

An Enhanced Dynamic Framed Slotted ALOHA Anti-collision Algorithm*

Su-Ryun Lee¹ and Chae-Woo Lee²

¹Samsung Electronics, Suwon, Korea

²School of Electrical and Computer Engineering, Ajou University
San 5 Wonchon-dong Yeoungtong-gu, Suwon, Korea
srlee@ajou.ac.kr, cwlee@ajou.ac.kr

Abstract. In RFID system, one of the problems that we must solve is the collision between tags that may occur when a reader attempts to read a number of tags simultaneously. Most of the popular anti-collision algorithms are based on the ALOHA-type algorithm, which are simple and show good performance when the number of tags to read is small. However, for ALOHA-type algorithms the time required to read the tags generally increases exponentially as the number of tag increases. Thus, they are not very efficient when the number of tags to read is large. In the paper, we propose a new anti-collision algorithm called Enhanced Dynamic Framed Slotted ALOHA (EDFSA) in which the time required to read the tags increases linearly with the number of tags. The proposed algorithm estimates the number of unread tags first and adjusts the number of responding tags or the frame size to give the optimal system efficiency. Simulation results show that the proposed algorithm improves the reading rate of data of the tags by 85~100% compared to the conventional algorithms.

1 Introduction

Recently RFID (Radio Frequency IDentification) attracts attention as an alternative to the bar code in the distribution industry, supply chain and banking sector. This is because RFID system reads data without close context to the object it intends to read and stores more data than the bar code does. One of the largest disadvantages in RFID system is its low tag identification efficiency by tag collision especially when the number of tags in its readable range is large. Tag collision is the event that the reader cannot identify the data of tag when more than one tag occupy the same RF communication channel simultaneously. For a solution to the problem, the existing methods have to increase data transmission speed by extending frequency bandwidth or tag identification efficiency by minimizing tag collision. However, it is impossible to extend a frequency

* This work was supported in part by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21th Century Frontier R&D Program in republic of Korea.

bandwidth because usable frequency bands are limited. Therefore we must reduce tag collision for increasing tag identification efficiency. So far, several tag anti-collision algorithms have been proposed. Among them, the most widely used are framed slotted ALOHA algorithm and binary search algorithm. Due to its simple implementation, framed slotted ALOHA algorithm is used frequently [1].

The ISO/IEC 18000-6 describes the parameters for the air interface communications at 860MHz to 960MHz. This standard specifies two tag types; TYPE A and TYPE B [2]. In the standard, the reader uses frequency hopping spread spectrum. To read the data of TYPE A tag, a reader uses the ALOHA Protocol as the tag anti-collision algorithm and PIE (Pulse Interval Encoding), which is a kind of ASK (Amplitude Shift Keying), as the carrier modulation method. To read the data of TYPE B tag, the reader applies Binary tree algorithm as the tag anti-collision algorithm and Manchester coding as the carrier modulation method. The tag transmits its data to the reader by modulating the incident energy and reflecting it back to the reader.

The EPC Class 1 Generation 2 describes the protocols of RFID system operating in the 860~960MHz frequency range [3]. The EPC Class 1 Generation 2 was adopted as ISO/IEC 18000-6 TYPE C standard. In this standard, the reader uses frequency hopping spread spectrum method to occupy the channel. The communication between the reader and tags in EPC Class 1 Generation 2 is similar to that of the ISO/IEC 18000-6 TYPE A. As the tag anti-collision algorithm the reader uses Slotted Random anti-collision algorithm that is a kind of ALOHA protocols and PIE as the carrier modulation method.

As the most RFID systems use passive tags, frame sizes are limited in the framed slotted ALOHA algorithm [1]. In the algorithm, a tag randomly selects a slot number in the frame and responds to the reader using the slot number it selected. In this method, when the number of tags is small, the probability of tag collision is low. Thus the time used to identify the all tags is relatively short. However, as the number of tags increases, the probability of tag collision becomes higher and the time used to identify the tags increases rapidly. This problem is inevitable if the number of tags that attempt to access the fixed number of ALOHA slots increases. To solve this problem, we propose an algorithm that limits the number of responding tags to the number that has the maximum efficiency when there are large number of tags. Therefore, this algorithm improves the efficiency of tag identification and then lineally increases the required time for tag identification even if there are a number of tags.

The remainder of this paper is organized as follows. Section 2 introduces a set of framed slotted ALOHA algorithms and points out their weakness. Section 3 proposes and analyzes new anti-collision algorithm called Enhanced Dynamic Framed Slotted ALOHA (EDFSA). Section 4 compares the proposed algorithm with existing algorithm. Finally, section 5 concludes the paper.

2 Previous Work

Slotted ALOHA algorithm is the tag identification method that each tag transmits its serial number to the reader in the slot of a frame and the reader identifies

the tag when it receives the serial number of the tag without collision. A time slot is a time interval that tags transmit their serial number. The reader identifies a tag when a time slot is occupied by only one tag. The current RFID system uses variants of slotted ALOHA known as framed slotted ALOHA algorithms. A frame is a time interval between requests of a reader and consists of a number of slots. This section briefly describes the existing framed slotted ALOHA anti-collision algorithms and compare their performance.

2.1 Basic Framed Slotted ALOHA (BFSA)

The BFSA uses a fixed frame and does not change its size. The reader offers information to the tags about the frame size and the random number which is used to select a slot in the frame. Each tag selects a slot to access using the random number [4].

Figure 1 is an example that shows how BFSA works. In the first read cycle, Tag 1 and Tag 3 simultaneously transmit their serial numbers in Slot 1. Tag 2 and Tag 5 transmit their serial numbers in Slot 2. As those tags are collided each other, Tag 1, 2, 3 and 5 must respond to the next request from the reader. The reader can identify Tag 4 in the first reader cycle because there is only one tag response in the time Slot 3. In the example, the frame size is set to three slots.

Since the frame size of the BFSA is fixed, its implementation is simple. However, the efficiency of tag identification is low. For instance, no tag may be identified in a read cycle if there are too many tags because all the slots may be filled with collision. Too many slots will be left idle if large frame size is used when the number of tags is small.

2.2 Dynamic Framed Slotted ALOHA (DFSA)

The DFSA changes the frame size for efficient tag identification. To determine the frame size, it uses the information such as the number of slots used to identify the tag, the number of the slots collided and so on. The DFSA has several versions depending on the methods changing the frame size. Among them, we will briefly explain the two popular methods described in [1].

The first algorithm regulates the frame size using the number of the empty slots, the slots with collision and the slots filled with one tag. When the probability of collision is higher than the upper threshold, the reader increases the frame

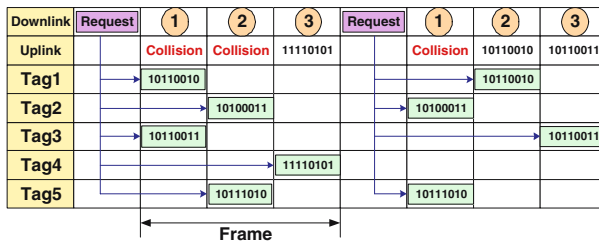


Fig. 1. The process of the BFSA

size. If the collision probability is lower than the lower threshold, the reader decreases the frame size. Because the reader starts a read cycle with the minimum frame size, when the number of tag is small it can identify the tags efficiently without increasing the frame size much. When the number of tags is large, the reader changes its frame size so as to decrease the collision probability.

The second algorithm starts a read cycle with the initial frame size which is either two or four. If no tag is identified during the previous read cycle, it increases the frame size and starts another read cycle. It repeats this until at least one tag is identified. If a single tag is identified it immediately stops the current read cycle and starts to read another tag with the initial frame size.

DFSA can identify the tag efficiently because the reader regulates the frame size according to the number of tags. However, the frame size change alone can not reduce sufficiently the tag collision when there are a number of tags because it can not increase the frame size indefinitely. In the second method, when the number of tags is small, then it can identify all the tag without too much collision. However, if the number of tags is large, it needs exponentially increasing number of slots to identify the tags because it always starts with the initial minimum frame size after identifying a tag, regardless how many tags are unread.

2.3 Advanced Framed Slotted ALOHA (AFSA)

The AFSA estimates the number of tags and determines a proper frame size that improves the efficiency [7][8]. Thus it has better performance than the BFSA.

In the AFSA, the number of tags is estimated using the result of a read cycle such as the number of empty slots, slots filled with one tag, and slots with collision. To estimate the number of tags, The AFSA uses an estimation function shown in Equation (1). According to Chebyshev’s inequality, the outcome of a random experiment involving a random variable X is most likely somewhere near the expected value of X . The estimation function uses this property. Thus it measures the difference between the real results and the expected values to estimate the number of tags for which difference becomes minimal [6].

The number of tags is estimated using both the frame size (N) used in the read cycle and the results of the previous read cycle as a triple of numbers $\langle c_0, c_1, c_k \rangle$ that quantify respectively the empty slots, slots filled with one tag, and slots with collision as Equation (1) [7][8].

$$\varepsilon_{vd}(N, c_0, c_1, c_k) = \min \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_{\geq 2}^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right| \tag{1}$$

In Equation (1), $\langle a_0^{N,n}, a_1^{N,n}, a_{\geq 2}^{N,n} \rangle$ are respectively the expected number of the empty slots, slots filled with one tag, and slots with collision where N and n respectively denote the frame size and the number of tags.

In the AFSA, it was assumed that the tags already read also respond to the reader during other read cycle. Varying the frame size, the AFSA calculates how many slots are need to read 99% of the tags. Then it selects the frame size which

gives the smallest number of slots. Because the AFSA estimates the number of tags and determines the frame size to minimize the collision probability, it is more efficient than the other algorithms. However, the AFSA has the same problem that it can not increase the frame size indefinitely as the number of tags increases. Thus, this algorithm works well if the number of tags is relatively small, however, if the number becomes large it begins to show poor performance [7][8]. Furthermore, this method can not be applied to the tag that is deactivated once it is read.

3 Enhanced Dynamic Framed Slotted ALOHA (EDFSA)

The previous framed slotted ALOHA algorithms change the frame size to increase the efficiency of the tag identification. However, as the number of tags becomes larger than the frame size, the probability of tag collision increases rapidly [5][9]. This problem can not be solved without restricting the number of responding tags approximately the same as the frame size as we will explain later in this paper. In the following subsection, we propose Enhanced Dynamic Framed Slotted ALOHA algorithm which solves this problem.

3.1 Description of the EDFSA

If we can estimate the number of unread tags, we can determine the frame size that will maximize the system efficiency or minimize the tag collision probability. In general, when the number of tags is large, we can reduce the probability of tag collision by increasing the frame size. Because we can not increase the frame size indefinitely, when the number of unread tags is too large to achieve high system efficiency, we must somehow restrict the number of responding tags so that the optimal number of tags responds to the given frame size. When the number of unread tags is too small to achieve the optimal system efficiency, we must reduce the frame size. The system efficiency is defined as the ratio of the slots filled with one tag to the current frame size. If we know the number of unread tags and the frame size, we can calculate the system efficiency. The estimated number of unread tags can be obtained by Equation (1).

The EDFSA estimates the number of unread tags first. If the number is much larger than the maximum frame size, to improve collision performance, it divides the unread tags into a number of groups and allows only one group of tags to respond. In the algorithm once the number of tags that should respond is determined, we can calculate the ratio of the responding tags to the total unidentified tags. With that ratio, the reader requests to respond to the all unidentified tags having zero remainder after the modulo operation. In every read cycle, the reader estimates the number of unread tags and calculates the number of groups that gives the maximum throughput during next read cycle. If the frame size is larger than the one that gives the optimal system efficiency, then the reader starts to reduce the frame size so that it can achieve the optimal efficiency with the estimated number of unread tags.

When the reader limits the number of responding tags, it transmits the number of tag groups and a random number to the tags when it broadcasts a request. The tag that received the request generates a new number from the received random number and its serial number and divides the new number by the number of tag groups. Only the tags having the remainder of zero respond to the request. When the number of estimated unread tags is below the threshold, the reader adjusts the frame size without grouping the unread tags. It means the reader broadcasts a read request with a frame size, a random number and the number of tags groups of one in this case. After each read cycle, the reader estimates the number of unread tags and adjust its frame size. This repeats until all the tags are read.

3.2 Analysis of the EDFSA

Generally in the framed slotted ALOHA anti-collision method, the system efficiency begins to increase as the the number of responding tags becomes larger and it begins to decrease if the number of responding tags increases more than a certain point. We will derive the optimum number of tags that should respond if the frame size is given.

When the reader uses a frame size of N and the number of responding tags is n , the probability that r tags exist in one given slot is a binomial distribution as follows.

$$B_{n, \frac{1}{N}}(r) = \binom{n}{r} \left(\frac{1}{N}\right) \left(1 - \frac{1}{N}\right)^{n-1} \quad (2)$$

The expected number of read tags during one read cycle is given as follows.

$$a_1^{N,n} = N \cdot B_{n, \frac{1}{N}}(1) = N \cdot n \left(\frac{1}{N}\right) \left(1 - \frac{1}{N}\right)^{n-1} \quad (3)$$

where $a_r^{N,n}$ denotes the number of slots with r tags with the frame size of N and n unread tags. Then the system efficiency is calculated as follows.

$$\text{System Efficiency} = \frac{\text{the number of slots filled with one tag}}{\text{current frame size}} = a_1^{N,n}/N \quad (4)$$

We can obtain the number of tags that gives the maximum system efficiency by differentiating Equation (3).

$$\begin{aligned} \frac{d(a_1^{N,n})}{dn} &= (1 - 1/N)^{n-1} + n(1 - 1/N)^{n-1} \ln(1 - 1/N) \\ &= (1 - 1/N)^{n-1} \{1 + n \ln(1 - 1/N)\} \\ &= 0 \end{aligned} \quad (5)$$

Solving the above equation, we can derive the optimal number of responding tags with the frame size of N as follows.

$$n = \left\lceil -\frac{1}{\ln(1 - 1/N)} \right\rceil \quad (6)$$

When the number of tags is n , the optimal frame size can be derived as follows.

$$N = \frac{1}{1 - e^{-\frac{1}{n}}} = \frac{e^{\frac{1}{n}}}{e^{\frac{1}{n}} - 1} \tag{7}$$

When n is large, using Taylor series we can simplify the above equation as follows.

$$N \simeq \frac{1 + \frac{1}{n}}{1 + \frac{1}{n} - 1} = n + 1, \quad n \gg 1 \tag{8}$$

The above equation tells us that when the number of tags and the frame size are approximately the same, the system efficiency becomes the maximum [5].

From Equation (8), we can conclude that if we restrict the number of responding tags similar to the frame size we can achieve maximum system efficiency. If the number of unread tags is sufficiently large (i.e., larger than the frame size), we can restrict the tag response by grouping the tags and allowing only one group to respond and this can be done by Modulo operation. The number of groups or the Modulo (M) is calculated as follows.

$$M = \left\lceil \frac{\text{The number of unread tags}}{N} \right\rceil \tag{9}$$

where N denotes the maximum frame size.

In this paper, considering the implementation complexity we assume that the EDFSA uses the power of two (2,4,8, ...) for grouping the tags. Then the modulo operation can be simply done using the shift register. Figure 2-(a) shows the system efficiency as the number of tags increase while the frame size is set to $N = 256$. In the figure we can see that the maximum system efficiency can be achieved when the number of unread tags and the frame size are approximately the same and it is 36.8%.

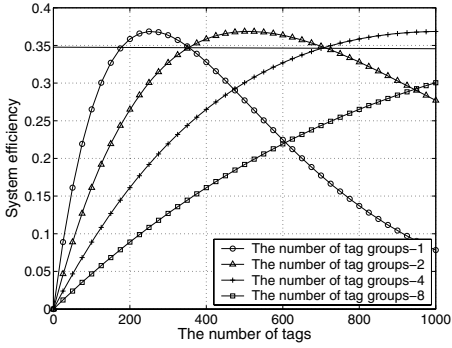
From the figure we can determine a specific number of tag groups which gives better system efficiency than others. When the number of unread tags is near or less than the frame size, we can achieve higher system efficiency if we do not use the Modulo operation and decrease the frame size. Figure 2-(b) shows how system efficiency changes when we vary the frame size.

The EDFSA chooses the frame size and the Modulo that give better performance than any other combination of the two may provide. For example, as we see in the figure, the number of tags that produces the same expected system efficiency whether we apply Modulo 2 operation or Modulo 1 operation can be obtained as follows.

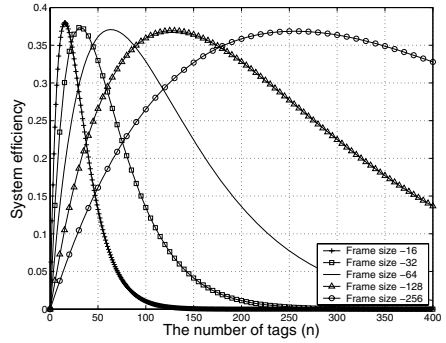
$$\frac{a_1^{256, n/2}}{256} = \frac{a_1^{256, n}}{256} \tag{10}$$

We can rewrite the above equation as follows.

$$\frac{n}{2} \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{\frac{n}{2} - 1} = n \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{n - 1} \tag{11}$$



(a) System efficiency vs. number of tags



(b) System efficiency vs. frame size

Fig. 2. System efficiencies

Therefore, we obtain

$$n = 354. \tag{12}$$

If the number of unread tags is slightly larger than 354, to achieve the optimal system efficiency we must divide the tags into two groups, and for the number of unread tags slightly smaller than 354 we must let every unread tag respond. By doing this, we can always obtain the expected system efficiency between 34.6% to 36.8 %. Table 1 shows example values.

Table 1. The number of unread tags vs. optimal frame size and Modulo

The number of unread tags	Frame Size	Number of tag groups (M)
708 – 1416	256	4
355 – 707	256	2
177 – 354	256	1
82 – 176	128	1
41 – 81	64	1
20 – 40	32	1

4 Performance Analysis of the EDFSA

We compare the EDFSA with the BFSa and the first algorithm of the Increase method of the DFSA. In the following, the first algorithm of the increase method of the DFSA is just called 'the Increase method'. We assume that the maximum frame size of each algorithm is 256 slots. We also assume that the BFSa uses the fixed frame size of 256 slots and the Increase method increases frames size from 16 slots to 256 slots. In the Increase method, we assume that the frame size doubles when the number of slots with collisions is more than 70% of the current frame size. We also assume that when the number of empty slots is more

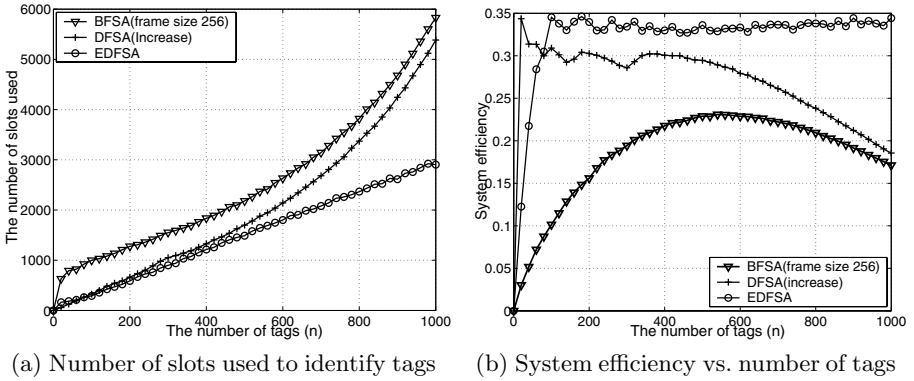


Fig. 3. Simulation results

than 30% of the current frame size, the reader reduces the current frame size in half. The EDFSA is assumed to have the initial frame size of 128 slots. When no tag is read during a read cycle, we assume that all the tags have been read and finished simulation.

From Figure 3, we can observe that as the number of tags increased, for both the BFSA and the increase methods the number of slots needed to read the tags increased exponentially while it increased linearly for the EDFSA. The system efficiency for the BFSA and Increase algorithms were below 20% because most of the slots were wasted by tag collision when the number of tags is large. When the the number of unread tags is small, most of the slots are left idle.

In the figure, the Increase method showed better performance than the BFSA because the Increase method can decrease the frame size when the number of unread tags becomes small, while the BFSA maintains its initial frame size of 256 regardless of the number of unread tags.

The number of slots used for the EDFSA to read the tags increased linearly as the number of tags did. This is because the EDFSA can partition the tag according to its maximum frame size when the number of unread tags was larger than its maximum frame size and reduces its frame size when the number of unread tags is smaller than the maximum frame size. Ideally the algorithm is able to maintain the system efficiency between 34.6% and 36.8% on the average regardless of the number of unread tags. In the simulation, the system efficiency of the algorithm showed the system efficiency of around 33% which is slightly less than the ideal, because of the initial frame size of 128 which was fixed regardless of the number of tags.

From Figure 3 we can also observe that when the number of tags is 1000, the EDFSA exhibits performance improvement of 100% and 85% compared with the BFSA and the Increase methods respectively. If the number of tags is larger we will be able to observe more dramatic performance improvement.

Though the simulation results use the number of slots as a performance metric, we believe the overall results will be very similar to the actual time it takes to read the tags because the reader generates a request just once every read cycle

and the time of the reader request is very small in the case of using a large frame size [4].

5 Conclusions

In this paper, we proposed an enhanced ALOHA anti-collision algorithm that is simple to implement but very efficient. The system efficiency becomes maximum when we set the frame size and the number of unread tags equal. For the simplicity of implementation, we used the power of two for the frame size and the number of tag groups. Despite the restrictions, we were able to maintain the system efficiency between 34.6 % and 36.8 %. This means that the number of slots needed to read the tags always increases linearly as the number of tags does. Theoretical maximum system efficiency is 36.8 % for framed slotted ALOHA. To verify the effectiveness of our algorithm we ran simulations and found that when the number of tags is 1000, our algorithm showed 85% to 100% the improvement of the reading rate over the other two comparing anti-collision algorithms.

In the algorithm though we may improve the performance of the proposed algorithm if we use natural numbers instead of the power of two when selecting the frame size and the number of tag groups. However, the performance improvement is not significant. When we use the number with the power of two, we are achieving the system efficiency of at least 34.6%, while we can do 36.8% if everything is set to optimal. Thus, the algorithm we proposed is simple to implement while achieving the performance close to theoretical maximum.

References

1. K. Finkenzerler. *RFID handbook - Second Edition*. JOHN WILEY & SONS, (2003)
2. ISO/IEC JTC 1/SC 32/WG 4: Information Technology automatic identification and data capture techniques - Part 6: Parameter for air interference communications at 860MHz to 960MHz, ISO/IEC FDIS 18000-6, November (2003)
3. EPCglobal: EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz - 960MHz, Ver. 1.0.9, Jan. (2005)
4. PHILIPS Semiconductor. *I-CODE1 System Design Guide: Technical Report*. May (2002)
5. R. Rom and M. Sidi. *Multiple Access Protocols/Performance and Analysis*. Springer-Verlag, (1990) 47-77
6. W. A. Shewhart and S. S. Wilks. *An Introduction to Probability Theory and Its Application - Second Edition*. Wiley publications, (1960)
7. H. Vogt.: Multiple Object Identification with Passive RFID Tags. 2002 IEEE International Conference on Systems, Man and Cybernetics. vol:3, October (2002)
8. H. Vogt.: Efficient Object Identification with Passive RFID Tags. Proc. Pervasive 2002. (2002) 98-113.
9. J.E. Wieselthier, A. Ephremides, and L.A. Michels.: An Exact Analysis and Performance Evaluation of Framed ALOHA with Capture. IEEE TRANSACTION ON COMMUNICATIONS, vol:37 February (1989) 125-137