

A Collision Resolution Algorithm for RFID using Modified Dynamic Tree with Bayesian Tag Estimation

S. K. Wijayasekara, S. Nakpeerayuth, R. Annur, W. Srichavengsup, K. Sandrasegaran, H.-Y. Hsieh and L. Wuttisittikulkij

Abstract—In this letter, a new tree-based anti-collision protocol for RFID systems is proposed to achieve a very high tag identification efficiency. The proposed algorithm works in two phases. In the first phase, the number of competing tags are estimated through the proposed Bayesian estimation technique, while in the second phase tags are identified using our modified dynamic tree algorithm. The system efficiency is mathematically derived and verified through simulation. Numerical results show that the proposed algorithm achieves a tag identification system efficiency of 45% and time system efficiency of 78.5%, thus outperforming any existing collision resolution algorithms.

Index Terms—Bayesian estimation, collision resolution, tag, tree

I. INTRODUCTION

An anti-collision protocol is an important component in radio frequency identification (RFID) systems to resolve conflicts that arise during a tag identification process. To initiate a tag identification session, the RFID reader issues a query command and waits for replies from tags within its vicinity, called interrogation zone. As these tags do not have a priori coordination among them, their responses to the reader occur at randomly selected time slots. If exactly one tag replies within a slot, the reader sends an acknowledgement and can then acquire the unique ID from the tag and this tag is successfully identified by the reader. On the other hand, if multiple tags reply in the same slot, collision occurs and none of these tags is identified. If not resolved properly, the tag collision problem can cause substantial delay in the tag identification process. In the literature, many anti-collision protocols have been proposed to resolve the tag identification problem.

Anti-collision protocols can be broadly classified into two categories, namely aloha-based [1] and tree-based [2]– [4] protocols. Aloha-based anti-collision protocols have received much attention and one of its variants known as the Q algorithm [5] has been officially adopted in the EPCglobal Gen-2 standard [5]. The Q algorithm is based on frame slotted aloha, in which its frame size can be adjusted dynamically in response to the number of competing tags and the maximum efficiency achievable is around 34%. To further enhance the

efficiency, several efficient tag estimation techniques have been proposed such as Šolić et al. [1] and these tag estimation techniques enable the system to get closer to the theoretical maximum efficiency of 36.8% for large number of tags. Recently, [6] suggested a Dynamic Frame Slotted Aloha (DFSA)-based anti-collision algorithm, called ds-DFSA, with an efficiency about 41%, which is a significant improvement over all known DFSA algorithms. In [7], dynamic sub-frame-based maximum a posteriori probability method (DS-MAP) is introduced to estimate the backlog in a sub-frame to decide the next frame sizes to use in DFSA.

On the other hand, tree-based anti-collision protocols originally proposed by [2] can fundamentally offer higher efficiency than aloha-based counterpart, due to its divide and conquer principle. For example, the conventional binary tree algorithm (BTA) is known to achieve a 34.6% efficiency without tag estimation, while the ternary tree algorithm (TTA) offers slightly higher efficiency of 36.6% [3]. A modified tree algorithm (MTA) presented in [8] suggested that in the binary tree structure if no tag is detected in the left subgroup, tags on the right subgroup could skip the slot to avoid unnecessary collision. When the skipping mechanism is used with a biased splitting probability of 0.582 for the right subgroup, the efficiency of 38.1% is achieved. Subsequent research aimed to enhance the tree-based anti-collision protocols by utilizing the estimated number of tags. TSA (tree slotted aloha) [9] is an early effort to perform tag estimation at every level of the tree for proper splitting in each colliding slot of a slotted aloha frame. However, the attainable efficiency is approaching 38%. Later the performance of TSA was significantly enhanced to about 41.5% as proposed in the Binary Search Tree Slotted Aloha (BSTSA) protocol [10] by introducing an elegant and powerful tag estimation method called binary tree splitting. The same binary tree splitting concept for tag estimation was later adopted by [4] and their most efficient Binary Tree Slotted Aloha (BTSA) protocol achieved 42% efficiency, which this to our knowledge is the most efficient and realizable algorithm to date. Recently, a new anti-collision protocol called ISE-BS (Idle Slot Elimination based Binary Splitting) that achieved around 40.65% efficiency without any tag estimation was proposed in [11]. ISE-BS introduced a special 1-bit Q feedback which enables a complete elimination of idle slots.

In this letter, we propose a new anti-collision protocol based on the tree algorithm, called Bayesian Estimated Modified Dynamic Tree algorithm (BE-MDT) which extends the concept of optimum dynamic tree (ODT) suggested in the original paper of [2], by introducing the Bayesian tag estimation and MTA.

S. K. Wijayasekara, S. Nakpeerayuth and L. Wuttisittikulkij were with the Department of Electrical and Engineering, Chulalongkorn University, Bangkok, Thailand, 10330.

R. Annur was with the Department of Computer and Communication Technology Kampar, Universiti Tunku Abdul Rahman- Perak Campus, Perak, Malaysia.

W. Srichavengsup was with the Computer Engineering, Robotics and Technology Research Laboratory, Thai-Nichi Institute of Technology, Bangkok, Thailand.

K. Sandrasegaran was with the School of Electrical and Data Engineering, University of Technology Sydney, Sydney, Australia.

H.-Y. Hsieh was with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

Corresponding author: L. Wuttisittikulkij (email: wlunchak@chula.ac.th).

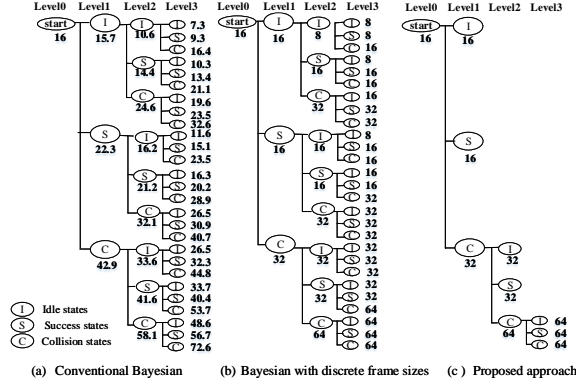


Fig. 1 Comparison of decision tree diagrams.

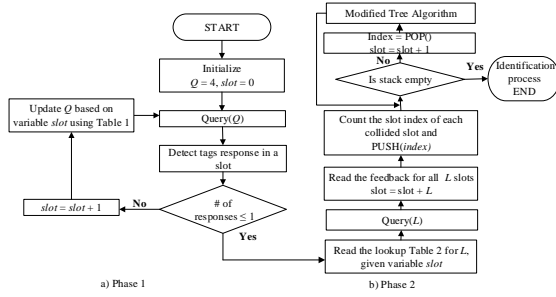


Fig. 2 Flow chart of the proposed BE-MDT algorithm.

TABLE I
LOOKUP TABLE FOR PRECOMPUTED BAYESIAN ESTIMATION.

Slot	1	2	3-6	7-13	14-28	29-59	60-120	121-244	245-491	>491
Q	4	5	6	7	8	9	10	11	12	13

TABLE II
INITIAL FRAME SIZES FOR MDT.

Slot	1-3	4-5	6-12	13-20	21-40	41-70	71-132	133-249	250-490	491-550	>550
Frame Size (L)	64	128	256	512	1024	2048	4096	8192	16384	32768	65536

II. THE PROPOSED BAYESIAN ESTIMATED MODIFIED DYNAMIC TREE (BE-MDT)

The proposed BE-MDT algorithm comprises two phases of operation. In the first phase, the algorithm aims to obtain the number of competing tags in the system. To acquire the estimated number as fast as possible, an efficient slot-based Bayesian tag estimation method is applied. In the second phase, we use an efficient technique to dynamically set the frame size of the first level of the tree, based on the estimated tag count from phase one. Then, we use MTA to resolve the conflict among the tags and identify them. This procedure in the second phase is referred to as MDT.

For the tag estimation, most well-known methods such as [1] rely on the feedback information in the previous frame, *i.e.* the number of idle, success and collision slots to estimate the number of competing tags. Since frame-based estimation can exhibit slow response even with the early frame breaking technique in which a frame can be terminated before the end of the frame is applied, we propose to perform the estimation at every slot to achieve fast tag estimation. In order to obtain good estimates, it is important to use a technique which can extract feedback information from each slot and accumulate them over a series of slots. Bayesian estimation method is

one of powerful mathematical tools that minimizes posterior expected value of a loss function while accumulating the prior information.

In this paper, we follow the Bayesian estimation principles introduced in [12], for the purpose of tag estimation, not for collision resolution. Based on their method, a decision tree with dynamically updated frame sizes is shown in Fig.1(a). This is displayed only for the first three slots. The decision tree grows exponentially with the number of slots, making it practically impossible to store all the pre-assigned frame sizes in a lookup table. Therefore, the reader must compute a new frame size at the end of every slot in real time. This is computationally too demanding for the reader. Fig.1(b) represents the modified decision tree from Fig.1(a) with an additional constraint that the frame sizes are limited to 2^k , where $\{k \in 1, 2, 3, \dots\}$. Such a constraint not only helps minimize the signaling from the reader to tags, but also substantially simplifies the tag estimation process.

In our work, we aim to acquire an accurate tag estimate using the smallest number of slots, and thus the decision tree should be terminated as soon as the tag estimate is deemed accurate enough. An appropriate termination criterion is based on the minimum ratio of the standard deviation over mean of the prior distribution of the number of tags. As this ratio gets smaller, the estimation becomes more accurate. Based on our extensive investigation over a broad range of tags from 2 to 65,535, we found that every colliding slot always exhibits a large standard deviation over mean which signifies that the estimation procedure should continue. In contrast, the idle and success slots give relatively small standard deviation over mean in prior distribution, which indicates that the estimation process can be terminated. Based on these investigations, the decision tree of the proposed method is shown in Fig.1(c). As we can see, the tree grows linearly with the number of slots, *i.e.* the number of leaves is two times the number of slots plus one. Consequently, proper values of Q can be precomputed as $\lceil \log_2 N_0 \rceil$ where N_0 is the mean number of tags as defined in [12] and stored in a lookup table as given in Table I. Thus, in our method, we get the benefit from the Bayesian estimation while the Bayesian computational complexity is no longer a problem.

Fig.2 shows the complete flow chart of the proposed algorithm. In the first phase, each unidentified tag accesses a slot with the probability of $1/2^Q$, where an initial broadcast frame size of $Q = 4$. When the reader experiences an idle or successful feedback from the current reading slot, it dismisses the tag estimation phase and moves to the second phase. Otherwise, the reader continues in the first phase with an updated value of Q using Table I. In the second phase, an initial frame size (L) is chosen based on the variable *slot* using the lookup table given in Table II. The reader sends the query command and waits for L slots. Only those collision slots are further resolved through MTA. The second phase is terminated when those collision slots are resolved. Note that Table II is constructed in such a way that the system efficiency can be maximized across a broad range of tags using only frame sizes that are the powers of two. An efficient means to identify the optimum frame sizes for every terminated slots is

to assign the mode of a posterior distribution of the number of tags at each terminating slot as an initial frame size. We can then fine-tune them to obtain suitable frame sizes, as given in Table II.

III. PERFORMANCE ANALYSIS

In this section, we analyze the average number of time slots required by the binary tree with biased splitting, MTA, MDT and BE-MDT algorithms to resolve all tags, so that their corresponding system efficiency can be determined.

Lemma 1: Let $C_1(N, k)$ be the average number of collision slots in the k^{th} level of the binary tree structure starting with N tags, $p_{k,j}$ be the probability that a tag arrives at the k^{th} level of the tree structure through the j^{th} path from all possible 2^k paths and p_0 and $p_1 = 1 - p_0$ be the probabilities that a tag selects the left and right branches respectively. Note that when $p_0 \neq p_1$, the binary tree is executed with biased splitting. Then, $C_1(N, k)$ is

$$C_1(N, k) = \sum_{j=0}^{2^k-1} \sum_{n=2}^N B(N, p_{k,j}, n), \quad (1)$$

where $B(n, p, i) = \binom{n}{i} p^i (1-p)^{n-i}$ and $p_{k,j} = \prod_{i=0}^{k-1} p_{\delta_i}$ with $j = dec(\delta_1 \delta_2 \dots \delta_k)$ and δ_i being a binary digits, indicating the subgroup selected at the i^{th} level, while $dec(\cdot)$ converts binary number to decimal.

Proof: A collision is said to take place when two or more tags choose the same slot. Therefore, the average number of collision slots in the k^{th} level of the binary tree structure is the summation of the collision probabilities of all slots in the k^{th} level and can be given as in (1).

Theorem 1: Let $T_{BTA}(N)$ be the average number of time-slots required by the binary tree protocol with biased splitting to recognize N tags. Then, $T_{BTA}(N)$ can be expressed as

$$T_{BTA}(N) = 2 \left[1 + \sum_{k=1}^{\infty} C_1(N, k) \right]. \quad (2)$$

Proof: Since the tree structure will continue to grow with collisions *i.e.*, every collided node will always produce two new nodes, by taking the summation of average number of collision slots in each level from Lemma 1 and multiply with the splitting factor of two, Theorem 1 can be yielded. Note also that

$$\sum_{n=2}^N B(N, p_{k,j}, n) = 1 - B(N, p_{k,j}, 0) - B(N, p_{k,j}, 1). \quad (3)$$

Hence, for the binary tree algorithm with fair splitting, $p_{k,j} = 1/2^k$ and

$$T_{BTA}(N) = 2 \left[1 + \sum_{k=1}^{\infty} \left[1 - \left[1 - \frac{1}{2^k} \right]^N - \frac{N}{2^k} \left[1 - \frac{1}{2^k} \right]^{N-1} \right] \right]. \quad (4)$$

This is exactly the same expression as (3.40) in [3], although it is derived from different approaches.

Lemma 2: The average number of definite collision slots that can be predicted at level k is

$$C_2(N, k) = \begin{cases} p_1^N, & k = 1, \\ \sum_{i=1}^{2^k-1} \sum_{n=2}^N [B(N, p_{k-1, (i+1)/2}, n) \times p_1^n], & k > 1. \end{cases} \quad (5)$$

Proof: For the binary tree structure, a collision will definitely occur on the right branch, if no tag selects the left branch. The probability of such a definite collision on a slot in the k^{th} level of the binary tree structure is the probability that there are more than two tags arriving at the $k-1^{th}$ level through the $(i+1)/2^{th}$ path and all of them select the right

branch with probability of p_1 . Note for the first level that all N tags are collided at the root slot. Therefore, $C_2(N, k)$ gives the average number of definite collision slots that can be predicted at level k .

Theorem 2: Let $T_{MTA}(N)$ be the average number of time slots consumed by MTA to identify all N tags. Then, $T_{MTA}(N)$ can be given by

$$T_{MTA}(N) = T_{BTA}(N) - \sum_{k=1}^{\infty} C_2(N, k). \quad (6)$$

Proof: In MTA, all inevitable collision slots that are predicted are skipped and tags involve in the collision proceed to the next tree level by splitting them into two subgroups. Thus, by deducting (2) by the average number of definite collision slots, (6) can be obtained.

Theorem 3: Let $T_{MDT}(N, L)$ denote the average number of time slots used in MDT to resolve N tags with the initial frame size of L slots. Then, $T_{MDT}(N, L)$ can be given by

$$T_{MDT}(N, L) = L \left[B(N, 1/L, 0) + B(N, 1/L, 1) + \sum_{n=2}^N B(N, 1/L, n) T_{MTA}(n) \right] \quad (7)$$

Proof: Tags are initially separated into L slots at random. Thus, the probabilities that each slot contains 0, 1 and more than one tag are $B(N, 1/L, 0)$, $B(N, 1/L, 1)$ and $B(N, 1/L, n)$ respectively. As only collision slots are resolved by MTA, (7) can be obtained.

Theorem 4: Let $T_{BE-MDT}(N, maxslot)$ be the average number of slots needed by BE-MDT to identify N tags, $p_i = 1/2^{Q_i}$ be the access probability that each tag uses in the i^{th} slot in the first phase where the Q_i value is given in Table I and L_i be the initial frame size used in the second phase for MDT. Then $T_{BE-MDT}(N, maxslot)$ can be calculated as

$$\begin{aligned} T_{BE-MDT}(N, maxslot) = & (1 - p_1)^N T_{MDT}(N, L_1) \\ & + N p_1 (1 - p_1)^{N-1} T_{MDT}(N-1, L_1) \\ & + \sum_{i=2}^{maxslot} [(1 - p_i)^N T_{MDT}(N, L_i) \\ & + N p_i (1 - p_i)^{N-1} T_{MDT}(N-1, L_i) \prod_{m=1}^{i-1} x_m] \\ & + T_{MDT}(N, L_{maxslot}) \prod_{q=1}^{maxslot} x_q, \end{aligned} \quad (8)$$

where $x_m = 1 - (1 - p_m)^N - N p_m (1 - p_m)^{N-1}$.

Proof: For BE-MDT, tags stay in the first phase as long as they experience collision slots. Hence, the probabilities that all tags arrive at the $(i-1)^{th}$ slot while still in the first phase is $\prod_{m=1}^{i-1} 1 - (1 - p_m)^N - N p_m (1 - p_m)^{N-1}$. If they transit into the second phase due to idleness or success, with the probability of $(1 - p_i)^N$ and $N p_i (1 - p_i)^{N-1}$ respectively, the number of tags entering the second phase are N and $N-1$ respectively and they will be resolved through MDT. By taking into account all possible events as shown in Fig.1(c), (8) can be yielded.

IV. RESULTS AND DISCUSSION

We compare the performance of the proposed BE-MDT with that of known tree-based anti-collision algorithms in terms of system efficiency and time system efficiency. The system

efficiency η_{SE} is the ratio of the number of tags N and the average number of time slots $T(N)$ required in an anti-collision algorithm to identify N tags as specified below:

$$\eta_{SE} = \frac{N}{T(N)}. \quad (9)$$

The time system efficiency η_{TSE} is defined as [10]

$$\eta_{TSE} = \frac{NT_s}{I(N)T_i + S(N)T_s + C(N)T_c}. \quad (10)$$

where $I(N)$, $S(N)$ and $C(N)$ are the expected number of idle, successful and collision slots respectively, while T_i , T_s and T_c are their corresponding time duration, according to EPCglobal C1 Gen2 standard [5]. Fig 3 presents the system efficiency comparison between the proposed BE-MDT algorithm and BTA [2], TTA [3], MTA [8], TSA [9], BSTSA [10], Dynamic BTSA [4] and Splitting BTSA [4]. It is clear that the BE-MDT consistently offers higher system efficiency than all other well-known tree-based algorithms for tags between 100-10,000 and actually beyond. Note that, for small number of tags, these numerical results for BTA, MTA, MDT and BE-MDT are obtained from our derived mathematical formula (2), (6), (7) and (8) respectively and confirmed through computer simulation. For larger number of tags, due to floating point precision limitations, numerical results for all algorithms are obtained solely from computer simulation. To ensure high accuracy of the simulation results, 10,000 runs of tag identification are executed for each case.

Although the system efficiency has been the most common measure of performance for anti-collision protocols, recent studies for RFID systems aim at maximizing the time system efficiency as it takes into account the fact that idle slots are shorter than successful and collision and various recommended time parameters as defined in the EPCglobal Gen 2 standard play a part in the actual time required for tag identification. Therefore, we apply the proposed algorithm to the RFID system according to the EPCglobal Gen 2 standard at data rate of 40 kbps with time parameters as given in Table 1 of [13]. Numerical results given in Fig 4 show that the BE-MDT can achieve better time system efficiency than all other algorithms, even the BSTSA which is primarily designed and optimized for maximum time system efficiency.

V. CONCLUSION

In this letter, we proposed a highly efficient anti-collision algorithm, with an efficiency of 45% for a broad range of number of tags using slot-based Bayesian estimation technique in conjunction with the modified dynamic tree algorithm for rapid tag identification. We also presented a new and complete mathematical analysis to derive the average number of time slots required by several tree-based algorithms including the proposed BE-MDT algorithm to resolve a group of tags. When applying the BE-MDT algorithm to the RFID system using EPCglobal C1 Gen2 standard, the achievable time system efficiency is at least 78.5%, which is higher than that of all other existing algorithms.

ACKNOWLEDGMENT

This work is supported in part by Smart Wireless Communication Ecosystem Research Group. The author would like to thank 100th anniversary Chulalongkorn University Fund for Doctoral Scholarship and 90th Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund).

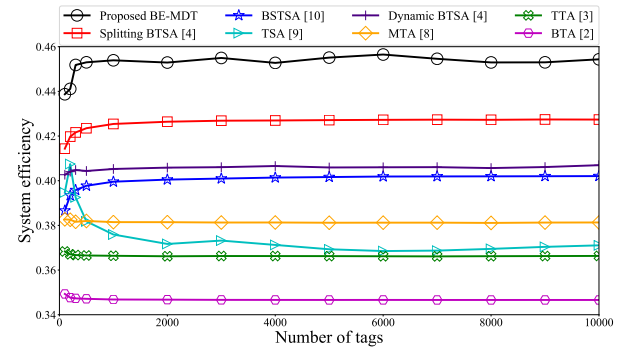


Fig. 3 Comparison of system efficiency of tree algorithms.

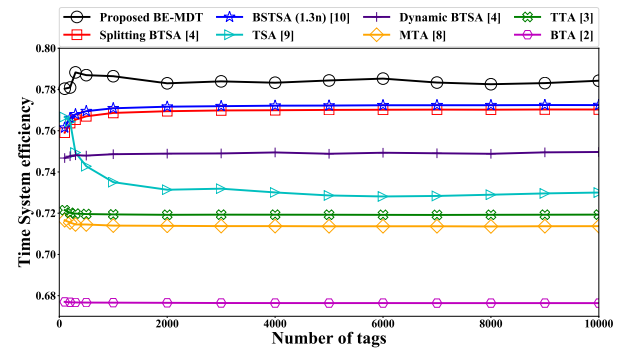


Fig. 4 Comparison of time system efficiency of tree algorithms.

REFERENCES

- [1] Šolić, Petar and Radić, Joško and Rožić, Nikola, "Energy efficient tag estimation method for ALOHA-based RFID systems," *IEEE Sensors J.*, vol. 14, no. 10, pp. 3637-3647, 2014.
- [2] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory.*, vol. 25, no. 5, pp. 505-515, 1979.
- [3] P. Mathys and P. Flajolet, "Q-ary collision resolution algorithms in random-access systems with free or blocked channel access," *IEEE Trans. Inf. Theory.*, vol. 31, no. 2, pp. 217-243, 1985.
- [4] 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, 2013.
- [5] EPCglobal, "EPCTM Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHz," version. 2.0.1, 2015.
- [6] J. Su, Z. Sheng, D. Hong and G. Wen, "An Effective Frame Breaking Policy for Dynamic Framed Slotted Aloha in RFID," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 692-695, April 2016.
- [7] Y. Chen, J. Su and W. Yi, "An Efficient and Easy-to-Implement Tag Identification Algorithm for UHF RFID Systems," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1509-1512, July 2017.
- [8] Yingqun Yu and G. B. Giannakis, "SICTA: a 0.693 contention tree algorithm using successive interference cancellation," *Proc. Int'l Conf. IEEE Comput. Commun. Soc.*, vol. 3, pp. 1908-1916, 2005.
- [9] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Tree slotted aloha: a new protocol for tag identification in RFID networks," *Proc. Int'l Symp. World of Wireless, Mobile and Multimedia Networks*, pp. 1-6, 2006.
- [10] 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.
- [11] J. Su, Z. Sheng, L. Xie, and G. Wen, "Idle-slots elimination based binary splitting anti-collision algorithm for RFID," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2394-2397, Dec. 2016.
- [12] R. Annur, W. Srichavengsup, S. Nakpeerayuth, and L. Wuttitsittikulkij, "Bayesian Method of Slotted Aloha Based Anti-Collision Protocol for RFID Systems," *Proc. Int'l Symp. Autom. Decent. Sys.*, pp. 87-90, 2015.
- [13] J. Su, Z. Sheng and L. Xie, "A Collision-Tolerant-Based Anti-Collision Algorithm for Large Scale RFID System," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1517-1520, July 2017.