Efficient Framed Slotted Aloha Protocol for RFID Tag Anticollision

Haifeng Wu and Yu Zeng

***Abstract*—In this paper, we propose a novel efficient frame slotted aloha (EFSA) protocol for radio frequency identification (RFID) tag anticollision in this paper. After successfully identifying each tag, the EFSA protocol will allocate the tag a slot number, which signifies when the tag could be identified during a read cycle. When no tags arrive and leave, idle slots and collision slots will not be produced in subsequent read cycles. In addition, if there is a collision slot in the EFSA protocol, colliding tags in the collision slot will be resolved by Q-ary splitting where Q is equal to the estimated number of colliding tags, while the other unidentified tags is in a waiting state until the colliding tags are successfully resolved. Since allocation of tags to slots is not random in the EFSA protocol, conventional tag quantity estimates are not suitable. Therefore, we also propose a novel tag quantity estimate for the EFSA protocol. Simulation results show that the EFSA protocol outperforms conventional protocols, in term of time slots of reidentifying tags, and the proposed estimate error is less than the conventional estimates in the EFSA protocol.**

***Note to Practitioners*—The main advantage of the proposed protocol is to retain information obtained from a previous cycle of tag identification, and hence skip many collisions to quickly reidentify tags. In many RFID applications where a reader may repeatedly identify tags, such as object tracking and locating, the proposed protocols can reduce time of reidentifying tags. Although some current anticollision protocols such as ABS, TCFSA can reduce time of reidentifying tags, they may not be optimal ways in reducing identification delay. The proposed protocol's performance of identification delay can surpasses the current protocols when colliding tags increase. In addition, the proposed protocol needs to estimate tag quantity. Since many published estimates all assume that tags distribution should be random, they may not be applicable to the proposed protocol. The novel estimate in this paper can help engineers to adopt the proposed protocol to RFID tag identification.**

***Index Terms*—Anticollision, framed aloha, radio frequency identification (RFID), tag identification.**

I. Introduction

RADIO FREQUENCY IDENTIFICATION (RFID) has been considered an affordable and viable technology for fast and reliable identification of massive number of objects. When an RFID reader identifies multiple tags attached on objects, the reader's communication with the tags is over a shared wireless channel and collisions of tags will happen. Since the RFID reader generally adopts a tag anticollision protocol to resolve the multitag collisions [1], [2], how to design the anticollision protocol will influence the performance of multitag identification. Conventional anticollision protocols can be grouped into three broad categories: tree-based protocols [3]–[5], aloha-based protocols [6]–[13], and hybrid protocols, such as tree slot protocol [23], [24] which combines the tree-based protocols and the aloha-based protocols. Aloha-based protocols randomize access time of tags to reduce collisions, and are suitable for systems with limited capabilities mobile nodes and a powerful base station or a reader. Now, 13.56 MHz ISM band EPC Class 1 [14], ISO 18000-6 Type A [15], Type C [16], [26], and EPCglobal Generation 2 (EPC Gen 2) [17] all use aloha-based protocols.

In many RFID applications, readers may repeatedly identify the tags that have been identified in a previous process of identification, i.e., a previous cycle. For example, in object tracking and locating, tagged objects will be repeatedly identified since the information of the objects need to be frequently read by monitors [18]. Myung *et al.* propose an anticollision protocol: adaptive binary splitting (ABS) [20]–[22], and Deng *et al.* propose an adaptive anticollision protocol based on aloha: tag count frame slotted aloha (TCFSA) protocol [25]. The two protocols above can use the information in a previous cycle of tag identification, and hence reduce identification time when reidentifying tags. In the ABS protocol, if there is a collision slot, colliding tags in the collision slot will be resolved by binary splitting, while the other unidentified tags is in a waiting state until the colliding tags are successfully resolved. Therefore, the colliding tags in the collision slot would not collide with the other unidentified tags in the other slots. For this reason, ABS's throughput is higher than pure binary splitting protocol, pure aloha protocol and TCFSA based on aloha protocol. However, when the number of colliding tags is much greater than two, ABS's binary splitting may not be an optimal method. Many protocols [6], [7], [9]–[11], [23] discuss that, throughput could achieve a maximum value when the number of tags is equal to that of slots which the tags could select. Thus, Q-ary splitting may be a better method than binary splitting, where Q is the number of the colliding tags.

If Q is set to the number of the colliding tags, estimate of the number should be considered because the number is generally unknown to a reader. The estimate problem is discussed much in many papers [7], [9]–[13], [27]. In these conventional tag estimates, such as lower bound estimate [10]–[13], Schoute esti-

Manuscript received June 22, 2010; revised November 30, 2010; accepted December 06, 2010. Date of publication January 28, 2011; date of current version July 07, 2011. This paper was recommended for publication by Associate Editor S. Sarma and Editor S. Sarma upon evaluation of the reviewers' comments. This work was supported in part by the Scientific Research Foundation of Yunnan Provincial Department of Education (2010Y423).

The authors are with the Department of Electronic and Communication Engineering, School of Electrical and Information Technology, Yunnan University of Nationalities, Kunming 650500, China (e-mail: whf5469@gmail.com; yv.zeng@gmail.com).

Digital Object Identifier 10.1109/TASE.2010.2101061

mate [9], [12], [13], idle slot estimate [12], Vogt estimate [11], [12], maximum *a posteriori* (MAP) estimate [7], and Bayesian estimate [27], they assume that the allocation of tags to slots should be random. However, the allocation of tags to slots is not random in the ABS and TCFSA protocol. Consider a moving tag set, where tags are categorized into staying tags, leaving tags and arriving tags, according to their states between two consecutive cycles [19]–[22], [25]. In ABS and TCFSA, the staying tags' allocation is not random because the allocation is according to information in the previous identification cycle. If there are many staying tags in a cycle, those conventional estimates above will not be suitable.

This paper proposes an efficient framed slotted aloha (EFSA) protocol for RFID tag anticollision. The EFSA protocol can allocate each tag a slot number, after identifying the tag. The slot number signifies when the tag could be identified during an identification frame of a cycle. In the next cycle, a reader can use the slot number to reduce reidentifying time. Moreover, modified from the tree slot protocol, the EFSA protocol adopts Q-ary splitting to resolve tag collision, where Q is equal to the estimated number of colliding tags. Compared with the ABS protocol, EFSA have a better throughput than ABS when the number of colliding tags increase. In addition, if we consider some staying tags in the EFSA protocol, distribution of tags is not random. Thus, this paper also proposes a novel estimate, whose estimate error is less than conventional estimates, such as Vogt and MAP estimate.

This paper is organized as follows. Section II proposes the EFSA protocol. Section III proposes our novel tag quantity estimate for the EFSA protocol. In Section IV, the performance of the EFSA protocol is analyzed. In Section V, we provide simulations to demonstrate the performance of the EFSA protocol and our estimate method. Finally, conclusions are drawn in Section VI.

II. EFSA PROTOCOL

The EFSA protocol configures an identification cycle with some frames that consist of slots. A reader initiates a read frame by broadcasting a request command to all tags in its range. This request command also includes a parameter, the frame length. Each tag selects its response slot and transmits its ID in such a slot. For a given time slot, there are only three possible types: no tags response, only one tag response and more than two tags response, respectively. No tags response and one tag response will produce an idle slot and a readable slot, but more than two tags response may not necessarily result in a collision slot. If a tag is decoded in such a slot, we can call it capture effect [26]. Here, in order to explain EFSA's tree structure, we assume that occurrence probability of capture effect is zero. That is, more than two tags response will result in a collision slot.

The tree structure of the EFSA protocol is similar to the tree slot protocol [23], [24]. Let L_0 be the size of the first frame in EFSA, and N_j be the number of tags transmitting their ID in slot j, with $j \leq L_0$, $N_i \geq 0$, $\Sigma_j N_j = n$, where n is the total number of tags . If $N_j \geq 2$, there is a collision in slot j. At the end of each reading frame, if the reader realizes that collisions occurred, it starts a Q-ary splitting, i.e., a new reading frame for each slot where there was a collision. This corresponds to adding new nodes in the tree, as sons of the node representing the above reading frame, one son for each slot with collisions. The size of such new frames needs to be estimated, which will be discussed in the next section. Obviously, in each new reading frame, collisions can occur. Each time a collision is sensed, a new node is inserted in the tree, and another reading frame is started. The whole process is recursively repeated until no collisions are detected.

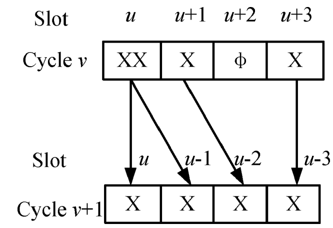

Fig. 1. EFSA: operation of allocating tags orderly slot number.

Algorithm 1. Reader operation in EFSA

```
Start, L0= RSC, RSC=0, LEVEL=0, SC=-1
collisionResolution(LEVEL, SC, L0) % call function
function collisionResolution(LEVEL, SLOT, Li) %    function definition
    SC=0, and broadcast(LEVEL, SLOT, Li)
    do {Transmit starting a slot command with SC and receive tag response
        if    tag collision
            channelStatus[SC]=busy, f=collision, and transmit f
        elseif no tag response
            channelStatus[SC]=empty, f=idle, and transmit f
        else only a tag response
            Receive ID from tag and store it
            RSC=RSC+1
            channelStatus[SC]=readable, f=readable, and transmit f
        end
        SC=SC+1 } while SC< Li
    [E, S, C]=performReadFrame(Li)
    % Counts empty, successful and collision slots in a frame with Li
    nEstimate=performTagEstimate(Li, E, S, C)    % Estimate tag quantity
    Li+1=floor((nEstimate-S)/C)
    SC=0, do { if channelStatus[SC]=busy
            collisionResolution(LEVEL+1, SC, Li+1)
        end
        SC=SC+1} while SC<Li
```

Fig. 2. EFSA protocol: pseudocode of reader's operation.

The EFSA protocol can reduce collision slots and idle slots when reidentifying tags. The reader adopting the EFSA protocol allocates each tag an orderly slot number after an identification cycle. The slot number signifies which slot within a frame the tag can select to transmit its ID. In the EFSA protocol, some collision slots and idle slots can be avoided because the reader can orderly identified the tags according to the slot number. Fig. 1 depicts the operation of allocating tags orderly slot number.

Fig. 2 shows pseudocode of a reader in the EFSA protocol. The reader has two counters, a *LEVEL* counter, a *slot counter* (*SC*) and a *readable slot counter* (*RSC*). *LEVEL* signifies a level of a tree of a reading cycle and is incremented by 1 when a new node is added to the tree. In each read cycle, the reader starts a frame by a command. The reader can send more than one command starting a frame during a cycle. *SC* is initialized to 0 at the beginning of a frame and is incremented by 1 at the end of each slot. When a frame's *SC* is greater than the frame length, the frame will be completed. *RSC* is initialized to 0 at the beginning of a read cycle and is incremented by 1 only in

Algorithm 2. Tag operation in EFSA

```
Energized by reader, enter Ready state and myLevel=0
while state=Ready & Receive starting command with (LEVEL, SLOT, L0)
      do { Enter Active state
            if LEVEL=0
                  if SSC=NULL or SSC >L0
                     SSC =random number from 0 to L0-1
                  end
                  myLevel=myLevel+1 and tagIdentify(SSC) % call function
            end
            if myLevel=LEVEL & SSC =SLOT
                  SSC =random number from 0 to Li-1
                  myLevel=myLevel+1, and tagIdentify(SSC) % call function
            end }
 function tagIdentify(SSC) % function definition
   while state=Active
         do { Receive starting a slot command with SC
               if SSC= SC
                     Transmit ID and receive feedback f from reader
                     if f=successful
                        Enter Identified state, and SSC=0
                     else
                        Enter Ready state
                     end
               end }
   while state=Identified
        do {Receive feedback f from reader
             if f=successful    SSC=SSC+1 end }
   return SSC
```

Fig. 3. EFSA protocol: pseudocode of tag's operation.

a readable slot. *RSC* will immediately count the number of tags that the reader has identified in a read cycle. According to tags' IDs received in a slot, the reader will know the type of the slot and inform all tags the type by transmitting a feedback. If the slot is readable, that means only a tag transmit its ID to the reader. The reader can identify the tag. An initialized frame length L_0 in a read cycle is set to the *RSC*'s value in a previous cycle. At the end of each frame, if the reader realizes that collisions occurred, it starts Q-ary splitting, i.e., a new reading frame for each slot where there was a collision. The length of the new read frame starting from a collision slot, L_{i+1} is set by

$$L_{i+1} = \left\lfloor \frac{n_i - S_i}{C_i} \right\rfloor \tag{1}$$

where n_i, S_i and C_i is the number of tags, the number of successful slots and collision slots in a read frame of i. Since the number of tags n_i is generally unknown, it needs to be estimated, which will be discussed in the next section. If there are no collision slots in all frames in a read cycle, the read cycle will be finished.

Fig. 3 shows the pseudocode of a tag operation in the EFSA protocol. At the beginning of a read cycle, a tag is energized by a reader and then enters a "Ready" state. After receiving a starting frame command with *LEVEL*, L_i and the slot number of a previous frame *SLOT*, the tag enters an "Active" state. Each tag has a *Selected Slot Counter* (*SSC*), which signifies which slot within a frame the tag can select to transmit its ID. Each tag also has a counter *myLevel*, which signifies the level of a tree where the tag is. When a frame's $LEVEL = 0$, this means that the frame is the first frame of a cycle. In the frame, a tag could transmit their IDs if its *SSC* is equal to *SC* received from the reader, where the tags' *SSC* value comes from a previous cycle. When a frame's $LEVEL \neq 0$, this means that the frame is not the first frame of a cycle. In the frame, if a tag's *mylevel* is equal to received *Level* and its *SSC* is equal to received *SLOT*, the tag's *SSC* will be reset to a random number from 0 to the frame length, and the tag also could transmit their IDs if its *SSC* is equal to *SC* received from the reader. If the ID collides with others, the tag will wait for being identifying in the next frame. If not, the tag can be identified, and its *SSC* will be set to 0 and then enter into an "Identified" state. The tag in "Identified" state increment its *SSC* if current timeslot is a readable one, but do nothing if not. Upon receiving a command ending the read cycle, the tag in the Identified state will not change the *SSC* value. Therefore, every tag holds a unique *SSC* value with respect to the number of tags recognized by the reader. The tag preserves the *SSC* value at the beginning of the next read cycle.

In EFSA protocol, some information is stored in tags counters, *SSC*. If the RF power should not be off during reidentifying tags, the information in the staying tags counter in RF filed will not be lost. For tags which leave RF field and reenter the field, the tags' information will be lost. However, if the leaving and reentering tags number is less, our algorithm can reduce reading time. Fig. 4 depicts an example of operation an idle slot and two collision slots of EFSA. In the $i-1$th cycle, there are five tags a, b, h, e, f, whose *SSC* is 0, 1, 2, 3, and 4, respectively. Since the tag h, whose *SSC* is 2 leaves in the ith cycle, this will cause an idle slot, and since two tag c, d, whose *SSC*s are both 1 and a tag g whose *SSC* is 4 are arriving in the ith cycle, this will also cause two collision slots. However, after all the staying and arriving tags are successfully identified, their *SSC*s will be reallocated as 6, 5, 4, 3, 2, 1 and 0, respectively. Thus, there are no unsuccessful slots if no tags arrive and leave in the $i+1$th cycle. In the example, EFSA can renumber tags in serial order according to their identified sequence, and each identified tag will get a unique *SSC*.

III. Tag Estimate

In EFSA protocol, if a reader realizes that collisions occurred, it starts Q-ary splitting, i.e., a new reading frame for each slot where there was a collision. Colliding tags in the collision slot will be identified in the new frame. The new frame length adjustment should be according to the tag quantity, and is set by (1). However, the tag quantity in (1) is usually unknown to a reader. In general, we can utilize read results collected at the reader, such as idle slot, collision slot and successful slot quantity in a frame to estimate the tag quantity. Most conventional tag estimates, such as Vogt estimate [11], [12], MAP estimate [7], and Bayesian estimate [27] are based on assumption that the allocation of tags to slots within a frame is random. In the first frame of a cycle of the EFSA protocol, however, there are some staying tags which are orderly allocated to slots by a reader in a previous cycle. These tags do not obey the random distribution. Therefore, Vogt estimate and MAP estimate cannot be extended to this condition. Certainly, if the first frame has colliding tags, the colliding tags will continue to be identified in some new frames and the allocation of the tags to slots within the frame will be random. Thus, Vogt and MAP estimates can be suitable. Next, we will mainly analyze tag quantity estimate in the first frame.

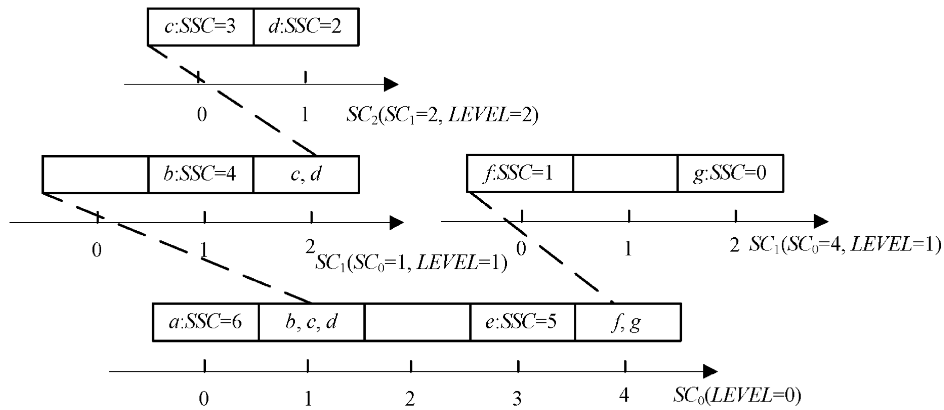

Fig. 4. EFSA: examples of operation in collision slot and idle slot.

Let A_i, $i > 1$ be the set of tags which are inside a reader's range in the ith read cycle of the EFSA protocol. To consider the tag's mobility, we defined staying tags set as $B = A_{i+1} \cap A_i$, leaving tags set as $C = A_i - B$ and arriving tags as $D = A_{i+1} - B$. We suppose an EFSA system with α arriving tags and β leaving tags at the beginning of the $i+1$th cycle, where $\alpha > 0, L > \beta > 0$, and L is the first frame length. If the arriving tags and leaving tags' *SSC* is a random number from 0 to L, the probability of finding r arriving tags in a slot of the first frame will be given by

$$p'_r = \binom{\alpha}{r}\left(\frac{1}{L}\right)^r\left(1-\frac{1}{L}\right)^{\alpha-r}, r = 0, 1, \ldots \tag{2}$$

and the probability of finding 0 and 1 staying tags in a slot will be given by

$$p''_0 = \frac{\beta}{L} \tag{3}$$

$$p''_1 = \frac{L-\beta}{L} \tag{4}$$

respectively. Therefore, the probability of finding 0, 1 and κ, $\kappa > 1$ tags in a slot are given by $p_0 = p'_0 p''_0$, $p_1 = p'_1 p''_0 + p'_0 p''_1$, and $p_\kappa = p'_\kappa + p'_1 p''_1$, respectively. Substituting (2)–(4) into the three formulas above, we have

$$p_0 = \frac{(L-1)^\alpha \beta}{L^{\alpha+1}} \tag{5-a}$$

$$p_1 = \frac{(L-1)^{\alpha-1}\alpha\beta + (L-1)^\alpha (L-\beta)}{L^{\alpha+1}} \tag{5-b}$$

$$p_\kappa = 1 - \frac{(L-1)^{\alpha-1}\alpha\beta + (L-1)^\alpha L}{L^{\alpha+1}}. \tag{5-c}$$

And the expected number of idle slots, collision slots and readable slots in a frame can be given by

$$a_0(\alpha,\beta) = \frac{(L-1)^\alpha \beta}{L^\alpha} \tag{6-a}$$

$$a_1(\alpha,\beta) = \frac{(L-1)^{\alpha-1}\alpha\beta + (L-1)^\alpha (L-\beta)}{L^\alpha} \tag{6-b}$$

$$a_\kappa(\alpha,\beta) = L - \frac{(L-1)^{\alpha-1}\alpha\beta + (L-1)^\alpha L}{L^\alpha}. \tag{6-c}$$

Thus, when the first frame with L time slots has c_0 idle slots, c_1 readable slots and c_κ collision slots, our estimate can be denoted by

$$\left(\hat{\alpha}, \hat{\beta}\right) = \arg \min_{\alpha\in\Omega,\beta\in\Xi} \|\mathbf{A}(\alpha,\beta) - \mathbf{C}\|^2 \tag{7-a}$$

$$\hat{n} = L + \hat{\alpha} - \hat{\beta} \tag{7-b}$$

where $\|\bullet\|$ is Euclidean norm and

$$\mathbf{C} = [c_0, c_1, c_\kappa]^{\mathrm{T}} \tag{8}$$

$$\mathbf{A}(\alpha,\beta) = [a_0(\alpha,\beta), a_1(\alpha,\beta), a_\kappa(\alpha,\beta)]^{\mathrm{T}}. \tag{9}$$

Since there is at least one arriving tag in a collision slot and one leaving tag in an idle slot, lower bound of α and β can be determined. Thus, the search tag range set, Ω and Ξ can be expressed as

$$\Omega = \{\alpha | c_\kappa \le \alpha \le \mathrm{A}\} \tag{10-a}$$

$$\Xi = \{\beta | c_0 \le \beta \le L\} \tag{10-b}$$

where we suppose that A is a maximum number of arriving tags that the RFID system can read.

To find a minimum, the estimate in (7) needs to search in the range of α, Ω and the range of β, Ξ. If we adopt a brute-force method, the search times will be $(A - c_\kappa + 1) \times (L - c_0 + 1)$, which is very high. Let

$$\delta(\alpha,\beta) = \|\mathbf{A}(\alpha,\beta) - \mathbf{C}\|^2. \tag{11}$$

From many simulations of computing $\delta(\alpha,\beta)$, we have the following results. For a given α, $\delta(\alpha,\beta)$ has a unique minimum at β_m and it will be monotonically decreasing for $\beta < \beta_m$ and monotonically increasing for $\beta > \beta_m$. Therefore, $\delta(\beta)$ is actually a V-shape curve with respect to β. Likewise, for a given β, $\delta(\alpha,\beta)$ is also a V-shape curve with respect to α. Based on the results, we can adopt a two-dimension (2D) binary search method to reduce the search times and hence lower the estimate computational complexity. The 2D binary search can be considered as a nest of two 1D binary searches. Thus, the 1D binary search's maximum estimate complexity is $O(3\log_2(\mathrm{A} - c_\kappa + 1))$, and the final 2D binary search's maximum is $O(3\log_2(\mathrm{A} - c_\kappa + 1) \times 3\log_2(L - c_0 + 1))$.

IV. Performance Analysis

A. System Throughput

For an RFID system, throughput can be defined as a ratio of average readable slots duration to total slots duration in a frame, which is given by

$$P_s = \frac{E(c_1)t_1}{E(c_0)t_0 + E(c_1)t_1 + E(c_\kappa)t_\kappa} \tag{12}$$

where t_0, t_κ, and t_1 is an idle, a collision, and a successful slot duration, respectively, and $E(\bullet)$ is an expected value. In EFSA protocol, the throughput can be discussed on two sides. First, all tags will be randomly allocated to slots in the first read cycle of EFSA because the tags *SSC*s could be random. In this scenario, average number of slots needed to identify n tags is $2.30n$ [23], [24]. Thus, the throughput can be given by 0.434. Second, in the first frame of the other read cycles in EFSA, some tags, such as staying tags are not randomly allocated to slots, but allocated by a reader in a previous read cycle. Therefore, the throughput analysis of tree slotted aloha [23], [24] is not suitable. In this section, we will mainly analyze the throughput under this condition.

Theorem 1: Suppose that durations of an idle, a collision and a readable slot are identical, and there are L identified tags in the ith cycle. If there are α arriving and β leaving tags at the beginning of the $i+1$th cycle, then the throughput of the first frame with L slots in the $i+1$th cycle of EFSA is

$$P_s(\alpha,\beta) = \frac{(L-1)^{\alpha-1}\alpha\beta + (L-1)^{\alpha}(L-\beta)}{L^{\alpha+1}} \tag{13}$$

where $\alpha > 0$, $L > \beta > 0$.

Proof: Substituting $E(c_r) = Lp_r$, $r = 0, 1, \kappa$, $t_0 = t_\kappa = t_1$ and (5) into (14), we will have (13). □

B. Analysis of Arriving Tags

Lemma 1: When the number of arriving tag $\alpha = L-1$, the throughput in the first frame of the $i+1$th cycle is equal to the maximum throughput of dynamic framed aloha [6], [7], [9]–[13], i.e.,

$$P_s(L-1,\beta) = 0.368 \tag{14}$$

no matter what the number of leaving tags β is.

Proof: Substituting $\alpha = L-1$ into (13), we have

$$\lim_{L\to+\infty}\left[P_s(L-1,\beta) = \left(1-\frac{1}{L}\right)^{L-1}\right] = \frac{1}{e}. \tag{15}$$

Then, we can obtain $P_s(L-1,\beta) = 0.368$. Since the maximum throughput of frame slotted aloha is 0.368, lemma 1 can be yielded. □

Lemma 2: When the number of arriving tags $\alpha_1 > L-1$, the first frame's throughput increases with the number of leaving tags β; when the number of arriving tags $\alpha_2 < L-1$, the first frame's throughput decreases with the number of leaving tags β.

Proof: From (13), let $\partial P_s/\partial\beta = 0$ and we have $\alpha = L-1$. If $\alpha_1 > L-1$, then $\partial P_s/\partial\beta > 0$. Hence, $P_s(\alpha_1,\beta)$ is a monotonically increasing function with respect to β. Likewise, if $\alpha_2 < L-1$, then $P_s(\alpha_2,\beta)$ is a monotonically increasing function with respect to β □

Theorem 2: When the number of arriving tags $\alpha_1 > L-1$, the first frame's throughput is not larger than the maximum throughput of dynamic framed aloha 0.368, i.e.,

$$P_s(\alpha_1,\beta) \le 0.368. \tag{16}$$

Proof: From Lemma2, $P_s(\alpha_1,\beta)$ is a monotonically increasing function of β when $\alpha_1 > L-1$. Since $0 \le \beta \le L$, then we have

$$P_s(\alpha_1,\beta) \le P_s(\alpha_1,L) = \left(1-\frac{1}{L}\right)^{\alpha_1-1}\frac{\alpha_1}{L}. \tag{17}$$

From $\alpha_1 > L-1$, we can obtain $\alpha_1 \ge L$. Since $P_s(\alpha_1,L)$ is monotonically decreasing function with respect to α_1 when $\alpha_1 \ge L$, we have

$$P_s(\alpha_1,L) \le P_s(L,L) = \left(1-\frac{1}{L}\right)^{L-1} \approx 0.368. \tag{18}$$

From (17) and (18), (16) can be yielded. □

C. Maximum Throughput

Lemma 3: When the number of leaving tags $\beta_1 > L/2$, the first frame's throughput achieve a maximum at the arriving tags number $\alpha = 2L - L^2/\beta$; when the number of leaving tags $\beta_2 < L/2$, the first frame's throughput achieve a maximum at the arriving tags number $\alpha = 0$.

Proof: From (13), let $\partial P_s/\partial\alpha = 0$ and we have

$$\lim_{L\to\infty}\left[\alpha = -\frac{1}{\ln\left(1-\frac{1}{L}\right)} - \frac{(L-1)(L-\beta)}{\beta}\right] = 2L - \frac{L^2}{\beta}. \tag{19}$$

If $\beta_2 < L/2$, substituting it into (19) can obtain $\alpha < 0$. Since the number of arriving tags cannot be less than 0, the throughput cannot achieve a maximum at $\alpha = 2L - L^2/\beta$ when $\beta_2 < L/2$. From (19), we know that $P_s(\alpha,\beta_2)$ is a monotonically increasing function with respect to α when $\beta_2 < L/2$ and $\alpha > 0$. Therefore, $P_s(\alpha,\beta_2)$ achieves a maximum at $\alpha = 0$, i.e.,

$$P_s(\alpha,\beta_2) \le P_s(0,\beta_2). \tag{20}$$

On the other hand, when $\beta_1 > L/2$, we have $\alpha > 0$ from (19). Thus,

$$P_s(\alpha,\beta_1) \le P_s\left(2L - \frac{L^2}{\beta_1,\beta_1}\right). \tag{21}$$

From (20) and (21), Lemma 3 can be given. □

Theorem 3: When both the numbers of arriving tags and leaving tags are 0, $\alpha = \beta = 0$, the first frame's throughput achievers a maximum 1.

Proof: From Theorem 2, when $\alpha_1 > L-1$, we have

$$P_s(\alpha_1,\beta) \le 0.368. \tag{22}$$

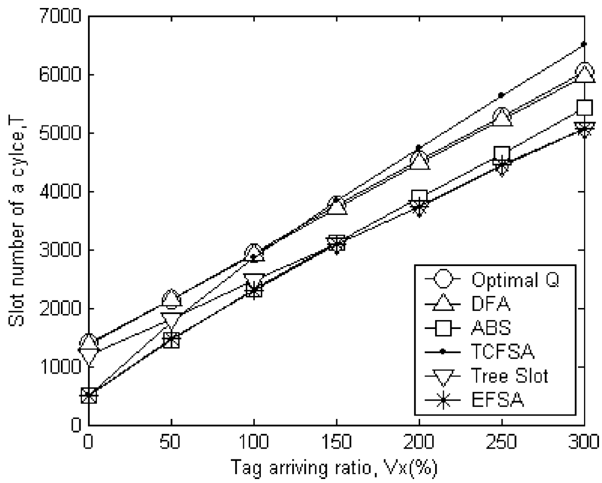

Fig. 5. Simulation result of slot number against V_x in (27) for protocol.

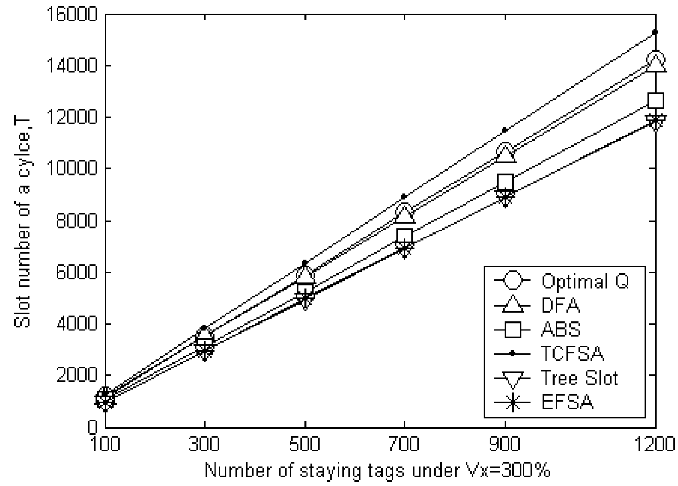

Fig. 6. Simulation result of slot number for protocol under $V_x = 300\%$ in (27).

From Lemma 2, when $\alpha_2 < L - 1$, we have

$$P_s(\alpha_2, \beta) \leq P_s(\alpha_2, 0). \tag{23}$$

From Lemma 3, we have

$$P_s(\alpha_2, 0) \leq P_s(0, 0). \tag{24}$$

Substituting $\alpha = \beta = 0$ into (13) can yield

$$P_s(0, 0) = 1. \tag{25}$$

From Lemma 1 and (22)–(25), we have

$$P_s(\alpha, \beta) \leq P_s(0, 0) = 1 \tag{26}$$

where $\alpha \geq 0$ and $0 \leq \beta \leq L$. □

V. Performance Evaluation

We evaluate the performance of the EFSA protocol by computer simulation. We individually perform each simulation 500 times, and average 500 simulation results into the final result.

A. Identification Time

In this subsection, we evaluate identification time of the EFSA protocol when a reader reidentifies a moving tag set. The first frame length was set to $L = 512$ slots in our simulations of this subsection. Fig. 5 presents identifying slot number T against V_x in a read cycle, for ABS [20]–[22], TCFSA [25], tree slot [23], dynamic framed aloha [6], [7], [9]–[13], optimal Q algorithm [26], and EFSA protocol. In Fig. 5, we define the tag arriving ratio V_x as

$$V_x = \frac{Q\left(X_{i+1} - X_i\right)}{Q\left(X_i\right)} \times 100\% \tag{27}$$

where X_i is a set of tags which are inside a reader's range in the ith read cycle, and $X_i \subset X_{i+1}$. $Q(\bullet)$ is defined as cardinality of tags set, $Q(X_i) = 512$ and the slot number of a cycle T is defined as [20]–[22]

$$T = E + S + C \tag{28}$$

in which E , S and C is the number of idle slot, successful slot and collision slot in a read cycle, respectively. From Fig. 5, we can see that the V_x loss of dynamic framed aloha and optimal Q algorithm from that of the EFSA protocol is approximately 50%. This result shows that EFSA would have less identification time when reidentifying more staying tags. And, as V_x increases, the gap of identification time between EFSA and ABS is much larger when $V_x > 150\%$. This result shows that, when colliding tags increase with arriving tags, EFSA's identification time is less than ABS. In addition, as V_x increases, the gap of identification time between EFSA and tree slot protocol is much larger when $V_x < 150\%$. The reason for this result is that, EFSA would have less identification time than tree slot protocol when reidentifying more staying tags.

Fig. 6 presents identifying slot number against the number of staying tags under $V_x = 300\%$, for ABS, TCFSA, tree slot, dynamic framed aloha, optimal Q algorithm, and the EFSA protocol, respectively. It is seen from Fig. 6 that EFSA shows a better performance in reducing identification delay than ABS under $V_x = 300\%$. This result shows that ABS's binary splitting is not a better way as the number of colliding tags increase. Fig. 7 presents identifying slot number against number of staying tags under $V_x = 50\%$. It is seen from Fig. 7 that EFSA shows a better performance in reducing identification delay than tree slot, dynamic framed aloha and optimal Q algorithm under $V_x = 50\%$. This result shows that EFSA would have less identification time when reidentifying more staying tags.

ISO 18000-6C can use SELECT commands to confirm a tag presence, and can reidentify the tag. However, if there are many moving tags, i.e., both leaving tags and arriving tags, the method of ISO 18000-6C may not be optimal. A leaving tag will produce an idle slot when SELECT is used to confirm a tag presence. Apparently, the number of idle slots will increase with the number of leaving tags, and many leaving tags will result in low throughput. In EFSA protocol, arriving tags randomly select slots, and they can have chance to occupy the slots produced by the laving tags. Therefore, EFSA can reduce the number of idle slots when there are both many leaving tags and arriving tags. Fig. 8 presents identifying slot number under different V_y

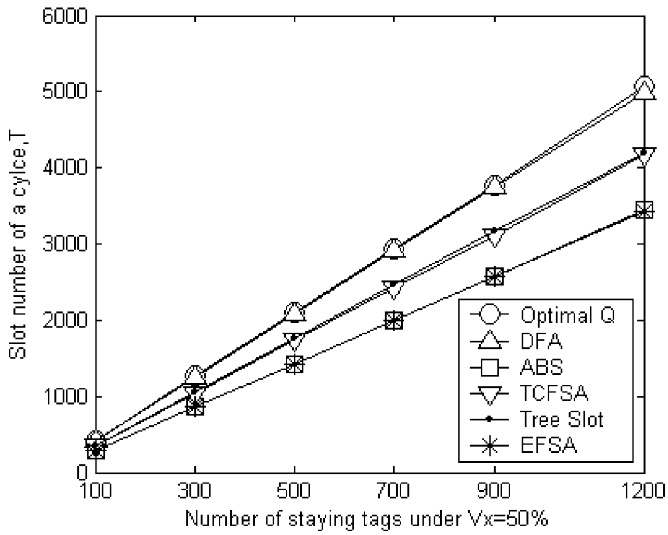

Fig. 7. Simulation result of slot number for protocol under $V_x = 50\%$ in (27).

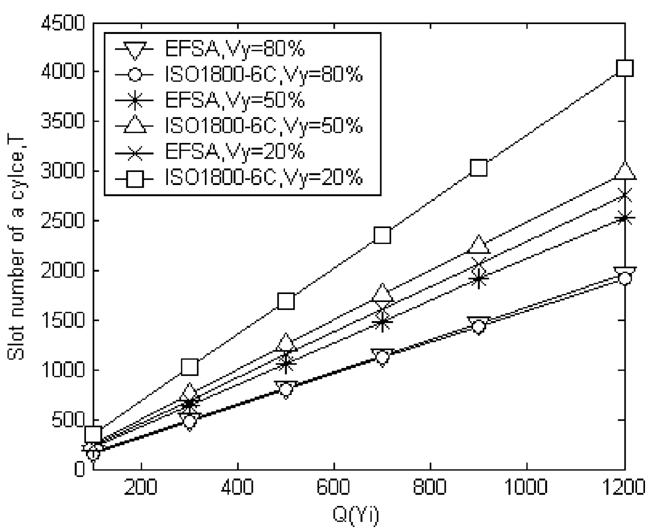

Fig. 8. Simulation result of slot number under different V_y in (29).

for ISO 18000-6C and EFSA, where V_y is defined as a ratio of staying tags and is given by

$$V_y = \frac{Q\left(Y_{i+1} \cap Y_i\right)}{Q\left(Y_i\right)} \times 100\% \tag{29}$$

and Y_i is a set of tags which are inside a reader's range in the ith read cycle. It is seen from Fig. 8 that, EFSA shows a better performance in reducing identification delay than ISO 18000-6C when $V_y = 50\%$ and 20%.

B. Tag Estimate

In this subsection, we evaluate tag quantity estimate in the EFSA protocol. Fig. 9 gives estimate error in the first frame of the ith cycle, for lower bound estimate [10]–[13], Schoute estimate [9], [12], [13], idle slot estimate [12], Vogt estimate [11], [12], MAP estimate [7], and the proposed estimate in this paper, where the estimate error ε is defined as

$$\varepsilon = \frac{\hat{n} - n}{n} \times 100\%. \tag{30}$$

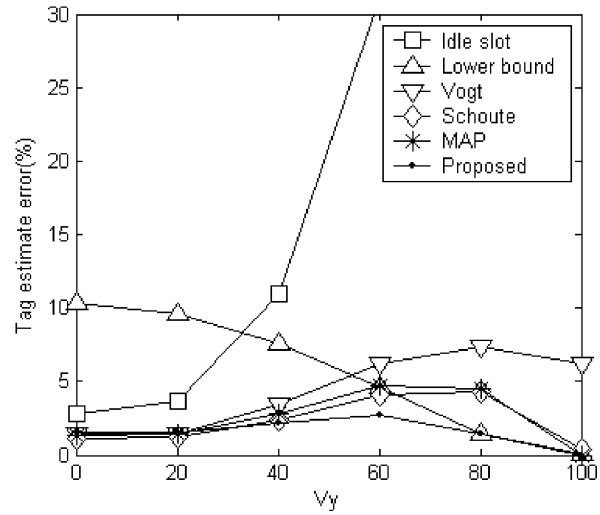

Fig. 9. Simulation result of estimate error against V_y in (28).

$Q\left(Y_i\right) = Q\left(Y_{i+1}\right) = 512$, frame length $L = 512$, tag number $n = 512$, maximum number of tags that the RFID system can read $N = 600$ and maximum number of arriving tags that the RFID system can read $\mathrm{A} = 600$. From Fig. 9, when $40\% < V_y < 80\%$, the proposed estimate error is less than the other estimates; when $80 < V_y < 100\%$, lower bound estimate error is close to the proposed estimate, while the other estimates errors are larger than the proposed estimate; when $V_y < 40\%$, MAP, Vogt ,and Schoute estimate errors are close to the proposed estimate. These results signify that the proposed estimate does not have a worse performance than the conventional estimates under any value of V_y.

VI. Conclusion

This paper proposes an efficient framed aloha protocol for RFID tag anticollision. The proposed protocol can retain information obtained from the last cycle of tag identification, and hence skip many collisions to quickly reidentify the tags. In many RFID applications where readers may repeatedly identify tags, such as object tracking and locating, the proposed protocols can reduce time of reidentifying tags. In addition, if there is a collision slot in the EFSA protocol, colliding tags in the collision slot will be resolved by Q-ary splitting where Q is equal to the estimated number of colliding tags. The Q-ary splitting method will have less identification time than binary splitting especially when the colliding tags increase. In the proposed protocol, this paper also proposes a novel tag quantity estimate, which have less estimated error and is suitable for EFSA.

References

[1] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*, 2nd ed. Oxford, U.K.: Wiley, 2003.
[2] D. H. Shin, P. L. Sun, D. C. Yen, and S. M. Huang, "Taxonomy and survey of RFID anti-collision protocols," *Comput. Commun.*, vol. 29, no. 11, pp. 2150–2166, 2006.
[3] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory*, vol. 25, no. 5, pp. 505–515, Sep. 1979.
[4] D. R. Hush and C. Wood, "Analysis of tree algorithm for RFID arbitration," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, U.K., 1998, p. 107.

[5] C. Law, K. Lee, and K. S. Siu, "Efficient memoryless protocol for tag identification," in *Proc. 4th ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Comput. Commun.*, Boston, MA, 2000, pp. 75–84.
[6] F. C. Schoute, "Dynamic frame length aloha," *IEEE Trans Commun.*, vol. Com-31, no. 4, pp. 565–568, Apr. 1983.
[7] W. T. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 9–15, Jan. 2009.
[8] C. Floerkemeier, "Bayesian transmission strategy for framed ALOHA based RFID protocols," in *Proc. IEEE Int. Conf. RFID*, 2007, pp. 228–235.
[9] J. R. Cha and J. H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," in *Proc. Int. Conf. Parallel and Distrib. Syst. Comput.*, Japan, 2005, pp. 63–67.
[10] H. Vogt, "Multiple object identification with passive RFID tags," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Hammamet, Tunisia, 2002, pp. 1–6.
[11] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. Int. Conf. Pervasive Comput.*, Zurich, Switzerland, 2002, pp. 98–113.
[12] G. Khandelwal, A. Yener, K. Lee, and S. Serbetli, "ASAP : A MAC protocol for dense and time constrained RFID systems," in *Proc. IEEE Int. Conf. Commun.*, Istanbul, Turkey, 2006, pp. 4028–4033.
[13] S. R. Lee, S. D. Joo, and C. W. Lee, "An enhanced dynamic framed ALOHA algorithm for RFID tag identification," in *Proc. Int. Conf. Mobile and Ubiquitous Systems: Networking and Services*, 2005, pp. 166–174.
[14] 13.56 MHz ISM band class 1 radio frequency identification tag interface specification: Recommended Standard, Auto-ID Center, Tech. Rep., 2003, 1.0.0.
[15] *Information Technology—Radio Frequency Identification (RFID) for Item Management—Part 6: Parameters for Air Interface Communications at 860 MHz to 960 MHz*, International Standard ISO/IEC 18000-6, 2004.
[16] *Information Technology—Radio Frequency Identification (RFID) for Item Management—Part 6: Parameters for Air Interface Communications at 860 MHz to 960 MHz, Amendment 1: Extension With Type C and Update of Types A and B*, International Standard ISO/IEC 18000-6, 2006.
[17] EPC Radio-Frequency Identification Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz ver. 1.1.0 Draft1, EPCglobal Inc., 2005.
[18] R. Angeles, "RFID technologies: Supply-chain applications and implementation issues," *Inf. Syst. Manage.*, vol. 22, no. 1, pp. 51–65, 2005.
[19] Y. C. Lai and C. C. Lin, "A pair-resolution blocking algorithm on adaptive binary splitting for RFID tag identification," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 432–434, Jun. 2008.
[20] M. Jihoon, L. Wonjun, and J. Srivastava, "Adaptive binary splitting for efficient RFID tag anti-collision," *IEEE Commun. Lett.*, vol. 10, no. 3, pp. 144–146, Mar. 2006.
[21] J. Myung, W. Lee, and T. K. Shih, "An adaptive memoryless protocol for RFID tag collision arbitration," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 1096–1101, Oct. 2006.
[22] J. Myung, W. Lee, J. Srivastava, and T. K. Shih, "Tag-splitting: Adaptive collision arbitration protocols for RFID tag identification," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 6, pp. 763–775, Jun. 2007.
[23] A. B. Maurizio, L. Francesca, and M. Francesca, "Instant collision resolution for tag identification in RFID networks," *Ad Hoc Networks*, vol. 5, no. 8, pp. 1220–1232, 2007.
[24] G. Maselli, C. Petrioli, and C. Vicari, "Dynamic tag estimation for optimizing tree slotted aloha in RFID networks," in *Proc. ACM MSWIM 2008*, Italy, 2008, pp. 315–322.
[25] X. Deng, M. Rong, T. Liu, Y. Yuan, and D. Yu, "Tag count frame slotted aloha: A novel anti-collision protocol in RFID systems," in *Proc. 67th IEEE Veh. Technol. Conf.*, Shanghai, China, 2008, pp. 2666–2670.
[26] Y. Maguire and R. Pappu, "An optimal Q-algorithm for the ISO 18000-6C RFID protocol," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 16–24, Feb. 2009.
[27] H. Wu and Y. Zeng, "Bayesian tag estimate and optimal frame length for anti-collision aloha RFID system," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 4, pp. 963–969, Oct. 2010.

Haifeng Wu received the M.S. degree in electrical engineering from Yunnan University, Kunming, China, in 2004, and the Ph.D. degree in electrical engineering from Sun Yat-Sen University, Guangzhou, China, in 2007.

He is currently an Assistant Professor at the Department of Information Engineering, Yunnan University of Nationalities. Prior to that, he was a Postdoctoral Scholar at the Kunchuan Institute of Technology from 2007 to 2009. His research interests include cooperative sensor networks, mobile communications, and RF engineering.

Yu Zeng received the M.S. degree in electrical engineering from Yunnan University, Kunming, China, in 2006.

She is currently an Assistant Professor at the Department of Information Engineering, Yunnan University of Nationalities. Prior to that, she was an Electrical Engineer at the Kunming Institute of Physics from 2006 to 2009. Her research interests include wireless network, mobile communications.