Resolving RFID Data Stream Collisions using Set-Based Approach

Prapassara Pupunwiwat, Bela Stantic

Institute for Integrated and Intelligent Systems Griffith University, Gold Coast Campus, Queensland 4222, Australia

{P.Pupunwiwat, B.Stantic}@griffith.edu.au

Abstract—Radio Frequency Identification (RFID) is a technology that incorporates the use of the electromagnetic spectrum to uniquely identify people or objects. RFID technology has major data stream issues caused by massive amounts of tags simultaneously being captured by the reader that results in collisions. This collision issue can be solved by using anticollision methods. The current ALOHA-based methods suffer from insufficient performance, especially when a substantial number of tags are present within a reader zone. In this work, we propose a Modified Dynamic Framed-Slotted ALOHA (MDFSA) technique to maximise the performance efficiency and to reduce the total number of slots queried during the tag identification process. The MDFSA approach creates new tag grouping rules using particular equations according to the optimal efficiency obtained for a specific number of tags. In this study, we have demonstrated that our proposed method has maintained its efficiency above other existing approaches, and has the most effective performance.

I. INTRODUCTION

RFID technology is a promising technology that has the potential to improve the efficiency of business processes by providing automatic identification through data capturing. In RFID systems, when numerous amounts of tags are simultaneously present in the reader zone, the reader is required to have the ability to read data from individual tags. The technical approach that handles *tag collision* without any interference is called an *anti-collision* method.

The RFID reader is a powerful device that has sufficient power and memory, while a *passive RFID tag* requires energy from the radio signals sent by a reader to output its data. If more than one tag sends their ID to the reader at the same moment, a collision occurs and the ID will need to be retransmitted according to an anti-collision scheme. The main focus of an anti-collision scheme is to read multiple tags as fast and reliably as possible. The two types of tag anti-collision algorithms widely used in RFID systems are the *Tree-based* anti-collision, and the *ALOHA-based* anti-collision techniques.

In this study, we propose a method called a *Modified Dynamic Framed-Slotted ALOHA* (MDFSA) to minimise the total number of *slots* queried during the tag identification process, and to maximise the performance efficiency. The results and analysis from our experimentation demonstrate that the MDFSA method maintained its efficiency above existing methods and has the most effective performance.

The remainder of this paper is organised as follows: in Section II, we provide general background information related to tag collisions and discuss different *ALOHA-based* anticollision approaches. We present our new methodology, the *MDFSA*, in Section III. In Section IV, we present our experimental evaluation, results and analysis; and conclude our paper in Section V.

II. RFID BACKGROUND

RFID technology is an automatic identification technology that transmits the identification of an object or person wirelessly using radio frequency waves. There are several methods of identification but the most common is to store a serial number that uniquely identifies a person or object such as *Electronic Product Code* (EPC). RFID may only consist of a tag and a reader but a complete RFID system involves many other components and software, such as network, middleware, and user applications.

Simultaneous transmissions in RFID systems lead to collisions as the readers and passive tags typically operate on the same channel. Three types of collisions are possible: Reader-Reader collision, Reader-Tag collision, and Tag-Tag collision. *Tag collision* problem is more complex than those within reader collision categories. The various types of tag anti-collision approaches for tag collision can be reduced to two basic types: *Tree-based* deterministic approach [1], [2], [3] and *ALOHA-based* probabilistic approach [4], [5].

In an ALOHA-based approach, tags respond at randomly generated times. If a collision occurs, colliding tags will have to identify themselves again after waiting a random period of time. This technique is faster than Tree-based but suffers from low throughput (efficiency) and tag starvation where not all tags can be identified due to the random nature of chosen time. The modern passive RFID tags use EPC Class 1 Generation 2 encoding scheme and ALOHA-based protocols for anticollision. Thus, in this study, we focus on the ALOHA-based techniques in order to improve throughput rates and increase performance efficiency.

A. Basic Framed-Slotted ALOHA

The Basic Framed-Slotted ALOHA (BFSA) is the simplest of all the protocols that use a fixed *frame-size* for all the

rounds. The reader offers information to the tags including the *frame-size* specification and the random number selected by each slot within the frame. Each tag selects a slot using the random number and then sends its ID back to the reader [6], [7], [8]. Since the *frame-size* of the BFSA is fixed, its implementation is simplistic. However, the system's efficiency drops significantly in the event of there being too large or too small tag counts.

In ALOHA-based anti-collision, there are three kinds of slot as shown in Figure 1: 1) *Successful slot* where there is only one tag reply; 2) *Empty slot* where there is no tag reply; and 3) *Collision slot* where there is more than one tag reply.

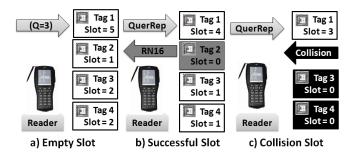


Fig. 1. Empty Slot, Successful Slot, and Collision Slot in EPC Class 1 Gen2 Protocol.

B. Dynamic Framed-Slotted ALOHA

The Dynamic Framed-Slotted ALOHA (DFSA) overcomes the problems associated with BFSA by dynamically changing the *frame-size* according to estimated number of *Backlog*, which is a number of tags that have not been read. In DFSA, each tag in an interrogation zone selects one of the given *N* slots to transmit its identifier; and all tags will be recognised after a few frames. Each frame is formed of specific number of slots that is used for the communication between the readers and the tags. DFSA can identify the tag efficiently because the reader adjusts the *frame-size* according to the estimated number of tags [6], [7], [9]. However, the *frame-size* change alone cannot reduce sufficiently the tag collision when there are large numbers of tags because it cannot increase the *frame-size* indefinitely. DFSA has various versions depending on different tag estimation methods used.

C. Enhanced Dynamic Framed-Slotted ALOHA

The DFSA algorithms change the *frame-size* to increase the performance efficiency of tag identification. However, as the number of tags become larger than the *frame-size*, the probability of tag collision increases rapidly. Since the *frame-size* cannot be increased indefinitely, when the number of unread tags is too large to achieve high system efficiency, the number of responding tags somehow must be restricted. This is so the optimal number of tags responds to the given *frame-size* [7].

The Enhanced Dynamic Framed-Slotted ALOHA (EDFSA) first estimates the number of unread tags. If the number of tags within the interrogation zone is larger than the maximum

frame-size, the EDFSA algorithm splits the number of Backlog into number of groups/sets and allows only one group of tags to respond. When the reader limits the number of responding tags, it transmits the number of tag sets and a random number to the tags when it issues the query. If the number of estimated Backlog is below the threshold, the reader adjusts the frame-size without grouping the unread tags. After each read cycle, the reader estimates the number of unread tags and adjusts its frame-size. This procedure repeats until all the tags are read [7]. The problem with EDFSA method is that it assumes that 256 is the optimal frame-size and splits tags into group set by using the power of two (2,4,8...). This results in a decrease in system efficiency when the number of tags is a fraction above the threshold and the number of group sets will be doubled.

D. Precise Tag Estimation Scheme

In order to predict accurate number of unread tags and determine the new *frame-size*; BFSA, DFSA, and EDFSA algorithms gather and use information such as number of successful slots, empty slots, and collision slots from the previous round to predict the appropriate *frame-size* for the next identification round. There have been several other methods mentioned in literature related to *Backlog* estimation, including Schoute Method [10], Lowerbound method, Precise Tag Estimation Scheme [11], Vogt method [12], and Bayesian method [13].

In this work, we have chosen a *Precise Tag Estimation Scheme (PTES)* [11] for Backlog estimation. PTES method is easy to implement with low overhead computation, and provides accurate tag estimation compared with other existing techniques. PTES approach assumes that for the current identification round, each collision slot has at least 2 tags collided. However, it is unknown how many tags actually caused the collision. There is exactly 1 tag per successful slot, thus successful slots is not taken into consideration. On the other hand, empty slots will continuously occur during the next rounds of identification despite the *frame-size*. Therefore, *Backlog* after the current frame is defined by equation:

$$Backlog = \lceil (V_1 \times c + V_2 \times e) \rceil$$

Where c is Collision Slot; e is Empty slot; V_1 and V_2 are variables for collision slot and empty prediction respectively.

III. MODIFIED DYNAMIC FRAMED-SLOTTED ALOHA

To address shortcomings of the current ALOHA-based approaches, we propose a MDFSA anti-collision technique to minimise number of *slots* queried and to optimise the performance efficiency. The Precise Tag Estimation Scheme (PTES) [11] is employed as accurate *frame-size* prediction for MDFSA.

A. The Fundamentals for ALOHA-based Tag Estimation

In the Framed-Slotted ALOHA based probabilistic scheme, to estimate the number of present tags, Binomial distribution has been utilised [14], [15], [16]. For a given initial Q in a

frame with F slots and n tags, the expected value of the number of slots with occupancy number x is as follows:

$$a_x = n \times C_n^x (\frac{1}{F})^x (1 - \frac{1}{F})^{n-x}$$

Therefore, the expected number of Empty slot e, Successful slot s, and Collision slot c are given by the following equations:

$$\begin{cases} e = a_0 = F(1 - \frac{1}{F})^n \\ s = a_1 = n(1 - \frac{1}{F})^{n-1} \\ c = a_k = F - a_0 - a_1 \end{cases}$$

Thus, the system efficiency is defined as the ratio between the number of Successful slot and the frame-size given by the following equations:

$$E = \frac{s}{F} = \frac{n(1 - \frac{1}{F})^{n-1}}{F} = n\frac{1}{F}(1 - \frac{1}{F})^{n-1}$$

It is proven that the highest efficiency can be obtained if the frame-size is equal to the number of tags, provided that all slots have the same fixed length:

$$F(optimal) = n$$

B. Foundation of MDFSA

The MDFSA method is based on the DFSA algorithm with PTES prediction except that it uses group splitting rule to split *Backlog* into different group set if the number of unread tags is higher than the maximum *frame-size*.

MDFSA approach first estimates the number of Backlog. If the number of Backlog within the reader zone is much larger than the specific frame-size, it splits the number of Backlog into a number of group sets and allows only one set of tags to respond. The reader then issues 'Query', which contains a 'Q' parameter to specify the frame-size. Each selected tag in the set will pick a random number between 0 to 2^Q - 1 and put it into its slot counter. Only the tag that picks zero as its slot counter responds to the request. When the number of estimated Backlog is below the threshold, the reader adjusts the frame-size without grouping the unread tags. After each read cycle, the reader estimates the number of Backlog using PTES algorithm and adjust its frame-size.

1) DFSA Algorithm using PTES: MDFSA approach first estimates the number of unread tags, then it decided if the number of tags needs to be spliced or not. The DFSA algorithm (using PTES) is then applied to each selected set of tag.

Algorithm 1 demonstrates the DFSA algorithm applied to each selected set of tags where only one set of tags responds to the reader, and the reader issues 'QueryRep' for the next slot. After 'QueryRep' command is received, each tag decreases its slot counter by 1. At the end of each frame, the reader checks if all tags have been identified, estimates the number of *Backlog* using PTES algorithm, and adjust its *frame-size*.

```
Reader sends Query
for (Identification procedure) do
    Every tags generate RN16 and slot counter;
    for (Current frame) do
        if (Slot\ counter == 0) then
             Tag replies its RN16;
             if (A single tag replies) then
                  Reader sends ACK(RN16) to a tag;
                  if (RN16 received by tag == RN16 tag saved data)
                  then
                      Tag sends (EPC+PC+CRC) to reader;
                  end
                  Reader sends QueryRep;
             end
             else if (Multiple tags reply) then
                 Reader sends QueryRep;
             end
             else if (No tag replies) then
                  Reader sends QueryRep;
             end
        if (Tag receives QueryRep) then
             slot counter = slot counter - 1;
        end
    end
    Reader uses PTES to adjust the size of the new frame;
    Reader sends QueryAdjust;
```

Algorithm 1: DFSA algorithm with PTES Frame-Size Prediction

2) MDFSA Rules: Instead of splitting tags into group by using the power of two (2,4,8...) as in EDFSA approach, the MDFSA approach derived new rules using particular equations according to the optimal system efficiency obtained for specific number of tags. We first conducted an experiment to acquire optimal frame-size for specific number of tags as shown in Figure 2. It can be seen that the optimal system efficiency achieved by the probabilistic ALOHA method is approximately 38% and the optimal number of tags are close to the maximum frame-size. Efficiency is calculated by:

$$Efficiency = (\frac{S}{S+C+E})$$

Where S is Successful slots, C is Collision slots, and E is Empty slots.

From Figure 2, we have derived equations 1, 2, 3, 4 and 5 to find a minimum and maximum number of tags suitable for particular *frame-size*. These equations are then used to exploit rules for MDFSA.

In this study, we proposed two rules for MDFSA: MDFSA and MDFSA-Extended (MDFSA-E). All rules split the number of *Backlog* into groups then used one of Q8 (frame-size 256), Q7 (frame-size 128), or Q6 (frame-size 64), to identify the current set of tags.

$$\begin{aligned} \max &= 2^Q + 2^{(Q-1)} - 2^{(Q-2)} + 2^{(Q-3)} \\ \min &= (2^{(Q-1)} + 2^{(Q-2)} - 2^{(Q-3)} + 2^{(Q-4)}) + 1 \end{aligned} \tag{1}$$

Equation 1 is used to calculate the minimum and maximum number of tags, in the case of only 'Q8' is applied during the identification cycle. For both MDFSA and MDFSA-E

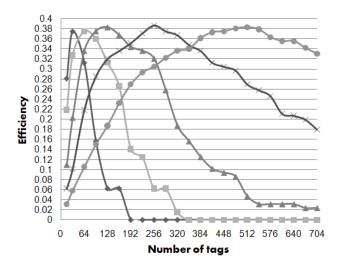


Fig. 2. Performance efficiency of different frame-size (2^Q) on different number of tags.

rules, we acquired the maximum number of 352 tags and the minimum number of 177 tags after rewritten Equation 1 (as displayed in Table I and II).

$$\begin{aligned} \max &= (2^Q + 2^{(Q-1)} - 2^{(Q-2)} + 2^{(Q-3)}) + \\ &\quad (2^{(Q-2)} + 2^{(Q-3)} - 2^{(Q-4)} + 2^{(Q-5)}) \\ \min &= (2^Q + 2^{(Q-1)} - 2^{(Q-2)} + 2^{(Q-3)}) + 1 \end{aligned} \tag{2}$$

Equation 2 is used to calculate the minimum and maximum number of tags, in the case of two types of 'Q8' and 'Q6' is applied in the MDFSA-E rule. Thus, after rewritten Equation 2, we obtained the maximum number of 440 tags and the minimum number of 353 tags (as shown in Table II).

$$\max = (2^{Q} + 2^{(Q-1)} - 2^{(Q-2)} + 2^{(Q-3)}) + (2^{(Q-1)} + 2^{(Q-2)} - 2^{(Q-3)} + 2^{(Q-4)})$$

$$\min = (2^{Q} + 2^{(Q-1)} - 2^{(Q-2)} + 2^{(Q-3)}) + 1$$
(3)

$$\max = (2^{Q} + 2^{(Q-1)} - 2^{(Q-2)} + 2^{(Q-3)}) + (2^{(Q-1)} + 2^{(Q-2)} - 2^{(Q-3)} + 2^{(Q-4)})$$

$$\min = [(2^{Q} + 2^{(Q-1)} - 2^{(Q-2)} + 2^{(Q-3)}) + (2^{(Q-2)} + 2^{(Q-3)} - 2^{(Q-4)} + 2^{(Q-5)})] + 1$$
(4)

Equation 3 and 4 are used to calculate the minimum and maximum number of tags for the MDFSA and MDFSA-E rules respectively, in the case of two types of 'Q8' and 'Q7' is used during the identification process. After rewritten Equation 3, we acquired the maximum number of 528 tags and the minimum number of 353 tags (MDFSA rule: Table I). In addition, after rewritten Equation 4, we obtained the maximum

number of 528 tags and the minimum number of 441 tags (MDFSA-E rule: Table II).

$$\begin{aligned} \max &= (2^Q + 2^{(Q-1)} - 2^{(Q-2)} + 2^{(Q-3)}) + \\ &\quad (2^{(Q-1)} + 2^{(Q-2)} - 2^{(Q-3)} + 2^{(Q-4)}) + \\ &\quad (2^{(Q-2)} + 2^{(Q-3)} - 2^{(Q-4)} + 2^{(Q-5)}) \\ \min &= [(2^Q + 2^{(Q-1)} - 2^{(Q-2)} + 2^{(Q-3)}) + \\ &\quad (2^{(Q-1)} + 2^{(Q-2)} - 2^{(Q-3)} + 2^{(Q-4)})] + 1 \end{aligned} \tag{5}$$

Equation 5 is used to calculate the minimum and maximum number of tags, in the case of three types of 'Q8', 'Q7', and 'Q6' is applied in the MDFSA-E rule. After rewritten Equation 5, we acquired the maximum number of 616 tags and the minimum number of 529 tags (as presented in Table II).

TABLE I MDFSA Rule - The Number of Unread Tags, Optimal Frame-Size (A and B), and Number of Group (A and B).

Backlogs	FS A	GS A	FS B	GS B
1233 to 1408	256	4	-	-
1057 to 1232	256	3	128	1
881 to 1056	256	3	-	-
705 to 880	256	2	128	1
529 to 704	256	2	-	-
353 to 528	256	1	128	1
177 to 352	256	1	-	-
89 to 176	128	1	-	-
45 to 88	64	1	-	-
23 to 44	32	1	-	-
12 to 22	16	1	-	-
6 to 11	8	1	-	-

TABLE II MDFSA-E Rule - The Number of Unread Tags, Optimal Frame-Size (A, B, C), and Number of Group (A, B, C).

Backlogs	FS A	GS A	FS B	GS B	FS C	GS C
1320 to 1408	256	4	-	-	-	-
1233 to 1320	256	3	128	1	64	1
1145 to 1232	256	3	128	1	-	-
1057 to 1144	256	3	-	-	64	1
969 to 1056	256	3	-	-	-	-
881 to 968	256	2	128	1	64	1
793 to 880	256	2	128	1	-	-
705 to 792	256	2	-	-	64	1
617 to 704	256	2	-	-	-	-
529 to 616	256	1	128	1	64	1
441 to 528	256	1	128	1	-	-
353 to 440	256	1	-	-	64	1
177 to 352	256	1	-	-	-	-
89 to 176	128	1	-	-	-	-
45 to 88	64	1	-	-	-	-
23 to 44	32	1	-	-	-	-
12 to 22	16	1	-	-	-	-
6 to 11	8	1	-	-	-	-

For both MDFSA and MDFSA-E rules, if the number of unread tags is larger than 352, to achieve the optimal system efficiency we must divide the tags into two or more groups.

For the number of unread tags smaller than 352, we must let every unread tag responds. By doing so, we can always achieve the expected system efficiency as demonstrated in Figure 2.

Table I shows the MDFSA rule. For instance, if the number of *Backlog* equals to 1100 tags, the MDFSA algorithm will split the unread tags into 3 groups of Q8 (256) and 1 group of Q7 (128).

Table II displays the MDFSA-E rule. For example, if the number of *Backlog* equals to 900 tags, the MDFSA-E algorithm will split the unread tags into 2 groups of Q8 (256), 1 group of Q7 (128), and 1 group of Q6 (64).

IV. EXPERIMENTAL EVALUATION

In order to show the significance of our proposed method, we conducted an experimental evaluation and compared our method to the existing approaches. In this section, we describe the data sets used in the experiment, present the results, and provide the analysis.

A. Data Sets

The aim of the experiment is to compare the performance of our proposed MDFSA method to the existing DFSA and EDFSA approaches. The number of tags simulated for the experiment is between 400 and 1400 tags. While performing each tag anti-collision algorithm, the number of tags is supposedly unknown.

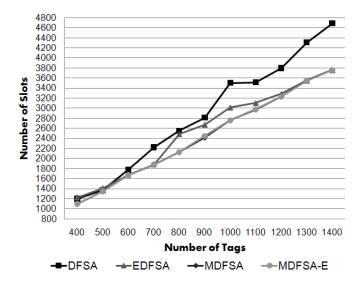


Fig. 3. Number of slots comparison for DFSA, EDFSA, and MDFSA methods on different number of tags.

B. Results and Analysis

Our experiment compares the performance of our proposed MDFSA method to the existing DFSA and EDFSA approaches. From Figure 3, it can be seen that both MDFSA and MDFSA-E produced minimal number of *slots* during the identification process compared to other methods. Specifically, the MDFSA and MDFSA-E techniques minimised the number of slots from the EDFSA approach; when the number of tags

is between 400 and 500 tags, and between 800 and 1200 tags. This is because the number of group sets for EDFSA will be doubled when the number of *Backlog* reached the specific threshold; while the MDFSA and MDFSA-E increased number of groups slowly according to the estimated number of unread tags. Thus, both MDFSA and MDFSA-E methods performed mutually in terms of number of *slots* minimisation.

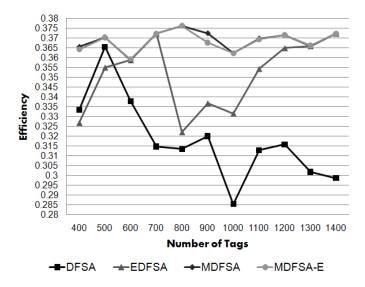


Fig. 4. Performance efficiency for DFSA, EDFSA, and MDFSA methods on different number of tags.

Figure 4 demonstrates that both MDFSA and MDFSA-E approaches maintained its system efficiency above other methods and has the most stable performance. Nevertheless, the MDFSA-E required additional number of group sets from the MDFSA method throughout the identification process (see Table I and II). As a result, the MDFSA-E approach required extra time to initiate a new group as opposed to the MDFSA method. On the other hand, the DFSA's efficiency dropped dramatically when the number of tags increase, while the EDFSA's efficiency become unstable during the time when number of group set twice over from 1 to 2 and from 2 to 4 (between 400 to 500, and 800 to 1200 tags).

TABLE III
PERCENTAGE IMPROVEMENT OF THE PROPOSED MDFSA VERSUS EXISTING EDFSA AND DFSA TECHNIQUES.

	MDFSA		MDFSA-E		
	% from	% from	% from	% from	
	EDFSA	DFSA	EDFSA	DFSA	
400	10.62	8.83	10.29	8.50	
500	4.12	1.32	4.12	1.32	
600	0.00	5.86	0.12	5.97	
700	0.00	15.47	0.00	15.47	
800	14.41	16.69	14.41	16.69	
900	9.58	14.08	8.38	12.94	
1000	8.49	21.23	8.49	21.23	
1100	4.19	15.42	4.06	15.30	
1200	1.76	15.00	1.76	15.00	
1300	0.00	17.55	0.06	17.60	
1400	0.00	19.80	0.00	19.80	
Average	4.83	13.75	4.70	13.62	

Table III shows the percentage of improvement of the proposed MDFSA and MDFSA-E methods versus the EDFSA and DFSA methods. It can be seen that during the time when the number of group doubled from 1 to 2 (400 tags) and from 2 to 4 (800 tags), the MDFSA and MDFSA-E show the highest percentage of improvement in comparison with the EDFSA method. On the other hand, the percentage of improvement increased more stably compared with the DFSA method, since the DFSA method does not imply group splitting rules.

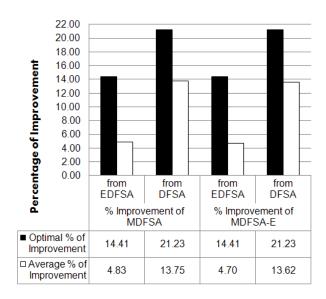


Fig. 5. Percentage of improvement of MDFSA compared to existing methods.

Both MDFSA and MDFSA-E methods have better performances than the EDFSA approach by about 5% on average, while it is approximately 14% better than the DFSA method as demonstrated in Figure 5. The optimal percentage of improvement of our proposed methods can achieve up to 14% and 21% compared with the EDFSA and DFSA respectively, depending on the number of tags within the interrogation zone. However, the MDFSA-E method required additional number of groups from the MDFSA method and acquired slightly lower percentage of improvement compared with the MDFSA method. Thus, we conclude that the MDFSA method is the most effective method in terms of system efficiency and number of slots minimisation.

V. CONCLUSION

In this study, we have identified the significance of RFID tag anti-collision and developed an efficient method to minimise the tag starvation problem. We have proposed the Modified Dynamic Framed-Slotted ALOHA (MDFSA) technique to maximise the performance efficiency and to reduce the total number of slots queried during the tag identification process. From the result and analysis of the experimentation, we have indicated that the proposed MDFSA has produced a minimal number of slots when compared with other methods. Specifically, the MDFSA displays higher performance efficiency when compared to EDFSA method by about 5% on average,

while it is approximately 14% better than the DFSA approach. Therefore, we conclude that the MDFSA is the most effective method. It has maintained its system efficiency above other methods, and has the highest achieving performance.

REFERENCES

- [1] J. Myung and W. Lee, "Adaptive splitting protocols for RFID tag collision arbitration," in *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2006, pp. 202–213.
- [2] J. Ryu, H. Lee, Y. Seok, T. Kwon, and Y. Choi, "A Hybrid Query Tree Protocol for Tag Collision Arbitration in RFID systems," in *MobiHoc* '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing. Glasgow, UK: IEEE Computer Society, 2007, pp. 5981–5986.
- [3] P. Pupunwiwat and B. Stantic, "Joined Q-ary Tree Anti-Collision for Massive Tag Movement Distribution," in *Thirty-Third Australasian Computer Science Conference (ACSC 2010)*, ser. CRPIT, B. Mans and M. Reynolds, Eds., vol. 102. Brisbane, Australia: ACS, 2010, pp. 99–108
- [4] C. H. Quan, W. K. Hong, and H. C. Kim, "Performance Analysis of Tag Anti-collision Algorithms for RFID Systems," in *Emerging Directions* in *Embedded and Ubiquitous Computing*, vol. 4097. Seoul, Korea: Springer Berlin/Heidelberg, 2006, pp. 382–391.
- [5] J. D. Shin, S. S. Yeo, T. H. Kim, and S. K. Kim, "Hybrid Tag Anticollision Algorithms in RFID Systems," in *Computational Science ICCS* 2007, vol. 4490. Beijing, China: Springer Berlin/Heidelberg, 2007, pp. 693–700.
- [6] J. Ding and F. Liu, "Novel Tag Anti-Collision Algorithm with Adaptive Grouping," Wireless Sensor Network (WSN), vol. 1, no. 5, pp. 475–481, 2009.
- [7] S. R. Lee and C. W. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Anti-collision Algorithm," in *Emerging Directions in Embedded and Ubiquitous Computing*. Seoul, Korea: Springer Berlin/Heidelberg, 2006, conf, pp. 403–412.
- [8] J. G. Lee, S. J. Hwang, and S. W. Kim, "Performance Study of Anticollision Algorithms for EPC-C1 Gen2 RFID Protocol," in *Informa*tion Networking. Towards Ubiquitous Networking and Services, vol. 5200/2008. Estoril, Portugal: Springer Berlin/Heidelberg, 2008, pp. 523–532.
- [9] M. R. Devarapalli, V. Sarangan, and S. Radhakrishnan, "AFSA: an efficient framework for fast RFID tag reading in dense environments," in QSHINE '07: The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness Workshops. New York, NY, USA: ACM, 2007, pp. 1–7.
- [10] F. C. Schoute, "Dynamic Frame Length ALOHA," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 565–568, 1983.
- [11] P. Pupunwiwat and B. Stantic, "Dynamic Framed-Slot ALOHA Anti-Collision using Precise Tag Estimation Scheme," in *Twenty-First Australasian Database Conference (ADC 2010)*, ser. CRPIT, H. T. Shen and A. Bouguettaya, Eds., vol. 104. Brisbane, Australia: ACS, 2010, pp. 19–28.
- [12] H. Vogt, "Efficient Object Identification with Passive RFID Tags," in Pervasive '02: Proceedings of the First International Conference on Pervasive Computing. London, UK: Springer-Verlag, 2002, pp. 98– 113.
- [13] C. Floerkemeier, "Bayesian Transmission Strategy for Framed ALOHA Based RFID Protocols," in RFID, 2007. IEEE International Conference on RFID Gaylord Texan Resort, Grapevine, TX, USA, 2007, pp. 228– 235.
- [14] B. Li, Y. Yang, and J. Wang, "Anti-collision Issue Analysis in Gen2 Protocol - Anti-collision issue analysis considering capture effect," *Auto-ID Labs White Paper.*, 2009, http://www.autoidlabs.org/single-view/dir/article/6/320/page.html.
- [15] X. Fan, I. Song, and K. Chang, "Gen2-based hybrid tag anti-collision Q algorithm using Chebyshev's inequality for passive RFID systems," in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC* 2008. IEEE 19th International Symposium on, Cannes, France, 2008, pp. 1–5.
- [16] X. Fan, I. Song, K. Chang, D. B. Shin, H. S. Lee, C. S. Pyo, and J. S. Chae, "Gen2-Based Tag Anti-collision Algorithms Using Chebyshev's Inequality and Adjustable Frame Size," vol. 30, no. 5, 2008, pp. 653–662.