 8760 ms and 4900 ms, respectively. 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. It gradually decreases when the missing rate increases, because more expected collision slots become empty. However, its performance is still worse than that of SFMTI, which shows that many time slots in IIP-revised2 are still wasted. CLS aims to identify multiple tags in one time slot. However, it is effective only when the missing tags account for a large proportion of total tags. 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. DLS always shows the optimal performance in this figure. Owing to the dynamic adjustment of the frame size, DLS slightly reduces the execution time when the missing rate is less than 0.9, and more apparently when the missing rate is greater than 0.9. When the missing rate is 0.95, the execution time of DLS is approximately 67.8% that of CLS and 36.3% that of SFMTI. EDFSA has the worst performance when the missing rate is small. When the missing rate is extreme large, it shows desirable performance because the number of present tags is small. According to our simulation results, it outperforms DLS when the missing rate is larger than 0.9985.

If we do not have prior knowledge of the missing rate, we can use LOF [22] to estimate it. We can invoke LOF to do 10 trials (a rough estimation of tags according to [23] and also used in the first phase of $SRC_S$ [23]) to calculate the missing rate, and then use DLS for stocktaking. We denote this approach as LOF-DLS. We compare the performances of LOF-DLS and DLS by varying the missing rate; the result is shown in Fig. 4. It can be seen that their performances are quite close. The execution time of LOF-DLS is 3.5%-7.7% more than that of DLS when the missing rate is between 0.70 and 0.95. This shows DLS can achieve desirable performance after a rough estimation of missing rate if there is no prior knowledge of the missing rate.

We further compare the performances of different approaches when the missing rate is between 0.994 and 0.999. The results are shown in Fig. 5 and Fig. 6. We sperate them into two figures to show the differences of them more clearly. Buzz [25] is also included in this comparison. Buzz may obtain wrong results of the present RFID tags. We repeat the processing and only count Buzz's execution time when the result is correct. Since Buzz obtains only a short ID of a tag, we use SFMTI or a polling approach to further obtain the full tag IDs, denoted by Buzz-SFMTI and Buzz-Polling. For Buzz-SFMTI, the time slots are computed based on the short IDs, and the responses from tags are the full tag IDs. For Buzz-Polling, a reader sends the short IDs one by one, and the corresponding tags reply their full tag IDs.

It can be observed that the execution times of IIP-revised1 of SMFTI are the same with those in Fig. 3. The execution time of IIP-revised2 is close with that of SMFTI because IIP-revised2 benefits from that many expected collision slots become empty. Their execution times are much more those of CLS and DLS. Buzz-SFMTI and Buzz-Polling have similar performance, which shows obtaining short IDs of tags dominate the execution time of Buzz. The execution time of Buzz (Buzz-SFMTI or Buzz-Polling) is less than those of SMFTI and CLS, but greater than that of DLS. It shows that the dynamic change of frame size is effective. The execution time of LOF-DLS is about 32 ms more than that of DLS, due to the cost of missing rate estimation. EDFSA outperforms DLS when the missing rate is larger than 0.998.

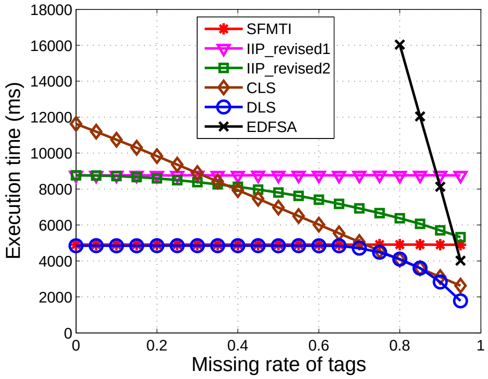


Fig. 3: Execution times of different approaches when varying the missing rate of tags from 0 to 1

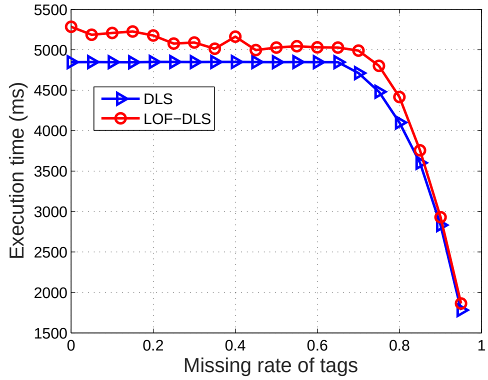


Fig. 4: Execution times of DLS and LOF-DLS when varying the missing rate of tags from 0 to 1

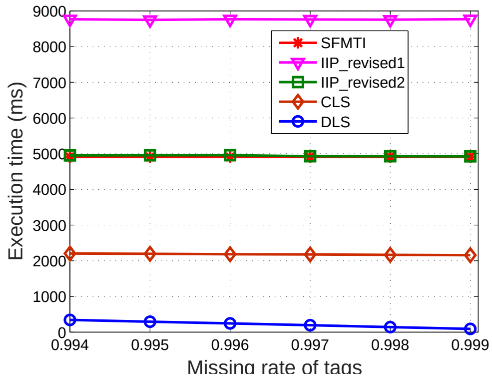


Fig. 5: Execution times of different approaches when varying the missing rate of tags from 0.994 to 0.999

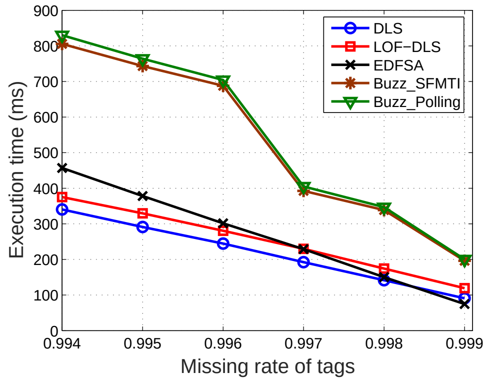


Fig. 6: Execution times of different approaches when varying the missing rate of tags from 0.994 to 0.999

### 4.2 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. 7.

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. DLS 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.

### 4.3 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, similar to the processing in SFMTI [14]. 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 use $k$-collision slots ($k > 3$) for identification. 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 3.3. We compare their execution times for different missing rates. The data of SFMTI is also plotted as the baseline. Since the missing rate changes in different rounds, we perform only the first round of identification with these algorithms and subsequently analyze the average execution time required to identify a candidate tag. Notably, in all these algorithms, the Huffman coding is not used. $N^*$ is set to 10000. The result is shown in Fig. 8.

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.

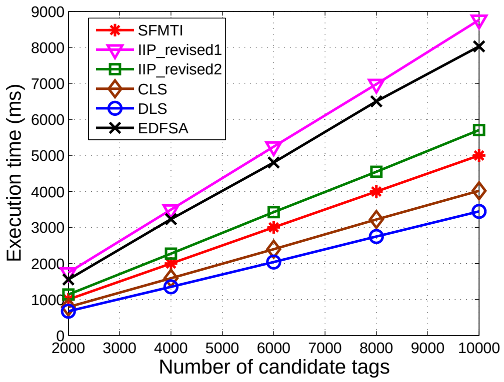


Fig. 7: Execution times of different approaches when varying the number of candidate tags

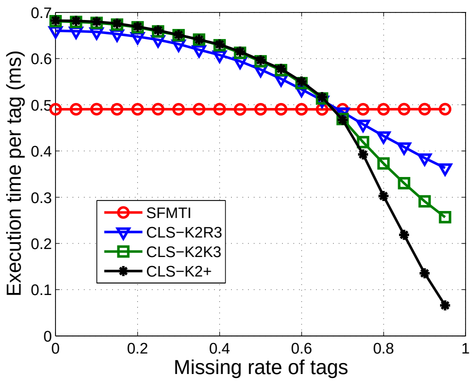


Fig. 8: Execution times of different hashing strategies used in CLS when varying the missing rate of tags

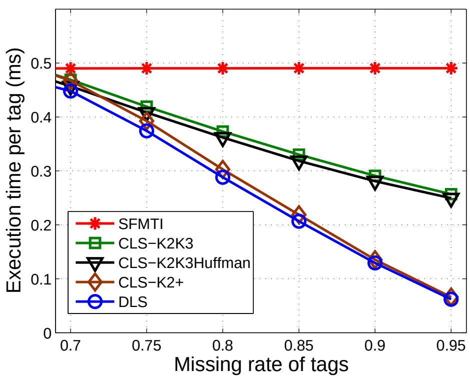


Fig. 9: Execution times of CLS variants with and without Huffman coding when varying the missing rate of tags

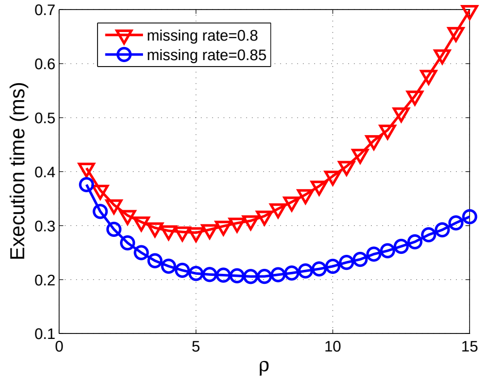


Fig. 10: Execution times of different algorithms when varying $\rho$

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.

### 4.4 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. This technique cannot be used for SFMTI and CLS-K2R3 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. 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, we refer to the algorithm as CLS-K2K3Huffman. DLS is equivalent to CLS-K2+ together with Huffman coding when the missing rate is large. We set $N^*$ to 10000. The result is shown in Fig. 9.

It can be observed that the performance of DLS 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 DLS 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.

### 4.5 Impact of Frame Size

The performance of CLS is affected by the frame size. According to our analysis in Section 3.3, 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 number of candidate tags is 10000 and 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. 10.

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 3.3. The result is shown in Table 1. It

TABLE 1: The optimal $\rho$ computed and obtained in simulations

| Missing rate | 0.65 | 0.7 | 0.75 | 0.8 | 0.85 | 0.9 | 0.95 |
|---|---|---|---|---|---|---|---|
| Theorical value | 1.3 | 2.2 | 3.4 | 4.8 | 6.7 | 10.0 | 20.0 |
| Simulation value | 1.0 | 2.0 | 3.5 | 5.0 | 7.0 | 10.0 | 19.5 |

can be observed that the difference between them is small, which shows the correctness of our computation of the optimal frame size.

## 5 Related Works

ID collection algorithms can be directly used for stocktaking. These algorithms are classified into tree-based algorithms [4], [5] and ALOHA-based algorithms [7], [8], [9], [10], [11], [26]. 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 [27], [28], [29], [30], [31], [32] 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 [12] 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 [14], these unused time slots account for approximately 48% of all the time slots. The authors in [12] also investigated other algorithms including TPP, TPP/TR, and TPP/CSTR by introducing a polling phase and/or eliminating some collision slots. The reported performances of them are worse than that of IIP. In [13], 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 [14] 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. Buzz [25] used compressive sensing to determine the RFID tags. There are two problems for this approach. One is that much memory space is required to perform 0-1 integer linear programming to obtain the results. The other is that Buzz only obtains a short ID of a tag, and additional processing (e.g., ID polling) is required to obtain the full tag ID. According to our simulations, its execution time is similar with that of EDFSA when the missing rate is large. PCMTI is proposed to arrange a pair of tags to reply in a time slot simultaneously and use manchester coding to determine which tags replied [24]. This protocol requires modification in the physical layer. Moreover, $k$-collisions ($k > 2$) are not utilized. RUN [17], HARN [3], OMTI [33] are proposed to solve the problem when some unexpected tags are present. These tags may enable some missing tags to be detected as present tags. iVEKI determines the missing of key tags among a number of ordinary tags using a privacy preserved way [34]. These problems are different from the problem investigated in this study.

## 6 Conclusion

In this study, we investigated RFID-based stocktaking with a high missing rate. It is different from previous missing tag identification problem because the candidate tags are significantly more than the present tags. We propose an algorithm called CLS (Coarse-grained Inventory List based Stocktaking) 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. We also propose an extension to CLS, called DLS (Dynamic Inventory List based Stocktaking) to adaptively handle the constantly changing missing rate during the identification process for the first time. We have performed extensive simulations for validating the effectiveness of the proposed approach. According to the results, when the missing rate is 0.95 and the number of candidate tags is 10000, our approach requires only 36.3% of the execution time compared with the state-of-the-art solutions.

## Acknowledgments

We thank Mingze Li, Tingxi Zou, and Xiaoyue Gao for developing the simulation programs. This research is supported in part by National Natural Science Foundation of China No. 61502351, National Key R&D Program of China No. 2018YFC1604000, Chutian Scholars Program of Hubei, China, Luojia Young Scholar Funds of Wuhan University No. 1503/600400001, and Alibaba Innovative Research (AIR) Program No. H-ZG6G.

## References

[1] K. Finkenzeller, *RFID handbook: Fundaments and Application in Contactless Smart card and Identification*. John Wiley & Sons, 2003.
[2] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.
[3] 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.
[4] 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.

[5] 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.
[6] 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.
[7] H. Vogt, "Multiple object identification with passive RFID tags," in *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, 2002, pp. 6–12.
[8] J. R. Cha and J. H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," in *Proc. of International conference on parallel and distributed systems (ICPADS)*, 2005, pp. 63–67.
[9] C. Floerkemeier, "Bayesian transmission strategy for framed ALOHA based RFID protocols," in *Proc. of IEEE International Conference on RFID*, 2007, pp. 228–235.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] 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.
[15] W. Zhu, X. Meng, X. Peng, J. Cao, and M. Raynal, "Time-efficient RFID-based stocktaking with a coarse-grained inventory list," in *Proc. of 2018 IEEE/ACM International Symposium on Quality of Service (IWQoS)*, 2018.
[16] 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.
[17] M. Shahzad and A. X. Liu, "Fast and reliable detection and identification of missing RFID tags in the wild," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3770–3784, December 2016.
[18] ——, "Probabilistic optimal tree hopping for rfid identification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 796–809, 2015.
[19] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li, and D. Chen, "Efficient tag identification in mobile RFID systems," in *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, 2010, pp. 1–9.
[20] K. Fyhn, R. M. Jacobsen, P. Popovski, and T. Larsen, "Fast capture recapture approach for mitigating the problem of missing rfid tags," *IEEE Transactions on Mobile Computing*, vol. 11, no. 3, pp. 518–528, 2012.
[21] 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.
[22] 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.
[23] Z. Zhou, B. Chen, and H. Yu, "Understanding RFID counting protocols," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 312–327, 2016.
[24] L. Zhang, W. Xiang, I. Atkinson, and X. Tang, "A time-efficient pair-wise collision-resolving protocol for missing tag identification," *IEEE Transactions on Communications*, vol. 65, no. 12, pp. 5348–5361, 2017.
[25] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low-power backscatter networks," *Acm Sigcomm Computer Communication Review*, vol. 42, no. 4, pp. 61–72, 2012.
[26] W. Zhu, J. Cao, H. Chan, X. Liu, and V. Raychoudhury, "Mobile RFID with a high identification rate," *IEEE Trans. on Computers*, vol. 63, no. 7, pp. 1778–1792, July 2014.
[27] 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.
[28] 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.
[29] 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.
[30] J. Yu, L. Chen, R. Zhang, and K. Wang, "On missing tag detection in multiple-group multiple-region RFID systems," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1371–1381, May 2017.
[31] ——, "Finding needles in a haystack: Missing tag detection in large RFID systems," *IEEE Transactions on Communications*, vol. 65, no. 5, pp. 2036–2047, May 2017.
[32] Y. Zhang, S. Chen, Y. Zhou, and O. Odegbile, "Missing-tag detection with presence of unknown tags," in *Proc. of the 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2018, pp. 1–9.
[33] Y. Wang, J. Liu, X. Wang, F. Zhu, and L. Chen, "Missing tag identification in open RFID systems," in *Proc. of the IEEE International Conference on Communications*, 2017, pp. 1–6.
[34] H. Chen, Z. Wang, F. Xia, Y. Li, and L. Shi, "Efficiently and completely identifying missing key tags for anonymous RFID systems," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2915–2926, Aug 2018.

**Weiping Zhu** is currently an associate professor of School of Computer Science at Wuhan University. He is a CHUTIAN Young Scholar and LUOJIA Young Scholar. He received the PhD degree from Hong Kong Polytechnic University, Hong Kong, in 2013. Before current position, he worked for Hong Kong Polytechnic University and The University of Hong Kong. He also visited IRISA-INRIA, France, from 2011 to 2012. His research interests include RFID, WSN, distributed computing, and pervasive computing. He has published over 40 papers in major international journals and conference proceedings.

**Xing Meng** is currently a post graduate student of Department of Industrial and Manufacturing Systems Engineering at The University of Hong Kong. He received the BSc degree from Wuhan University, China in 2017 in applied mathematics. His research interests include RFID and supply chain management.

**Xiaolei Peng** was a post graduate student of the School of Computer Science at Wuhan University from 2016 to 2018. He received the BS degree from Chongqing University of Science & Technology, China in 2016 in computer science. His research interests include RFID and big data analysis.

**Jiannong Cao** is currently a chair professor of the Department of Computing at Hong Kong Polytechnic University. He is also the director of the Internet and Mobile Computing Lab in the department and the director of Universitys Research Facility in Big Data Analytics. He received the BSc degree from Nanjing University, China, in 1982, and the MSc and PhD degrees from Washington State University, USA, in 1986 and 1990, all in computer science. His research interests include parallel and distributed computing, wireless sensing and networks, pervasive and mobile computing, and big data and cloud computing. He has co-authored 5 books, co-edited 9 books, and published over 600 papers in major international journals and conference proceedings. He has directed and participated in 90 research and development projects and, as a principal investigator, obtained over HK$43 million grants. He served the Chair of the Technical Committee on Distributed Computing of IEEE Computer Society 2012-2014 and a member of IEEE Fellows Evaluation Committee of the Computer Society and the Reliability Society. He is a fellow of IEEE and ACM distinguished member.

**Michel Raynal** is a professor of computer science at IRISA-INRIA, Rennes, France. His main research interests are the basic principles of distributed computing systems. He is a world leading researcher in the domain of distributed computing. He is the author of numerous papers on distributed computing (more than 120 in journals and 250 papers in international conferences) and is well-known for his distributed algorithms and his nine books on distributed computing. He has chaired the program committee of the major conferences on the topic (e.g., ICDCS, DISC, SIROCCO, and OPODIS). He has also served on the program committees of many international conferences, and is the recipient of several “Best Paper” awards (ICDCS 1999, 2000 and 2001, SSS 2009, Europar 2010). His h-index is 57. Since 2010, he has been a senior member to the prestigious “Institut Universitaire de France”. He is a member of Academia Europaea.