An efficient Anti-Collision Algorithm for RFID System

TZAY-FARN SHIH† and WEN-LI HSU
Department of Computer Science and Information Engineering,
Chaoyang University of Technology,
Taichung, Taiwan 41349, R.O.C
tfshih@cyut.edu.tw, s9627619@cyut.edu.tw

Abstract: -Radio frequency identification (RFID) is an automatic identification system working through radio frequency (RF). Tag collision occurs while two or more than two tags deliver their unique identification (UID) to the reader. Since collision, which would significantly affect the efficacy of RFID, has been one of the major study issues for RFID technology, we have developed a solution, bit competed algorithm (BCA), to solve the collision problem occurred in RFID system and consequently improve the efficacy of RFID. While receiving the UIDs, our reader will identify the tags through Boolean OR operation for bit-by-bit priority competition. The massive computer simulation results and the comparison with other algorithms for collision prevention proved our solution can effectively solve the collision problem and improve the efficacy of RFID system.

Key-Words: -RFID, Reader, Tag, Collision, Anti-collision algorithm.

1 Introduction

Radio frequency identification (RFID) system, which is composed by two components – reader and tag, is an automatic identification system working through radio frequency (RF) communication. Figure 1 illustrates an UID format of a tag with n bits length. Through the RF communication between reader and tags, the UIDs on tags would be identified [1-3]. However, if more than one tag delivered their UIDs at the same time, the collision or error problem occurred.

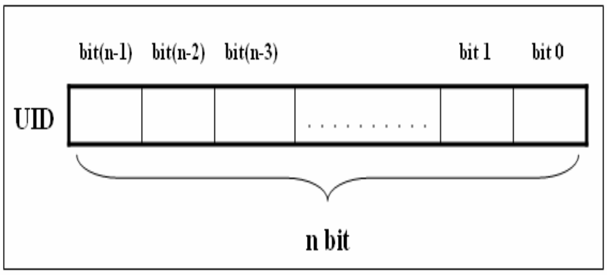
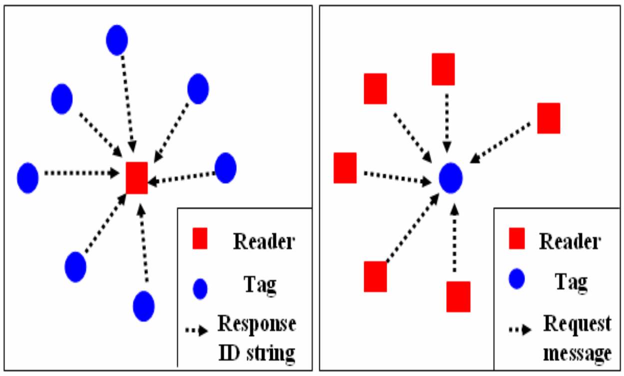


Figure 1. UID format

There are two types of collision in RFID, tag Collision [4-14] and reader Collision [15-18], which were described in Figure 2.

Tag collision is occurred when more than one tag was responding the message to the same reader at the same time and causing tags competition, as in Figure 2(a).

Reader collision is occurred when more than one reader in a certain area delivered the request message to the same tag, as in Figure 2(b).



(a) (b)

Figure 2. RFID Collision: (a) Tag Collision (b)Reader Collision

Either reader collision or tag collision established a significant study subject, the solution for collision prevention and increase the identification efficiency of the system. Since most of the readers have sufficient capacity for communication and coordination in between, reader collision could be prevented in advance.

However, as tags have only limited ability of calculation, the design for tag collision prevention algorithm is not easy. Although there were many studies focusing on solutions for tag collision, a

† Corresponding author.

better solution is still expected and we have made efforts on this subject.

The structure of this paper is as follows: the second section will revisit previous related research, the third section will describe the collision prevention algorithm; the fourth section shows simulation analysis as a result, the fifth part of our conclusions.

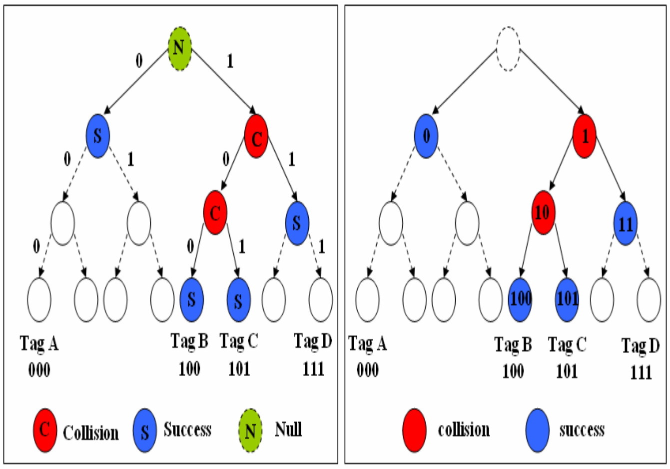
2 Related Work

Multiple access technology has been evolved to avoid the collision and the limitation of sharing channels in wireless networks, such as space division multiple access (SDMA), frequency division multiple access (FDMA), time division multiple access (TDMA), and code division multiple access (CDMA). However, considering the cost of reader, not all the technology can be applied at the same time in a RFID system and the TDMA is mostly applied to avoid collision [1-3].

Tree-based algorithm [4-9] and aloha-based algorithm [10-14], described as follows, are two main measures among all the studies to solve the tag collision problem in RFID system.

2.1 Tree-based algorithm

Tree-based algorithm: Collision occurred when the request of reader received more than one UID from tags with the same bit length. Reader would complete identification by requesting the response bit by bit till only one tag response is received. The most common algorithms of tree-based algorithms are Binary Tree Algorism (BTA) and Query Tree Algorithm (QTA). BTA identifies one bit at a time, in a sequence of highest bit to lowest bit. Every time reader would broadcast a bit value to all the tags, and then the tag would compare this requested bit value with the comparable UID bit value and deliver response to reader or enter the sleep mode. This process would repeat till one tag is identified by the reader, and the same for the other tags identification till all the tags are read. Taking an example of an UID with 3 bits in a sequence of bit2 ~ bit0, the bit value broadcasted by the reader would be in a sequence as follows: 0(bit2), 1(bit2), 0(bit1), 1(bit1), 0(bit0), and then 1(bit0). BTA is illustrated in Figure 3(a), where data transfer of Tag A, Tag B, Tag C, Tag D were proceeded. Collision occurred when more than one tag responded the same bit value after the reader had broadcasted null message. First of all, the reader would broadcast a bit value 0 to all the tags to identify the tag with bit value 0 at highest bit, which is Tag A.



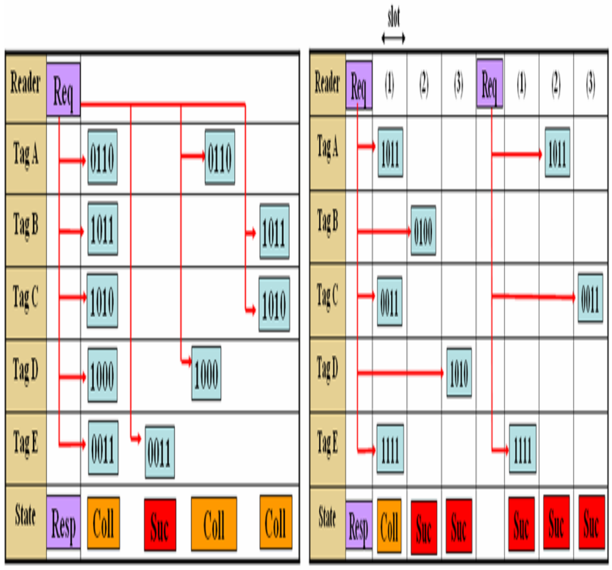
(a) (b)

Figure 3.Tree-based algorithm: (a) Binary tree algorithm (b) Query tree algorithm

Secondly, the reader would broadcast a bit value 1 to all the tags to identify the tag with bit value 1 at highest bit, which are Tag B and Tag C and the collision occurred again. Then the reader would broadcast a bit value 0 to all the tags to identify the tag with bit value 0 at the second highest bit, and go through the same logic till all the tags are identified. However, the long tags could generate too much collision and delay of identification, so other improvement like QTA in Figure 3(b) was developed to solve this problem. QTA is identifying one bit string at a time, in a sequence of highest bit to lowest bit and increasing the length of bit series. Every time the reader would broadcast a bit series, and then the tag would respond if its first bit series are the same. The collision occurred when more than one tag response happened, and then the reader would broadcast a longer bit series for the next identification and repeat this procedure till only one tag identified. Taking an example of an UID with 3 bits, the bit value broadcasted by the reader would be in a sequence as follows: 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111. Query Tree Algorithm is illustrated in Figure 3(b), where data transfers of Tag A, Tag B, Tag C Tag D were preceded. First of all, the reader would broadcast a bit value 0 to all the tags to identify the tag with bit value 0 at highest bit, which is Tag A. Secondly, the reader would broadcast a bit value 1 to all the tags to identify the tag with bit value 1 at highest bit, which are Tag B and Tag C and the collision occurred. Then the reader would broadcast a bit series with value 00 to all the tags to identify the tag with bit value 00 at first two digits, and go through the same logic till all the tags are identified. This method may also have the problems like overread and too much idle nodes.

2.2 Aloha-based algorithm

Aloha-based algorithm: Identification is successful only when one UID responds without any collision with another UID, after the reader broadcasting the request. The collision occurred when two or more than two tags delivered responses at the same time, and the tags should re-deliver response after a while, till the tag is identified by the reader. Pure-Aloha Algorithm and Slotted-Aloha Algorithm are the most well-know Aloha-based algorithm. Identification by Pure-Aloha Algorithm is successful only when one UID responds without any collision, after the reader broadcasting the request. The collision occurred when two or more than two tags delivered responses at the same time, and the tags should re-deliver response after a while, till the tag is identified by the reader. As the tags can always deliver the UIDs to the reader, the collision occurrence is high and the system efficacy is low. Pure-Aloha Algorithm process is illustrated in Figure 4(a), where Tag A, Tag B, Tag C, Tag D, and Tag E responded. First of all, the reader broadcasted the request message to all the tags, and all 5 tags responded at the same time and the collision occurred. The tags should re-respond after waiting for a while, and then the reader identified Tag E while UID of Tag E was delivered and no another tag's UID delivered at the same time. When Tag D's UID was delivering, the delivering of Tag A created the collision and both have to wait for a while and re-deliver the UIDs. Identification is repeated and completed through the same process. However, the high collision occurrence may limit the delivery of the tags through this algorithm, so improved algorithm like Slotted-Aloha Algorithm in Figure 4(b) has been developed, to avoid the interruption of another UID delivery while there is one UID delivering. Slotted-Aloha Algorithm splits the delivery period too many time slots, and defaults the tags delivery can be started at the starting time of time slots. The system efficacy doubled as the new tag delivery cut-in was avoided and the collision occurrence was consequently decreased. The reader controls the numbers and the assignments of time slots. The reader would randomly assigned time slots to every tag and broadcast a request routinely and the tag should respond at the time slot assigned. The collision occurred when more than one tag responded at the same time, and the tags should wait till the new time slot assigned by the reader. This process would repeat till all the tags are identified. Slotted-Aloha Algorithm is illustrated in Figure 4(b), where Tag A, Tag B, Tag C, Tag D and Tag E responded. First of all, the reader broadcasted the request message to all the tags and assigned slot 1 to Tag A, Tag C and Tag E. Tag A, Tag C and Tag E delivered at the same time



(a) (b)

Figure 4. Aloha-based algorithm: (a) Pure-Aloha algorithm (b) Slotted-Aloha algorithm

slot so the collision occurred. The tags should re-respond after waiting for a while, and then the reader identified Tag B while only Tag B was assigned to time slot 2 and delivered its UID at time slot 2. Tag E, Tag A, and Tag C were identified after

3 Bit Competed Algorithm

Both algorithms above create too many collisions or too many idle nodes while the reader is identifying too many tags at the same time. Therefore, the collision can't be avoided and the system efficacy is decreased by delay. We have developed a algorithm, bit competed algorithm (BCA), to solve the collision problem through decrease collision occurrence and consequently improve the system efficacy. BCA is working through the Boolean OR operation in reader to prioritize the UIDs according to the results of bit-by-bit priority competition. The reader would periodically broadcast Request Null message to all the tags, and all the tags would deliver their UIDs at the same time. Collision occurred when the tag response is more than 1; then the reader would go through Boolean OR operation of UIDs bit-by-bit, from the highest bit to the lowest bit, to identify the UID with the highest priority.

The procedure of Boolean OR operation is described as follows. Reader would analysis the bit values of all UIDs received, if the Boolean OR algorithm result is 1, all the tags with bit value 0 would give up the competition and only tags with bit value 1 would remain for next bit competition. If the algorithm result is 0, the reader would precede the

Step 1. The reader periodically sends Request Null message to the tags in the effective range and waiting for the tags' response.

Step 2. Is there a tag collision? If a tag collision occurs, go to step 3; otherwise, go to step 6.

Step 3. Boolean ORed the highest UID bit of all competed tags.

Step 4. If the value of Boolean ORed operation equal to 0, go to step 5. If the Boolean ORed value equal to 1, all tags with the competing bit equal to 0 will stop to compete and waiting for a new request message. If the number of tag with the competing bit value 1 greater than 1, go to step 5; otherwise, go to step 6.

Step 5. Boolean ORed the next UID bit of all competed tags, go to step 4.

Step 6. Tag identification succeeded.

Figure 5. (b) BCA algorithm

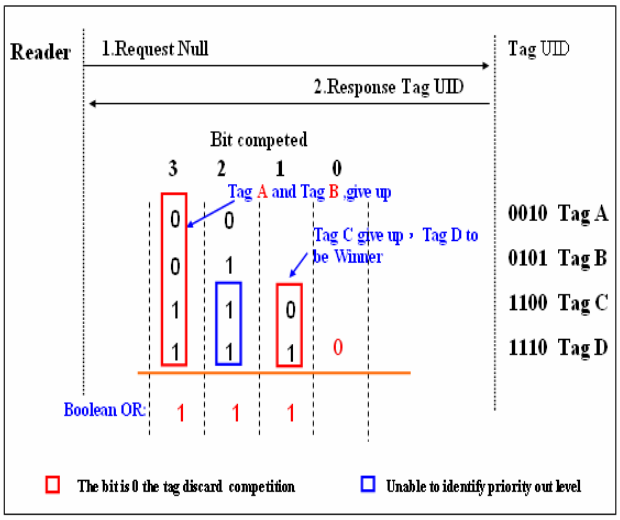


Figure 6. An example of BCA competition process

next bit competition. This bit-by-bit competition would identify only one tag finally. Taking a 3-bits UID tag as an example, the reader would precede the Boolean OR algorithm in the sequence of bit 2, bit 1, and bit 0, from the highest bit to the lowest bit. Our BCA algorithm, using Boolean OR operation bit-by-bit, would effectively decrease the collisions and requests-responses between reader and tag. The procedure of BCA is as Figure 5.

Figure 6 shows an example of BCA collision competition process for an identification between one reader and 4 tags with UID as follows: Tag A(00100)、Tag B(11011)、Tag C(11001)、Tag D(11100). After the reader broadcasting request null message to all tags, the collision occurred as there are 4 tags response their UIDs to reader in the same time. The reader uses BCA algorithm to determine the priority of all UIDs through Boolean OR operation bit-by-bit, following the steps hereafter:

•Step 1: The highest UID bit (bit 4) of every tag was read, which were 0, 1, 1, 1, respectively. The Boolean OR operation of these 4 bits generated bit value 1, and then the tag with bit value 0 at bit 4, such as Tag A(00100) would give up and wait for the next request. Tag B(11011),Tag C(11001) and Tag D(11100) had bit value 1 at bit 4 would remain for competition.

•Step 2: The second highest bit (bit 3) of Tag B(11011),Tag C(11001) and Tag D(11100) were 1, 1, respectively, and the Boolean OR algorithm result would be 1. However, the bit values of bit 3 were the same, so the competition had to be determined by next bit (bit 2).

•Step 3: The third highest bit (bit 2) of Tag B(11011)Tag C(11001) and Tag D(11100) were 0,0,1, respectively, and the Boolean OR algorithm result would be 1. Tag B and Tag C would give up as its third highest bit was 0, and Tag D(11100) would be the only tag left and identified by the reader. If the priority can be identified at higher bit, the computation is less and the identification would be faster.

•Step 4: The algorithm repeats till all the tags were identified. The maximal number of request-response would be n-1 when there were n tags delivering UIDs at the same time, and this is better than the other algorithms.

4 SIMULATION RESULTS

The results of simulation show our BCA can effectively decrease the collision and increase the efficacy of RFID system. Nevertheless, as every algorithm is different with its only features, this may benefit some algorithm in certain cases. In order to have a fair comparison, we would compare our BCA with BTA and QTA as they are all bit-by-bit competition algorithms.

In the simulation analysis, we consider three criteria: collision occurrence, request-response, total numbers of bits delivered for different algorithms while two factors, tag amounts and bits length of UID were applied. The results were compared with the studies of BTA and QTA to evaluate the improvement of RFID efficacy by our BCA.

4.1 Number of Tag and identification potency relations

In this section, the effect of the increased tag amounts on collision occurrence, request-response, and the total number of bits delivered was investigated. Tag amounts were randomly generated between 100 and 1000 tags, with UID at 12 bits. In Figure 7(a), the relationship between tag amount and collision occurrence in our BCA revealed better efficacy than BTA and QTA. As BCA can always identify one tag at each collision through Boolean OR operation, BTA and QTA may not have the same efficiency and usually identify one tag only after several collisions. Therefore, collision occurrence of BCA is much less than of BTA and QTA. BCA can guarantee identification completed in k-1 collisions while there are k tags take part in the competition.

In Figure 7(b), the effect of increased tag amounts on request-response has revealed BCA has better efficacy as the request-response of BCA is far less than of BTA and QTA, since the collision occurrence is decreased by BCA. The fewer collision occurs, the less request-response packets transmit. It is reasonable that BCA is also less affected by tag amount while BTA and QTA are compared.

4.2 Tag UID length and identification potency relations

Since the bits length of UID would affect the identification as well as tag amounts, the effect of the increased bits length on collision occurrence, request-response, and the total number of bits delivered was studied. Bits length of UID would be limited less than 96 bits, which is the standard tag bits length in EPC Global. BCA, BSA and QTA were compared though the effect of the different bits length, such as 16 bits, 64 bits and 96 bits, on collision occurrence, request-response, and the total number of bits delivered. The results in Figure 8(a) revealed that BSA and QTA would be more affected on collision occurrence by bits length of UID, while Figure 7(a) revealed that the collision occurrence of BCA was mainly resulting from tag amounts. Therefore, different bits length of UID proceeded by BCA have the same curve in Figure 8(a).

As request-response is related to collision occurrence, BCA has better request-response since its collision occurrence is mainly affected by the tag amounts, instead of bits length of UID. As one tag would be identified at every collision, the bits length of UID would not affect much the request-response of BCA, as in Figure 8(b), while the other algorithms were affected more by bits length of UID.

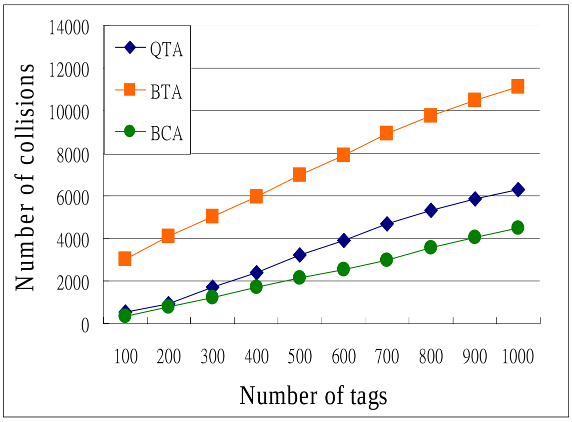


Figure 7(a) Number of tags and collision relations

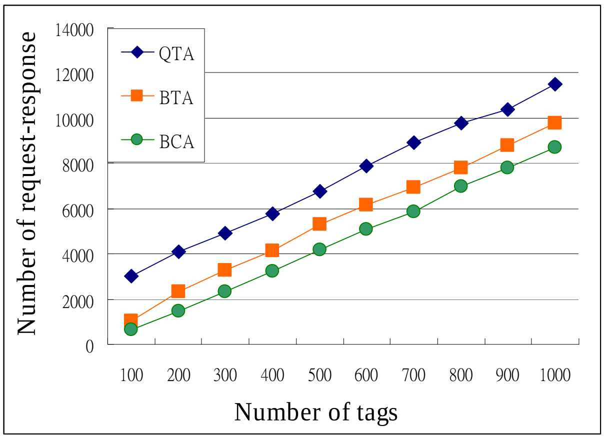


Figure 7(b) Number of tags and request-response relations

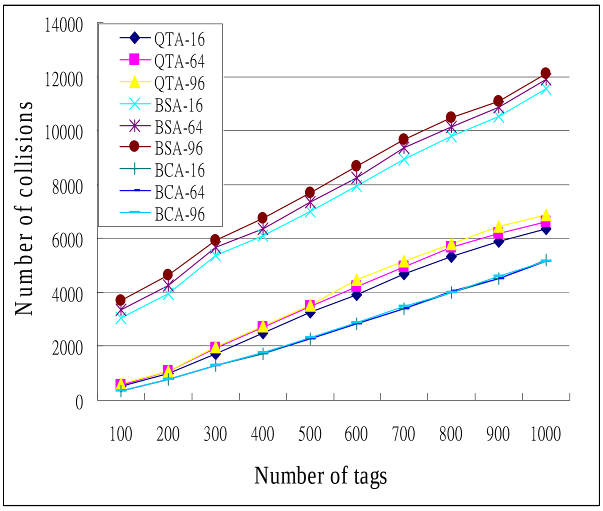


Figure 8(a) Tag UID length and collision relations

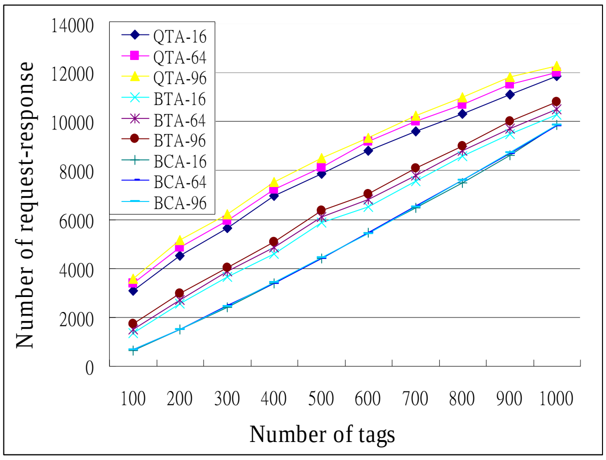


Figure 8 (b) Tag UID length and request-response relations

5 Conclusion

As collision would significantly affect the RFID efficacy, a good algorithm for collision prevention should eliminate the collision and consequently improve the identification rate and speed to contribute an efficient system. We have proved a better solution, bit competed algorithm (BCA), to solve the tag collision problem of RFID and therefore improve the efficacy of RFID. BCA is working through Boolean OR operation in reader with bit-by-bit priority competition of delivered UIDs. As one prioritized tag would be produced in every collision, this method would decrease collision occurrence and the overhead of identification. Moreover, collision occurrence and overhead would not be easily affected by bits length of UIDs. According to computer simulation and the comparison, it is revealed that our algorithm has better efficacy than Binary Tree Algorithm and Query Tree Algorithm.

References:

[1] D. Shih, P.L. Sun, D.C.Yen and S. M.Huang, “Taxonomy and Survey of RFID Anti-collision protocols,” Computer and communications, Vol.29, 2006, pp. 2150–2166.

[2] K. Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, 2nd ed. New York: Wiley, 2003.

[3] Tang, Zhijun and He, Yigang, “Research of Multi-access and Anti-collision Protocols in RFID Systems,” Anti-counterfeiting, Security, Identification, IEEE International Workshop on, April 2007 pp.377 – 380.

[4] J. H. Chio, D. Lee, H. Lee, “Query tree-based reservation for efficient RFID tag anti-collision,” IEEE Communications. Letters, Vol. 10, No. 4, April 2006, pp. 236-238.

[5] Ji Hwan Choi,Dongwook Lee and Hyuckjae Lee, “Bi-slotted tree based anti-collision protocols for fast tag identification in RFID systems,” *IEEE Communications Letters*, Vol 10, December 2006, pp.861 – 863.

[6] J. Myung,W. Lee, and J. Srivastava, “Adaptive binary splitting for efficient RFID tag anti-collision,” *IEEE Communications. Letter*, vol. 10, No. 3,Mar 2006, pp. 144-146.

[7] Jihoon Myung and Wonjun Lee, “Adaptive Splitting Protocols for RFID Tag Collision Arbitration,” *MobiHoc’06, Florence, Italy*, 2006, pp. 202–213.

[8] Sung Hyun Kim and PooGyeon Park, “An Efficient Tree-Based Tag Anti-Collision Protocol for RFID Systems,” *IEEE Communications Letters*, Vol.11, May 2007, pp.449 – 451.

[9] T. Wang, “Enhanced binary search with cut-through operation for anti-collision in RFID systems,” *IEEE Communications Letter*, Vol. 10, No. 4, April 2006, pp.236-238.

[10] C. S. Kim, K. L. Park, H. C. Kim, and S. D. Kim, “An efficient stochastic anti-collision algorithm using bit-slot mechanism,” *PDPTA04*, 2004.

[11] J. Cha and J. Kim, “Novel Anti-collision Algorithms for Fast Object Identification in RFID System,” *in Proc. of Parallel and Distributed System*, Vol. 2, Jul 2005, pp. 63-67.

[12] J. Park, M. Chung, and T.-J. Lee, “Identification of RFID Tags in Framed-Slotted ALOHA with Robust Estimation and Binary Selection,” *IEEE Communications Letters*, Vol. 11, No.5, 2007, pp. 452-454.

[13] S. Lee, S. Joo, and C. Lee, “An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification,” *in Proc. MobiQuitous2005*, pp. 166-172.

[14] Wong, C.P and Quanyuan Feng, “Grouping based bit-slot ALOHA protocol for tag anti-collision in RFID systems,” *IEEE Communications Letters*, Vol.11, December 2007 pp.946 – 948.

[15] J. Kim, W.Lee, J. Yu, J. Myung, E. Kim, C.Lee, “Effect of Localized Optimal Clustering for Reader Anti-Collision in RFID Networks:Fairness Aspects to the Readers, ” *Proceedings 14th International Conference on Computer Communications and Networks*, 2005, pp. 497-502.

[16] Junius Ho, Daniel W. Engels, and Sanjay E. Sarma, "HiQ: A Hierarchical Q-Learning Algorithm to Solve the Reader Collision Problem," *SAINTW'06*, 2006, 4 pp.

[17] J. Waldrop, D.W. Engels, S.E. Sarma, "Colorwave: an anticollision algorithm for the reader collision problem," *in: IEEE Wireless Communications and Networking Conference,* March, 2003, pp.1206–1210.

[18] Shailesh M.Birari "Mitigating the Reader Collision Problem in RFID Networks with Mobile Readers"Dissertation for Master of Technology, Indian Institute of Technology, India, 2005.