CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2013; **25**:2080–2097
Published online 28 March 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.2835

SPECIAL ISSUE PAPER

Efficient protocol design for dynamic tag population monitoring in large-scale radio frequency identification systems ‡

Qingjun Xiao [1], Kai Bu [1], Bin Xiao [1,*,†] and Limin Sun [2]

[1]*Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China*
[2]*Institute of Software, Chinese Academy of Sciences, Beijing, China*

SUMMARY

As radio frequency identification (RFID) tags become more ubiquitously available, they will stay in dynamic environments where tags can freely enter or leave RFID readers' interrogation range. With such a dynamic tag population, there arises a problem of population monitoring, whose purpose is to identify the *missing tags* that have departed from the reading range and the *new tags* that have newly entered. This problem is a new problem which cannot be well solved by the conventional tag identification protocols. In this paper, we first show that this traditional approach is inefficient, because it collects all the tag IDs in each scan and ignores the ready-for-use knowledge of the tag population in a previous scan. To be more efficient, we present three protocols: (i) a baseline protocol that improves the traditional tag identification protocol by optimizing its length of random number used for collision detection; (ii) a novel one-phase protocol with easy labor to identify exactly the new tags and the missing tags by fully utilizing the knowledge of previous tag population; and (iii) a hybrid protocol that smartly combines the baseline protocol and the one-phase protocol. Its purpose is to deal with the situation that the knowledge of previous tag population is highly inconsistent with the current tag population. This hybrid protocol, as shown by our analysis, can improve the tag monitoring accuracy by 25%, and improve the time efficiency by 55.3%, as compared with a recent work (called two-phase protocol), which also identifies the population changes. Copyright © 2012 John Wiley & Sons, Ltd.

Received 21 December 2011; Revised 21 February 2012; Accepted 22 February 2012

KEY WORDS: ubiquitous computing; RFID; dynamic tag population; tag population monitoring

1. INTRODUCTION

About a decade ago, researchers envisioned radio frequency identification (RFID) as one of the enabling techniques for the future of ubiquitous computing [1]. Nowadays, RFID has arguably become one of the most successful technologies in the computing history, which has been applied to a wide range of applications, for example, warehouse management, logistic control, and traffic management. The success of RFID is largely due to its clear advantages over the classical barcode system. Barcode-attached products can be read only by moving them close to readers, but an RFID tag can be accessed wirelessly from a distance, that is, tens of feet for passive tags or even hundreds of feet for active tags. Moreover, a group of RFID tags can be inventoried as a *population* at the speed of several milliseconds per tag, whereas barcode can only be checked manually one by one.

As the wide spread of RFID tags, they will inevitably be deployed in dynamic environments, where there exist two kinds of tags: the *missing tags* that disappeared from the readers' range and

*Correspondence to: Bin Xiao, Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China.
†E-mail: csbxiao@comp.polyu.edu.hk
‡A shorter version of the paper was presented at the 9th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2011.

the *new tags* that are previously unknown and newly appear. Identifying these two kinds of tags can bring solid business benefits. For example, imagine a large warehouse with tens of thousands of stocks. Every night, the warehouse manager needs to check the inventories and locate the missing items and the new items. This is because (i) the discovery of missing items probably indicates inventory theft or vendor fraud, and (ii) the identified new items may help expose management faults, for example, unregistered stocks and misplaced stocks. However, manually counting the large inventory for the two kinds of tags is prohibitively laborious. Instead, if each stock is attached with an RFID tag, then the readers can detect and identify the stocks' changes automatically. In summary, we focus on the important RFID problem of monitoring a dynamic tag population, with the purpose of identifying the missing tags and the new tags.

In recent years, various other RFID problems have been investigated by researchers. The *tag identification* problem is perhaps the most extensively studied problem, which collects all the tag IDs in a tag population as quickly as possible [2–9]. Another well-known research 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]. The *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 these previous research topics. (i) It may appear that if we can collect all the tag IDs in the current population (i.e., the tag identification problem), then by comparing the current population with the population in the previous scan, we will easily learn the population changes. However, usually, there are a large number of tags that exist in both the previous scan and the current scan, called *remaining tags*. Recollecting the remaining tag IDs in the current scan is a waste of time, especially when each tag ID can be as long as 96 bits, according to EPC (Electronic Product Code) UHF (Ultra High Frequency) RFID specification [3]. (ii) Can we detect the tag population change by the tag estimation problem? Intuitively, if we can detect the change in the number of tags, then we can know the existence of newly arrived tags or 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 few tags, we can hardly assert the change based on the estimated population size that is statistically variant. (iii) Finally, our tag population monitoring problem is also different from the missing tag identification problem, which determines only the IDs of missing tags. In contrast, our problem identifies both missing tags and new tags.

The most related is a recent paper [14], which also identifies missing tags and new tags. However, its proposed protocol can be significantly improved in time efficiency for the following three reasons. (i) The protocol uses only empty slots in the previous scan (or in the current scan) for the detection. But we find that both empty slots and singleton slots can be used. (ii) The protocol uses two separated phases to detect missing tags and new tags. This is time-wasting because both phases need the remaining tags to respond, which is unnecessary and can be avoided. (iii) To guarantee the final detection accuracy, the protocol must rely on the multiple-round execution, which is time consuming. In fact, only one round can suffice the need for accuracy, if we can adjust the ALOHA frame size to a large enough value. More importantly, this protocol ignores an important factor—the *remaining tag ratio* (or the percentage of tags shared by two adjacent scans). When the remaining tag ratio is too small, the protocol that identifies population changes is no longer efficient, and the traditional protocol becomes more attractive, which collects all the IDs in current population.

We propose protocols to efficiently 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 our contributions to the efficient identification of new tags and missing tags.

(1) We describe the protocol that identifies current tag population as the *baseline protocol*. We improve its time efficiency by 10%, by optimizing the length of the transmitted random numbers used for collision detection.

(2) We propose an improved *one-phase protocol* that identifies only the population changes. This protocol is 70% more efficient than the baseline protocol, when the remaining tag ratio is 0.5 or higher. This protocol is 25% more accurate than state-of-the-art two-phase protocol proposed in [14], because we utilize both empty slots and singleton slots in the previous scan (or in the current scan) to detect population changes.
(3) We analyze the accuracy-efficiency trade-off of the one-phase protocol. We show that, for our one-phase protocol, the increase of accuracy requirement will cause the decrease of time efficiency (i.e., time cost per identified new tag or missing tag). This trade-off analysis can help us to achieve the minimum identification time that can just satisfy the accuracy requirement. We also show that, at the same accuracy level, our one-phase protocol can be 55% more efficient than the two-phase protocol proposed in [14].
(4) We propose a *hybrid protocol* that combines the baseline protocol and the one-phase protocol. This hybrid protocol is to address the problem that the one-phase protocol is low efficient, when there is only a small overlap between the previous population and the current population. In this case, the hybrid protocol will switch to the more efficient baseline protocol. Our analysis shows that our hybrid protocol can be 65.33% more efficient than the two-phase protocol [15] when the remaining tag ratio is as small as 0.2.

The rest of this paper is organized as follows. In Section 2, we introduce the necessary background knowledge of RFID systems. In 3, we formulate our tag monitoring problem. For this problem, we present three monitoring protocols: the baseline protocol in Section 4, the one-phase protocol in Section 5, and the hybrid protocol in Section 7. In Section 6, we analyze the accuracy-efficiency tradeoff of the one-phase protocol. The hybrid protocol in Section 7 combines the one-phase protocol and the baseline protocol, to improve the efficiency when the remaining tag ratio is small. Finally, we review the related work in Section 8 and conclude our paper in Section 9.

2. RADIO FREQUENCY IDENTIFICATION BACKGROUND AND SYSTEM MODEL

In this section, we introduce the necessary background knowledge of RFID systems. We focus on the canonical tag identification problem.

2.1. Framed slotted ALOHA protocol

We assume that an RFID system consists of a single reader and many tags, and this reader has a single antenna. This simplified scenario has been adopted by most existing RFID research [5, 6, 8, 9]. If the single reader has multiple antennas or multiple readers are used, it is easy to extend our protocols using existing reader and antenna scheduling algorithms [16].

The traditional tag identification problem is to collect the tag IDs within the reader's radio range. For this problem, a key challenge is the *collision*. That is, multiple tags may respond simultaneously and collide when they hear the reader's query. To solve this problem, a plethora of protocols can be found in literature, which can be classified into two major categories: *tree-traversal* algorithms [2, 7] and *framed slotted ALOHA* [3–6, 8, 9, 17]. The latter provides higher time efficiency than the former and thus is more pervasively used in large-scale RFID systems [3, 9].

We employ the framed slotted ALOHA scheme in this paper. Its basic idea is that the reader starts a frame with f consecutive time slots, and each tag picks a slot randomly based on a uniform probability distribution. This uniform slot selection can reduce the chance that a slot contains multiple tag responses. The tags that still collided will be scheduled to respond in the next frame.

In an ALOHA frame, the number of tag responses in a time slot follows *Poisson distribution*. If we denote that number as k, then its probability function is $P(k) = \frac{\rho^k}{k!} \cdot e^{-\rho}$, where ρ is the load factor. *Load factor* of a frame is defined as the ratio of the number n of tags to the number f of time slots, that is, $\rho = \frac{n}{f}$.

2.2. Slot states in ALOHA frames

For a slot in an ALOHA frame, there are basically three possible states according to the number of tags replying in this slot. If there are no tags replies, this slot is empty which is noted as number 0.

DOI: 10.1002/cpe

If there is one and only one tag reply, this slot is singleton noted as number 1. If there are at least two tag responses, this slot is collision noted as number 2.

We use the following probability notations throughout this paper:

- P_0 denotes the probability for a slot to be empty;
- P_1 denotes the probability for a slot to be singleton;
- P_2 denotes the probability for a slot to be collision.

For these three probabilities, 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 as we stated before.

We should also note that the singleton slots can be further classified into singleton-with-ID slots and singleton-without-ID slots. In a singleton-without-ID slot, a tag replies only with a random number to inform the reader about its presence; in a singleton-with-ID slot, a tag replies with both a random number and a tag ID according to EPC UHF RFID specification [3].

3. TAG POPULATION MONITORING PROBLEM

In this section, we formulate our problem precisely. A tag population is inevitably dynamic because the tags can enter or leave the reader's range for various reasons. Such a tag population can be modeled 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 monitoring protocol is to derive an estimate of the current population T_{t+1}, with the prior knowledge of the previous population T_t. Either 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 Figure 1.

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

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

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

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

For tag monitoring protocols, we consider two performance metrics: *accuracy* and *time efficiency*.

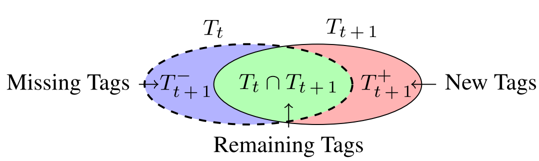

Figure 1. Missing tags, remaining tags, and new tags.

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, that is, $\gamma = \frac{t_{\text{total}}}{|T_t - \hat{T}_{t+1}| + |\hat{T}_{t+1} - T_t|}$, 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.

4. BASELINE PROTOCOL

For the tag monitoring problem, the most straightforward solution is to collect all the tag IDs in the current population and then compare the collected IDs with the memory of the previous population to identify the changes. The accuracy of this baseline protocol can be close to one, because 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.

4.1. Time cost for different slot states

Many previous works analyze time efficiency of tag ID leveraging ALOHA frames. However, these studies assume that the time cost of all slots is the same. In practice, the time costs of the four slot states (i.e., empty, collision, singleton-without-ID, and singleton-with-ID) are 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 [§] and waiting time between transmissions[¶].

	Definition	**Value**
t_e	Time cost of an empty slot	184 μs
t_s	Time cost of a singleton-without-ID slot	$184 + 16\,\upsilon$ μs
t_{ID}	Time cost of a singleton-with-ID slot	$2128 + 32\,\upsilon$ μs
t_c	Time cost of a collision slot, that is,	$(1 - 2^{-\upsilon})t_s + 2^{-\upsilon}t_{ID}$

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 [‖]. 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, that is, $(1 - 2^{-\upsilon})t_s + 2^{-\upsilon}t_{ID}$. 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}.

4.2. Protocol time efficiency

4.2.1. Tag ID efficiency For a tag ID protocol, its time efficiency is usually defined as the time cost per collected tag ID, that is, $\frac{t_e P_0 + t_{ID} P_1 + t_c P_2}{P_1}$. This is because for any slot i, it has P_0

[§]Note: $RT_{rate} = 64$kbps, $TR_{rate} = 62.5$kbps, QueryRep $= 4$bits, Ack $= 2 + \upsilon$bits.
[¶]Assume $RT_{cal} = 31.25$ μs, $T_{pri} = 8$ μs, $T_1 = 80$ μs, $T_2 = T_3 = 40$ μs.
[‖]Because a tag ID is as long as 96-bit EPC plus 16-bit CRC and each bit requires 16 μs to transmit [3]

DOI: 10.1002/cpe

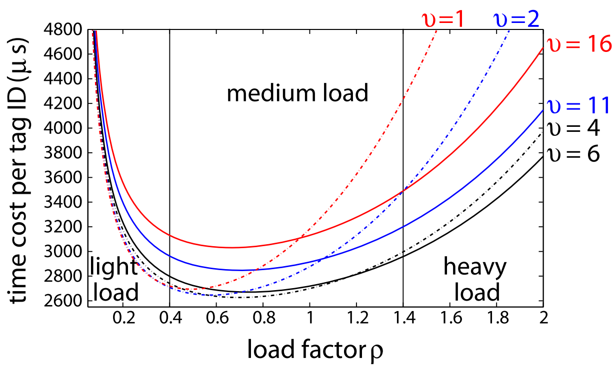

Figure 2. Time efficiency of tag identification protocol against load factor.

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 $t_{ID} + \rho^{-1} t_e + (e^{\rho} - (1+\rho))\, t_c$.

We plot in Figure 2 the tag ID efficiency against load factor ρ. Figure 2 shows that when the load factor of the frame is between 0.6 and 0.8, the time efficiency reaches its peak about 3000 μs if the RAND length υ is 16. We argue that although Figure 2 uses the slot time cost parameters t_e, t_{ID}, t_c of EPC RFID system in Section 4.1, our conclusions can be easily adapted to other RFID systems by adjusting these parameters accordingly.

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 ID efficiency.

Figure 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 3000 μs to 2700 μs (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.

4.2.2. Tag monitoring efficiency Different from the tag ID efficiency, tag-monitoring efficiency is the total time cost 2700 $\mu s \cdot |T_{t+1}|$ divided by the size of population changes, that is, $|T_t - T_{t+1}| + |T_{t+1} - T_t|$. This tag-monitoring efficiency can be rewritten as follows.

$$\gamma_{\text{baseline}}(\beta^-, \beta^+) = \frac{|T_t \cap T_{t+1}| + |T_{t+1} - T_t|}{|T_t - T_{t+1}| + |T_{t+1} - T_t|} \cdot 2700\ \mu\text{s} = \frac{1 - \beta^-}{\beta^- + \beta^+} \cdot 2700\ \mu\text{s} \tag{1}$$

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

5. ONE-PHASE PROTOCOL

We observe that the major drawback of baseline protocol is the recollection of the remaining tag IDs, which are known in the previous population. To avoid such redundant collection and improve

DOI: 10.1002/cpe

time efficiency, we adopt an *incremental update* strategy that identifies only the IDs of new tags and missing tags. We assume that 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.

5.1. 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 5.3). This 'one frame' is depicted in Figure 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 the 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, for example, tags 3–7.
- A missing tag replies only in expected frame, for example, tags 0–2.
- A new tag responds only in the true frame, for example, tags 8–10.

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 the remaining tags, in which we have no interests. But it is also possible that (i) this slot contains equal number of missing tags and new tags, which will escape from our detection, or (ii) 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 in Section 6.

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, because 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, because there are no remaining tags indicated by $s_i = 0$ (e.g., slot1).

For the other two cases, the identification is difficult because 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 5.3.

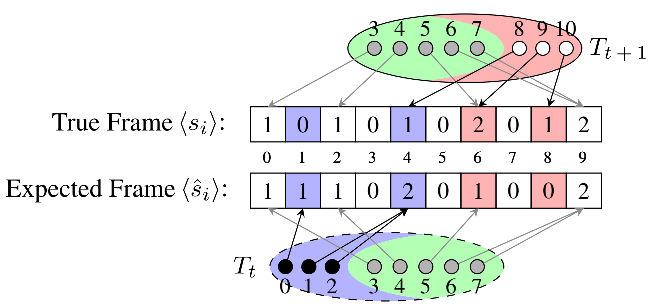

Figure 3. One-phase protocol to detect population changes by one frame.

DOI: 10.1002/cpe

- 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).

s_i	$\hat{s}_i$	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 nonempty.

- 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).

5.2. 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 ID 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 a load factor of the expected frame, which is equal to $|T_t|/f$. 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 (Expected accuracy of tag monitoring protocols)
The expected accuracy of our one-phase protocol can be calculated as

$$\alpha_{\text{one-phase}}(\rho^{\cup}, \beta^-, \beta^+) \approx \frac{\beta + (\hat{P}_0 + \hat{P}_1)\cdot\beta^+}{1-(P_0+P_1)\cdot\beta^-} = \frac{\beta + (1+\hat{\rho})\, e^{-\hat{\rho}}\cdot\beta^+}{1-(1+\rho)\, e^{-\rho}\cdot\beta^-}, \tag{2}$$

where β^- is the missing tag ratio, β is the remaining tag ratio, β^+ is the new tag ratio, $\hat{\rho}$ is the load factor of the expected frame, ρ is the load factor of the true frame, and $\rho^{\cup}$ is the union load factor that is equal to $\frac{|T_t \cup T_{t+1}|}{f}$. 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 and expected frames.

This function $\alpha_{\text{one-phase}}$ only needs three parameters $\rho^{\cup}, \beta^-, \beta^+$. This is because for the true frame load factor ρ, we have

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

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

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

In contrast, the expected accuracy of the two-phase protocol in [14] can be calculated as

$$\alpha_{\text{two-phase}}(\rho^{\cup}, \beta^-, \beta^+) \approx \frac{\beta + e^{-\hat{\rho}}\cdot\beta^+}{1-e^{-\rho}\cdot\beta^-}.$$

DOI: 10.1002/cpe

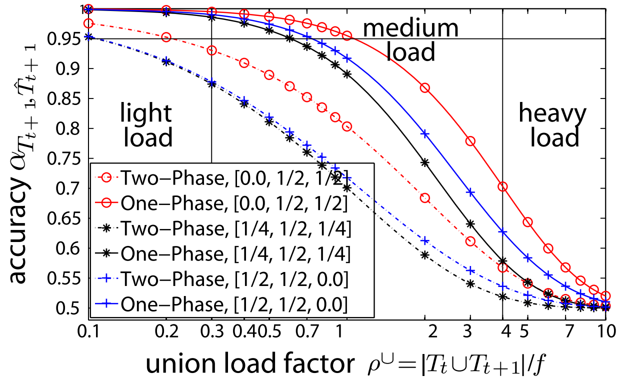

Figure 4. Compare accuracy between our one-phase protocol and the two-phase protocol in [14].

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 the 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. □

We plot in Figure 4 both the accuracy of our one-phase protocol and the accuracy of the two-phase protocol. Figure 4 adopts three typical scenarios with different combinations of $[\beta^-, \beta, \beta^+]$. 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.

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 $\rho^{\cup}$ is equal to 1, the accuracy of the two-phase protocol is 70%, whereas the accuracy of our one-phase protocol is 89%, which means 27.14% improvement. Figure 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 the remaining tags is larger than 2 remaining tags per slot. In slots with 2 or more remaining tags, we cannot 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.

5.3. *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 of tags are mixed in one slot. The input of the protocol is the prior knowledge of the 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.

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).

DOI: 10.1002/cpe

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 command 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 command (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 $\hat{T}_{t+1} := (T_t - T^-_{t+1}) \cup T^+_{t+1}$

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 Figure 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, for example, tag 5 that should respond in changed slot 6 in Figure 3. Such remaining tags that are wrongly marked as missing tags will not 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 are responded by RAND and showed their presence in slot i to the new tag set T^+_{t+1}. For example, in Figure 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, for example, tag 5. But this will not 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 the 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 is 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

DOI: 10.1002/cpe

to a new frame (see Ln. 15). The sole purpose of this new frame is to collect the 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 should not. The answer is to use the session flag feature** 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.

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.

6. ACCURACY-EFFICIENCY TRADE-OFF

In this section, we analyze the trade-off between the accuracy and the efficiency for the proposed one-phase protocol.

6.1. Time efficiency analysis

We analyze the time efficiency of our one-phase protocol as follows.

Theorem 2 (Time efficiency of one-phase protocol)
The time efficiency of one-phase protocol can be calculated as

$$\gamma_{\text{one-phase}}(\rho^{\cup}, \beta^-, \beta^+) = \frac{t_{\text{slot}}}{n^-_{\text{slot}} + n^+_{\text{slot}}}, \tag{3}$$

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.
- n^+_{slot} is the expected number of new tags that can be identified in a slot.

To calculate n^-_{slot} and n^+_{slot}, we can use the following equations:

$$\begin{aligned} n^-_{\text{slot}}(\rho^{\cup}, \beta^-, \beta^+) &= \rho^-(P^{\cap}_0 + P^{\cap}_1)P^+_0 + \rho^-(1 - e^{-\rho^-})P^{\cap}_0 P^+_1 + P^-_1 P^{\cap}_0 P^+_2, \\ n^+_{\text{slot}}(\rho^{\cup}, \beta^-, \beta^+) &= P^-_0(P^{\cap}_0 + P^{\cap}_1)\rho^+ + P^-_2 P^{\cap}_0 P^+_1 + P^-_1 P^{\cap}_0 \rho^+(1 - e^{-\rho^+}), \end{aligned}$$

where P^-_0, P^-_1, or P^-_2 are the probabilities of 0, 1, or 2+ missing tags in a slot, respectively; $P^{\cap}_0$, $P^{\cap}_1$, or $P^{\cap}_2$ are the probabilities of 0, 1, or 2+ remaining tags in a slot, respectively; P^+_0, P^+_1, or P^+_2 are the probabilities of 0, 1, or 2+ new tags in a slot. These probabilities can be calculated from the load factors ρ^-, $\rho^{\cap}$, ρ^+, where

- ρ^- is the missing tag load factor that is equal to $\beta^- \cdot \rho^{\cup}$,
- $\rho^{\cap}$ is the remaining tag load factor that is equal to $(1 - \beta^- - \beta^+) \cdot \rho^{\cup}$,
- ρ^+ is the new tag load factor that is equal to $\beta^+ \cdot \rho^{\cup}$.

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.

DOI: 10.1002/cpe

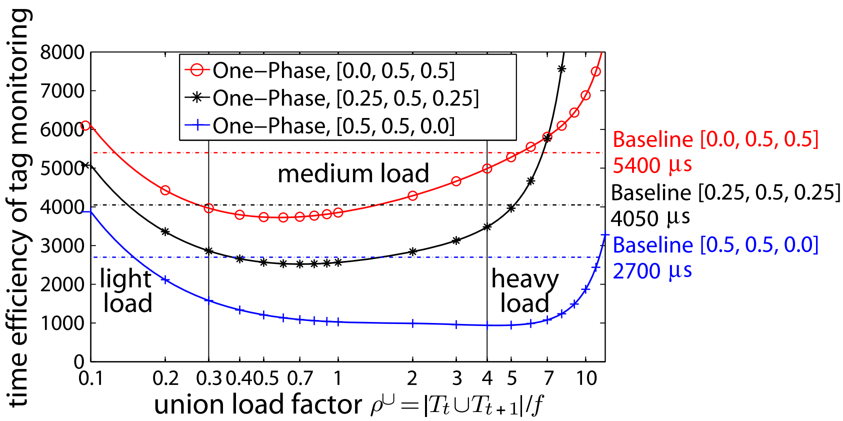

Figure 5. One-phase protocol versus baseline protocol in monitoring efficiency.

The expression to calculate t_{slot} can be found in Appendix A, which however is complicated. Its basic idea is to consider the five kinds of slots in the true frame: (i) empty slots; (ii) singleton slots whose previous state is also singleton; (iii) singleton slots whose previous state is non-singleton; (iv) collision slots whose previous state is also collision; and (v) collision slots whose previous state is non-collision. The time costs for the five cases are different. For the first case, it is t_e. For the second case, it is t_s. For the third case, it is t_{ID}. For the fourth case, it is t_c. For the last case, it is about 2700 μs per tag, because these tags will be collected in subsequent frames by an identification protocol. For all the five cases, we also need to calculate their probabilities and combine them linearly with the time cost to obtain t_{slot}.

This efficiency $\gamma_{\text{one-phase}}$ is a function of union load factor $\rho^{\cup}$, which is plotted in Figure 5. This figure adopts three scenarios: $[0.0, 0.5, 0.5]$, $[0.25, 0.5, 0.25]$, and $[0.5, 0.5, 0.0]$.

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 μs per changed tag (i.e., $\gamma_{\text{baseline}} = 2700\ \mu\text{s} \cdot \frac{\beta+\beta^{+}}{1-\beta}$), whereas the best efficiency of our one-phase protocol is roughly 1000 μs, that is, 70% reduction in time cost. In the $[0.0, 0.5, 0.5]$ scenario, the efficiency of baseline protocol is 5400 μs, whereas the best time efficiency of our protocol is about 4000 μs, that is, 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 does not need to transmit tag IDs.

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 the 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.

As a conclusion, when we use the one-phase protocol, we need to avoid the light-load region and especially the high-load region which experiences rapid efficiency degradation. Instead, the best practice is to let the union-load factor $\rho^{\cup}$ fall into the medium-load region by adjusting the frame size f, to achieve high-time efficiency.

6.2. *Accuracy-efficiency trade-off*

For the one-phase protocol, we need to keep a balance between the accuracy and the efficiency. The existence of such a balance can be perceived, if we check two figures jointly. Figure 4 shows that the accuracy can be improved by reducing the union load factor $\rho^{\cup}$ towards the light-load region. However, the high accuracy above 99% can only be achieved in the light-load region. Meanwhile, Figure 5 shows that when entering the light-load region, reducing $\rho^{\cup}$ will degrade the time efficiency. Thus, we need to analyze the accuracy-efficiency trade-off.

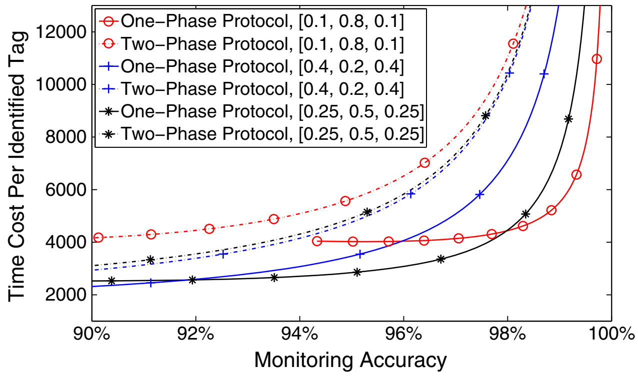

Figure 6. Trade-off between efficiency and accuracy.

We analyze the functional relation between accuracy and efficiency as follows. We have already proved that for the one-phase protocol, its accuracy and efficiency are both functions of the union load factor $\rho^{\cup}$ (see Theorems 1 and 2). Thus, by changing $\rho^{\cup}$, we can get the functional curves of efficiency against accuracy, which are plotted in Figure 6. This figure adopts the three following scenarios with different remaining tag ratio β:

- scenario $[0.1, \ 0.8, \ 0.1]$ that configures β as 0.8,
- scenario $[0.25, 0.5, 0.25]$ that configures β as 0.5,
- scenario $[0.4, \ 0.2, \ 0.4]$ that configures β as 0.2.

Figure 6 does not show the portion with accuracy lower than 90% because we believe that RFID users have no interests in such poor accuracy.

Figure 6 shows that, to achieve higher accuracy, we have to sacrifice the time efficiency, which is consistent with our expectation. Especially, if the required accuracy exceeds 98%, the time cost skyrockets with the increase of accuracy. The explanation is that our one-phase protocol is based on a *randomized algorithm*. This algorithm detects new tags by randomly distributing them to slots where they can be detected, for example, pre-empty slots and pre-singleton slots. If ultra-high accuracy is needed, we have to increase the percentage of such slots by configuring large frame size. For missing tag detection, the case is similar.

Figure 6 also shows that our one-phase protocol is more efficient than the two-phase protocol proposed in [14]. 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.

7. HYBRID PROTOCOL

We know that, to achieve high accuracy ($> 98\%$), our one-phase protocol has to severely sacrifice the time efficiency. In this section, we will further show that, in extreme cases, this protocol may become even less efficient than the baseline protocol. Thus, we will propose a hybrid protocol to make a smart switch to the baseline protocol, when the efficiency of the one-phase protocol is poor.

7.1. Motivation

We compare the time efficiency of one-phase protocol with that of baseline protocol in Figure 7. This figure shows that the one-phase protocol can be less efficient than the baseline protocol in the following two situations.

DOI: 10.1002/cpe

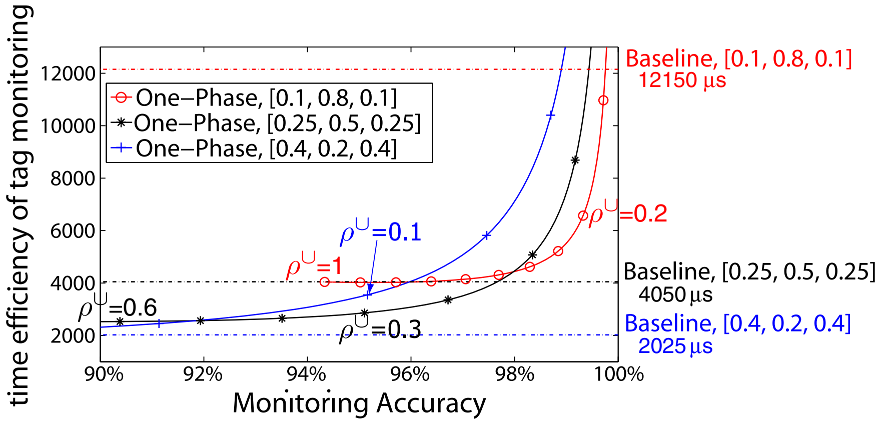

Figure 7. One-phase protocol vs. baseline protocol in time efficiency.

(1) The one-phase protocol can be less efficient than the baseline protocol, when the remaining tag ratio β is overly small. This is because the one-phase protocol adopts an *incremental update* strategy that identifies only the population changes. Such a strategy is no longer efficient, when the remaining tags become the minority (e.g., $\beta = 0.2$) and the population changes becomes the majority (e.g. $1 - \beta = 0.8$). Because now, there exist a non-negligible possibility that new tags and missing tags are mapped to one slot with equal numbers. In such slots, the reader cannot detect state changes and thus cannot detect the missing tags and new tags. Therefore, to minimize the possibility of such non-detectable cases, the one-phase protocol has to configure extra large frame size when β is as small as 0.2, to achieve satisfactory accuracy. However, this will reduce the efficiency of one-phase protocol to around 3800 µs when the accuracy is roughly 95%, as shown in Figure 7. In contrast, the baseline protocol can provide better time efficiency of 2025 µs when the remaining tag ratio β is 0.2.

(2) The one-phase protocol can be less efficient than the baseline protocol, when there is an ultra-high accuracy requirement that exceeds a certain degree. Figure 7 shows that, in the $[0.25, 0.5, 0.25]$ scenario, when the accuracy requirement exceeds 98%, the time efficiency of one-phase protocol gets close to or becomes even worse than that of baseline protocol. The reason is that our one-phase protocol is a *randomized algorithm*. This algorithm detects missing tags and new tags by randomly distributing them to slots where they can be detected. If ultra-high accuracy is needed, it has to use extra large frame size to increase the chance of detection, which however will cause the inflation of time cost per detected new (or missing) tag.

7.2. Hybrid protocol

To improve the tag-monitoring efficiency, we propose a hybrid protocol. This protocol invokes one-phase protocol in normal situations, and it makes a smart switch to the baseline protocol, when it finds that the one-phase protocol would provide worse efficiency than the baseline protocol. To facilitate better understanding of our hybrid protocol, we illustrate its effect in Figure 8. This figure, as usual, adopts the three scenarios of $[0.1, 0.8, 0.1]$, $[0.25, 0.5, 0.25]$, $[0.4, 0.2, 0.4]$ with different remaining tag ratio. For each scenario, there is an *accuracy turning point*. When the accuracy requirement exceeds this turning point, the one-phase protocol will become less efficient than the baseline protocol. Thus, our hybrid protocol switches from the one-phase protocol to the baseline. For example, in the $[0.25, 0.5, 0.25]$ scenario, this turning point is about 97%; in the $[0.1, 0.8, 0.1]$ scenario, this turning point is about 99.5%. In the $[0.4, 0.2, 0.4]$ scenario, it seems that there does not exist such a turning point, and the hybrid protocol completely adopts the baseline protocol. But in fact, the hybrid protocol still has a turning point that is smaller than 90%. Thus, this turning point for the $[0.4, 0.2, 0.4]$ scenario cannot be seen in Figure 8.

DOI: 10.1002/cpe

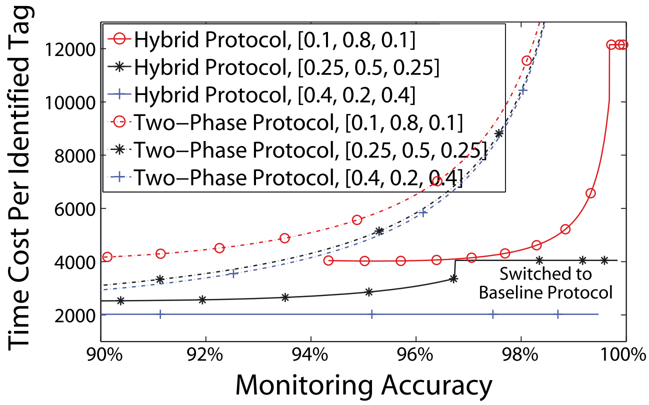

Figure 8. Our hybrid protocol vs two-phase protocol.

We present as follows the algorithm to calculate the accuracy turning point used by our hybrid protocol. We firstly calculate the union load factor $\rho^{\cup}_{\text{critical}}$ that corresponds to this accuracy turning point. It is based on solving the following equation:

$$\gamma_{\text{one-phase}}(\rho^{\cup}_{\text{critical}}, \beta^-, \beta^+) = \gamma_{\text{baseline}}(\beta^-, \beta^+)$$

where γ_{baseline} is the function to calculate the time efficiency of baseline protocol in Equation (1), and $\gamma_{\text{one-phase}}$ is the function to calculate the time efficiency of one-phase protocol in Equation (3). We assume that the missing tag ratio β^- and the new tag ratio β^+ are fixed and known. Thus, we can solve this equation and obtain the critical value of union load factor $\rho^{\cup}_{\text{critical}}$ where the one-phase protocol has the same efficiency with the baseline protocol. Then, our second step is to calculate the accuracy turning point by calling the function in Equation (2) with parameters $\rho^{\cup}_{\text{critical}}$, β^-, β^+.

Finally, we show in Figure 8 the advantage of our hybrid protocol over the two-phase protocol proposed in [14]. In $[0.25, 0.5, 0.25]$ scenario, our hybrid protocol can be 55.43% more efficient than the two-phase protocol at the same accuracy level (e.g., 96%), because our hybrid protocol adopts the one-phase protocol with better efficiency. Figure 8 also shows that the efficiency improvement of our hybrid protocol is even more prominent (i.e., about 65.33%) when the remaining tag ratio is as small as 0.2. This is because our hybrid protocol smartly switches to the baseline protocol.

8. RELATED WORK

Radio Frequency IDentification 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 [3–6, 8, 9]. The former organizes all tag IDs in a tree of ID prefixes, whereas 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, because we can establish an estimate of the current tag population, with the knowledge of the 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 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.

9. CONCLUSION

This paper focused on the problem of monitoring a dynamic tag population and tracking the population changes. By solving this problem, we make the following contributions. (i) We derived an optimal configuration of RAND length for the traditional tag ID protocol, which is neglected before. (ii) We proposed a one-phase solution which is more efficient and accurate than the previous two-phase protocol. Our solution is more is efficient because it uses only one frame to detect both missing tags and new tags. Our solution is more accurate because it uses both empty slots and singleton slots in the true frames (or in the expected frames) to detect the population changes. (iii) We proposed a hybrid protocol that can achieve high efficiency when the remaining tag ratio is small. This hybrid protocol can adaptively switch between our one-phase protocol and the traditional tag ID protocol.

APPENDIX A: TIME EFFICIENCY ANALYSIS OF ONE-PHASE PROTOCOL

Firstly, we calculate per slot time cost t_{slot} by Theorem 3.

Theorem 3 (Per-slot time cost)
For our one-phase protocol, the expected per-slot time cost t_{slot} can be calculated as:

$$t_{\text{slot}} = t_{\text{e}}\, P_0 + t_{\text{s}}\, P_1^{=} + t_{\text{ID}}\, P_1^{\neq} + t_{\text{c}}\, P_2^{=} + t_{\text{cID}}\, P_2^{\neq} \tag{4}$$

where P_0 is the probability for slot i to be empty with $s_i = 0$, $P_1^{=}$ (or $P_1^{\neq}$) is the probability for slot i to be singleton with $s_i = 1$ and unchanged with $\hat{s}_i = s_i$ (or changed with $\hat{s}_i \neq s_i$), $P_2^{=}$ (or $P_2^{\neq}$) is the probability for slot i to be collision with $s_i = 2$ and unchanged with $\hat{s}_i = s_i$ (or changed with $\hat{s}_i \neq s_i$). The symbols t_{e}, t_{s}, t_{ID}, t_{c}, t_{cID} are the time cost for the five situations, which are listed in the table below.

States of slot i	**Time cost**	**Probability**
$s_i = 0$	$t_{\text{e}} = 184\ \mu\text{s}$	$P_0 = e^{-\rho}$
$s_i = 1 \wedge \hat{s}_i = s_i$	$t_{\text{s}} = 280\ \mu\text{s}$	$P_1^{=} = P_1^{-} P_0^{\cap} P_1^{+} + P_0^{-} P_1^{\cap} P_0^{+}$
$s_i = 1 \wedge \hat{s}_i \neq s_i$	$t_{\text{ID}} = 2320\ \mu\text{s}$	$P_1^{\neq} = P_1 - P_1^{=}$
$s_i = 2 \wedge \hat{s}_i = s_i$	$t_{\text{c}} = 311.875\ \mu\text{s}$	$P_2^{=} = P_2^{-} P_0^{\cap} P_2^{+} + (1 - P_0^{-}) P_1^{\cap} (1 - P_0^{+}) + P_2^{\cap}$
$s_i = 2 \wedge \hat{s}_i \neq s_i$	$t_{\text{cID}} = t_{\text{c}} + 2700\ \mu\text{s} \cdot \rho(1 - e^{-\rho})/P_2$	$P_2^{\neq} = P_2 - P_2^{=}$

Proof
We analyze the probabilities of the five situations as follows. If $s_i = 0$, the time cost for this empty slot in the true frame is t_{e} (see Section 4.1 for definitions of t_{e}, t_{s}, t_{ID}, t_{c} with $\upsilon = 6$). If $s_i = 1$, the time cost of this singleton slot in the true frame depends on whether we need to collect ID. As described at Ln. 11-13 of Protocol 1, we need to collect ID in each changed singleton slot with $\hat{s}_i \neq s_i$. Thus, the time cost is t_{ID} when $\hat{s}_i \neq s_i$, and is t_{s} when $\hat{s}_i = s_i$. If $s_i = 2$, the time cost for this collision slot in the true frame is t_{c}. But when changes are detected with $\hat{s}_i \neq s_i$, we additionally need to collect the multiple IDs in this slot by another frame using tag ID protocols, whose cost is at best 2700 μs per tag when $\upsilon = 6$ as depicted in Figure 2.

DOI: 10.1002/cpe

Symbol	Definition
ρ^-	Missing tag density in a frame: $\rho^- = \lvert T_t - T_{t+1}\rvert / f$
$\rho^\cap$	Remaining tag density in a frame: $\rho^\cap = \lvert T_t \cap T_{t+1}\rvert / f$
ρ^+	New tag density in a frame: $\rho^+ = \lvert T_{t+1} - T_t\rvert / f$
P_0^- (or P_1^-, or P_2^-)	Probability of 0 (or 1, or at least 2) missing tags in a slot: $P_0^- = e^{-\rho^-}$; $P_1^- = \rho^- e^{-\rho^-}$; $P_2^- = 1 - P_0^- - P_1^-$
$P_0^\cap$ (or $P_1^\cap$, or $P_2^\cap$)	Probability of 0 (or 1, or at least 2) remaining tags in a slot: $P_0^\cap = e^{-\rho^\cap}$; $P_1^\cap = \rho^\cap e^{-\rho^\cap}$; $P_2^\cap = 1 - P_0^\cap - P_1^\cap$
P_0^+ (or P_1^+, P_2^+)	Probability of 0 (1, at least 2) new tags in a slot: $P_0^+ = e^{-\rho^+}$; $P_1^+ = \rho^+ e^{-\rho^+}$; $P_2^+ = 1 - P_0^+ - P_1^+$

We analyze the probabilities of the five situations as follows. The probability of $s_i = 0$ is no doubt P_0. For the situation $\hat{s}_i = s_i = 1$, the probability $P_1^=$ is $P_1^- P_0^\cap P_1^+$ (i.e., the probability of one missing tag, zero remaining tags, and one new tag in slot i) plus $P_0^- P_1^\cap P_0^+$ (i.e., the probability of zero missing tags, one remaining tag, and zero new tags in slot i). For the situation $\hat{s}_i = s_i = 2$, the probability $P_2^=$ is $P_2^- P_0^\cap P_2^+$ (i.e., the probability of two missing tags, zero remaining tags, and two new tags in slot i) plus $(1 - P_0^-) P_1^\cap (1 - P_0^+)$ (i.e., the probability of at least one missing tags, one remaining tag, and at least one new tag in slot i) plus $P_2^\cap$ (i.e., the probability of at least two remaining tags). □

Secondly, we calculate the expected number of per-slot identified changed tags as $n_{\text{slot}} = n_{\text{slot}}^- + n_{\text{slot}}^+$, where n_{slot}^- is the expected number of identified missing tags in a slot and n_{slot}^+ is the expected number of identified new tags in a slot.

$$\begin{aligned} n_{\text{slot}}^- &= \rho^-(P_0^\cap + P_1^\cap)P_0^+ + \rho^-(1 - e^{-\rho^-})P_0^\cap P_1^+ + P_1^- P_0^\cap P_2^+, \\ n_{\text{slot}}^+ &= P_0^-(P_0^\cap + P_1^\cap)\rho^+ + P_2^- P_0^\cap P_1^+ + P_1^- P_0^\cap \rho^+(1 - e^{-\rho^+}), \end{aligned}$$

where P_0^-, P_1^-, P_2^- are the probabilities of 0, 1, $\geqslant$2 missing tags in a slot, respectively; $P_0^\cap$, $P_1^\cap$, $P_2^\cap$ are the probabilities of 0, 1, $\geqslant$2 remaining tags in a slot, respectively; P_0^+, P_1^+, P_2^+ are the probabilities of 0, 1, $\geqslant$2 new tags in a slot, respectively.

Now, we can give out a more accurate estimate of tag monitoring accuracy than Equation (2), that is, $\alpha_{\text{one-phase}} = \frac{\rho^\cap + n_{\text{slot}}^+}{\rho^\cup - n_{\text{slot}}^-}$. This equation considers the interferences between new tags and missing tags and thus can work well when the remaining tag density $\rho^\cap$ is low. It can be converted to the following form.

$$\alpha_{\text{one-phase}} = \frac{\beta + (n_{\text{slot}}^+/\rho^+)\cdot\beta^-}{1 - (n_{\text{slot}}^-/\rho^-)\cdot\beta^+} \tag{5}$$

Finally, we can calculate the time efficiency of the one-phase protocol by $\gamma_{\text{one-phase}} = \frac{t_{\text{slot}}}{n_{\text{slot}}^- + n_{\text{slot}}^+}$. Note that this efficiency $\gamma_{\text{one-phase}}$ is a function of only one parameter, that is, the union load factor $\rho^\cup$, because the missing tag density ρ^-, the remaining tag density ρ^*, and the new tag density ρ^+ are all determined by the union load factor.

ACKNOWLEDGEMENTS

This work was partially supported by the project HK RGC PolyU 5299/11E.

DOI: 10.1002/cpe

REFERENCES

1. Weiser M. The computer for the 21st century. *Proceedings of ACM SIGMOBILE*, Vol. 3, New York, USA, 1999; 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. Semiconductors P. I-CODE smart label RFID tags, 2004.
5. Vogt H. Efficient object identification with passive RFID tags. *Proceedings of IEEE PERCOM*, Zurich, Switzerland, 2002; 98–113.
6. Zhen B, Kobayashi M, Shimizu M. Framed ALOHA for multiple RFID objects identification. *IEICE Trans. on Communications* 2005; **E88–B**(3):991–999.
7. Myung J, Lee W. Adaptive splitting protocols for RFID tag collision arbitration. *Proceedings of ACM MOBIHOC*, New York, 2006; 202–213.
8. Lee S-R, Joo S-D, Lee C-W. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. *Proceedings of IEEE MOBIQUITOUS*, San Diego, USA, 2005; 166–174.
9. Qian C, Liu Y, Ngan H, Ni LM. ASAP: Scalable identification and counting for contactless RFID systems. *Proceedings of IEEE ICDCS*, Genoa, Italy, 2010; 52–61.
10. Qian C, Ngan H, Liu Y. Cardinality estimation for large-scale RFID systems. *Proceedings of IEEE PerCom*, Hong Kong, 2008; 30–39.
11. Kodialam M, Nandagopal T. Fast and reliable estimation schemes in RFID systems. *Proceedings of ACM MobiCom*, LA, USA, 2006; 322–333.
12. Tan CC, Sheng B, Li Q. How to monitor for missing RFID tags. *Proceedings of IEEE ICDCS*, Beijing, China, 2008; 295–302.
13. Li T, Chen S, Ling Y. Identifying the missing tags in a large RFID system. *Proceedings of ACM MOBIHOC*, Chicago, USA, 2010; 1–10.
14. Sheng B, Li Q, Mao W. Efficient continuous scanning in RFID systems. *Proceedings of IEEE INFOCOM*, San Diego, USA, 2010; 1–9.
15. Xiao Q, Bu K, Xiao B. Efficient monitoring of dynamic tag populations in RFID systems. *Proceedings of IEEE/IFIP EUC (best paper award)*, Melbourne, Australia, 2011; 106–113.
16. Tang S, Li X, Chen G, Liu Y, Zhao J. Raspberry: A stable reader activation scheduling protocol in multi-reader RFID systems. *Proceedings of IEEE ICNP*, Princeton, NJ, 2009; 304–313.
17. Bu K, Xiao B, Xiao Q, Chen S. Efficient pinpointing of misplaced tags in large RFID systems. *Proceedings of IEEE SECON*, Salt Lake City, Utah USA, 2011; 260–268.

DOI: 10.1002/cpe