2011 Ninth IEEE/IFIP International Conference on Embedded and Ubiquitous Computing

Efficient Monitoring of Dynamic Tag Populations in RFID Systems

Qingjun Xiao, Kai Bu, Bin Xiao
Department of Computing
The Hong Kong Polytechnic University
Email: {csqjxiao, cskbu, csbxiao}@comp.polyu.edu.hk

***Abstract*—As RFID tags become more ubiquitously available, e.g., in a supermarket, it is necessary to monitor larger-scale tag populations in a dynamic environment to get updated tag information. This paper considers the problem of monitoring a dynamic tag population, to identify both the missing tags and new tags. Traditional approach can solve the problem by collecting all tag IDs in the current population, which could be slow because it ignores the knowledge of the tag population in a previous scan. To be more efficient, this paper presents two protocols: (1) a baseline protocol with optimized length of random number bits, (2) an improved one-phase protocol with easy labor to identify only the new and missing tags in ALOHA frames by fully utilizing previous tag population knowledge. Our analysis shows that the one-phase protocol can improve the monitoring accuracy by 25% and improve the time efficiency by 55%, as compared with the two-phase protocol proposed in a recent paper which also identifies population changes.**

***Keywords*-Ubiquitous Computing, RFID, Dynamic Tag Population, Tag Population Monitoring**

I. INTRODUCTION

About a decade ago, RFID (radio-frequency identification) was envisioned as one of the enabling techniques towards the future of ubiquitous computing [1]. Nowadays, RFID has arguably become one of the most successful technologies in the computing history, which has been adopted by a wide range of applications, e.g. warehouse management, object tracking and inventory control. RFID's success is largely due to its clear advantages over the classical barcode system. RFID extends the operation distance from inches to tens of feet (for passive tags) or even hundreds of feet (for active tags). RFID also dramatically improves the tag reading speed: barcode attached products can only be checked manually, one at a time, while all RFID tags in the range of a reader can be inventoried as a *population*, at the speed of several milliseconds per tag.

This paper studies an important RFID problem of *monitoring a dynamic tag population*, which identifies the population changes, including both the *missing tags* that disappear within the readers' range and the *new tags* that are previously unknown and newly appear. Such a population change identification ability can bring solid benefits to many industries. For example, imagine a large warehouse with tens of thousands of stocks. Every night, the warehouse manager needs to check the inventories and find the missing items and the new items, because the existence of missing items probably indicates inventory thief or vendor fraud, and the presence of new items may expose management faults, e.g. unregistered stocks and misplaced stocks. However, manual counting is laborious. If each stock item is attached with a tag, then the RFID readers can know the stocks changes automatically.

In recent years, researchers also investigated various other RFID problems. Much prior work concentrates on *tag identification* problem which is to collect all the tag IDs in a tag population as quickly as possible [2], [3], [4], [5], [6], [7], [8], [9]. Another extensively studied topic is the *tag estimation* problem which is to give a rough estimate of the tag population size in a time-efficient manner [10], [11]. *Missing tag identification* problem also attracted much attention recently, which is to identify the IDs of missing tags in a tag population [12], [13].

Our tag population monitoring problem is different from the previous research topics. It may appear that, if we can collect all the tag IDs in the current population (i.e. the tag identification problem), then we will learn the population changes by comparing current population with the population in the previous scan. However, there can be a large number of tags that exist in both the previous scan and the current scan, called remaining tags. Recollecting the IDs of these remaining tags in the current scan is a waste of time, especially when each tag ID can be as long as 96 bits, according to EPC UHF RFID specification [3]. Can we detect the tag population change by the tag estimation problem? If we cannot detect any change in the number of tags, then we have no need to further identify those newly-arrived tags and missing tags. However, the tag number estimate is just a rough estimate with at least 10% error [10], [11]. If the population change only involves a few tags, we can hardly assert the change based on the estimated population size that is statistically variant. Finally, our tag population monitoring problem that identifies both missing tags and new tags is also different from the missing tag identification problem which determines only the IDs of missing tags.

The most related is a recent paper [14] which designed a protocol to detect both missing tags and new tags. However, the time efficiency of this protocol can be improved, because it uses only empty slots in the previous scan (or in the current scan) for the detection. To guarantee the final detec-

978-0-7695-4552-3/11 $26.00 © 2011 IEEE
DOI 10.1109/EUC.2011.30

tion accuracy, it must rely on the multiple-round execution which is time consuming. Moreover, this protocol uses two separated phases to detect missing tags and new tags. This is time-wasting since both phases need the remaining tags to respond, which is unnecessary and can be avoided.

We propose protocols that not only detect the tag population change events but also identify which tags are new tags or missing tags. The most important performance criterion is to minimize the *identification time* while meeting the *accuracy lower bound*. Otherwise, if the protocol execution takes too long, the normal operation (e.g. relocating goods in a warehouse) may alter the current population during the scan, and misleads the protocol to report wrong sets of missing tags and new tags, which would cause confusion to warehouse management.

We highlight the contributions of this paper to identifying new tags and missing tags. (1) We describe the protocol that identifies the current tag population as the *baseline protocol*, and we improve its time efficiency by 10%, by optimizing the length of the random number bits used for collision detection. (2) We propose a *one-phase protocol* that identifies only the population changes. This protocol is 70% more efficient than the baseline, when the remaining tag ratio is 0.5 or above. This protocol is 25% more accurate than state-of-the-art two-phase protocol in [14], by utilizing both empty slots and singleton slots in the previous scan (or in the current scan) to detect population changes. (3) We analyze the tradeoff between accuracy and efficiency. We show that, for our one-phase protocol, the increase of accuracy requirement will cause time efficiency decrease, i.e. higher time cost per identified new tag or missing tag. We also show that our one-phase protocol is 55% more efficient in execution time than the two-phase protocol in [14] at the same accuracy level.

The rest of this paper is organized as follows. We introduce some preliminary knowledge about RFID systems in Section II. We formulate our tag monitoring problem in Section III. For this problem, we present two monitoring protocols: the baseline protocol in Section IV and the one-phase protocol in Section V. We analyze the accuracy-efficiency tradeoff of the one-phase protocol in Section VI. We review the related work in Section VII and conclude our paper in Section VIII.

II. RFID BACKGROUND AND SYSTEM MODEL

In this section, we introduce the technical background in RFID systems. An RFID system consists of RFID tags and RFID readers. Each tag stores a unique ID in its memory, and each reader can read the IDs of its surrounding tags by wireless connections. We consider the scenario where there is only one reader covering all the tags. This is the most common scenario adopted by most existing RFID research [5], [6], [8], [9]. If multiple readers are used, we believe that it is easy to extend our protocols using existing reader scheduling protocols [15].

A. Framed Slotted ALOHA Protocol

One major challenge in the RFID research is the *collision* problem, i.e. when multiple tags hear the reader's query and respond, their responses will overlap and the reader may fail to decode the overlapped waveform. There exist a plethora of RFID anti-collision protocols, which can be classified into two major categories: tree-traversal algorithms [2], [7] and *framed slotted ALOHA* [3], [4], [5], [6], [8], [9], [16]. We adopt the latter because it has higher time-efficiency than the former in large RFID systems [3], [9]. The basic idea of the latter is to start an ALOHA frame with many time slots, and distribute the many tags uniformly to these slots to reduce the chance that multiple tags respond in a same slot. This *uniform tag distribution* is implemented by each tag selecting its own slot autonomously and pseudo-randomly, using a hash function $h_f(id, r)$, where id is the tag ID, f is the number of slots in the ALOHA frame, and r is a random seed. The parameters f and r are broadcast by the reader when starting the frame.

B. Slot States in Framed Slotted ALOHA

A slot in a frame has three possible states, according to the number of replying tags in this slot. If there are no tags replies, the slot is *empty* (noted as number 0). If there is one and only one tag reply, it is called a *singleton* slot (noted as 1). If there are at least two tag responses, it is a *collision* slot (noted as 2). Singleton slots can be further classified as *singleton-with-ID* slots and *singleton-without-ID* slots, according to whether tag ID transmission occurs. The singleton-without-ID slots is for the reader to know the presence of tags without ID collection. We denote the probability for a slot to be empty as P_0, the probability of being singleton as P_1, and the probability of being collision as P_2. Then we have $P_0 = e^{-\rho}$, $P_1 = \rho \cdot e^{-\rho}$, and $P_2 = 1 - P_0 - P_1$, because the number of replying tag in a slot follows *Poisson distribution* whose expected number of occurrences equals the frame's load factor ρ, i.e. the number of tags divided by the number of slots.

III. TAG POPULATION MONITORING PROBLEM

In this section, we formulate our problem precisely. A tag population is inevitably dynamic because tags can move in or out of the reader's range for various reasons. Such a dynamic tag population can be modelled as a dynamic process $[T_0, \ \ldots, \ T_t, \ T_{t+1}, \ \ldots \ ]$, where T_t is the tag population at discrete time t. A tag population monitoring protocol is, with the knowledge of previous tag population T_t, to derive an estimate of the current tag population T_{t+1}. Each tag population is represented by a set of tag IDs.

If comparing the previous population and the current population, we have three kinds of tags, as depicted in Fig. 1.

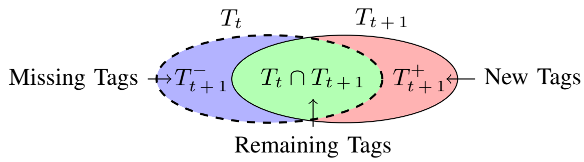

Fig. 1. Missing Tags, Remaining Tags, and New Tags.

- *Missing Tags*: the tags that were found in previous population but no longer exist in current population, which are noted as $T^-_{t+1} = T_t - T_{t+1}$.
- *Remaining Tags*: the tags that exists in both the previous population and the current population, i.e. $T_t \cap T_{t+1}$.
- *New Tags*: the tags that are unknown in previous population but appear in current population, which are noted as $T^+_{t+1} = T_{t+1} - T_t$.

We introduce a vector $[\beta^-, \beta, \beta^+]$ to formalize the relation between the previous population T_t and the current population T_{t+1}.

- β^- is the *missing tag ratio* that equals $\frac{|T^-_{t+1}|}{|T_t \cup T_{t+1}|}$,
- β is the *remaining tag ratio* that equals $\frac{|T_t \cap T_{t+1}|}{|T_t \cup T_{t+1}|}$,
- β^+ is the *new tag ratio* that equals $\frac{|T^+_{t+1}|}{|T_t \cup T_{t+1}|}$.

Of course, the sum of the three ratios is equal to one. For example, vector $[0.0, 0.5, 0.5]$ means no missing tags, 50% remaining tags, 50% new tags. Vector $[0.5, 0.5, 0.0]$ means 50% missing tags, 50% remaining tags, no new tags.

We consider two performance metrics for evaluating tag monitoring protocols: *accuracy* and *time efficiency*.

1) Tag monitoring accuracy α is the degree of similarity between the current population T_{t+1} and the generated estimate $\hat{T}_{t+1}$. We define $\alpha = \frac{|T_{t+1} \cap \hat{T}_{t+1}|}{|T_{t+1} \cup \hat{T}_{t+1}|}$. The maximum value of the accuracy α is equal to one, when the estimate $\hat{T}_{t+1}$ is absolutely accurate and identical to T_{t+1}. A key concern of this paper is to guarantee the estimation accuracy α to be above a threshold α_0.

2) Time Efficiency γ is defined as the total execution time t_{total} divided by the size of identified population changes, i.e. $\gamma = \frac{t_{\text{total}}}{|T_t - \hat{T}_{t+1}| + |\hat{T}_{t+1} - T_t|}$, where $|T_t - \hat{T}_{t+1}|$ is the number of identified missing tags and $|\hat{T}_{t+1} - T_t|$ is the number of identified new tags. We do not define γ as the total time cost divided by the size of current population, because we believe that the users of our RFID systems are mainly interested in the changes rather than the current population.

IV. BASELINE PROTOCOL

The most straightforward protocol to solve this tag monitoring problem is to collect all the IDs in the current population, and then compare the collected IDs with the memory of previous population to identify the changes. The accuracy of this baseline protocol can be close to one, since the traditional tag identification methods usually support the multiple-round execution, in which the next round can collect the tag IDs that fails to collect in the previous round. The time efficiency of this baseline protocol is analyzed by the following subsections.

A. Time Cost for Different Slot States

Much previous analysis of protocol time efficiency assumes the time cost of all slots is the same. But in practice the time cost of the four slot states (i.e. empty, collision, singleton-without-ID and singleton-with-ID) is different. We list the four time cost in the following table, where υ is the length of the random number (RAND for short) sent by tags to facilitate the detection of collisions at the reader side. When two tags send different RANDs (whose possibility is $1 - 2^{-\upsilon}$), the reader can detect the collision at waveform level. Note that the listed time cost assumes EPC RFID protocol [3], and include both data transmission time[1] and waiting time between transmissions[2].

	Definition	**Value**
t_{e}	time cost of an empty slot	$184\ \mu\text{s}$
t_{s}	time cost of a singleton-without-ID slot	$184 + 16\upsilon\ \mu\text{s}$
t_{ID}	time cost of a singleton-with-ID slot	$2128 + 32\upsilon\ \mu\text{s}$
t_{c}	time cost of a collision slot, i.e.	$(1 - 2^{-\upsilon})\, t_{\text{s}} + 2^{-\upsilon}\, t_{\text{ID}}$

Time cost t_{e} of an empty slot is the smallest. Time cost t_{s} of a singleton-without-ID slot almost doubles t_{e}. Time cost of a singleton-with-ID slot t_{ID} is at least ten times larger than t_{e}, because a tag ID is as long as 96-bit EPC plus 16-bit CRC and each bit requires 16µs to transmit[1]. Therefore, reducing the number of ID transmissions is a key concern for many RFID protocols. The calculation of time cost of a collision slot is complicated, i.e. $(1 - 2^{-\upsilon})\, t_{\text{s}} + 2^{-\upsilon}\, t_{\text{ID}}$. Here, $1 - 2^{-\upsilon}$ is the probability for the reader to detect the collision when receiving RANDs. If two tags send identical RANDs with $2^{-\upsilon}$ probability, the collision can be detected by the reader only when the two tags send out their IDs, whose time cost is t_{ID}.

B. Protocol Time Efficiency

1) Tag Identification Efficiency: For a tag identification protocol, its time efficiency is usually defined as the time cost per collected tag ID, i.e. $\frac{t_{\text{e}} P_0 + t_{\text{ID}} P_1 + t_{\text{c}} P_2}{P_1}$. This is because for any slot i, it has P_0 chance to be empty whose time cost is t_{e}, P_1 chance to be singleton-with-ID whose time cost is t_{ID}, and P_2 chance to be collision whose time cost is t_{c}. This time efficiency can be rewritten as $t_{\text{ID}} + \rho^{-1} t_{\text{e}} + \big(e^{\rho} - (1 + \rho)\big) t_{\text{c}}$.

We plot in Fig. 2 the tag identification efficiency against load factor ρ. Fig. 2 shows that when the load factor of the frame is between 0.6 and 0.8, the time efficiency reaches its peak about 3000µs if the RAND length υ is 16. We argue that, although Fig. 2 uses the slot time cost parameters $t_{\text{e}}, t_{\text{ID}}, t_{\text{c}}$ of EPC RFID system in Section IV-A,

[1]Note: RT_{rate}=64kbps, TR_{rate}=62.5kbps, QueryRep=4bits, Ack=2+υbits.
[2]Assume $\text{RT}_{\text{cal}} = 31.25$µs, $\text{T}_{\text{pri}} = 8$µs, $\text{T}_1 = 80$µs, $\text{T}_2 = \text{T}_3 = 40$µs.

our conclusions can be easily adapted to other RFID systems by adjusting these parameters accordingly.

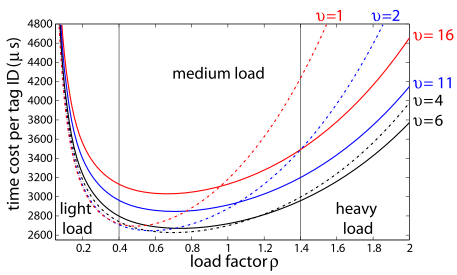

Fig. 2. Time Efficiency of Tag Identification Protocol against Load Factor.

Beside the appropriate choice of load factor, another interesting issue is the optimal configuration of RAND length υ. It is true that it is not a specification compliant feature that the RAND length is adjustable. For example, EPC UHF RFID protocol defines RAND to have 16 bits [3], and Philips I-Code system defines RAND to be 10 bits [4]. We investigate what is the optimal RAND length for tag identification efficiency.

Fig. 2 shows that the optimal RAND length is 6. If reducing the RAND length from 16 to 6, the time efficiency can be improved from 3000μs to 2700μs (i.e. 10% improvement). We regard 6 as the optimal length because a value smaller than 6 will lead to worse performance in heavy load region where the load factor is larger than 1.4 (i.e. the $\upsilon = 4$ curve is higher than the $\upsilon = 6$ curve in the heavy load region). The explanation is that, when RAND length is 6, collisions can be detected with $1 - 2^{-6} = 98.44\%$ probability. If RAND length is reduced to 4, the detection probability will drop to $1 - 2^{-4} = 93.75\%$. Note that an undetected collision will be followed by a time-consuming tag ID transmission. Such a trend of performance degradation in heavy load region is even more prominent when RAND length is reduced to 1 or 2. Therefore, we configure RAND length to 6 to optimize the performance in medium load regions and heavy load regions.

2) Tag Monitoring Efficiency: Different from the tag identification efficiency, tag monitoring efficiency is the total time cost $2700\mu s \cdot |T_{t+1}|$ divided by the size of population changes, i.e. $|T_t - T_{t+1}| + |T_{t+1} - T_t|$. This tag monitoring efficiency can be rewritten as

$$\gamma_{\text{baseline}} = \frac{|T_t \cap T_{t+1}| + |T_{t+1} - T_t|}{|T_t - T_{t+1}| + |T_{t+1} - T_t|} 2700\mu s = \frac{\beta + \beta^+}{1 - \beta} 2700\mu s.$$

As a summary, the monitoring efficiency of the baseline protocol degrades with the increase of remaining tag ratio β which means the baseline protocol will waste more time in collecting the already-known remaining tag IDs.

V. ONE-PHASE PROTOCOL

We observe that the major drawback of baseline protocol is the re-collection of remaining tag IDs which are known in the previous population. To avoid such redundant collection and improve time efficiency, we adopt an *incremental update* strategy that identifies only the IDs of new tags and missing tags. We assume the set of identified new tag IDs is T_{t+1}^+ and the set of identified missing tag IDs is T_{t+1}^-. Then we can generate an estimate of the current population by the equation: $\hat{T}_{t+1} = (T_t - T_{t+1}^-) \cup T_{t+1}^+$. Therefore, the pivot is shifted to the efficient collection of new tag IDs and the efficient identification of missing tags whose IDs are already known. We present our solution in this section.

A. Detection of New Tags and Missing Tags

We propose a one-phase protocol that detects the presence of new tags or missing tags in one frame (note: the new tag ID collection will be described in Section V-C). This "one frame" is depicted in Fig. 3 as the "true frame" which invites the participation of all tags in the current population T_{t+1}. Besides this true frame, we construct a so-called "expected frame" that involves the tags in previous population T_t. This expected frame is generated by pure computation without any wireless transmission involved, and it uses the same hash function as the true frame.

- A remaining tag responds in both the true frame and the expected frame and at the same slot, e.g. tags 3-7.
- A missing tag replies only in expected frame, e.g. tags 0-2.
- A new tag responds only in the true frame, e.g. tags 8-10.

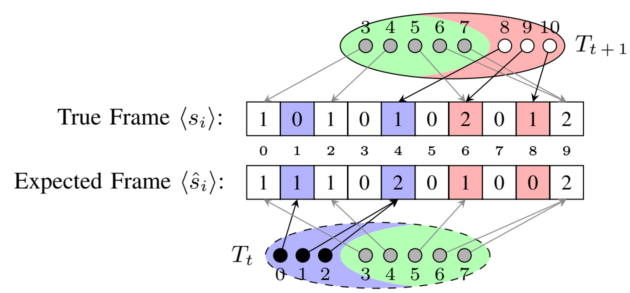

Fig. 3. One-Phase Protocol to Detect Population Changes by One Frame.

We detect the presence of new tags and missing tags by scanning the two frames and comparing the slot states. For the slot i, we denote its state in the true frame by s_i and denote its state in the expected frame by $\hat{s}_i$. If the slot state changes with $s_i \neq \hat{s}_i$, it indicates the presence of either new tags or missing tags, or even both. This is because when a slot contains only remaining tags which are mapped to both frames, its state will have no change. If there is no state change with $s_i = \hat{s}_i$, most probably this slot contains only remaining tags, in which we have no interests. But it is also possible that (1) this slot contains equal number of missing tags and new tags, which will escape from our detection, or (2) this slot contains at least two remaining tags, which will shield our detection. We will analyze the impacts of these two abnormal cases on detection accuracy later.

We list in the following table all the possibilities for a slot whose state changes with $s_i \neq \hat{s}_i$. For the first two cases, the identification of new tags and missing tags is easy, because the tags in such slots are either all new tags or all missing tags.

- If the true state s_i is nonempty and the expected state $\hat{s}_i$ is empty, all the tags in this slot are new tags, since there are no remaining tags as indicated by $\hat{s}_i = 0$ (e.g. slot 8).
- If the true state s_i is empty and the expected state $\hat{s}_i$ is nonempty, all the tags in this slot are missing tags, since there are no remaining tags indicated by $s_i = 0$ (e.g. slot1).

$\mathbf{s_i}$	$\mathbf{\hat{s}_i}$	**Detection Notes**
1^+	0	all the tags that respond in slot i are new tags
0	1^+	all the tags that should respond in slot i are missing tags
2	1	some tags that respond in slot i are new tags; also possibly the old tag is missing and all tags that respond are new tags
1	2	some tags that should respond disappear; also possibly all tag that should respond disappear and one new tag comes

NOTE: 0 is empty, 1 is singleton, 2 is collision, and 1^+ is non-empty.

For the other two cases, the identification is difficult since the tags in such slots are a mixture of missing tags, remaining tags and new tags, which needs to differentiate. We solve this differentiation problem by a technique called population change recalculation which is detailed in Section V-C.

- If the true state s_i is collision and the expected state $\hat{s}_i$ is singleton, the slot contains a remaining tag and a new tag (e.g. slot 6 which has remaining tag 5 and new tag 9).
- If the true state s_i is singleton and the expected state $\hat{s}_i$ is collision, the slot may contain a remaining tag and a missing tag, or even contain a new tag and two missing tags (e.g. slot 4 with new tag 8 and missing tags 1,2).

B. Accuracy Analysis

We analyze the accuracy that can be achieved by utilizing all the four kinds of changed slots. A recent paper that also studies new tag and missing tag detection uses only the first two cases whose tag identification is easy [14]. We will highlight the improvement we made in accuracy (i.e. about 25%) by also using the other two cases.

We assume that $\hat{\rho}$ is load factor of the expected frame which equals $|T_t|/f$. Thus the probability of empty slots (or singleton slots) in the expected frame is $\hat{P}_0 = e^{-\hat{\rho}}$ (or $\hat{P}_1 = \hat{\rho}\, e^{-\hat{\rho}}$). Similar parameters can be assumed for the true frame: load factor $\rho = |T_{t+1}|/f$, empty slot probability $P_0 = e^{-\rho}$, and singleton slot probability $P_1 = \rho\, e^{-\rho}$.

Theorem 1. *The accuracy of our one-phase protocol is*

$$E(\alpha) \approx \frac{\beta + (\hat{P}_0 + \hat{P}_1)\cdot\beta^+}{1 - (P_0 + P_1)\cdot\beta^-} = \frac{\beta + (1+\hat{\rho})\, e^{-\hat{\rho}}\cdot\beta^+}{1 - (1+\rho)\, e^{-\rho}\cdot\beta^-}, \quad (1)$$

where β^- is missing tag ratio, β is remaining tag ratio and β^+ is new tag ratio. In contrast, the accuracy of the two-phase protocol in [14] is only $E(\alpha) = \frac{\beta + e^{-\hat{\rho}}\cdot\beta^+}{1 - e^{-\rho}\cdot\beta^-}$.

Proof: Consider an arbitrary new tag in the set $T_{t+1} - T_t$, it has $\hat{P}_0 + \hat{P}_1$ probability to be mapped to a non-collision slot in the expected frame (i.e. $\hat{s}_i = 0$ or 1), where it can change the slot state and get identified. An arbitrary missing tags in the set $T_t - T_{t+1}$ has $P_0 + P_1$ probability to be mapped to a non-collision slots in the true frame (i.e. $s_i = 0$ or 1) and get identified. Thus, the expected number of identified new tags is $(\hat{P}_0 + \hat{P}_1)\cdot|T_{t+1} - T_t|$, and the expected number of identified missing tag is $(P_0 + P_1)\cdot|T_t - T_{t+1}|$. The accuracy of one-phase protocol can be estimated as $E(\alpha) \approx \frac{|T_t \cap T_{t+1}| + (\hat{P}_0+\hat{P}_1)\cdot|T_{t+1}-T_t|}{|T_t \cup T_{t+1}| - (P_0+P_1)\cdot|T_t - T_{t+1}|}$, where the nominator is the number of remaining tags $|T_t \cap T_{t+1}|$ plus the number of identified new tags, and the denominator is the union population size $|T_t \cup T_{t+1}|$ with the identified missing tag removed. This accuracy can be further rewritten to the form in Theorem 1. ■

The accuracy equation in Theorem 1 appears to be a function with two variables: load factor ρ of the true frame, and load factor $\hat{\rho}$ of the expected frame. In fact, it is a function with only one variable, i.e. the union load factor $\rho^{\cup} = \frac{|T_t \cup T_{t+1}|}{f}$. This is because for the true frame load factor ρ, we have

$$\rho = \frac{|T_{t+1}|}{f} = \frac{|T_t \cup T_{t+1}|}{f}(\beta + \beta^+) = \rho^{\cup}(\beta + \beta^+),$$

and for the expected frame load factor $\hat{\rho}$, we have

$$\hat{\rho} = \frac{|T_t|}{f} = \frac{|T_t \cup T_{t+1}|}{f}(\beta^- + \beta) = \rho^{\cup}(\beta^- + \beta),$$

where missing tag ratio β^-, remaining tag ratio β and new tag ratio β^+ are all constants. The physical meaning of the union load factor $\rho^{\cup}$ is that $e^{-\rho^{\cup}}$ is the expected ratio of slots that are empty both in true frame and in expected frame.

We plot in Fig. 4 both the accuracy of our one-phase protocol and the accuracy of the two-phase protocol in [14]. Fig. 4 adopts three typical scenarios with different combinations of $[\beta^-, \beta, \beta^+]$. The remaining tags ratio β is fixed to 50%. Vector $[0.25, 0.5, 0.25]$ means 25% missing tags and 25% new tags. Vector $[0.0, 0.5, 0.5]$ means no missing tags and 50% new tags. Vector $[0.5, 0.5, 0.0]$ means 50% missing tags and no new tags.

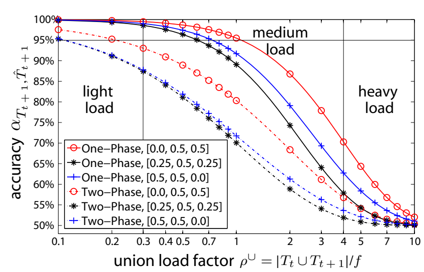

Fig. 4. Our One-Phase Protocol vs. Two-Phase Protocol [14] in Accuracy.

Figure 4 shows that the accuracy of our one-phase protocol is roughly 25% better than the accuracy of two-phase protocol in [14]. For example, in the $[0.25, 0.5, 0.25]$ scenario, when the union load factor $\rho^{\cup}$ equals 1, the accuracy of the two-phase protocol is 70%, while the accuracy of our one-phase protocol is 89%, which means 27.14%

improvement. Fig. 4 also shows that the accuracy of both protocols is poor and lower than 0.7 in the heavy load region with union load factor $\rho^{\cup}$ above 4. This is because when $\rho^{\cup}$ is above 4 and remaining tag ratio β is 0.5, the density of remaining tags is larger than 2 remaining tags per slot. In slots with 2 or more remaining tags, we can not detect the presence of new tags and missing tags. However, when the union load factor is lower than 0.3, the accuracy of our one-phase protocol can be higher than 95%, which can satisfy the needs of many RFID applications.

C. Identification of New Tags and Missing Tags

We present our one-phase protocol in Protocol 1. This protocol provides high accuracy by utilizing all the four kinds of slot state changes, and it addresses the problem of differentiating missing tags and new tags when all kinds tags mixed in one slot. The input of the protocol is the prior knowledge of previous population T_t, and the output is an estimate of the current population $\hat{T}_{t+1}$. Note that when the prior knowledge T_t is an empty set, our protocol will degrade to the baseline protocol that neglects the prior knowledge.

Protocol 1: One-Phase Population Monitoring Protocol

input : the prior knowledge of tag population T_t at time t
output : estimate $\hat{T}_{t+1}$ of current tag population at time $t+1$

1 Reader generates frame size f and random seed r
2 Reader obtains state belief $\hat{s}_i$ of each slot, by T_t and $h_f(id, r)$
3 Reader resets the session flags of all tags in T_{t+1} to 0
4 Reader starts a frame by broadcasting f and r to all tags
5 **for** *slot* $i \leftarrow 0$ **to** $f-1$ **do**
6   **if** *a tag selects slot* i *by* $h_f(id, r)$ **then** it replies RAND
7   Reader obtains the state s_i of slot i when receiving RAND
8   **if** $s_i = \hat{s}_i$ **then** Reader closes slot i by a special QueryRep that forces tags in slot i to invert session flags
9   **else**
10   Reader updates the missing tag set by $T^-_{t+1} := T^-_{t+1} \cup \{id \in T_t \mid \text{tag } id \text{ should respond in slot } i\}$
11   **if** *slot* i *is singleton with* $s_i = 1$ **then**
12   Reader sends ACK, and the tag replies its id
13   Reader updates new tag set by adding id to T^+_{t+1}
14   Reader closes slot i by QueryRep (note: the single tag in slot i automatically inverts its session flag [3])
15 Reader uses a new frame or new frames to collect IDs of tags whose session flags are 0, and adds the collected IDs to T^+_{t+1}
16 Reader re-identifies the missing tags by $T^-_{t+1} := T^-_{t+1} - T^+_{t+1}$
17 Reader re-identifies the new tags by $T^+_{t+1} := T^+_{t+1} - T_t$
18 **return** tag population estimate $\hat{T}_{t+1} := (T_t - T^-_{t+1}) \cup T^+_{t+1}$

Firstly, the reader detects population changes. The reader starts a frame by broadcasting frame size f and random seed r to the current population T_{t+1} (see Ln. 4). The tags use hash function $h_f(id, r)$ to choose their slots in which they respond with RAND to show their presence (see Ln. 6). The reader, after receiving the RANDs, can obtain the state s_i of each slot i (see Ln. 7). The reader can also establish a prior belief $\hat{s}_i$ about slot i's state (see Ln. 2), using the knowledge of the previous tag population T_t and the hash function $h_f(id, r)$. Then the reader compares the true state s_i with the state belief $\hat{s}_i$ (see Ln. 8). If no change can be found, the reader closes the current slot i instantly by the QueryRep command which is defined in [3]; otherwise, this slot i is detected as a changed slot which contains new tags or missing tags.

Secondly, the reader further identifies new tags and missing tags in the changed slot i by the following three steps.

Step 1 (*Missing Tag Identification*). The reader adds all the tags in population T_t that should respond in slot i to the missing tag set T^-_{t+1} (see Ln. 10). For example, in Fig. 3, the slots $\{0, 4, 6, 8\}$ are changed slots, and the old tags $\{0, 1, 2, 5\}$ that should respond in these changed slots are marked as missing tags. It is possible that this set T^-_{t+1} may contain remaining tags, e.g. tags 5 that should respond in changed slot 6 in Fig. 3. Such remaining tags that are wrongly marked as missing tags won't ruin our final tag population estimate $\hat{T}_{t+1}$, because their responses will be heard by the reader in step 2 and be re-identified as new tags.

Step 2 (*New Tag Identification*). The reader will add the IDs of all the tags that responded by RAND and showed their presence in slot i to the new tag set T^+_{t+1}. For example, in Fig. 3, tags $\{5, 8, 9, 10\}$ that responded in the changed slots $\{0, 4, 6, 8\}$ are regarded as new tags. It is possible that a tag in the set T^+_{t+1} is in fact a remaining tag, e.g. tags 5. But this won't ruin our final population estimate $\hat{T}_{t+1}$ at step 3.

Different from the missing tags whose IDs are contained in T_t, the IDs of new tags are unknown and need to be collected. The reader will use two methods to collect new tag IDs. Firstly, if a changed slot i is singleton in the true frame (e.g. slots $4, 8$), then the reader collects the single tag ID in slot i directly (see Ln. 11-13). This because as defined in [3], the reader can send an ACK command to notify the tag to propagate back its ID. Secondly, if a changed slot i is collision in the true frame, then the multiple tag IDs cannot be collected in the current slot and should be delayed to a new frame (see Ln. 15). The sole purpose of this new frame is to collect IDs of the new tags mapped to the changed slots that are collision in true frame. But the question is how can we let these tags know they should participate in this new frame and let other tags know they shouldn't. The answer is to use the session flag feature[3] defined in [3]. Initially, the session flags of all tags are zero (see Ln. 3). Then, the flags of the tags in unchanged slots will be be forced to invert (see Ln. 8). The flags of the tags in singleton changed slots will invert automatically (see Ln. 14) according to [3]. Thus only the tags in collision changed slots do not invert their flags and will participate the ID collection frame(s) at Ln. 15.

[3]By section 6.3.2.2 of [3], each tag has four session flags S0-S3. The frame start command Query contains two parameters: a session flag id and a desired value (e.g. session flag S2 and value 0). A tag will participate in this frame only if the tag's corresponding session flag matches the desired value.

Step 3 (*Population Change Recalculation*). To remove the remaining tags wrongly contained in the missing tag set T^-_{t+1}, the reader recalculates the set of missing tags by $T^-_{t+1} = T^-_{t+1} - T^+_{t+1}$ (see Ln. 16). To remove the old tags wrongly contained in the new tag set T^+_{t+1}, the reader recalculates the set of new tags by $T^+_{t+1} = T^+_{t+1} - T_t$ (see Ln. 17). Finally, with the recalculated missing tag set and new tag set, we give the population estimate $\hat{T}_{t+1}$ at Ln. 18.

VI. TIME-EFFICIENCY TRADEOFF

This section analyzes the tradeoff between monitoring accuracy and efficiency of the proposed one-phase protocol.

A. Time Efficiency Analysis

We analyze the time efficiency of our one-phase protocol, which is another performance metric besides accuracy. We calculate the time efficiency as $\gamma_{\text{one-phase}} = \frac{t_{\text{slot}}}{n^-_{\text{slot}} + n^+_{\text{slot}}}$, where t_{slot} is the expected time cost of a slot, n^-_{slot} is the expected number of missing tags that can be identified in a slot, and n^+_{slot} is the expected number of new tags that can be identified in a slot. This efficiency $\gamma_{\text{one-phase}}$ is a function of union load factor $\rho^{\cup}$, which is plotted in Fig. 5. This figure adopts three scenarios: $[0.0, 0.5, 0.5]$, $[0.25, 0.5, 0.25]$, $[0.5, 0.5, 0.0]$.

We omit the expression to calculate t_{slot} which is complicated. This is because the true frame has five kinds slots with different time cost: *empty* slots whose cost is t_{e}, *singleton unchanged* slots whose cost is t_{s}, *singleton changed* slots whose cost is t_{ID}, *collision unchanged* slots whose cost is t_{c}, and *collision changed* slots whose cost is about $2700\mu\text{s}$ per tag (see Section IV-A for time cost definitions). We need to calculate their corresponding probabilities, and combine them linearly to obtain t_{slot}. The equations for n^-_{slot} and n^+_{slot} are as follows:

$$n^-_{\text{slot}} = \rho^-(P^{\cap}_0 + P^{\cap}_1)P^+_0 + \rho^-(1 - e^{-\rho^-})P^{\cap}_0 P^+_1 + P^-_1 P^{\cap}_0 P^+_2\,,$$
$$n^+_{\text{slot}} = P^-_0(P^{\cap}_0 + P^{\cap}_1)\rho^+ + P^-_2 P^{\cap}_0 P^+_1 + P^-_1 P^{\cap}_0 \rho^+(1 - e^{-\rho^+})\,,$$

where P^-_0, P^-_1, P^-_2 are the probabilities of 0, 1, ≥ 2 missing tags in a slot respectively; $P^{\cap}_0, P^{\cap}_1, P^{\cap}_2$ are the probabilities of 0, 1, ≥ 2 remaining tags in a slot respectively; P^+_0, P^+_1, P^+_2 are the probabilities of 0, 1, ≥ 2 new tags in a slot.

Figure 5 shows that the time efficiency of our protocol is prominently higher than that of the baseline protocol in the medium load region. For example, in the $[0.5, 0.5, 0.0]$ scenario, the efficiency of baseline protocol is $2700\mu\text{s}$ per changed tag (i.e. $\gamma_{\text{baseline}} = 2700\mu\text{s} \cdot \frac{\beta + \beta^+}{1 - \beta}$), while the best efficiency of our one-phase protocol is roughly $1000\mu\text{s}$, i.e. 70% reduction in time cost. In the $[0.0, 0.5, 0.5]$ scenario, the efficiency of baseline protocol is $5400\mu\text{s}$, while the best time efficiency of our protocol is about $4000\mu\text{s}$, i.e. 26% improvement. Our one-phase protocol has much better performance in $[0.5, 0.5, 0.0]$ scenario than in $[0.0, 0.5, 0.5]$ scenario, because $[0.5, 0.5, 0.0]$ scenario has only missing tags whose identification is does not need to transmit tag IDs.

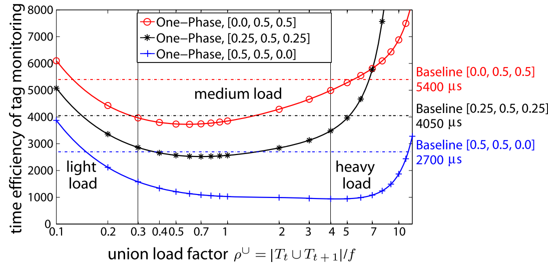

Fig. 5. One-Phase Protocol vs. Baseline Protocol in Monitoring Efficiency.

Figure 5 also shows that the time efficiency of one-phase protocol degrades rapidly in the high-load region with $\rho^{\cup} > 4$. This is because in the high-load region, the expected number of remaining tags in a slot $\rho^{\cap}$ is larger than 2 (note: $\rho^{\cap} = \beta \cdot \rho^{\cup}$). A slot with at least two remaining tags will have its states in the true frame and in the expected frame to be both collision, which can hide the presence of new tags and missing tags. In contrast, in light-load region with few such slots, our protocol only degrades mildly in time efficiency.

B. Accuracy-Efficiency Tradeoff

The one-phase protocol needs to keep a balance between accuracy and time efficiency. Although the accuracy can be improved by reducing union load factor (see Fig. 4), this meanwhile will degrade the time efficiency when entering the light-load region (see Fig. 5). Therefore, we analyze the functional relation between accuracy and efficiency, and plot our analysis results in Fig. 6. It adopts three scenarios with different remaining tag ratio β: the scenario $[0.1, 0.8, 0.1]$ with $\beta = 0.8$, the scenario $[0.25, 0.5, 0.25]$ with $\beta = 0.5$, and the scenario $[0.4, 0.2, 0.4]$ with $\beta = 0.2$. Fig. 6 does not show the portion with accuracy lower than 90% since we believe that RFID users have no interests in such poor accuracy.

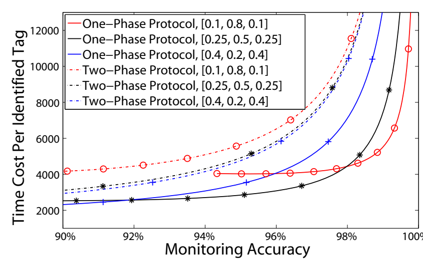

Fig. 6. Tag Monitoring Efficiency γ vs. Accuracy Requirement α_0.

As we have expected, Fig. 6 shows that the time cost of our one-phase protocol increases as the required accuracy goes higher. If the required accuracy exceeds 98%, the time cost inflates nearly straight up with the increase of accuracy. The explanation is that our one-phase protocol is based on a *randomized algorithm* that detects missing tags and new tags by randomly distributing them to slots where they can be detected. If ultra-high accuracy is needed, we may have to use extra-large frame size to increase the chance of detection.

However, we believe that the accuracy between 90% and 99% can already satisfy the need of most RFID applications.

Finally, we show the advantage of our one-phase protocol over the two-phase protocol [14] in Fig. 6. It shows that at the same accuracy level (e.g. 96%), our one-phase protocol can be 55.43% more efficient than the two-phase protocol in $[0.25, 0.5, 0.25]$ scenario. This is because the two-phase protocol has lower accuracy than our one-phase protocol. If the two-phase protocol needs to achieve the same accuracy level as ours, it must use much larger frame size (or even multiple round execution) to create more empty slots for effective detection of new tags and missing tags, which however reduces protocol efficiency.

VII. RELATED WORK

RFID technology has been considered in many applications. The most traditional application is the *tag identification* problem which collects all tag IDs in a population. The proposed solutions can be classified into two major categories: tree-based [2], [7] and ALOHA-based [3], [4], [5], [6], [8], [9]. The former organizes all tag IDs in a tree of ID prefixes, while the latter distributes all tag IDs uniformly in an ALOHA frame. The major difficulty of ALOHA-based protocols is how to choose the optimal frame size which should be roughly equal to the number of tags. Therefore, *tag population size estimation* problem becomes another hot topic for RFID research [10], [11]. Another important problem that attracts academic interests is, given the prior knowledge of the tag population, to *identify the missing tags* [12], [13].

However, in practice, besides missing tags, there may also exist new tags whose IDs are unknown. The problem of identifying both of these tags is called *tag population monitoring* problem, since we can establish an estimate of the current tag population, with the knowledge of previous tag population, new tags and missing tags. A relevant study on this problem is a recent paper [14] which can detect a new tag when it is mapped to an empty slot in the expected frame, and detect a missing tag when it is mapped to an empty slot in the true frame. However, the accuracy of this protocol can be improved by at least 25%, if we can also make use of the massive singleton slots in the expected frame or true frame to detect of missing tags and new tags. Moreover, the two-phase protocol is inefficient, because it uses two separated phases to detect missing tags and new tags. The remaining tags thus need to respond in both two phases, which waste precious execution time.

VIII. CONCLUSION

This paper focused on the problem of monitoring a dynamic tag population. We proposed a one-phase solution which is efficient by using only one frame to detect both missing tags and new tags. This solution is also accurate because it uses both empty slots and singleton slots in the true frames (or in the expected frames) to detect the population changes. Another contribution we made is that we derived an optimal configuration of RAND length for the traditional tag identification protocol, which is neglected before.

ACKNOWLEDGEMENT

The authors would like to thank the reviewers for comments. This work is supported by HK RGC PolyU 5314/10E.

REFERENCES

[1] M. Weiser, "The computer for the 21st century," in *Proc. of ACM SIGMOBILE*, vol. 3, 1999, pp. 3–11.

[2] "ISO/IEC 18000 information technology - RFID for item management - Part 6: Parameters for air interface communications at 860MHz-960MHz," 2004.

[3] "EPC™ radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860MHz-960MHz v1.2.0," 2008.

[4] P. Semiconductors, "I-CODE smart label RFID tags," 2004.

[5] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. of IEEE PERCOM*, 2002, pp. 98–113.

[6] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for multiple RFID objects identification," *IEICE Trans. on Communications*, 2005.

[7] J. Myung and W. Lee, "Adaptive splitting protocols for RFID tag collision arbitration," in *Proc. of ACM MOBIHOC*, 2006.

[8] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. of IEEE MOBIQUITOUS*, 2005, pp. 166–174.

[9] C. Qian, Y. Liu, H. Ngan, and L. M. Ni, "ASAP: Scalable identification and counting for contactless RFID systems," in *Proc. of IEEE ICDCS*, 2010, pp. 52–61.

[10] C. Qian, H. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," in *Proc. of IEEE PerCom*, 2008.

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

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

[13] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. of ACM MOBIHOC*, 2010.

[14] B. Sheng, Q. Li, and W. Mao, "Efficient continuous scanning in RFID systems," in *Proc. of IEEE INFOCOM*, 2010.

[15] S. Tang, X. Li, G. Chen, Y. Liu, and J. Zhao, "Raspberry: A stable reader activation scheduling protocol in multi-reader RFID systems," in *Proc. of IEEE ICNP*, 2009.

[16] K. Bu, B. Xiao, Q. Xiao, and S. Chen, "Efficient pinpointing of misplaced tags in large RFID systems," in *Proc. of IEEE SECON*, 2011.