This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

Finding Needles in a Haystack: Missing Tag Detection in Large RFID Systems

Jihong Yu, Lin Chen, Rongrong Zhang, Kehao Wang

***Abstract*—Radio frequency identification (RFID) technology has been widely used in missing tag detection to reduce and avoid inventory shrinkage. In this application, promptly finding out the missing event is of paramount importance. However, existing missing tag detection protocols cannot efficiently handle the presence of a large number of unexpected tags whose IDs are not known to the reader, which shackles the time efficiency. To deal with the problem of detecting missing tags in the presence of unexpected tags, this paper introduces a two-phase Bloom filter-based missing tag detection protocol (BMTD). The proposed BMTD exploits Bloom filter in sequence to first deactivate the unexpected tags and then test the membership of the expected tags, thus dampening the interference from the unexpected tags and considerably reducing the detection time. Moreover, the theoretical analysis of the protocol parameters is performed to minimize the detection time of the proposed BMTD and achieve the required reliability simultaneously. In addition, we derive a critical threshold on the unexpected tag size for the execution of first phase in BMTD. Extensive experiments are then conducted to evaluate the performance of the proposed BMTD. The results demonstrate that the proposed BMTD significantly outperforms the state-of-the-art solutions.**

I. Introduction

A. Background

Recent years have witnessed an unprecedented development of the radio frequency identification (RFID) technology. As a promising low-cost technology, RFID is widely utilized in various applications ranging from inventory control [27] [28] [4], supply chain management and logistics [14] [32] [7] [26] [16] to tracking/location [24] [34] [10]. In these applications, an RFID system typically consists of one or several RFID readers and a large number of RFID tags. Specially, the RFID reader is a device equipped with a dedicated power source and an antenna and can collect and process the information of tags within its coverage area. An RFID tag, on the other hand, is a low-cost microchip labeled with a unique serial number (ID) to identify an object and can receive and transmit the radio signals via the wireless channel. More specifically, the tags are generally classified into two categories: passive and active tags. Passive tags are energized by the radio wave of the reader, while active tags have power sources and relatively long communication range.

J. Yu and L. Chen (corresponding author) are with Lab. Recherche Informatique (LRI-CNRS UMR 8623), Univ. Paris-Sud, 91405 Orsay, France, {jihong.yu, chen}@lri.fr. R. Zhang is with LIPADE, Univ. Paris Descartes, France, rongrong.zhang@parisdescartes.fr. K. Wang is with Dept. Inform. Eng., Wuhan University of Technology, China, kehao.wang@whut.edu.cn. The work of K. Wang is supported by National Natural Science Foundation of China under grant 61672395.

B. Motivation and problem statement

According to the statistics presented in [23], inventory shrinkage, a combination of shoplifting, internal theft, administrative and paperwork error, and vendor fraud, resulted in 44 billion dollars in loss for retailers in 2014. Fortunately, RFID technology can be used to reduce the cost by monitoring products for its low cost and non-line-of-sight communication pattern. Obviously, the first step in the application of loss prevention is to determine whether there is any missing tag. Hence, quickly finding out the missing tag event is of practical importance.

The presence of unexpected tags, however, prolongs the detection time and even leads to miss detection. Here, we present two examples to motivate the presence of unexpected tags in realistic scenarios.

- *Example 1.* Consider a retail store with expensive goods and a much larger amount of inexpensive goods, and an RFID system is deployed to monitor the goods. Because of the higher value of expensive products, they are expected to be detected more frequently, but the tags of inexpensive goods also response the interrogation of readers, which influences the decision of readers.
- *Example 2.* Consider a large warehouse rented to multiple companies where the products of the same company may be placed in different zones according to their individual categories, such as child food and adult food, chilled food and ambient food. When detecting the tags identifying products from one company, readers also receive the feedbacks from the tags of other companies.

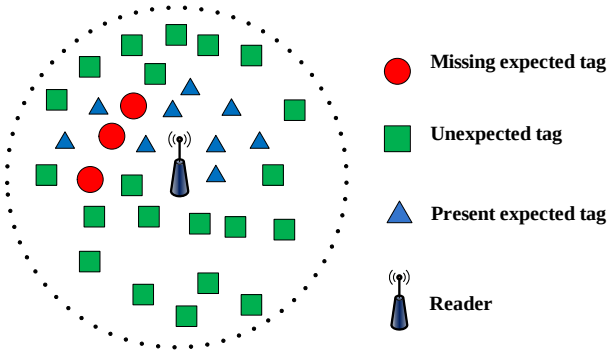


Fig. 1. Missing tag detection with the presence of unexpected tags.

In both examples, how to effectively reduce the impact of unexpected tags is of critical importance in missing tag detection. In this paper, we consider a scenario, as depicted in Fig. 1, where each product is affixed by an RFID tag. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

reader stores the IDs of expected tags. The problem we address is how to detect missing expected tags in the presence of a large number of unexpected tags in the RFID systems in a reliable and time-efficient way.

C. Prior art and limitation

Prior related work can be classified into three categories from the perspective of detecting missing tags: missing tag detection protocols, tag identification protocols, and tag estimation protocols.

There are two types of missing tag detection protocols: probabilistic [33] [18] [19] [31] and deterministic [15] [35] [17]. The probabilistic protocols find out a missing tag event with a certain required probability if the number of missing tags exceeds a given threshold, thus they are more time-efficient but return weaker results in comparison with the deterministic protocols that report all IDs of the missing tags. Actually, they can be used together such that a probabilistic protocol is executed in the first phase as an alarm that reports the absence of tags and then a deterministic protocol is executed to report IDs of missing tags. Unfortunately, all missing tag detection protocols except RUN [31] work on the hypothesis of a perfect environment without unexpected tags and thus fail to effectively detect missing tags in the presence of unexpected tags. Although RUN [31] is tailored for missing tag detection in the RFID systems with unexpected tags, all unexpected tags may always participate in the interrogation, which leads to the significant degradation of the performance when the unexpected tag population size scales.

Tag identification protocols [21] [22] [13] [30] can identify all tags in the interrogation region. To detect missing tags, tag identification protocols can be executed to obtain the IDs of the tags present in the population and then the missing tags can be found out by comparing the collected IDs with those recorded in the database. However, they are usually time-consuming [15] and fail to work when it is not allowed to read the IDs of tags due to privacy concern.

Tag estimation protocols [25] [29] [37] [5] are used to estimate the number of tags in the interrogation region. If many expected tags are absent in RFID systems without unexpected tags, a missing tag event may be detected by comparing the estimation and the number of expected tags stored in the database. However, the estimation error may be misinterpreted as missing tags and cause detection error, especially when there are only a few missing tags. Moreover, the estimation protocol cannot handle the case with a large number of unexpected tags.

D. Proposed solution and main contributions

Motivated by the detrimental effects of unexpected tags on the performance of missing tag detection, we devise a reliable and time-efficient protocol named Bloom filter-based missing tag detection protocol (BMTD). Specifically, BMTD consists of two phases, each consisting of a number of rounds.

- In each round of the first phase, the reader fist constructs a Bloom filter by mapping all the expected tag IDs into it such that each tag has multiple representative bits. Then the constructed Bloom filter is broadcasted to all tags. If at least one representative bit of a tag is '0's, it finds itself unexpected and will not participate in the rest of BMTD. Thus, the number of active unexpected tags is considerably reduced.
- Subsequently, in each round of the second phase, the reader constructs a Bloom filter by aggregating the feedbacks from the remaining tags and uses it to check whether any expected tag is absent from the population.

The major contributions of this paper can be articulated as follows. First, we propose a new solution for the important and challenging problem of missing tag detection in the presence of a large number of unexpected tags by employing Bloom filter to filter out the unexpected tags and then detect the missing tags. Second, we perform the theoretical analysis for determining the optimal parameters used in BMTD that minimize the detection time and also meet the required reliability, among which we unveil the fundamental relationship between the performance of the detection algorithm and unexpected tag size and derive a critical threshold on the unexpected tag size for the execution of filtering phase. Third, we perform extensive simulations to evaluate the performance of BMTD. The results show that BMTD significantly outperforms the state-of-the-art solutions.

The remainder of the paper is organised as follows. Section II gives a brief overview of related work. In Section III, we formally present the missing tag detection problem and describe the design goal and requirements. In Section IV and V, we elaborate the designed protocol and perform the theoretical analysis of the parameter configuration, respectively. In Section VI, we introduce the method to estimate the unexpected tag population size. Then the extensive simulations are conducted in Section VII. Finally, we conclude our paper in Section VIII.

II. Related Work

Extensive research efforts have been devoted to detecting missing tags by using probabilistic method [33] [18] [19] [31] and deterministic method [15] [35] [17]. Next, we briefly review the existing solutions of missing tag detection problem.

The objective of probabilistic protocols is to detect a missing tag event with a predefined probability. Tan *et al.* initiate the study of probabilistic detection and propose a solution called TRP in [33]. TRP can detect a missing tag event by comparing the pre-computed slots with those picked by the tags in the population. Different from our BMTD, TRP does not take into account the negative impact of unexpected tags. Follow-up works [18] [19] employ multiple seeds to increase the probability of the singleton slot. Same to TRP, they are required to know all the tags in the population. The latest probabilistic protocol called RUN is proposed in [31]. The difference with previous works lies in that RUN considers the influence of unexpected tags and can work in the environment with unexpected tags. However, RUN does not eliminate the interference of unexpected tags fundamentally such that the false positive probability does not decrease with respect to the unexpected tag population size, which shackles the detection

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

efficiency especially in the presence of a large number of unexpected tags. In addition, the first frame length is set to the double of the cardinality of the expected tag set in RUN, which is not established by theoretical analysis and leads to the failure of estimation method in RUN when the number of the unexpected tags is far larger than that of the expected tags.

The objective of deterministic protocols is to exactly identify which tags are absent. Li *et al.* develop a series of protocols in [15] which intend to reduce the radio collision and identify a tag not in the ID level but in the bit level. Subsequently, Zhang *et al.* propose another series of determine protocols in [35] of which the main idea is to store the bitmap of tag responses in all rounds and compare them to determine the present and absent tags. But how to configure the protocol parameters is not theoretically analyzed. More recently, Liu *et al.* [17] enhance the work by reconciling both 2-collision and 3-collision slots and filtering the empty slots such that the time efficiency can be improved. None of existing deterministic protocols, however, have been designed to work in the chaotic environment with unexpected tags. In this scenario, because unexpected tags may reply in the same slots with missing expected tags, the reader cannot detect missing tags in these slots, resulting in the failure of existing protocols.

Bloom filter is employed to solve tag searching problem in RFID systems [36], [6], while we address a different problem of detecting missing event in RFID system, the tag searching protocols thus cannot be used directly to efficiently solve our problem.

III. System Model and Problem Formulation

A. System model

Consider a large RFID system consisting of a single RFID reader and a large number of RFID tags. The reader broadcasts the commands and collects the feedbacks from the tags. In the RFID system, the tags can be either battery-powered active ones or lightweight passive ones that are energized by radio waves emitted from the reader. In this paper, we first take account of the single-reader case and then extend the proposed protocol to the multi-reader case.

The communications between the readers and the tags follow the *Listen-before-talk* mechanism [9]: A reader initiates communication first by sending commands and broadcasting the parameters to tags, such as the frame size, random seeds, and then each tag responds in its chosen time slot. Consider an arbitrary time slot, if no tag replies in this slot, it is called an *empty slot*; otherwise, it is called a *nonempty slot*. Only one bit is needed to distinguish an empty slot from a nonempty slot: '0' for an empty slot with an ideal channel while '1' for a nonempty slot with a busy channel.

During the communications, the tag-to-reader transmission rate and the reader-to-tag transmission rate may differ with each other and are subject to the environment. In practice, the former can be either 40kb/s $\sim$ 640kb/s in the FM0 encoding format or 5kb/s $\sim$ 320kb/s in the modulated subcarrier encoding format, while the later is normally about 26.7kb/s $\sim$ 128kb/s [8].

B. Problem formulation

In the considered RFID system, we use $\mathbb{E}$ to denote the set of IDs of the expected tags which are expected to be present in a population and target tags to be monitored. In the RFID system, we assume that an unknown number of tags, m, out of these $|\mathbb{E}|$ tags are missing. Note that $|\cdot|$ stands for the cardinality of a set. Denote by $\mathbb{E}_r$ the set of IDs of the remaining $|\mathbb{E}| - m$ tags that are actually present in the population. Let $\mathbb{U}$ be the set of IDs of unexpected tags within the interrogation region of the reader which does not need to be monitored. The reader may neither knows exactly the IDs of unexpected tags nor does it know the cardinality of $\mathbb{U}$.

Let M be a threshold on the number of missing expected tags. We use P_{sys} to denote the probability that the reader can detect a missing event. The optimum missing tag detection problem is formally defined as follows.

Definition 1 (Optimum missing tag detection problem)**.** *Given $|\mathbb{U}|$ unexpected tags where both $|\mathbb{U}|$ and the IDs of tags in $\mathbb{U}$ are unknown, the optimum missing tag detection problem is to devise a protocol of minimum execution time capable of detecting a missing event with probability $P_{sys} \geq \alpha$ if $m \geq M$, where α is the system requirement on the detection reliability.*

Table I summaries the main notations used in the paper.

TABLE I
Main Notations

Symbols	Descriptions
$\mathbb{E}$	set of target tags that need to be monitored
$\mathbb{E}_r$	tags that are actually present in the population
$\mathbb{U}$	set of unexpected tags
α	required detection reliability
m	number of missing expected tags
M	threshold to detect missing tags
P_{sys}	prob. of detecting a missing event in BMTD
J	number of rounds in Phase 1
l_j	length of the j-th frame of Phase 1
k_j	number of hash functions in the j-th frame of Phase 1
s_j	random seed used in the j-th frame of Phase 1
$\mathbb{U}_r$	set of remaining active unexpected tags after Phase 1
N^*	number of remaining active tags after Phase 1
$P_{1,j}$	false positive rate in the j-th frame of Phase 1
T_1	time cost of Phase 1
W	number of rounds in Phase 2
f_w	length of the w-th frame of Phase 2
R_w	number of hash functions in the w-th frame of Phase 2
d_w	random seed used in the w-th frame of Phase 2
$P_{2,w}$	false positive rate in the w-th frame of Phase 2
T_2	time taken to execute W rounds in Phase 2
T	theoretical execution time
q	prob. of detect a missing tag in a given slot of Phase 2
Z	random variable for slot of the first detection
$E[T_D]$	expected detection time of BMTD

IV. Bloom Filter-based Missing Tag Detection Protocol

A. Design rational and protocol overview

To improve the time efficiency of detecting missing tags in the presence of a large number of unexpected tags in the population, we limit the interference of unexpected tags in our

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

protocol. To achieve this goal, we employ a powerful technique called *Bloom filter* which is a space-efficient probabilistic data structure for representing a set and supporting set membership queries [3] to rule out the unexpected tags in the set $\mathbb{U}$, which efficiently reduces their interference and thus the overall execution time. Following this idea, we propose a *Bloom filter-based Missing Tag Detection protocol* (BMTD), by which Bloom filters are sequentially constructed by the reader and by the feedbacks from the active tags in the RFID system.

The BMTD consists of two phases: 1) the unexpected tag deactivation phase and 2) the missing tag detection phase.

- The first phase is divided into J rounds where the reader constructs J Bloom filters by mapping the recorded IDs in the reader to deactivate the unexpected tags after identifying them.
- The second phase is divided into W rounds. The reader constructs W Bloom filters according to the responses of the remaining active tags and uses the Bloom filters to detect any missing event. Our protocol either detects a missing event or reports no missing event if the reader does not detect a missing event after W rounds.

We elaborate the design of the BMTD in the rest of this section.

B. Phase 1: unexpected tag deactivation

In Phase 1, we use Bloom filters to reduce the number of active unexpected tags. Specifically, in the j-th round of Phase 1 ($j = 1, 2, ..., J$), the reader first constructs a Bloom filtering vector by mapping the expected tags in set $\mathbb{U}$ into an l_j-bit array using k_j hash functions with random seed s_j. Here, we denote the l_j-bit Bloom filter vector as $BF_{1,j}(\mathbb{E})$. How the values of l_j, k_j are chosen and how J is calculated are analysed in Sec. V on parameter optimisation.

Then, the reader broadcasts the l_j-bit Bloom filtering vector, k_j and s_j to all tags. Upon receiving $BF_{1,j}(\mathbb{E})$, k_j, and s_j, each tag maps its ID to k_j bits pseudo-randomly at positions $h_1(ID), h_2(ID), \cdots, h_{k_j}(ID)$, and checks the corresponding positions in $BF_{1,j}(\mathbb{E})$. If all of k_j bits are 1, then the tag regards itself expected by the reader. If any of k_j bits is 0, the tag regards that it is unexpected and then remains silent in the rest of the time.

Let $\mathbb{U}_j$ denote the set of the remaining active unexpected tags after the j-th round of Phase 1, and let $\mathbb{U}_j \cap BF_{1,j}(\mathbb{E})$ denote the set of unexpected tags that pass the membership test of $BF_{1,j}(\mathbb{E})$. Since the Bloom filter has no false negatives, the set of remaining active tags can be represented as $\mathbb{E}_r \cup \mathbb{U}_{j-1} \cap BF_{1,j}(\mathbb{E})$.

After J rounds when Phase 1 is terminated, the number of remaining active unexpected tags, termed as $|\mathbb{U}_r|$, is $|\mathbb{U}_J \cap BF_{1,J}(\mathbb{E})|$. The present tag population size can be written as $|\mathbb{E}_r \cup \mathbb{U}_r|$. Subsequently, the reader enters Phase 2.

C. Phase 2: missing tag detection

In the second phase, we still employ Bloom filter to detect a missing tag event. Note that the parameters that the reader broadcasts in each round in Phase 2 except random seeds are identical. In the w-th round of Phase 2 ($w = 1, 2, ..., W$), the reader first broadcasts the parameters containing the Bloom filter size f_w, the number of hash functions R_w, and a new random seed d_w. How their values are chosen and how W is calculated are analysed in Sec. V on parameter optimisation.

After receiving the configuration parameters, each tag in the set $\mathbb{E}_r \cup \mathbb{U}_r$ selects R_w slots at the indexes $h_v(ID)$ ($1 \le v \le R_w$) in the frame of f_w slots and transmits a short response at each of the R_w corresponding slots. As a consequence, a Bloom filter is formed in the air by the responses from the remaining active tags. In each round, there are two types of slots: empty slots and nonempty slots.

According to the responses from the tags, the reader encodes an f_w-bit Bloom filter as follows: If the i-th slot is empty, the reader sets i-th bit of the f_w-bit vector to be '0', otherwise '1'. Consequently, a virtual Bloom filter is constructed using which the reader then performs membership test. Let $BF_{2,w}(\mathbb{E}_r \cup \mathbb{U}_r)$ denote the constructed Bloom filter in w-th round.

To perform membership test, the reader uses tag IDs from the expected tag set $\mathbb{E}$. Specifically, for each ID in $\mathbb{E}$, the reader maps it into R_w bits at positions $h_v(ID)$ ($1 \le v \le R_w$) in $BF_{2,w}(\mathbb{E}_r \cup \mathbb{U}_r)$. If all of them are '1's, then the tag is regarded as present. Otherwise, the tag is considered to be missing. If a missing event is detected in w-round, the reader terminates the protocol without executing the remaining rounds. Otherwise, the reader initiates a new round until the protocol runs W rounds. If the reader does not detect a missing event after W rounds, it reports no missing event, i.e., the number of missing tags m is less than the threshold M.

D. An illustrative example of BMTD

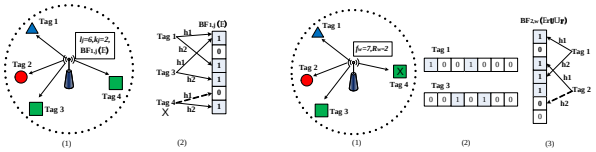
We present an illustrative example to show the execution of BMTD. Consider an RFID system with 4 tags. We assume that the reader needs to monitor tag 1 and tag 2 and thus knows their IDs, i.e., $\mathbb{E}$={ID1, ID2}, but it is not aware of the presence of tag 3 and tag 4, who are unexpected, i.e., $\mathbb{U}$={ID3, ID4}. In the example, tag 2 is missing from the population.

As shown in (1) of Fig. 2(a), the reader first constructs a Bloom filter $BF_{1,j}(\mathbb{E})$ by mapping IDs in $\mathbb{E}$ and broadcasts a message containing $BF_{1,j}(\mathbb{E})$ and the values of k_j and l_j. Here we assume $J = 1$, $k_j = 2$ and $l_j = 6$. After receiving $BF_{1,j}(\mathbb{E})$, each tag checks if it is an expected tag. As shown in (2) of Fig. 2(a), tag 1 finds itself expected due to the fact that both h_1(ID1) and h_2(ID1) are equal to 1. However, tag 4 realizes that it is unexpected for $h_1(\text{ID4}) = 0$ and deactivates itself. Different from tag 4, actually unexpected tag 3 passes the test and will participate in the rest of BMTD.

As depicted in (1) of Fig. 2(b), after the first phase, the reader starts to detect missing tags by broadcasting parameters f_w and R_w. Here we assume $W = 1$, $R_w = 2$ and $f_w = 7$. By using f_w and R_w, tag 1 and tag 3 generate a Bloom filter vector, respectively, which is shown in (2) of Fig. 2(b). Then they transmit following their individual Bloom filter vector. By sensing the channel, the reader can encode a Bloom filter and use it to check the IDs in $\mathbb{E}$ one by one. As shown in (3) of Fig. 2(b), since the Bloom filter is constructed based on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

the responses of tag 1 and tag 3, tag 1 passes the test but tag 2 fails and is regarded as absent. Then the protocol reports a missing event.



(a) Phase 1: unexpected tag deactivation

(b) Phase 2: missing tag detection

Fig. 2. Example illustrating BMTD

V. Performance optimisation and parameter tuning

In this section, we investigate how the parameters in the BMTD are configured to minimise the execution time while ensuring the performance requirement.

A. Tuning parameters in Phase 1

According to the property of Bloom filter, false negatives are impossible. The false positive rate of the Bloom filter $BF_{1,j}(\mathbb{E})$ in the j-th round in Phase 1, defined as $P_{1,j}$, can be calculated as follows [3]:

$$P_{1,j} = \left[1 - \left(1 - \frac{1}{l_j}\right)^{|\mathbb{E}|k_j}\right]^{k_j} \approx (1 - e^{-|\mathbb{E}|k_j/l_j})^{k_j}. \quad (1)$$

By rearranging (1), we can express the Bloom filter size in the j-th round as

$$l_j = \frac{-|\mathbb{E}|k_j}{\ln(1 - P_{1,j}^{\frac{1}{k_j}})}. \quad (2)$$

The total time spent in this round can thus be calculated as $l_j * t_r$, where t_r denotes the per bit transmission time from reader to tags.

We denote C_j the cost to detect and deactivate an unexpected tag as follows:

$$C_j = \frac{l_j t_r}{|\mathbb{U}|(1 - P_{1,j})} = \frac{-t_r|\mathbb{E}|k_j}{|\mathbb{U}|(1 - P_{1,j})\ln(1 - P_{1,j}^{\frac{1}{k_j}})}. \quad (3)$$

From the expression of C_j, it can be noted that C_j represents the average time consumed to detect and deactive an unexpected tag in the j-th round. In our design we minimize C_j so as to achieve the optimal time-efficiency. To minimize C_j, we first compute the derivative of C_j with respect to k_j as follows:

$$\frac{\mathbf{d}C_j}{\mathbf{d}k_j} = \frac{|\mathbb{E}|t_r\left(P_{1,j}^{\frac{1}{k_j}}\ln P_{1,j} - k_j(1 - P_{1,j}^{\frac{1}{k_j}})\ln(1 - P_{1,j}^{\frac{1}{k_j}})\right)}{|\mathbb{U}|(1 - P_{1,j})k_j(1 - P_{1,j}^{\frac{1}{k_j}})\ln^2(1 - P_{1,j}^{\frac{1}{k_j}})}. \quad (4)$$

Furthermore, let $\frac{\mathbf{d}C_j}{\mathbf{d}k_j} = 0$, we can obtain

$$P_{1,j}^{\frac{1}{k_j}} = \frac{1}{2}, \quad (5)$$

and the unique minimiser $k_j^* = \frac{-\ln P_{1,j}}{\ln 2}$ as $\frac{\mathbf{d}C_j}{\mathbf{d}k_j} > 0$ when $k_j > \frac{-\ln p_{1,j}}{\ln 2}$, and $\frac{\mathbf{d}C_j}{\mathbf{d}k_j} < 0$ when $k_j < \frac{-\ln p_{1,j}}{\ln 2}$. Therefore, C_j reaches the minimum value when $P_{1,j}^{\frac{1}{k_j^*}} = \frac{1}{2}$. The optimum Bloom filter size, denoted as l_j^*, can be computed as

$$l_j^* = \frac{|\mathbb{E}|k_j^*}{\ln 2}. \quad (6)$$

The time spent in the j-th round can be computed as $\frac{|\mathbb{E}|t_r k_j^*}{\ln 2}$. Therefore, the total execution time of Phase 1, denoted as T_1, can be derived as

$$T_1 = \sum_{j=1}^{J} \frac{|\mathbb{E}|t_r k_j^*}{\ln 2}. \quad (7)$$

k_j^* $(1 \le j \le J)$, as well as J, are set with the parameters in Phase 2 to minimize the global execution time, as analyzed in Sec. V-C and Sec. V-D.

Let N^* be the number of tags still active after Phase 1 (i.e., J rounds), it holds that

$$N^* = |\mathbb{E}| - m + |\mathbb{U}_r|, \quad (8)$$

where $\mathbb{U}_r$ is the set of unexpected tags still active after Phase 1. Recall (5), the expectation of N^* can be derived as

$$E[N^*] = |\mathbb{E}| - m + |\mathbb{U}|\prod_{j=1}^{J} P_{1,j} = |\mathbb{E}| - m + |\mathbb{U}|\left(\frac{1}{2}\right)^{\sum_{j=1}^{J} k_j^*}. \quad (9)$$

B. Tuning parameters in Phase 2

Similar to Phase 1, the false positive rate of the w-th round in Phase 2, defined as $P_{2,w}$, can be calculated as

$$P_{2,w} = \left[1 - \left(1 - \frac{1}{f_w}\right)^{N^* R_w}\right]^{R_w} \approx (1 - e^{-N^* R_w/f_w})^{R_w}. \quad (10)$$

Therefore, the Bloom filter size is

$$f_w = \frac{-N^* R_w}{\ln(1 - P_{2,w}^{\frac{1}{R_w}})}.$$

Moreover, the probability that at least one missing tag can be detected in w-th round, denoted as $P_{d,w}$, can be computed as

$$P_{d,w} = 1 - P_{2,w}^m. \quad (11)$$

Following the analysis above, the probability P_{sys} that the reader is able to detect a missing event after at most W rounds in Phase 2, can thus be written as

$$P_{sys} = 1 - \prod_{w=1}^{W}(1 - P_{d,w}) = 1 - P_{2,w}^{mW}. \quad (12)$$

It follows from the system requirement that

$$P_{sys} = 1 - P_{2,w}^{mW} = \alpha. \quad (13)$$

As a result, we can obtain

$$f_w = \frac{-N^* R_w}{\ln(1 - (1 - \alpha)^{\frac{1}{mWR_w}})}. \quad (14)$$

In the following lemma, we derive the optimum frame size of the Bloom filter f_w which is broadcast by the reader in each round of Phase 2.

Lemma 1. *Let $y \triangleq WR_w$, the optimum Bloom filter frame size, denoted by f_w^*, that achieves the detection requirement while minimising the execution time of Phase 2, is as follows:*

$$f_w^* = \frac{-N^* R_w}{\ln(1 - (1 - \alpha)^{\frac{1}{my^*}})} \quad (15)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

where $y^* = \frac{\ln(1-\alpha)}{m \ln \frac{1}{2}}$.

Proof. Denote by f the total length of all W Bloom filters in the second phase, we thus have

$$f = \sum_{w=1}^{W} f_w = \frac{-N^* W R_w}{\ln(1-(1-\alpha)^{\frac{1}{mWR_w}})}. \tag{16}$$

It can be checked that f depends on the product of W and R_w which is the total number of hash functions used in Phase 2. To minimize the execution time, let $y \triangleq WR_w$, we first calculate the derivation of f with respect to y as follows:

$$\frac{\mathbf{d}f}{\mathbf{d}y} = \frac{N^*(1-\alpha)^{\frac{1}{my}} \ln(1-\alpha)}{my(1-(1-\alpha)^{\frac{1}{my}}) \ln^2(1-(1-\alpha)^{\frac{1}{my}})} - \frac{N^*}{\ln(1-(1-\alpha)^{\frac{1}{my}})}.$$

Imposing $\frac{\mathbf{d}f}{\mathbf{d}y} = 0$ yields

$$y = \frac{\ln(1-\alpha)}{m \ln \frac{1}{2}}.$$

Moreover, when $y < \frac{\ln(1-\alpha)}{m \ln \frac{1}{2}}$, it holds that $\frac{\mathbf{d}f}{\mathbf{d}y} < 0$; when $y > \frac{\ln(1-\alpha)}{m \ln \frac{1}{2}}$, it holds that $\frac{\mathbf{d}f}{\mathbf{d}y} > 0$. Therefore, f achieves the minimum at $y^* = \frac{\ln(1-\alpha)}{m \ln \frac{1}{2}}$. The minimum of f_w, denoted by f_w^* can be computed by injecting $y = y^*$ into (14). The proof is thus completed. □

Remark. *As the reader does not have prior knowledge on m, the number of missing tags, in the design of BMTD, we require that the detection performance requirement to be hold for any $m \geq M$. Hence, f_w^* and y^* are as follows:*

$$f_w^* = \frac{-N^* R_w}{\ln(1-(1-\alpha)^{\frac{1}{My^*}})}, \tag{17}$$

$$\text{where } y^* = \frac{\ln(1-\alpha)}{M \ln \frac{1}{2}}, \tag{18}$$

where we use $m = M$ in N^* and y^*, which is the hardest case. Since $N^* = |\mathbb{E}| - m + |\mathbb{U}_r|$, it can be checked that the detection probability P_{sys} is monotonically increasing and $P_{2,w}$ is monotonically decreasing with respect to the number of missing tags m, meaning that $m = M$ makes the detection hardest and any greater m will ease the hardness, it is thus reasonable to use $m = M$ in the rest of the analysis, because if the reader can detect a missing tag event with probability α when $m = M$, it will fulfill the detection with probability $P_{sys} > \alpha$ when $m > M$.

In addition, since y^* is the total number of hash functions used in Phase 2 and at least one round is executed so as to detect a missing event, y^* needs to be a positive integer. Therefore, we set $y^* = \lceil \frac{\ln(1-\alpha)}{M \ln \frac{1}{2}} \rceil$, which guarantees the required detection performance requirement. Note that R_w and W can be set as arbitrary positive integers.

Under the optimum parameter setting derived above, we can calculate the time needed to execute W rounds of Phase 2, denoted by T_2, as follows:

$$T_2 = \frac{-t_t N^* y^*}{\ln(1-(1-\alpha)^{\frac{1}{My^*}})}, \tag{19}$$

where t_t is the time needed by the tags to transmit one bit to the reader. T_2 sets an upper-bound on the execution time of Phase 2.

C. Tuning k_j^* and J to minimize worst-case execution time

In this subsection, we study how to set k_j^* and J to minimize the worst-case execution time, which corresponds to the experience of the execution time where no missing event is detected and hence all the W rounds in the second round need to be executed. We denote the worst-case execution time by T. In the following theorem, we derive the minimiser of $\mathbb{E}[T]$.

Theorem 1. *Denote $x \triangleq \sum_{j=1}^{J} k_j^*$, x need to be set to x^* as follows to minimise the worst-case execution time of the BMTD:*

$$x^* = \begin{cases} 0 & |\mathbb{U}| \leq U_0 \\ \frac{\ln \frac{-t_r |\mathbb{E}| \ln(1-(1-\alpha)^{\frac{1}{My^*}}))}{t_t y^* |\mathbb{U}| \ln^2 2}}{-\ln 2} & |\mathbb{U}| > U_0 \end{cases}, \tag{20}$$

where $U_0 \triangleq \frac{|\mathbb{E}| t_r \ln(1-(1-\alpha)^{\frac{1}{My^*}})}{-t_t y^* \ln^2 2}$. That is, in regard to minimise the worst-case execution time, when the number of unexpected tags does not exceed a threshold U_0, Phase 1 is not executed, otherwise Phase 1 is executed with the parameters k_j^* and J set to $\sum_{j=1}^{J} k_j^* = x^*$.

Proof. Recall the two phases of BMTD and (7), we can derive the expectation of T as follows:

$$\begin{aligned} \mathbb{E}[T] = T_1 + T_2 &= \sum_{j=1}^{J} \frac{|\mathbb{E}| t_r k_j^*}{\ln 2} + \frac{-t_t y^* E[N^*]}{\ln(1-(1-\alpha)^{\frac{1}{My^*}})} \\ &= \frac{|\mathbb{E}| t_r}{\ln 2} \sum_{j=1}^{J} k_j^* + \frac{-t_t y^* \left(|\mathbb{E}| - M + |\mathbb{U}| \left(\frac{1}{2}\right)^{\sum_{j=1}^{J} k_j} \right)}{\ln(1-(1-\alpha)^{\frac{1}{My^*}})}. \end{aligned} \tag{21}$$

From (21), it can be noted that $E[T]$ is a function of $x = \sum_{j=1}^{J} k_j^*$. We then calculate the optimum x^* that minimizes $E[T]$. To that end, we compute the derivation of $E[T]$ with respect to x:

$$\frac{\mathbf{d}E[T]}{\mathbf{d}x} = \frac{|\mathbb{E}| t_r}{\ln 2} + \frac{t_t y^* |\mathbb{U}| \ln 2}{\ln(1-(1-\alpha)^{\frac{1}{My^*}})} \left(\frac{1}{2}\right)^x. \tag{22}$$

Since $\left(\frac{1}{2}\right)^x \leq 1$, it thus holds for all $x \geq 0$ that $\frac{\mathbf{d}E[T]}{\mathbf{d}x} \geq 0$ if $\frac{|\mathbb{E}| t_r}{\ln 2} + \frac{t_t y^* |\mathbb{U}| \ln 2}{\ln(1-(1-\alpha)^{\frac{1}{My^*}})} \geq 0$, i.e.,

$$|\mathbb{U}| \leq \frac{|\mathbb{E}| t_r \ln(1-(1-\alpha)^{\frac{1}{My^*}})}{-t_t y^* \ln^2 2} = U_0. \tag{23}$$

It is worth noticing that $E[T]$ is a monotonic nondecreasing function in this case with respect to x, we thus set $x = 0$ to minimize the execution time, which means that if the number of unexpected tags is smaller than the threshold U_0, we should remove the Phase 1 and only execute Phase 2.

In contrast, if $|\mathbb{U}| > U_0$, $\frac{\mathbf{d}E[T]}{\mathbf{d}x}$ can be negative, zero, or positive. Setting $\frac{\mathbf{d}E[T]}{\mathbf{d}x} = 0$, the optimal value of x to minimise $E[T]$, defined as x^*, can be calculated as

$$x^* = \frac{\ln \frac{-t_r |\mathbb{E}| \ln(1-(1-\alpha)^{\frac{1}{My^*}})}{t_t y^* |\mathbb{U}| \ln^2 2}}{-\ln 2}.$$

□

Remark. *Since x^* represents the total number of hash functions used in Phase 1, it needs to be a non-negative integer. Therefore, we set x^* either to its ceiling or floor*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

integer depending on which one leads to a smaller $E[T]$. The parameters k_j^* and J are set such that $\sum_{j=1}^{J} k_j^* = x^*$.

D. Tuning k_j^* and J to minimize expected detection time

The parameters derived in Theorem 1 establish that the BMTD is able to detect a missing event with probability equal to or greater than the system requirement α after W rounds of Phase 2. However, in many practical scenarios, the missing event may be detected in the round $w < W$ when the algorithm can be terminated. In this subsection, we derive the parameter configuration (i.e., k_j^* and J) that minimises the expected detection time. To that end, we first calculate the probability that at least one of the missing tags can be detected for the first time in a given slot and use it to formulate the expectation of the missing event detection time.

Lemma 2. *The probability that a missing tag can be detected in a given slot of Phase 1, denoted by q, is as follows:*

$$q = \left(1-(1-(1-\alpha)^{\frac{1}{y^*M}})^{\frac{M}{N^*}}\right)\cdot\left(1-(1-\alpha)^{\frac{1}{y^*M}}\right). \quad (24)$$

A loose lower-bound for q, denoted as q_{min}, can be established as follows:

$$q_{min} = \left(1-(\frac{1}{2})^{\frac{M}{|\mathbb{E}|-M+|\mathbb{U}|}}\right)(1-(1-\alpha)^{\frac{1}{y^*M}}). \quad (25)$$

Proof. A missing tag can be detected in a given slot only when at least one missing tag is hashed to this slot and no tag in $\mathbb{E}_r \cup \mathbb{U}_r$ selects the same location. Consider the hardest case for detecting a missing tag event, i.e., $m = M$, the probability that at least one missing tag maps to the given slot can be given by $\left(1-(1-\frac{1}{f_w^*})^{MR_w}\right)$. The probability that no tag in $\mathbb{E}_r \cup \mathbb{U}_r$ maps to that slot is equal to $(1-\frac{1}{f_w^*})^{N^*R_w}$. Consequently, multiplying the former by the later leads to q, i.e.:

$$\begin{aligned} q &= \left(1-(1-\frac{1}{f_w^*})^{MR_w}\right)\cdot(1-\frac{1}{f_w^*})^{N^*R_w} \\ &\approx (1-e^{-\frac{MR_w}{f_w^*}})\cdot e^{-\frac{N^*R_w}{f_w^*}} \\ &= \left(1-(1-(1-\alpha)^{\frac{1}{y^*M}})^{\frac{M}{N^*}}\right)\cdot(1-(1-\alpha)^{\frac{1}{y^*M}}). \end{aligned}$$

We then derive the lower-bound q_{min}. To that end, noticing that q is negatively correlated with N^* which falls into the range $\left[|\mathbb{E}|-M, |\mathbb{E}|-M+|\mathbb{U}|\right]$, we have

$$q \geq \left(1-(1-(1-\alpha)^{\frac{1}{y^*M}})^{\frac{M}{|\mathbb{E}|-M+|\mathbb{U}|}}\right)\cdot(1-(1-\alpha)^{\frac{1}{y^*M}}).$$

On the other hand, noticing that $y^* = \lceil\frac{\ln(1-\alpha)}{M\ln\frac{1}{2}}\rceil \geq \frac{\ln(1-\alpha)}{M\ln\frac{1}{2}}$, we have $q \geq q_{min} = \left(1-(\frac{1}{2})^{\frac{M}{|\mathbb{E}|-M+|\mathbb{U}|}}\right)(1-(1-\alpha)^{\frac{1}{y^*M}})$. □

After calculating q, we next derive the expected missing event detection time, denoted by $\mathbb{E}[T_D]$.

Theorem 2. *The expected missing event detection time $\mathbb{E}[T_D]$ is given by the following equation:*

$$\mathbb{E}[T_D] = \frac{|\mathbb{E}|t_r x}{\ln 2} + t_t \sum_{N^*=|\mathbb{E}|-M}^{|\mathbb{E}|-M+|\mathbb{U}|} \frac{1-(1-q)^f - fq(1-q)^f}{q} \binom{|\mathbb{U}|}{N^*-|\mathbb{E}|+M}\left(\frac{1}{2^x}\right)^{N^*-|\mathbb{E}|+M}\left(1-\frac{1}{2^x}\right)^{|\mathbb{U}|-N^*+|\mathbb{E}|-M}. \quad (26)$$

Proof. Recall (16), it holds that there are $f = \frac{-N^*y^*}{\ln(1-(1-\alpha)^{\frac{1}{My^*}})}$ slots in Phase 2. We next calculate the number of slots before detecting the first missing tag. It is easy to check that the event that in slot z the reader detects the first missing tag happens if no missing tags is detected in the first $z-1$ slots while at least one missing tag is detected in slot z. Let Z denote the random variable of z, we have

$$P\{Z=z\} = (1-q)^{z-1} * q, \quad (27)$$

which is geometrically distributed.

We can then compute the expectation of Z, conditioned by N^*, as follows:

$$\begin{aligned} E[Z|N^*] &= \sum_{z=1}^{f} z\cdot P\{Z=z\} \\ &= \frac{1-(1-q)^f - fq(1-q)^f}{q}. \end{aligned} \quad (28)$$

Moreover, it follows from the analysis of Phase 1 that the probability that an unexpected tag is still active after Phase 1 is $\prod_{j=1}^{J} P_{1,j}$. On the other hand, since $\mathbb{U}_r$ represents the ID set of active unknown tags after Phase 1, recall (5) and $\sum_{j=1}^{J} k_j^* = x$, we can compute the probability of having u active unexpected tags after Phase 1 as follows:

$$\begin{aligned} P\{|\mathbb{U}_r| = u\} &= \binom{|\mathbb{U}|}{u}\left(\prod_{j=1}^{J} P_{1,j}\right)^u\left(1-\prod_{j=1}^{J} P_{1,j}\right)^{|\mathbb{U}|-u} \\ &= \binom{|\mathbb{U}|}{u}\left(\frac{1}{2^x}\right)^u\left(1-\frac{1}{2^x}\right)^{|\mathbb{U}|-u}. \end{aligned} \quad (29)$$

It can be noted that $|\mathbb{U}_r|$ follows the binomial distribution. Recall the relationship between N^* and $|\mathbb{U}_r|$ in (7), it holds that

$$E[Z] = \sum_{N^*=|\mathbb{E}|-M}^{|\mathbb{E}|-M+|\mathbb{U}|} E[Z|N^*]\binom{|\mathbb{U}|}{N^*-|\mathbb{E}|+M}\cdot \left(\frac{1}{2^x}\right)^{N^*-|\mathbb{E}|+M}\cdot\left(1-\frac{1}{2^x}\right)^{|\mathbb{U}|-N^*+|\mathbb{E}|-M} \quad (30)$$

Therefore, $E[T_D]$ can be derived as

$$E[T_D] = T_1 + E[Z]\cdot t_t = \frac{|\mathbb{E}|t_r x}{\ln 2} + E[Z]\cdot t_t. \quad (31)$$

Injecting $E[Z]$ into $E[T_D]$ completes the proof. □

After deriving $E[T_D]$ as a function of x, we seek the optimum, denoted by x_e^*, which minimizes $E[T_D]$. To this end, we first establish an upper-bound of x_e^* in the following lemma.

Lemma 3. *It holds that $x_e^* \leq \frac{2t_t\ln 2}{t_r|\mathbb{E}|q_{min}}$.*

Proof. We write $E[T_D]$ as a function of x. Specifically, let $E[T_D] = g(x)$. To prove the lemma, we show that for any $x > 2x_0$ it holds that $g(x) \geq g(x_0)$ where $x_0 \triangleq \frac{t_t\ln 2}{t_r|\mathbb{E}|q_{min}}$.

To this end, we first derive the bounds of $g(x)$. Recall (27),(28), (30) and (31), we have

$$\begin{aligned} g(x) &> \frac{|\mathbb{E}|t_r x}{\ln 2}, \\ g(x) &\leq \frac{|\mathbb{E}|t_r x}{\ln 2} + \frac{t_t}{q_{min}}. \end{aligned}$$

For any $x > 2x_0$, we then have

$$g(x) > \frac{|\mathbb{E}|t_r x}{\ln 2} > \frac{2|\mathbb{E}|t_r x_0}{\ln 2} = \frac{|\mathbb{E}|t_r x_0}{\ln 2} + \frac{t_t}{q_{min}} \geq g(x_0)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

The lemma is thus proved. □

Lemma 3 shows that x_e^* falls into the range $[0, 2x_0]$. We can thus search $[0, 2x_0]$ to find x_e^* that minimises $E[T_D]$ and then set J and k_j^* such that $\sum_{j=1}^{J} k_j^* = x_e^*$.

E. BMTD parameter setting: summary

We conclude this section by streamlining the procedure of the parameter setting in the BMTD:

1) *Set parameters in Phase 2:* given $|\mathbb{E}|$, M, α and $|\mathbb{U}|$, compute f_w^* and y^* by (17) and (18), respectively, and set R_w and W such that $R_w W = y^*$;
2) *Set parameters in Phase 1:* compute x^* by Theorem 1 if the objective is to minimise the worst-case execution time; compute x_e^* if the objective is to minimise the expected detection time; then the set of k_j^* and J is given such that $\sum_{j=1}^{J} k_j^* = x^*$ or $\sum_{j=1}^{J} k_j^* = x_e^*$.

Following the above two steps, we can obtain all parameters in the BMTD.

We conclude this section by remarking a limitation of BMTD, the incompatibility with the current RFID standard. We note that all the state-of-the-art solutions except RUN are not compatible with the current RFID standard, which dates back to 2005 and did not envisioned advanced functionalities such as missing tag detection at that time. However, BMTD can be easily implemented on programmable tags, such as WISP [2] or OpenBeacon [1], by programming tags to interpret the filtering vector and select response slots which are lightweight operations.

VI. Cardinality estimation

In order to execute the BMTD, the reader needs to estimate the number of unexpected tags $|\mathbb{U}|$. In our work, we use the SRC estimator which is designed in [5] and is the current state-of-the-art solution. Denote by $\overline{|\mathbb{E}| - m + |\mathbb{U}|}$ the estimated total number of tags in the system, then the cardinality $|\mathbb{U}|$ can be approximated as $\overline{|\mathbb{U}|} = \overline{|\mathbb{E}| - m + |\mathbb{U}|} - |\mathbb{E}|$ if $m << |\mathbb{E}|, |\mathbb{U}|$. Because the number of bits that set to one in Bloom filter is concentrated tightly around the mean [20] and [11], once the estimation $\overline{|\mathbb{U}|}$ is obtained, we can calculate the expectation of N^* according to (9) with $m = M$ and use it as the estimator of N^*.

The SRC estimator consists of two phases: rough estimation and accurate estimation. It is proven in [5] that SRC can obtain a rough estimation $\hat{n}$ which at least equals to $0.5(|\mathbb{E}| - m + |\mathbb{U}|)$ after its first phase. In the second phase, SRC can achieve that the relative estimation error is not greater than ϵ which is referred to as confidence range with the settings as follows: the frame size $L_{est} = \frac{65}{(1-0.04^{\epsilon})^2}$ and the persistence probability $p_{pe} = \min\{1, 1.6L_{est}/\hat{n}\}$.

We then analyse the overhead introduced to estimate the cardinality of $\mathbb{U}$. As proven in [5], the overhead of SRC estimator is at most $O(\frac{1}{\epsilon^2} + \log\log(|\mathbb{U}| + |\mathbb{E}|))$, which is moderate for large-scale RFID systems with large $|\mathbb{U}|$ and $|\mathbb{E}|$.

A. Fast detection of missing event

In our estimation approach, we require that $m \ll |\mathbb{E}|, |\mathbb{U}|$. In case where m is close to $|\mathbb{E}|, |\mathbb{U}|$, the estimation may not be accurate. Luckily, in this case, we can quickly detect a missing event in the cardinality estimation phase due to large m.

Specifically, we analyze the SRC estimator's capability of detecting missing event under large m by comparing the pre-computed slots with those selected by the present tags. Recall the proof of Lemma 2, we can derive the detection probability in any given slot, defined as q_{pre}, as

$$q_{pre} = \left(1 - \left(1 - \frac{p_{pe}}{L_{est}}\right)^m\right) * \left(1 - \frac{p_{pe}}{L_{est}}\right)^{(\mathbb{U}+\mathbb{E}-m)}. \quad (32)$$

Since the detections in different slots are independent of each other, the probability of detecting at least one missing tag event by the SRC estimator can be calculated as $1 - (1 - q_{pre})^{L_{est}}$ which is a increasing function of m.

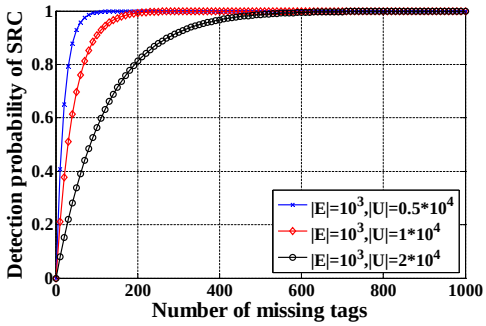


Fig. 3. q_{pre} vs. m.

Fig. 3 illustrates the detection probability of SRC with the various number of missing tags under different unexpected tag population sizes. To obtain the figure, we set $|\mathbb{E}| = 10^3$ and $\epsilon = 0.1$. It is observed that in the cases that $|\mathbb{U}| = 0.5 * 10^4$, $1 * 10^4$, $2 * 10^4$, SRC is able to detect at least a missing tag event with probability one when m is not less than 100, 200, 600, which means that a missing event is detected by SRC and the reader does not need to invoke the BMTD. In the other side, in the cases that m is less than 100, 200, 600, it holds that $|\frac{\overline{|\mathbb{U}|}}{|\mathbb{U}|} - 1| \leq 0.138, 0.132, 0.128$, respectively. With reference to the conclusion drawn from the Fig. 4, the BMTD can tolerate these levels of estimation error.

B. Sensibility to estimation error

The estimation algorithm we use inevitably introduces error on $|\mathbb{U}|$, which may have a negative impact on the performance of the BMTD. In order to investigate this impact, we next illustrate the sensitivity of the detection time to the estimation error.

Fig. 4 shows the theoretically calculated expected detection time from (26) under different unexpected tag population sizes and various levels of estimation error for $M = 1$. All results here are normalized with respect to the expected detection time without estimation error, which can be represented as $|\frac{\overline{E[T_D]}}{E[T_D]} - 1|$. As shown in the figure, the relative error of detection time increases with the estimation error in all range

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

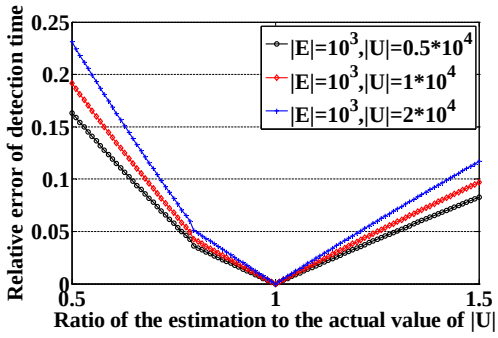


Fig. 4. $\left|\frac{\overline{E[T_D]}}{E[T_D]}-1\right|$ vs. $\frac{\overline{|\mathbb{U}|}}{|\mathbb{U}|}$.

of unexpected tag population. But it is worth noticing that the relative error of detection time only increases by 5% at most when $|\frac{\overline{|\mathbb{U}|}}{|\mathbb{U}|}-1|\leq 0.2$, which is nearly same with that without estimation error. Note that the slope before $\frac{\overline{|\mathbb{U}|}}{|\mathbb{U}|}=1$ changes because the calculated x_e changes from 2 to 3 which is optimal.

C. Enforcing detection reliability

Estimation error also has impact on the reliability of the BMTD as P_{sys} is calculated base on the estimated cardinality.

To enforce the detection reliability, we introduce more rounds to execute additional Bloom filters. The scheme works as follows: After receiving the Bloom filtering vector constructed by the active tags in the set $\mathbb{E}_r\cup\mathbb{U}_r$ in each round of Phase 2, the reader first counts the actual number of '1' bits in the filtering vector, defined as s_1 and uses it to compute the actual false positive probability, denoted by $\hat{P}_{2,w}$, as follows:

$$\hat{P}_{2,w}=\frac{s_1}{f_w^*}, \tag{33}$$

because an arbitrary unexpected tag maps to a '1' bit with a probability of s_1 out of f_w^*.

Following (13), we have the observed protocol reliability, denoted by $\hat{P}_{sys}$, as follows:

$$\hat{P}_{sys}=1-\hat{P}_{2,w}^{MW}. \tag{34}$$

If $\hat{P}_{sys}<\alpha$, the reader adds one more round in Phase 2 to further detect the missing tag event until $\hat{P}_{sys}\geq\alpha$.

D. Discussion on Multi-reader Scenario

In large-scale RFID systems deployed in a large area, multiple readers are deployed to ensure the full coverage for a larger number of tags in the interrogation region. In this scenario, we leverage the approach proposed in [12] and employed in [31]. The main idea is that a back-end server is used to synchronize all readers such that the RFID system with multiple readers operates as the single-reader case.

Specially, the back-end server calculates all the parameters involved in BMTD and constructs Bloom filter and sends them to all readers such that they broadcast the same parameters and Bloom filter to the tags. Because the parameters are identical across readers, a tag in the overlapped region will choose the same slots as in the single-reader case. Furthermore, each reader sends its individual Bloom filtering vector back to the back-end server. When the back-end server receives all Bloom filtering vectors, it applies logical OR operator on all received Bloom filtering vectors, which eliminates the impact of the duplicate readings of tags in the overlapped interrogation region. Consequently, a virtual Bloom filter is constructed by the back-end server.

VII. Performance Evaluation

The problem addressed in this paper is to detect the missing expected tags in the presence of a large number of unexpected tags in a time-efficient and reliable way. In this section, we evaluate the performance of the proposed BMTD. It has been shown in [31] that existing missing detection protocols cannot achieve the required reliability when there are unexpected tags in the RFID systems except the latest RUN [31]. We thus compare our proposed BMTD to RUN which is the only one considering the presence of the unexpected tags in terms of the actual reliability and the detection time. Note that the detection time can be interpreted as the time taken to either detect the fist missing tag event if a missing tag is found or complete the execution if no missing tag is found.

The simulation parameters are set with reference to [19] and [31]. Specifically, since both transmission rates from the tags to the reader and the reader to the tags depend on physical implementation and interrogation environment, we make the same assumption as in [19] that $t_r=t_t$. Moreover, because RUN is the baseline protocol, we use the similar simulation scenarios and the same performance metrics as in [31] where the time needed to detect a missing tag event is shown in terms of the number of slots. To that end, we, without loss of generality, assume $t_r=t_t=1$ in (26) in the simulation. Besides, we compute the optimal parameter values for RUN by following its specifications.

In the simulation, we use SRC [5] armed with missing tag detection function in this paper to estimate the unexpected tag population size with the confidence rang $\epsilon=0.1$. And all presented results are obtained by taking the average value of 100 independent trials under the same simulation setting.

We start by evaluating the performance of the BMTD by optimizing the worst-case execution time and the expected detection time.

A. Comparison between two strategies of BMTD

In this subsection, we compare the performance of two strategies of the BMTD which are abbreviated to Worst-M and Expected-M here, respectively. We set $|\mathbb{E}|=1000$, $m=100$, $\alpha=0.9$, $|\mathbb{U}|=10000:5000:30000$, $M=1$ and 50.

Table II lists the results where the first and second elements in the two-tuple $(\cdot,\cdot)$ denote the actual reliability and detection time, respectively. It can be seen that Expected-M costs less time than Worst-M to achieve the same reliability which is greater than the system requirement on the detection reliability, especially when M is small. Specifically, compared with Worst-1, Expected-1 reduces the detection time by up to 51.92% when $|\mathbb{U}|=10000$. This is because $x^*=5$ is too large for Phase 1 by optimizing the worst-case execution time,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

which wastes time. In contrast, minimizing the expected detection time relieves the influence of unexpected tag population size on the time of Phase 2 and thus outputs a smaller $x_e^* = 2$. In the rest of our simulation, we configure the parameters of the BMTD to minimise the expected detection time.

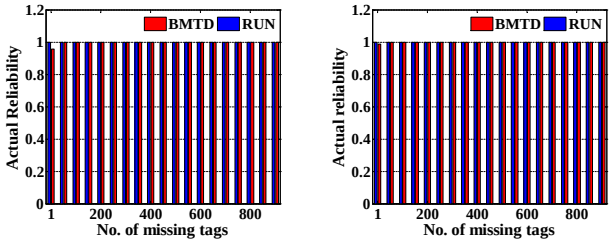
TABLE II
ACTUAL RELIABILITY AND DETECTION TIME OF BMTD

Strategy	Number of unexpected tags				
	10000	15000	20000	25000	30000
Worst-1	(1,4108)	(1,4441)	(1,5013)	(1,5453)	(1,5510)
Expected-1	(1,1975)	(1,3187)	(1,3569)	(1,3828)	(1,4191)
Worst-50	(1,1357)	(1,1841)	(1,2753)	(1,2762)	(1,2995)
Expected-50	(1,1353)	(1,1618)	(1,2272)	(1,2472)	(1,2815)

B. Comparison between BMTD and RUN

1) Comparison under different number of missing tags: In this subsection, we evaluate the performance of BMTD under different number of missing tags, which stands for the effectiveness and efficiency of BMTD. To that end, we set $|\mathbb{E}| = 1000$, $|\mathbb{U}| = 10000$, $m = 1 : 50 : 901$, $\alpha = 0.9$ and 0.99. Moreover, we set the threshold to $M = 1$.

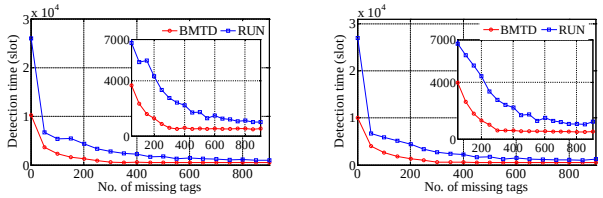
Actual reliability: BMTD achieves the required reliability for any missing tag population size when there are a large number of unexpected tags in the RFID systems. Fig. 5(a) and 5(b) illustrate the actual reliability of BMTD and RUN for $\alpha = 0.9$ and 0.99, respectively. It can be observed that both BMTD and RUN achieve the reliability more than that required by the system.



(a) $\alpha = 0.9$ (b) $\alpha = 0.99$

Fig. 5. Actual reliability vs. number of missing tags

Detection time: BMTD is more time-efficient in comparison to RUN. Fig. 6(a) and 6(b) show the detection time for $\alpha = 0.9$ and 0.99, respectively. For clearness, we further highlight the caves from $m = 51$ to 901. As shown in the figures, the detection time of BMTD is far shorter than that of RUN and decreases with the number of missing tags significantly. This is unsurprising. BMTD is able to deactivate major unexpected tags, which greatly reduces the number of active tags in the population, such that the presence of more missing tags makes the detection much easier. In contrast, RUN does not take into account the impact of unexpected tag population size, leading to longer detection delay in the presence of large number of unexpected tags.

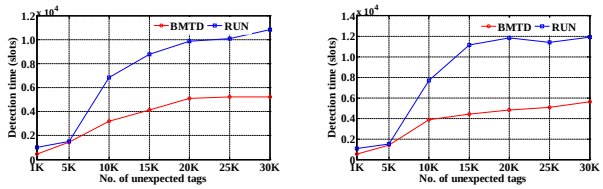


(a) $\alpha = 0.9$ (b) $\alpha = 0.99$

Fig. 6. Detection time vs. number of missing tags

2) Comparison under different number of unexpected tags: In this subsection, we evaluate the performance of BMTD under different number of unexpected tags, which represents the generality of BMTD. To that end, we set $|\mathbb{E}| = 1000$, $m = 50$, $M = 1$, $\alpha = 0.9$ and 0.99. Moreover, we select such $|\mathbb{U}| = 1000, 5000 : 5000 : 30000$ that various values of $\frac{|\mathbb{U}|}{|\mathbb{E}|}$ are covered in the simulation.

Actual reliability: BMTD achieves the reliability greater than the required reliability for different cardinalities of unexpected tag set. In the simulation, tt can be observed that the actual reliability achieved by both BMTD and RUN when $\alpha = 0.9$ and 0.99, respectively, is equal to one.



(a) $\alpha = 0.9$ (b) $\alpha = 0.99$

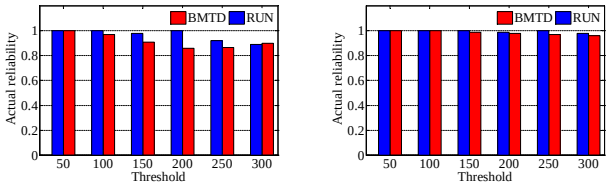
Fig. 7. Detection time vs. number of unexpected tags

Detection time: The BMTD outperforms the RUN considerably in terms of detection time even in the scenario with the small number of unexpected tag. Fig. 7(a) and 7(b) show the detection time for $\alpha = 0.9$ and 0.99, respectively. As shown in the figures, BTMD is able to save time especially when more unexpected tags are present in the population. Moreover, the increase in detection time of BTMD is more slow than that of RUN. This is due to the ability of BTMD that it can detect the missing tag event when estimating the $|\mathbb{U}|$ and determine whether to execute the unexpected tag deactivation phase following Lemma 3, which is exactly ignored in RUN.

3) Comparison under different values of threshold: In this subsection, we evaluate the performance of BMTD under different thresholds, which represents the tolerability of BMTD. To that end, we set $|\mathbb{E}| = 1000$, $|\mathbb{U}| = 10000$, $m = 100$, $\alpha = 0.9$ and 0.99. Moreover, we choose such $M = 50 : 50 : 300$ that the threshold can be greater or smaller than or equal to the number of missing tags in the simulation.

Actual reliability: BMTD achieves better reliability than the required reliability when $m \geq M$. As shown in Fig. 8(a) and 8(b), BMTD fails to achieve the required reliability only when $m < M$, which does not have negative impact because the objective of the missing tag detection protocol is to detect

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2666790, IEEE Transactions on Communications

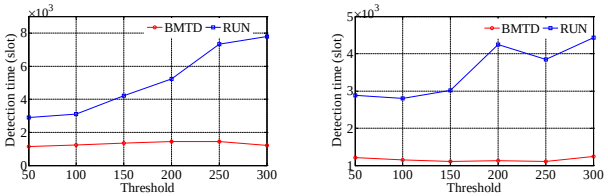


(a) $\alpha = 0.9$ (b) $\alpha = 0.99$

Fig. 8. Actual reliability vs. threshold

the missing tags only if the number of missing tags exceeds the threshold M.

Detection time: BMTD can tolerate the deviation from the threshold in terms of the detection time even when $m < M$. Fig. 9(a) and 9(b) show the detection time for $\alpha = 0.9$ and 0.99, respectively. It can be seen from the figures that the detection time of BMTD almost does not vary with the deviation. The detection time of RUN, by contrast, increases substantially as the deviation increase when $m < M$. This is because RUN terminates only when it runs optimal number of frames since the first frame when the estimated value of $|\mathbb{U}|$ does not vary by 0.1% in consecutive 10 frames if it does not detect any missing tag in any frame, while BMTD stops once the observed reliability $\hat{P}_{sys}$ exceeds α.



(a) $\alpha = 0.9$ (b) $\alpha = 0.99$

Fig. 9. Detection time vs. threshold

C. Impact of different estimation errors

In this subsection, we investigate the impact of estimation error on the actual reliability and detection time. We set $|\mathbb{E}| = 1000, |\mathbb{U}| = 10000$, $\alpha = 0.9$, $M = 1$ and $m = 1, 10, 50$.

Table III lists the results where the first and second elements in the two-tuple $(\cdot, \cdot)$ denote the actual reliability and detection time, respectively. It can be seen that the overall performance is best when estimation error is 0.1. When the estimation error is very low, namely 0.01, while BMTD has higher reliability, it spends huge time on achieving the stringent requirement on the estimation and is time-consuming. When the estimation is relatively rough, namely 0.3, though BMTD achieves the lower latency in some cases, it suffers from poor reliability which even cannot satisfy the system requirement when $m = 1$.

TABLE III
ACTUAL RELIABILITY AND DETECTION TIME UNDER DIFFERENT ESTIMATION ERRORS

Estimation error	Number of missing tags		
	1	10	50
0.01	(0.98,35311)	(1,32694)	(1,33185)
0.1	(0.931,9966)	(1,5467)	(1,3467)
0.3	(0.88,9348)	(0.98,5055)	(1,4864)

VIII. CONCLUSIONS

This paper has investigated an important problem of detecting missing tags in the presence of a large number of unexpected tags in large-scale RFID systems. Specifically, we aim at detecting a missing tag event in a reliable and time-efficient way. This paper has proposed a two-phase Bloom filter-based missing tag detection protocol (BMTD). In the first phase, we employed Bloom filter to screen out and then deactivate the unexpected tags in order to reduce their interference to the detection. In the second phase, we further used Bloom filter to test the membership of the expected tags to detect missing tags. We also showed how to configure the protocol parameters so as to optimize the detection time with the required reliability. Furthermore, we conducted extensive simulation experiments to evaluate the performance of the proposed protocol and the results demonstrate the effectiveness and efficiency of the propose protocol in comparison with the state-of-the-art solution.

REFERENCES

[1] OpeaBeacon. [Online]. Available: http://www.openbeacon.org/.
[2] WISP platform. [Online]. Available: https://wisp.wikispaces.com/.
[3] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.
[4] K. Bu, B. Xiao, Q. Xiao, and S. Chen. Efficient misplaced-tag pinpointing in large RFID systems. *IEEE Transactions on Parallel and Distributed Systems*, 23(11):2094–2106, 2012.
[5] B. Chen, Z. Zhou, and H. Yu. Understanding RFID counting protocols. In *ACM MobiHoc*, pages 291–302. ACM, 2013.
[6] M. Chen, W. Luo, Z. Mo, S. Chen, and Y. Fang. An efficient tag search protocol in large-scale rfid systems with noisy channel. *IEEE/ACM TON*, 24(2):703–716, 2016.
[7] S. Chen, M. Zhang, and B. Xiao. Efficient information collection protocols for sensor-augmented RFID networks. In *IEEE INFOCOM*, pages 3101–3109. IEEE, 2011.
[8] EPCglobal Inc. Radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 mhz - 960 mhz version 1.0.9. [Online], 2005. Available: http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_0_9-standard-20050126.pdf.
[9] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu. Counting RFID tags efficiently and anonymously. In *IEEE INFOCOM*, pages 1–9. IEEE, 2010.
[10] J. Han, C. Qian, X. Wang, D. Ma, J. Zhao, P. Zhang, W. Xi, and Z. Jiang. Twins: Device-free object tracking using passive tags. In *IEEE INFOCOM*, pages 469–476. IEEE, 2014.
[11] F. Hao, M. Kodialam, and T. Lakshman. Building high accuracy bloom filters using partitioned hashing. In *ACM SIGMETRICS*, pages 277–288. ACM, 2007.
[12] M. Kodialam, T. Nandagopal, and W. C. Lau. Anonymous tracking using RFID tags. In *IEEE INFOCOM*, pages 1217–1225. IEEE, 2007.
[13] T. F. La Porta, G. Maselli, and C. Petrioli. Anticollision protocols for single-reader RFID systems: temporal analysis and optimization. *IEEE Transactions on Mobile Computing*, 10(2):267–279, 2011.
[14] C.-H. Lee and C.-W. Chung. Efficient storage scheme and query processing for supply chain management using RFID. In *ACM SIGMOD*, pages 291–302. ACM, 2008.
[15] T. Li, S. Chen, and Y. Ling. Identifying the missing tags in a large RFID system. In *ACM MobiHoc*, pages 1–10. ACM, 2010.
[16] J. Liu, B. Xiao, K. Bu, and L. Chen. Efficient distributed query processing in large rfid-enabled supply chains. In *IEEE INFOCOM*, pages 163–171. IEEE, 2014.
[17] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu. Completely pinpointing the missing RFID tags in a time-efficient way. *IEEE Transactions on Computers*, 64(1):87–96, 2015.

[18] W. Luo, S. Chen, T. Li, and Y. Qiao. Probabilistic missing-tag detection and energy-time tradeoff in large-scale RFID systems. In *ACM MobiHoc*, pages 95–104. ACM, 2012.

[19] W. Luo, S. Chen, Y. Qiao, and T. Li. Missing-tag detection and energy–time tradeoff in large-scale RFID systems with unreliable channels. *IEEE/ACM Transactions on Networking*, 22(4):1079–1091, 2014.

[20] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[21] J. Myung and W. Lee. Adaptive splitting protocols for RFID tag collision arbitration. In *ACM MobiHoc*, pages 202–213. ACM, 2006.

[22] V. Namboodiri and L. Gao. Energy-aware tag anticollision protocols for rfid systems. *IEEE Transactions on Mobile Computing*, 9(1):44–59, 2010.

[23] National Retail Federation. National retail security survey. [Online], 2015. Available: https://nrf.com/resources/retail-library/national-retail-security-survey-2015.

[24] L. M. Ni, D. Zhang, and M. R. Souryal. RFID-based localization and tracking technologies. *IEEE Wireless Communications*, 18(2):45–51, 2011.

[25] C. Qian, H. Ngan, Y. Liu, and L. M. Ni. Cardinality estimation for large-scale RFID systems. *IEEE Transactions on Parallel and Distributed Systems*, 22(9):1441–1454, 2011.

[26] Y. Qiao, S. Chen, T. Li, and S. Chen. Energy-efficient polling protocols in RFID systems. In *ACM MobiHoc*, page 25. ACM, 2011.

[27] RFID Journal. DoD releases final RFID policy. [Online], 2004. Available: http://www.rfidjournal.com/article/articleview/1080/1/1.

[28] RFID Journal. DoD reaffirms its RFID goals. [Online], 2007. Available: http://www.rfidjournal.com/article/articleview/3211/1/1.

[29] M. Shahzad and A. X. Liu. Every bit counts: fast and scalable RFID estimation. In *ACM Mobicom*, pages 365–376, 2012.

[30] M. Shahzad and A. X. Liu. Probabilistic optimal tree hopping for RFID identification. In *ACM SIGMETRICS*, volume 41, pages 293–304. ACM, 2013.

[31] M. Shahzad and A. X. Liu. Expecting the unexpected: Fast and reliable detection of missing RFID tags in the wild. In *IEEE INFOCOM*, pages 1939–1947. IEEE, 2015.

[32] B. Sheng, C. C. Tan, Q. Li, and W. Mao. Finding popular categories for RFID tags. In *ACM MobiHoc*, pages 159–168. ACM, 2008.

[33] C. C. Tan, B. Sheng, and Q. Li. How to monitor for missing RFID tags. In *IEEE ICDCS*, pages 295–302. IEEE, 2008.

[34] P. Yang, W. Wu, M. Moniri, and C. C. Chibelushi. Efficient object localization using sparsely distributed passive RFID tags. *IEEE Transactions on Industrial Electronics*, 60(12):5914–5924, 2013.

[35] R. Zhang, Y. Liu, Y. Zhang, and J. Sun. Fast identification of the missing tags in a large RFID system. In *IEEE SECON*, pages 278–286. IEEE, 2011.

[36] Y. Zheng and M. Li. Fast tag searching protocol for large-scale rfid systems. *IEEE/ACM TON*, 21(3):924–934, 2013.

[37] Y. Zheng and M. Li. Zoe: Fast cardinality estimation for large-scale RFID systems. In *IEEE INFOCOM*, pages 908–916. IEEE, 2013.

Jihong Yu received the B.E degree in communication engineering and M.E degree in communication and information systems from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2010 and 2013, respectively, and is currently pursuing the Ph.D. degree in computer science at the University of Paris-Sud, Orsay, France. His research interests include wireless communications and networking and RFID technologies.

Lin Chen (S07-M10) received his B.E. degree in Radio Engineering from Southeast University, China in 2002 and the Engineer Diploma from Telecom ParisTech, Paris in 2005. He also holds a M.S. degree of Networking from the University of Paris 6. He currently works as associate professor in the department of computer science of the University of Paris-Sud. He serves as Chair of IEEE Special Interest Group on Green and Sustainable Networking and Computing with Cognition and Cooperation, IEEE Technical Committee on Green Communications and Computing. His main research interests include modeling and control for wireless networks, distributed algorithm design and game theory.

Rongrong Zhang received the B.Eng and M.Eng degrees in communication and information systems from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2010 and 2013, respectively, and is currently pursuing the Ph.D. degree in Computer Communication in the Faculty of Mathematics and Computer Science at the University of Paris Descartes, France. Her research interests focus on Wireless Area Body Networks (WBANs) for healthcare and wireless communications.

Kehao Wang received the B.S degree in Electrical Engineering, M.s degree in Communication and Information System from Wuhan University of Technology, Wuhan, China, in 2003 and 2006, respectively, and Ph.D in the Department of Computer Science, the University of Paris-Sud XI, Orsay, France, in 2012. He currently works as associate professor in the department of Information Engineering of the Wuhan University of Technology. His research interests are cognitive radio networks, wireless network resource management, and data hiding.