Passive RFID Tag Anticollision Algorithm for Capture Effect

Haifeng Wu and Yu Zeng

Abstract—**Capture effect is very common in passive radio frequency identification systems. Even though multiple tags transmit their signals to a reader simultaneously, one of the tags will be successfully identified due to capture effect. However, capture effect may hide some tags. Many existing algorithms that assume no capture effect will miss the hidden tags. In this paper, we extend an existing algorithm to the capture effect environment and propose a novel tag anticollision algorithm to enhance the efficiency of tag identification. The proposed algorithm adopts an allocation strategy to reduce collision between the hidden tags by capture effect and the other tags. The strategy lets the collided tags resolved in a current cycle immediately. On the other hand, the hidden tags will enter the next cycle. Thus, the resolved tags will not collide with the hidden tags in the next cycle, and the collision will be reduced. The experiment results show that the proposed protocol's identification efficiency outperforms other existing protocols when the occurrence probability of capture effect varies from 0.1 to 0.6.**

Index Terms—**RFID, tag anti-collision, capture effect.**

I. INTRODUCTION

INTERNET of Things (IoT) has been considered as the third wave of world information industry, after waves of computer and internet. IoT needs to be based on widely used Radio Frequency Identification (RFID) tags because plenty of objects information is collected into network only by the tags [1]. In a passive RFID system, a reader broadcasts a request command to all tags in its range. And then, the tags select their response slots to transmit their signals [2]. For a given time slot, there are only three possible types: no tag responses, only one tag response and more than two tag responses, respectively. Many existing anti-collision algorithms [3]–[17] assume that the three possible types will result in an idle slot, a successful slot and a collision slot, respectively. However, the assumption sometimes may be invalid. A near-far problem [18], [19] in the passive system is very common. A tag that is close will transmit a much stronger signal than one that is far away. The reader is likely to decode only the near tag, instead of the far tag. Therefore, more than two tags responses will not necessarily result in a collision slot. If there is a tag decoded in such a slot, the phenomenon may be called capture effect. Other un-decoded tags in the slot are called hidden tags [18].

For the problem of capture effect, there have been various solutions proposed in [18]–[21]. However, the previous solutions are not very efficient for the following reasons. First, when capture effect occurs, a hidden tag will re-select another slot for being re-identified. Thus, the identification efficiency with capture effect would depend on how the hidden tags select slots. Unfortunately, the previous solutions do not reasonably allocate slots to the hidden tags. The unreasonable allocation produces excessive collision.

The second reason is to set a frame length. In existing algorithms for passive RFID tag anti-collision, aloha-based algorithms are very popular [4]–[10], [20]–[24]. The idea of aloha-based algorithms is to divide access time of tags into a number of slots. Each tag responds at a randomly selecting slot. An optimal value of identification efficiency can be achieved only when the number of slots in a frame, i.e. a frame length is set to the number of tags [5]–[8]. If capture effect occurs, however, such a frame length can not guarantee the optimal efficiency. Under capture effect environment, an optimal frame length requires not only the information of the number of tags but also the occurrence probability of capture effect [25]. Generally, the probability of capture effect is related to the identification environment, such as the distance between tags and a reader and the strength of the tags' backscattering signal [5], [18]. Due to the variance of the environment, the probability of capture effect is not easy to be obtained.

In [8] and [16], we propose a novel aloha-based algorithm and a novel tree-based algorithm to enhance the efficiency of RFID tag identification, respectively. However, the two works cannot be applied to capture effect environment because the hidden tags will be missed. In [25], we derive an optimal frame length under capture effect. However, the work is based on an ideal theoretical model of capture effect. It is difficult to be applied in practice. This paper extends our previous work in [16] to the capture effect environment and proposes a novel algorithm, adaptive BTSA algorithm for capture effect (ABTSA). The proposed algorithm firstly utilizes the sequences generated by 16-bit pseudo-random numbers (RN16) in EPC C1 Gen2 [21] to detect the hidden tags. Then, the proposed algorithm adopts an allocation strategy to reduce collisions between the hidden tags and the other tags.

Manuscript received April 26, 2014; revised July 5, 2014; accepted July 6, 2014. Date of publication July 15, 2014; date of current version November 6, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61262091, in part by the Applied and Basic Research Foundation of Yunnan Province under Grant 2011FB083, in part by the Major Special Project of Scientific Research Foundation, Yunnan Provincial Department of Education under Grant ZD2011009, in part by the Innovative Research Team, Yunnan University of Nationalities, Kunming, China, and in part by the 17th Batches of Young and Middle-aged Leaders in Academic and Technical Reserved Talents Project of Yunnan Province. The associate editor coordinating the review of this paper and approving it for publication was Prof. Kiseon Kim.

The authors are with the 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/JSEN.2014.2339653

Finally, an adaptive technique is used to adjust the frame length close to an optimal size.

Our contributions are summarized as follows.

- We extend a previous algorithm to the capture effect environment. The performance of the proposed algorithm's identification efficiency outperforms the existing algorithm when the capture effect occurs.
- Under capture effect, we propose an allocation strategy to reduce collision between the hidden tags and the other tags. The strategy lets the collided tags resolved in a current cycle immediately while the hidden tags enter the next cycle. Thus, the resolved tags will not collide with the hidden tags and the collision will be reduced.

II. Capture Effect Problem

For capture effect, at least two problems should be considered. The first problem is how to detect the hidden tags by capture effect. For a slot with capture effect, the hidden tags may be missed because a reader could think that the slot is successful. Due to capture effect, a reader can successfully decode one of several responding tags' signals. If the reader returns the decode signal to all of the responding tags, any tags which find that the decoded signal does not match their own should be hidden tags [18], [23]. Since the method successfully solve the problem of detecting the hidden tags, this paper will not focus more attention on it.

The second problem that we should consider is about the identification efficiency. Here, we defined the efficiency as a ratio between the duration of successful identification slots and the total identification duration [5]–[10], seen also in Section V. Since the total duration is the sum of the durations of successful slots, idle slots and collision slots, fewer idle slots and collision slots will produce higher slot efficiency. Generally, the efficiency with capture effect is some different from that without capture effect for the following reasons. First, capture effect changes an assumed collision slot into a successful one. Thus, the efficiency with capture effect will be higher than that without capture effect. Second, the hidden tags may be collided with the other unidentified tags. When capture effect occurs, a hidden tag can not be identified. It will re-select another slot for being re-identified. Besides, the other unidentified tags in the current cycle, such as collided tags also need to re-select slots. It is very likely that the unidentified tags may select the same slot that the hidden tag selects. In this case, collision will happen. Surely, if the unidentified tags and the hidden tags select different slots, collision may be avoided and the efficiency will be enhanced. Therefore, the efficiency with capture effect will be related to how the hidden tags and the unidentified tags select slots.

III. Related Work

General binary tree (GBT) protocol [18] is a BT-based protocol [11], [12] to cope with capture effect. In BT, the collided slot will be continuously split into two slots until each slot is successful or idle. Since capture effect will make a slot having more than two tags into a successful slot, BT will not split the slot into two sub-slots. Consequently, the hidden tags in the slot will be missed. To solve the problem, GBT protocol let a reader return a decoded ID to all responding tags. A hidden tag will find that the ID does not match its own. Then, the hidden tag in a current BT cycle will enter the next one. From this, all tags will be divided into successfully identified ones and hidden ones after the first cycle completing. In GBT, however, all of hidden tags will enter a single slot in the second cycle and then will be split. If the single slot has too many hidden tags, this may produce excessive collision among the hidden tags.

EPC C1 G2 [21] is a very popular standard for ultra-high frequency (UHF) RFID systems. The standard utilizes Q algorithm to resolve the tag collision. Q algorithm is a framed slotted aloha-based algorithm [4]–[8], which randomizes access time of tags to reduce collisions. The access time constitutes several frames and the frame also constitutes several slots. As we know, there must be too many idle slots if a frame length is much larger than the number of tags; on the contrary, there must be too many collision slots if the length is much smaller than the number. In Q algorithm, when a frame has excessive idle slots, a reader will end the frame prematurely and start another shorter frame. On the other hand, a reader will start another longer frame. The purpose of the adjustment is to obtain a reasonable length for the number of tags, and thus reduce excessive idle or collision slots. Q algorithm considers the influence of capture effect on the tag identification. It could adopt RN16 to detect hidden tags (the details can be seen in Section IV-A). Whenever a frame is ended, any hidden tags will enter the next frame. Therefore, the hidden tags will not be missed. In Q algorithm, however, not only the hidden tags but also the other unidentified tags, e.g. the tags collided in a previous frame will enter the next frame. Since the two kinds of tags may select the same slots, excessive collision between them will also happen.

Optimal Q algorithm [19] is also a framed slotted aloha-based algorithm under capture effect. Like Q algorithm, Optimal Q lets the hidden tags enter the next frame to be identified again. However, optimal Q algorithm adopts the idea of a dynamic frame length ALOHA algorithm. That is, each frame's length is equal to the number of tags. When capture effect happens, such a frame could not guarantee optimal efficiency. An optimal length should be related not only to the number of tags, but also to the occurrence probability of capture effect (details seen in (13) and (14), Section V). Therefore, Optimal Q algorithm does not work better under capture effect environment.

In [25] and [26], two capture-aware estimation algorithm are proposed, respectively. The algorithms derive an optimal frame length about the number of tags and the probability of capture effect for the dynamic frame length ALOHA protocol. Based on the frame length, an optimal value of the identification efficiency would be achieved under capture effect. However, the algorithms assume that the probability of capture effect is invariant. The model of the probability is ideal and is not easy to be applied in real industrial environment.

Previously, we also propose a tag anti-collision algorithm, adaptive BTSA in [16]. The advantage of the algorithm is that, it does not need the estimation of the number of tags, and its

TABLE I
MISSED RATE (%) UNDER CAPTURE EFFECT

OCCURRENCE PROBABILITY OF CAPTURE EFFECT	0	0.2	0.4	0.6	0.8	1.0
ADAPTIVE BTSA	0	25	47	62	83	97
BSTSA	0	29	51	69	86	99
MODIFIED Q	0	39	68	85	94	97

Missed Rate is defined as a ratio between the number of missed tags by a reader and that of total tags during the reader's range.

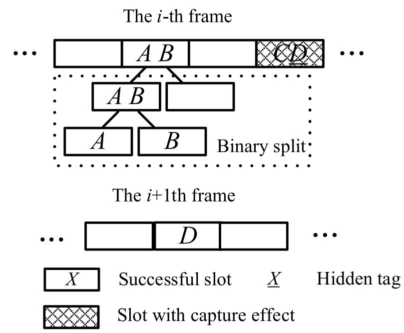


Fig. 1. An example of allocation strategy for capture effect.

efficiency is not affected by the variance of the number of tags. However, like many existing algorithms such as BSTSA [17] and Modified Q [27], adaptive BTSA does not consider capture effect. The hidden tags will be missed in the algorithm when capture effect occurs. Table I gives several existing algorithms' missed rate, R which is defined by

$$R = \frac{n_{\text{missed}}}{n} \times 100\%$$

where n is the number of tags in a reader's range and set to 500, and n_{missed} is the number tags that the reader have missed. Here, the missed tags are in fact the hidden tags by capure effect. It is seen from Table I that, R increase with the occurrence probability of capture effect. For this reason, adaptive BTSA, BSTSA and Modified Q are hard to be applied to industry and commerce. For example, an RFID reader running adaptive BTSA at a supermarket checkout counter misses some tags on products due to capture effect. Customers will not pay for the missed products. This will bring commercial losses to the supermarket.

In this paper, we extend adaptive BTSA to capture effect environment and propose ABTSA algorithm. The proposed algorithm can detect the hidden tags, and thus cannot miss any hidden tags. In order to enhance the identification efficiency with capture effect, furthermore, ABTSA circumvents how to reasonably allocate slots to the hidden tags and the other unidentified tags.

IV. ABTSA ALGORITHM FOR TAG ANTI-COLLISION

A. RN16 for Anti-Collision Under Capture Effect

RN16 is firstly proposed for anti-collision in EPC C1 Gen2 [21]. In this paper, we adopt RN16 to resolve collision under capture effect environment. RN16 is randomly generated and should be fewer bits in length than an ID sequence. Before ID, tags firstly transmit the RN16: (i) if collided, the tags do not need to transmit the longer ID. (ii) if idle, a reader will wait for only a duration of the shorter RN16. (iii) if successful, a reader will return a decoded RN16 and an acknowledge (ACK) command. A tag whose RN16 matches the decoded RN16 will transmit its ID to the reader. The total transmitted bits in this case are still fewer than those of the ID and ACK.

Using RN16, ABTSA can successfully detect the hidden tags. When multiple tags transmit their RN16 to a reader simultaneously, the reader can decode one tag's RN16 if capture effect occurs. According to the case (iii) above, the reader will return the decoded RN16 to all responding tags. Thus, the responding tags can check whether the RN16 match its own or not and then know whether they are hidden or not. Instead of RN16, GBT protocol [18] returns an ACK with a decoded ID to detect the hidden tags. Since the RN16 is shorter than ID, the proposed method expends less time than GBT.

B. Allocation Strategy for Capture Effect

In ABTSA algorithm, the allocation strategy for capture effect involves two parts: (i) the allocation of the hidden tags and (ii) the allocation of the collided tags. In the former, a reader determines when to identify the hidden tags. Like conventional framed slotted aloha [4]–[8], ABTSA configures a read process with some continuous frames consisting of slots. A tag will select a random slot in a frame to transmit its RN16. If the tag is hidden, the tag will wait until the current frame completed. Only when the next frame starts, the hidden tag can re-select a slot to transmit its RN16. This will cancel the collision between the hidden tag and the other unidentified tags in the current frame.

In the second allocation part, the reader will not let the collided tags enter the next frame. If collision occurs, the collided tags will be resolved by binary split immediately. Therefore, the resolved tags will not collide with the hidden tags entering the next frame.

Fig. 1 gives an example of the allocation strategy. In the ith frame, tags A, B, C and D select their slots to transmit their RN16, respectively. If A's RN16 is collided with B, they are resolved by binary split immediately. D is hidden by capture effect and will enter the $i+1$th frame. Since A and B are successfully resolved and will not enter the $i+1$th frame, it will not be likely that D is collided with A or B in the $i+1$th frame.

C. Adaptive Scheme

The adaptive scheme in ABTSA protocol consists of an adaptive adjustment of frame length and an adaptive tag location. The former can slot-by-slot adjust frame length and keep the frame length proportional to the number of tags. The latter can let each of tags obtain the knowledge of location, i.e. an order number of slot. Next, we will discuss them, respectively.

1) Adaptive Adjustment of Frame Length: The adaptive adjustment of frame length is based on Q algorithm. When there are excessive collision slots in a frame, a reader will end the frame prematurely and start a new frame with larger length. On the contrary, a reader will start a new frame with less length. The algorithm can be realized by parameters Q, Q_{fp} and a step c. The value of Q is round (Q_{fp}), where Q_{fp} is a floating representation of Q. That is, a reader rounds Q_{fp} to an integer value and substitutes this integer value for Q. A frame length is $L = 2^Q$, where $Q = Q_0$, and Q_0 is an initial value of Q. Firstly, a reader will transmit a command with Q. Then, a tag's counter will select a random integer number from 0 to 2^Q-1, and decrease its counter by 1 after each slot ending. When a tag's counter is 0, the tag will transmit its RN16 to the reader. For a given time slot, there are only three possible cases: an idle, a successful and a collision slot, respectively. According to the kth slot type, $slot_k$, a reader will adjust Q_{fp} by

$$Q_{fp}(k+1) = Q_{fp}(k) + cg(k) \tag{1}$$

where generally $0.1 \le c \le 0.5$, and c selects a small value when Q is large, and a larger value when Q is small [21]. $g(k)$ is given by

$$g(k) = \begin{cases} 1, & \text{if } slot_k = C \\ -1, & \text{if } slot_k = I \\ 0, & \text{if } slot_k = S \end{cases} \tag{2}$$

where C, I and S denotes that the kth slot, $slot_k$ is collision, idle and successful, respectively. A reader starts a new frame by judging whether round $(Q_{fp}) = Q$ or not. If round (Q_{fp}) is not equal to Q, the reader will start a new frame, whose frame length is set by a new value of Q.

The main difference between the proposed adaptive adjustment and Q algorithm is the binary split discussed in Section IV-B. When round $(Q_{fp}) = Q$, collided tags will be resolved by binary split immediately. The other unidentified tags will wait until the collided tags are successfully resolved. Thus, the resolved tags will not collide with the other tags entering the next frame.

2) Adaptive Tag Location: In the adaptive tag location, tags firstly transmit their RN16 to a reader, and then the reader feeds back the outcomes of the received RN16, i.e. a collision, a successful or an idle slot to the tags. According to the feedback, each of the tags can use the adaptive algorithm to obtain the knowledge of location, i.e. which slot to select. The knowledge can be stored in a tag's counter. Initially, a counter's value is a number from 0 to the frame length L-1. When a tag's counter is changed into 0, the tag will transmit its RN16. Suppose that a tag's counter value in the kth slot is $counter(k)$. Then, the adaptive tag location can be shown as

$$counter(k+1) = counter(k) + g'(k) \tag{3}$$

where the function $g'(k)$ is as follows. When $counter(k) > 0$

$$g'(k) = \begin{cases} 1, & \text{if } slot_k = C \\ -1, & \text{if } slot_k = I \text{ or } S \end{cases} \tag{4}$$

When $counter(k) = 0$

$$g'(k) = \begin{cases} \text{a random 0 or 1}, & \text{if } \text{slot}_k = C \\ -1, & \text{if } slot_k = S \text{ and RN16 matches} \end{cases} \tag{5}$$

TABLE II

AN EXAMPLE OF OPERATION IN ADAPTIVE ALGORITHM FOR TAG LOCATION

Slot	tag: value of counter			Slot type
k=0	A: 1	B: 1	C: 3	idle
k=1	A: 0	B: 0	C: 2	collision
k=2	A: 1	B: 0	C: 3	successful
k=3	A: 0	/	C: 2	successful
k=4	/	/	C: 1	idle
k=5	/	/	C: 0	successful

"/" denotes the tag's counter does not work

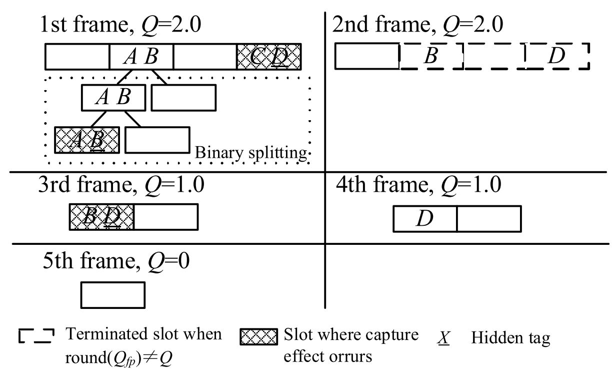


Fig. 2. An example of ABTSA protocol identifying tags in a cycle.

where $counter(k) < 0$ means that the tag has been identified and does not work.

In the anti-collision algorithm, a tag selects a slot to respond via its counter value. Only when the counter value decreases to zero, the tag will respond. The value of counter from (3)–(5) could make each tag locate a slot without collision. Table II gives an example of the adaptive location. Suppose that initial counter values of three tags A, B and C are 1, 1 and 3, respectively. After the algorithm completed, tags C, B and A will locate slots $k = 2$, 3 and 5 to be identified, respectively.

D. An Example of Algorithm

Fig. 2 depicts an example of ABTSA identifying tags in a cycle, where $c = 0.5$ and $Q_0 = 2.0$. We assume that tag A, B, C and D's initial *Counter* in the cycle is 1, 1, 3 and 3, respectively. Thus, A is collided with B in the second slot of the first frame. Since round $(Q_{fp}) = Q$, the collided tags in the slot will be resolved by binary split, where A and B select a same slot again, but A's RN16 is decoded because of capture effect. In the final slot of the first frame, since capture effect occurred again, tag C is decoded. In the first slot of the second frame, since round $(Q_{fp}) \ne Q$, the rest slots of the frame will be finished prematurely. In the third frame, Q is changed from 2.0 to 1.0, and B is decoded and D is hidden due to capture effect. In the fifth frame, all slots are idle. That indicates that there are no tags. The tag identification cycle completes. Table III gives the procedure of the four tags' *Counter* change in the example.

TABLE III
THE PROCEDURE OF COUNTER CHANGE IN THE EXAMPLE

Frame	Q	Slot	Q_{fp}	counter A	counter B	counter C	counter D	Feedback
1	2.0	1	1.5	1	1	3	3	idle
		2	2.0	0	0	2	2	collision
		3	2.0	0	0	3	3	collision
		4	2.0	0	*	4	4	successful
		5	2.0	/	*	3	3	idle
		6	2.0	/	*	2	2	idle
		7	1.5	/	*	1	1	idle
		8	1.5	/	*	0	*	successful
2	2.0	1	1.0	/	1	/	3	idle
3	1.0	1	1.0	/	0	/	*	successful
		2	0.5	/	/	/	*	idle
4	1.0	1	0.5	/	/	/	0	successful
		2	0	/	/	/	/	idle
5	0	1	0	/	/	/	/	idle

"" denotes a hidden tag whose counter waits until a current frame completed. "/" denotes a tag's counter does not work*

V. ANALYSIS OF IDENTIFICATION EFFICIENCY

In this section, we analyze the total time and the efficiency for ABTSA identifying all tags under capture effect environment. The total time is defined by a sum of the duration of idle slots, successful slots and collision slots, and the identification efficiency is defined by a ratio between the duration of successful slots and the total time.

Definition 1: Let T_{TIME} denotes the total time which ABTSA expends on identifying tags inside a reader's range. Then, the time T_{TIME} can be given by

$$T_{\text{TIME}} = T_{\text{C}} t_{\kappa} + T_{\text{I}} t_0 + T_{\text{S}} t_1 \tag{6}$$

where T_{C}, T_{I} and T_{S} denote the number of collision slots, idle slots and successful slots, respectively and t_0, t_{κ} and t_1 is an idle, a collision and a successful slot duration, respectively.

Definition 2: Let P denotes the identification efficiency when a reader identifies all tags inside a reader's range. Then, the efficiency P can be given by

$$P = \frac{E\left(T_{\text{S}} t_1\right)}{E\left(T_{\text{TIME}}\right)} \tag{7}$$

where $E\left(\bullet\right)$ denotes an expectation value.

Definition 3: Let p denotes the occurrence probability of capture effect. Then, the probability p can be given by

$$p = \frac{E\left(T_1\right)}{E\left(T_{\text{K}}\right)} \tag{8}$$

where T_{K} denotes the number of slots which having two or more than two tags and T_1 denotes the number of slots where capture effect occurs.

It is seen from Section IV that, ABTSA adopts the adaptive frame length adjustment to resolve collision under capture effect environment. Next, we will analyze the relationship between the adaptive adjusted length and the identification efficiency. Firstly, we derive an ALOHA system's identification efficiency. Given one of the time slots, the number of tags allocated in the slot is a binomial distribution with n Bernoulli experiments and $1/L$ occupied probability. The probability of finding r tags in the slot is therefore given by [5]–[8]

$$p(L, n, c_r) = \binom{n}{r} (\frac{1}{L})^r (1 - \frac{1}{L})^{n-r} \tag{9}$$

The probability applies to all L slots, thus the expected value of the number of slots with occupancy number is given by

$$E(L, n, c_r) = \binom{n}{r} (\frac{1}{L})^{r-1} (1 - \frac{1}{L})^{n-r} \tag{10}$$

When $r = 0$

$$E(L, n, c_0) = L(1 - \frac{1}{L})^n \tag{11-a}$$

When $r = 1$

$$E(L, n, c_1) = n(1 - \frac{1}{L})^{n-1} \tag{11-b}$$

When $r = \kappa, \kappa \geq 2$

$$E(L, n, c_{\kappa}) = L - L(1 - \frac{1}{L})^n - n(1 - \frac{1}{L})^{n-1} \tag{11-c}$$

From (8), (10) and (11), the expected number of idle slots, successful slots and collision slots is

$$E(T_{\text{I\_ALOHA}}) = L(1 - \frac{1}{L})^n \tag{12-a}$$

$$E(T_{\text{S\_ALOHA}}) = n(1 - \frac{1}{L})^{n-1} + p[L - L(1 - \frac{1}{L})^n - n(1 - \frac{1}{L})^{n-1}] \tag{12-b}$$

$$E(T_{\text{C\_ALOHA}}) = (1 - p)[L - L(1 - \frac{1}{L})^n - n(1 - \frac{1}{L})^{n-1}] \tag{12-c}$$

Thus, from (7) and (12), an ALOHA system's identification slot efficiency is

$$P_{\text{ALOHA}} = \frac{E(T_{\text{S\_ALOHA}}) t_1}{E(T_{\text{I\_ALOHA}}) t_0 + E(T_{\text{S\_ALOHA}}) t_1 + E(T_{\text{C\_ALOHA}}) t_{\kappa}} \tag{13}$$

From (13), a frame length which could let P_{ALOHA} arrive at a maximum value will be an optimal length. Thus, we have

$$L_{\text{OPT}} = \arg \max_{L \in \Xi} P_{\text{ALOHA}} \tag{14}$$

where $\Xi = \{1, 2, \ldots L_{\max}\}$. Here, $L_{\max}$ does not need to be selected as an infinite number because $L = L_{\max} \to +\infty$ will have $P_{\text{ALOHA}} \to 0$ from (13). Thus, Ξ should be a finite set. That means that, we can find a maximum value of P_{ALOHA} by the finite number of searching.

Observation 1: The adaptive adjusted frame length in (1-2) may come close to the optimal frame length in (14) regardless of the occurrence probability of capture effect.

In adaptive length adjustment in (1-2), since a frame length is adjusted slot by slot, the length is not a fixed value.

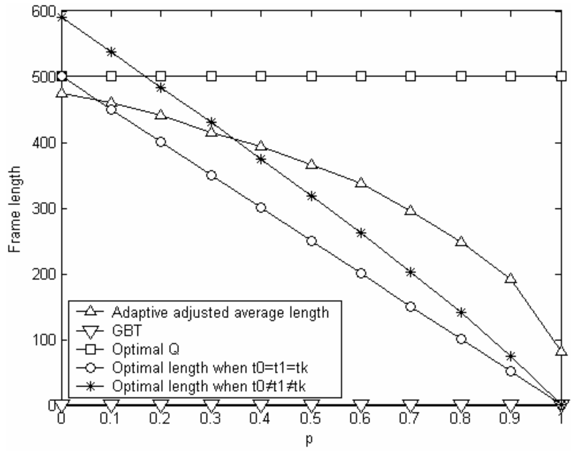


Fig. 3. Observation results: the frame length, $0 \leq p \leq 1$, $N = 500$.

Here, we use L_{AVE} to denote an average length during K slots, which is given by

$$L_{\text{AVE}} = \frac{\sum_{k=1}^{K} L(k)}{K} \tag{15}$$

Fig. 3 gives the average adaptive adjusted length when $n = 500$, $Q_0 = 4.0$, $K = 2000$, $0 \leq p \leq 1$. Since the first slot of GBT can be regarded as an ALOHA process with a frame length $L = 1$, GBT's frame length is 1 in Fig. 3. For Optimal Q algorithm, its frame length $L \approx n$ [19]. Hence, Optimal Q algorithm's length is set to 500. Fig. 3 gives two optimal lengths computed by (14) under $t_0 = t_\kappa = t_1$ and $t_0 \neq t_\kappa \neq t_1$, respectively. System parameters under $t_0 \neq t_\kappa \neq t_1$ is specified in EPC C1 Gen2, and the detailed parameters can be referenced to [24]. It is seen from Fig. 3 that, GBT's curve intersects the optimal length's curves both at $p = 1$, and will gradually deviate from the optimal length's curves as p decrease. Optimal Q's curve intersects the optimal length's curves at $p = 0$ and about $p = 0.15$, respectively and will gradually deviate from the optimal lengths as p increase. On the other hand, the average adjusted length's curve always comes close to the optimal lengths regardless of p.

Our work [25] and reference [26] both prove that if a frame length is set to L_{OPT} in (14), the identification efficiency can arrive at an optimal value. However, such an optimal frame is not easy to be guaranteed in practice. From (14), L_{OPT} requires the information of the number of tags, n and the probability of capture effect, p. Without loss of generality, we obtain the observation values of T_{C}, T_{I} and T_{S} in the ith frame, and estimate the value of n and p. The number of tags in the $i+1$th frame could be computed by n-T_{S}. However, it is difficult to obtain p because p is not invariable in the $i+1$th frame. Generally, p is related to the environment of identification, such as distance between tags and a reader and strength of the tags' backscattering signal [5], [18]. Since the environment will vary during each frame, it is not easy to obtain the exact value of p. On the contrary, the adaptive adjusted length does not require the information of p. It will come close to the optimal length regardless of p. Therefore, the adaptive adjusted length may achieve better performance of identification efficiency.

VI. Experiment

A. Experiment Setting

In experiments, we consider a scenario with a single reader and a set of passive tags that enter the reader's zone and do not leave until all the tags are successfully identified. To evaluate the performance of algorithms, we consider the following metrics:

- Identification efficiency: We measure the efficiency in a read cycle. The efficiency is defined as a ratio between durations of successful slots and total slots, seen in (7). This metric comes from reference [6], [7], [9], [10], [16], [18], and assumes that the durations of a successful, idle and collision slot are all equal, i.e. $t_0 = t_\kappa = t_1$. Fewer collision slots and idle slots can produce higher efficiency.
- Identification efficiency': We measure the efficiency in a read cycle, under EPC C1 Gen2 standard [20], [21], [24]. The efficiency is also defined in (7), but the durations of a successful, idle and collision slot are different, i.e. $t_0 \neq t_\kappa \neq t_1$. The detailed parameters in the standard EPC C1 Gen2 are referenced to [24]. Less duration of collision slots and idle slots can produce higher efficiency.
- Identification time: We measure the total time for reading tags in a read cycle, under EPC C1 Gen2 standard. The time is defined as duration where a reader successfully identifies all tags in its range, seen in (6). The detailed parameters are also referenced to [24]. Less identification time will produce fast identification.

Note that, we choose two different metrics for identification efficiency. The reason is that, we are difficult to find a uniform parameterization for all algorithms analyzed. For example, the identification efficiency in [6], [7], [9], [10], [16], and [18] is defined under $t_0 = t_\kappa = t_1$ while reference [20], [21], and [24] specified $t_0 \neq t_\kappa \neq t_1$. Therefore, this section will show our algorithm's performance under these two different metrics.

We do experiments by computer simulations. We implement algorithms by M-files in MATLAB, version 6.5 Release 13. We individually perform each experiment 500 times, and average 500 experiment results into the final results.

We divide the experiments into two parts. In the first one, we compared the performance of ABTSA with Q algorithm, GBT and Optimal Q algorithm, seen in Fig. 4-8. The parameter in each algorithm is selected as follows.

- Q algorithm: According to EPC C1 Gen2 [21], an initial value of Q, Q_0 selects 4.0. And, step c selects a small value when Q is large and selects a larger value when Q is small, generally $0.1 \leq c \leq 0.5$. Thus, step c selects values as follows. If $0 \leq Q \leq 2$, $c = 0.5$; if $Q \geq 10$, $c = 0.1$; else, $c = 1/Q$.
- GBT: Initially, all tag's counters values are 0. The hidden tags by capture effect in a current binary tree cycle will enter the next binary tree cycle [18]. A binary tree cycle means that all collision slots are split into successful or idle ones.

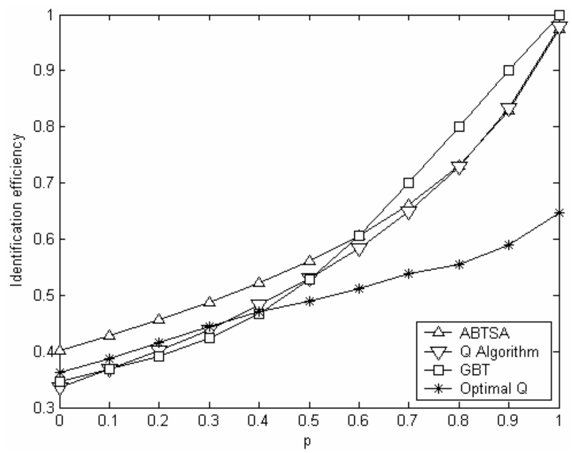


Fig. 4. Experiment result: identification efficiency, $0 \leq p \leq 1$, $n = 500$.

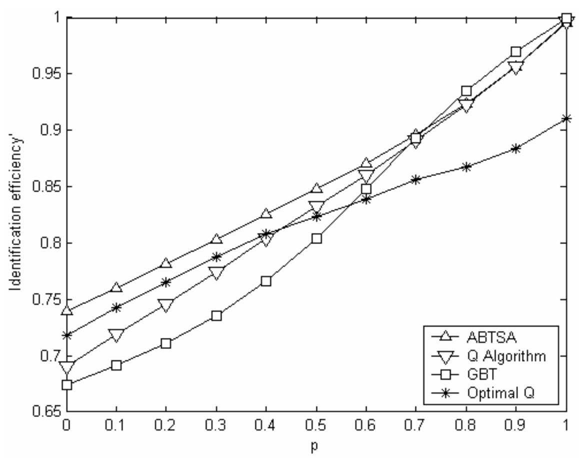


Fig. 5. Experiment result: identification efficiency', $0 \leq p \leq 1$, $n = 500$.

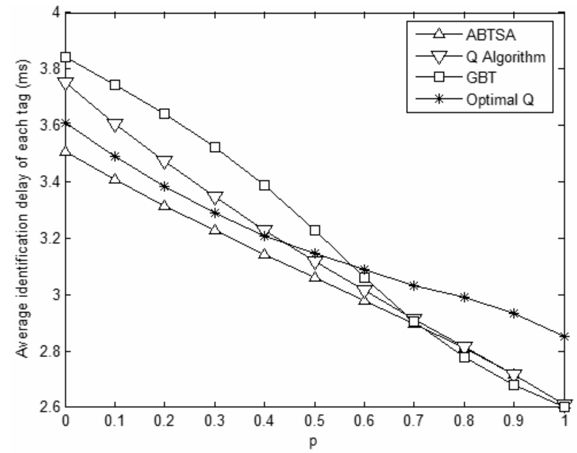


Fig. 6. Experiment result: average delay, $0 \leq p \leq 1$, $n = 500$.

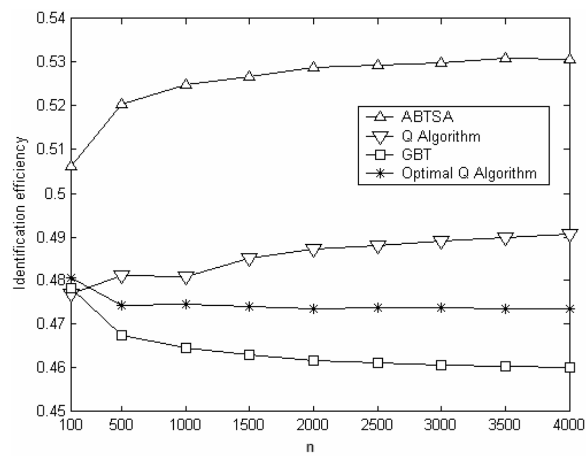


Fig. 7. Experiment result: identification efficiency, $100 \leq n \leq 4000$, $p = 0.4$.

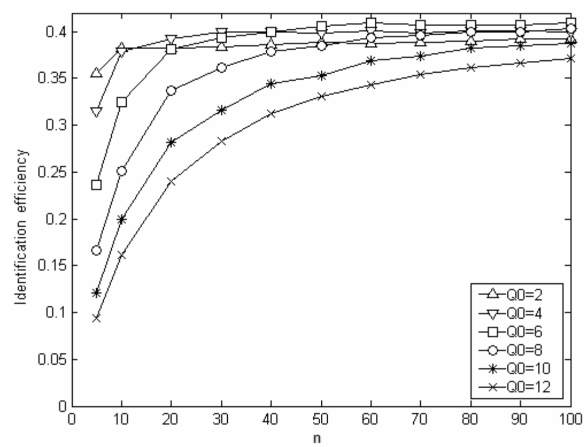


Fig. 8. Simulation result: ABTSA's identification efficiency of different Q_0 under $p = 0$.

- Optimal Q algorithm: Q_0 selects the same value as in Q algorithm above, 4.0. And, we assume that the number of tags T_n in the nth frame is known. Thus, the nth frame's Q_n is $\lfloor \log_2 T_n \rfloor$ [19].
- ABTSA: Q_0 selects the same value as in Q algorithm and Optimal Q algorithm above, 4.0. And, step c also selects the same values as in Q algorithm above. That is, if $0 \leq Q \leq 2$, $c = 0.5$; if $Q \geq 10$, $c = 0.1$; else, $c = 1/Q$.

n the second part, we present the identification efficiency of ABTSA under different Q_0, seen in Fig. 8. The purpose of the experiments is to indicate how p and Q_0 impact on the performance of ABTSA.

B. Comparison With Existing Algorithms

In this subsection, we compare the proposed algorithm with the existing algorithms. Fig. 4 presents the identification efficiency for Q algorithm, Optimal Q algorithm, ABTSA and GBT protocol when the occurrence probability of capture effect p varies from 0 to 1 and the number of tags $N = 500$. It is seen from Fig. 4 that, ABTSA's efficiency increases from 0.4 to 1 when p varies from 0 to 1. The efficiency of ABTSA outperforms that of Q algorithm, Optimal Q algorithm and GBT protocol when $0 < p < 0.6$. There are the following reasons for the result. First, ABTSA lets hidden tags select random slots in the next frame. On the contrary, GBT lets all of the hidden tags enter a single slot of the next binary tree cycle, which will cause excess collision. Second, collided tags in ABTSA can be resolved by binary split immediately and other unidentified tags will wait until the collided tags successfully resolved. However, Q algorithm and Optimal Q algorithm let both of the hidden tags and unidentified tags enter the next frame, which also may cause excess collision. Third, ABTSA adopts an adaptive technique to maintain a reasonable frame length for the number of tags and the probability of capture effect. On the other hand, Optimal Q algorithm lets the frame length set to the number of tags, which cannot guarantee the system achieves an optimal efficiency.

Besides, ABTSA still outperform the other algorithms when $p = 0$, i.e. no capture effect. The reason can be seen from Fig. 3. It is seen from Fig. 3 that, the adaptive adjusted length is closer to the optimal length than GBT when $p = 0$. Hence, ABTSA adopting the adaptive adjusted length would outperform GBT. Although Q algorithm and Optimal Q also adopt the adaptive length, they do not adopt the binary split. Thus, ABTSA adopting the binary split would outperform the two algorithms.

Of course, we also can see from Fig. 4 that, when $p > 0.6$, GBT's efficiency can surpass ABTSA. The reason can be seen from Fig. 3 and formula (14). When $p = 1$, we can obtain the optimal frame length is 1 under $t_0 = t_\kappa = t_1$. Since the first slot of GBT can be regarded as an ALOHA process with a frame length $L = 1$, its initial frame length is in fact 1. Therefore, GBT's frame length is closer to the optimal length than ABTSA when $p > 0.6$, and have higher efficiency. However, the value $p > 0.6$ means that, at least one slot will produce capture effect in every two assumed collision slots. This is not a common occurrence in RFID application [19]. In addition, Fig. 4 shows that the four algorithms' efficiency increase with p. The reason is that, larger value of p will produce more successful slots, and thus have higher efficiency.

Fig. 5 gives the identification efficiency' for the algorithms above, under EPC C1 Gen2 standard [21]. The typical system parameters in EPC are referenced to [24]. Note that instead of RN16, GBT adopts ID sequences to detect hidden tags [18]. Hence, the durations of slots in GBT should be more than that under the system parameters inEPC. In order to use a uniform metrics, we here assume that GBT also adopts RN16 to detect hidden tags. From Fig. 5, ABSTA still has the highest efficiency' when p $\leq$ 0.7. Even though p $>$ 0.7, GBT's efficiency' is only similar to ABTSA and not much higher than ABTSA.

Fig. 6 presents the average identification delay of each tag for Q algorithm, Optimal Q algorithm, ABTSA and GBT protocol, under EPC C1 Gen2 standard. In Fig. 6, the average delay of ABTSA decreases from 3.5 to 2.9ms when p varies from 0 to 0.7. ABTSA outperforms Q algorithm, Optimal Q algorithm and GBT protocol. The result is consistent with Fig. 5.

Fig. 7 presents the identification efficiency curves when $p = 0.4$ and $100 \leq N \leq 4000$. The four efficiency curves range from the highest to the lowest as follows: ABTSA, Q algorithm, Optimal Q algorithm and GBT.

C. Impact of Q_0

Fig. 8 presents the identification efficiency for ABTSA algorithm under different Q_0 when the number of tags increases from 100 to 1000. Q_0 is the initial value of Q and will decide an initial frame length's value because the initial value $L_0 = 2^{Q_0}$. The initial frame length has an effect on the identification efficiency. When the number of tags is large, a short initial frame will produce excessive collision slots; when the number is small, a long initial frame will produce excessive idle slots. The two cases both decrease the identification efficiency. The objective of Fig. 8 is to find an appropriate Q_0 which will not produce lower efficiency both under larger number or smaller number of tags.

It is seen from Fig. 8 that when the number of tags $n \leq 20$, the identification efficiency under $Q_0 \geq 6$ does not surpass that under $Q_0 \leq 4$. On the other hand, the identification efficiency under $Q_0 \leq 4$ will not be higher than that under $8 \geq Q_0 \geq 6$. Therefore, it is seen from the results that, the initial value of Q, 4.0 may be a compromise. They let the efficiency have a higher value regardless of the number of tags.

VII. Conclusion

In this paper, we propose ABTSA algorithm for RFID tag anti-collision under capture effect. The proposed algorithm mainly adopts the adaptive frame length adjustment and binary-split allocation strategy to resolve collision. From the theoretical analysis of performance, the two strategies will produce less idle slots and collision slots. Experiment results show that, when the occurrence probability of capture effect varies from 0.1 to 0.6, the identification efficiency of ABTSA will outperform the existing algorithms, Q algorithm, GBT and Optimal Q algorithm. On the other hand, when the probability is larger than 0.6, ABTSA's efficiency will not be higher than the existing algorithms. However, the occurrence probability of capture effect larger than 0.6, is not common in RFID application. If we consider typical system parameters in EPC C1 Gen2 standard, furthermore, the existing algorithms' efficiency will not be higher than ABTSA. Finally, an initial value of Q, 4.0 can make that ABTSA's efficiency have a higher value regardless of the number of tags.

References

[1] E. Welbourne *et al.*, "Building the internet of things using RFID: The RFID ecosystem experience," *IEEE Internet Comput.*, vol. 13, no. 3, pp. 48–55, May/Jun. 2009.

[2] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*. 2nd ed. London, U.K.: Wiley, 2003.

[3] 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, Jul. 2006.

[4] F. C. Schoute, "Dynamic frame length ALOHA," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 565–568, Apr. 1983.

[5] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. Int. Conf. Pervas. Comput.*, Zurich, Switzerland, Aug. 2002, pp. 98–113.

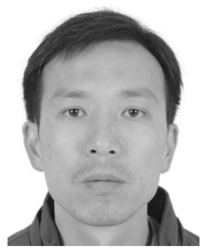
[6] B. Knerr, M. Holzer, C. Angerer, and M. Rupp, "Slot-wise maximum likelihood estimation of the tag population size in FSA protocols," *IEEE Trans. Commun.*, vol. 58, no. 2, pp. 578–585, Feb. 2010.

[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] 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.

[9] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Tree slotted aloha: A new protocol for tag identification in RFID networks," in *Proc. Int. Symp. World Wireless, Mobile Multimedia Netw.*, 2006, pp. 1–6.

[10] H. Wu and Y. Zeng, "Efficient framed slotted aloha protocol RFID tag anti-collision," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 3, pp. 581–588, Jan. 2011.
[11] Y. Cui and Y. Zhao, "Performance evaluation of a multi-branch tree algorithm in RFID," *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1356–1364, May 2010.
[12] D. R. Hush and C. Wood, "Analysis of tree algorithm for RFID arbitration," in *Proc. IEEE Int. Symp. Inform. Theory*, Cambridge, MA, USA, Aug. 1998.
[13] Z. Lei and T.-S. P. Yum, "The optimal reading strategy for EPC gen-2 RFID anti-collision systems," *IEEE Trans. Commun.*, vol. 58, no. 9, pp. 2725–2733, Sep. 2010.
[14] 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.
[15] C. Qian, H. Ngan, Y. Liu, and L. M. Ni, "Cardinality estimation for large-scale RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1441–1454, Sep. 2011.
[16] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary tree slotted ALOHA for passive RFID tag anticollision," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 19–31, Jan. 2013.
[17] T. F. La Porta, G. Maselli, and C. Petrioli, "Anticollision protocols for single-reader RFID systems: Temporal analysis and optimization," *IEEE Trans. Mobile Comput.*, vol. 10, no. 2, pp. 267–279, Feb. 2011.
[18] Y.-C. Lai and L.-Y. Hsiao, "General binary tree protocol for coping with the capture effect in RFID tag identification," *IEEE Commun. Lett.*, vol. 14, no. 3, pp. 208–210, Mar. 2010.
[19] 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, Jan. 2009.
[20] *Information Technology-Radio Frequency Identification (RFID) for Item Management—Part 6: Parameters for Air Interface Communications at 860 MHz to 960 MHz, Amendment1: Extension With Type C and Update of Types A and B*, International Standard ISO/IEC 18000-6, 2006.
[21] *EPC Radio-Frequency Identification Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz Version 1.1.0 Draft1*, EPCglobal Inc, Lawrenceville, NJ, USA, 2005.
[22] 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, Jun. 2006, pp. 4028–4033.
[23] *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, 2013.
[24] C. Wang, M. Daneshmand, K. Sohraby, and B. Li, "Performance abalysis of RFID generation-2 protocol," *IEEE Trans. Wireless Commun.*, vol. 8, no. 5, pp. 2592–2601, May 2009.
[25] X. Yang, H. Wu, Y. Zeng, and F. Gao, "Capture-aware estimation for the number of RFID tags with lower complexity," *IEEE Commun. Lett.*, vol. 17, no. 10, pp. 1873–1876, Oct. 2013.
[26] B. Li and J. Y. Wang, "Efficient anti-collision algorithm utilizing the capture effect for ISO 18000–6C RFID protocol," *IEEE Commun. Lett.*, vol. 15, no. 3, pp. 352–354, Mar. 2011.
[27] Y. Cui and Y. Zhao, "A modified Q-parameter anti-collision scheme for RFID systems," in *Proc. ICUMT*, Moscow, Russia, Oct. 2009, pp. 1–4.



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 a Professor with the Department of Information Engineering, Yunnan University of Nationalities, Kunming. Prior to that, he was a Post-Doctoral Scholar with 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 with the Department of Information Engineering, Yunnan University of Nationalities, Kunming. Prior to that, she was an Electrical Engineer with the Kunming Institute of Physics, Kunming, from 2006 to 2009. Her research interests include wireless network and mobile communications.