Novel Dynamic Framed-Slotted ALOHA Using Litmus Slots in RFID Systems

Soon-Bin YIM†, Jongho PARK†, Nonmembers, and Tae-Jin LEE†, Member

1. Introduction

Radio Frequency IDentification (RFID) is a wireless technology that identifies objects by the tags attached to them. It is expected that RFID systems will be widely deployed to automatically identify various objects [1]. An RFID system typically consists of RFID tags and a reader; the RFID reader tries to read out the information (IDs) of tags. One of the main objectives of RFID systems is to read the tags within a certain identification range as quickly as possible, e.g., a checking out system of a supermarket and an inventory management system of a warehouse [2]. The speed of identification is very important for such systems. Even if the RF reading environment for an RFID tag is ideal, its still an engineering challenge to support multiple collocated tags. Consider two tags situated next to each other and equidistant from the reader. On hearing the readers signal, both would acquire enough power to turn on and transmit a response back to the reader, resulting in a collision. The data from both tags would be superimposed and garbled.

When many tags transmit their IDs simultaneously in the reading range of a single reader, the reader may not identify their IDs due to collision. So, an anti-collision protocol is required to identify multiple tags quickly within a reader’s coverage area. In general, tag anti-collision protocols can be categorized into tree-based [4] and ALOHA-based [3] protocols. In binary tree protocols, collided tags are separated into two groups by selecting random numbers, which makes binary branching, until a reader identifies tags without further collisions [5], [7]. At the start of the identification process, a reader sends a start command. Then all tags transmit their IDs, and a collision occurs. The reader recognizes the collision and informs that to the tags. The collided tags randomly select 0 or 1 and add it to their own counters (the counters are set to 0 initially). The reader now sends a progress command, and all tags decrease their counters. The tags with 0 counter value transmit their IDs. This process is repeated until all tags are identified. In [14], m-ary tree protocol based on breadth-first tree search is proposed. In query tree protocols [6], [8], a reader transmits a query frame and the tags with the matching prefix with that in the query frame reply to the reader. If multiple tags have the same matching prefix, a collision occurs. Then the reader adds additional bit (‘0’ or ‘1’) to the prefix until only one tag responds and iterates this process until it identifies all tags [20]. The query tree-based m-ary tree protocol [15] proposes a hybrid query tree protocol which combines a tree-based query protocol with slotted backoff mechanism.

Slotted ALOHA-based protocols are designed to reduce the probability of tag collisions by making tags to respond at random time slots, and it can be classified into slotted ALOHA [16], [18], [21] and framed-slotted ALOHA [17], [19]. The pure ALOHA protocol is a simple anti-collision method based on ALOHA protocol. When tags receive a query frame from a reader, each tag randomly selects its response time and transmits its response message to the reader. The pure ALOHA protocol is simple but inefficient. To improve efficiency, modified version of pure ALOHA, slotted ALOHA is developed. In the slotted ALOHA protocol [18], tags transmit their IDs in randomly selected synchronous time slots.

The framed-slotted ALOHA protocol, e.g., Framed Slotted ALOHA (FSA) and Dynamic Framed-Slotted ALOHA (DFSA), divides a frame into time slots. The FSA protocol uses a fixed frame size. In FSA, a reader broadcasts the information about frame size to tags. Each tag selects a random time slot in a frame and transmits its response message in the time slot of the frame. The implementation of the FSA protocol is simple, however, if there are too many tags compared to the frame size, many time slots in a frame may experience collisions. And, if a frame size is larger than the number of tags, many time slots may be idle, which wastes time slots. The DFSA protocol changes the frame size for efficient tag identification. To determine the frame size, the readers need to estimate the number of tags, since the optimal frame size is known to be optimal when it is the same as the number of tags [13].

Many of the existing DFSA protocols [9]–[13] require
an appropriate tag estimation method to estimate an appropriate frame size. The mechanism in [9], [10] estimates the number of tags based on the number of collisions in a frame. The estimated number of tags is decided by multiplying the number of collisions by 2.39 or 2 in [9], [10]. In [11], [12], estimation of the number of tags is based on the success, collision and idle probability. However, when the measured probability of collision becomes high, the estimated number of tags may be inaccurate. The reason is that the probability of collision ($P_c$) becomes saturated as the number of tags increases. So for a small range of estimation error ($\varepsilon$) ($P_c - \varepsilon$, $P_c + \varepsilon$), a wide range of the number of tags is mapped, leading to a large estimation error (see Fig. 1, when $P_c > 0.9$). Generally, the identification time of DFSA is less than that of the basic binary tree algorithm when the estimation of tags is accurate. Since the basic binary tree algorithm separates collided tags into two groups, which may cause many collisions especially at the early stages of a binary collision tree. If the tag number of tags is estimated accurately with an appropriate estimation method, the performance of DFSA approaches that of the optimal DFSA. So we propose an accurate tag estimation method to enhance the performance of DFSA.

In this paper, we propose a new DFSA anti-collision protocol. The proposed DFSA with Litmus slots (DFSA-L) reduces the identification time by reducing the number of slots to determine an appropriate frame size regardless of the initial frame size. While other existing methods measure the probability of collision in a frame, DFSA-L measures the probability of collision in small litmus (test) slots and then decide the appropriate frame size. The proposed DFSA-L quickly and accurately estimates the number of tags and the appropriate frame size, so that the tag identification time is reduced. This paper is organized as follows. In Sect. 2, we explain the proposed DFSA-L using litmus slots. In Sect. 3, mathematical analysis for tag identification time of optimal DFSA, the existing methods and the proposed DFSA-L are presented. In Sect. 4, performance evaluation by simulations is presented. We conclude in Sect. 5.

2. Proposed DFSA-L

We present existing DFSA protocols for tag identification and propose our DFSA-L.

2.1 Existing Tag Estimation Methods in DFSA

Most DFSA protocols estimate the number of tags to be identified from the number of collision, success and idle slots or the probability of collision. A reader can distinguish the success slots and the collision slots by using Frame Check Sequence (FCS) in the tags’ IDs. The reader replies an ACK message for the success. In this way, the reader and the tags can recognize the collision. The Schoute method [9] assumes that the number of next frame size is exactly equal to the number of tags not identified in the current frame. The number of tags that choose a time slot of a frame is distributed by a Poisson distribution with mean $1$.

$$P[k \text{ tags choose a time slot}] = p_k = \frac{1}{k!} \cdot e^{-1}. \quad (1)$$

If the time slot experiences a collision, it can be inferred that the number of tags that choose the time slot is at least 2. The probability $p_{k,c}$ that $k$ tags choose a time slot conditioned on collision is

$$P[k \text{ tags choose a time slot | collision}] = p_{k,c} = \frac{p_k}{1 - p_0 - p_1}, \quad (2)$$

where $p_0$ and $p_1$ are idle and success probability, respectively. The expected number of tags that choose a time slot is

$$\sum_{k=2}^{\infty} k \cdot p_{k,c} = \frac{\sum_{k=2}^{\infty} e^{-1} \cdot \frac{1}{k!} \cdot e^{-1}}{1 - 2 \cdot e^{-1}} = \frac{e - 1}{e - 2} \approx 2.39, \quad (3)$$

which means that on average 2.39 tags transmit in a collision time slot. Let $m_c$ be the number of collisions in a frame. The expected number of tags to be identified in the next frame is

$$N = m_c \times 2.39, \quad (4)$$

which is the next frame size.

In the lower bound method [10], the estimation comes from the observation that a collision involves at least two different tags. Therefore, the expected number of tags to be identified in the next frame is

$$N = m_c \times 2. \quad (5)$$

The Vogt method [11] is based on Chebyshev inequality and aims to find the next frame size to minimize the distance between the observed vector, i.e., number of idle, success and collision slots $< m_i, m_s, m_c >$ in a frame, and the computed vector $< a_i^K, a_s^K, a_c^K >$ with the frame size $K$.

$$e_{cd}(K, m_i, m_s, m_c) = \arg \min_n \left| \left( \begin{array}{c} a_i^K \\ a_s^K \\ a_c^K \end{array} \right) - \left( \begin{array}{c} m_i \\ m_s \\ m_c \end{array} \right) \right|, \quad (6)$$

$$a_i^K = K \cdot \left(1 - \frac{1}{K}\right)^n, \quad (7)$$

$$a_s^K = K \cdot n \left(1 - \frac{1}{K}\right)^{n-1}, \quad (8)$$

$$a_c^K = K(1 - a_i^K - a_s^K). \quad (9)$$

Then, the expected number of tags to be identified in the next frame is

$$N = e_{cd}(K, m_i, m_s, m_c) - m_s. \quad (10)$$

The collision ratio (C-ratio) method [12] estimates the number of tags by the observed collision probability (collision ratio) in a frame. After a frame, we can measure the collision ratio $\hat{C}_{ratio}$ and estimate the number of tags $n$ from
\[ \hat{C}_{\text{ratio}} = 1 - \left(1 - \frac{1}{K}\right)^n \left(1 + \frac{n}{K-1}\right). \]  

(11)

where \( n \) is the number of tags to be estimated and \( K \) is the frame size. Therefore, the number of tags to be identified in the next frame is

\[ N = n - m_s, \]  

(12)

2.2 Proposed DFSA-L Protocol

The proposed DFSA-L decides the next frame size using small litmus (test) slots. The test slots indicate the small number of slots in the beginning of a frame. Tags within reading range of a reader randomly select their time slots in a frame and transmit IDs to the reader. After test slots, the reader measures the probability of collision during the test slots. Note that the probability of collision in the test slots represents that in the frame. If the probability of collision in the test slots is not within an appropriate threshold range (i.e., too small or too large), the reader can infer that the probability of collision in the current frame is too small or too large. So, the reader immediately stops the operation right after the test slots in the current frame and estimates the size of a new frame. The new frame size is decided based on the Schoute method using the measured number of collided slots in the test slots.

The Schoute method is simple and shows similar tag estimation performance with other methods when frame size is large enough. Since the number of test slots of the proposed DFSA-L is small, the measured success and collision probabilities of the probability-based methods (e.g., Vogt method and C-ratio method) may not be accurate, and the estimated number of tags may be inaccurate. Thus Schoute method is appropriate for proposed method which uses small test slots. Then the next frame immediately starts right after the test slots. If the probability of collision in the test slots is within the threshold range, the reader can confirm that the probability of collision in the frame is appropriate, and the size of the current frame is appropriate. Therefore, the tags stay in the current frame for identification and after the current frame, the reader changes the size of the next frame using the number of collided slots in the current frame.

The threshold defines the upper and lower bound of the probability of collision. In the optimal DFSA (when \( K = n \)), the highest probability of success is about 0.37. So, we assume that the frame size is properly estimated when the probability of success is greater than the required probability of success \( P_s \), \( P_s \) is in the range of \([0, 0.37]\), and it denotes the minimum identification efficiency. We assume 0.3 as \( P_s \) to achieve a higher identification efficiency, 30%.

If \( P_s \) is 0.3, the range of probability of collision is in the range of \([0.1, 0.5]\) (See Fig. 1). Let’s assume that we have \( n \) tags and the size of a frame is \( K \). Then the probability of success \( P_s \) is

\[ P_s = n \cdot \frac{1}{K} \left(1 - \frac{1}{K}\right)^{n-1}. \]  

(13)

So, the range

\[ P_s \geq 0.3, \]  

(14)

corresponds to the range of the probability of collision \( P_c \)

\[ 0.1 \leq P_c \leq 0.5, \]  

(15)

where

\[ P_c = 1 - \left(1 - \frac{1}{K}\right)^n - n \cdot \frac{1}{K} \left(1 - \frac{1}{K}\right)^{n-1}. \]  

(16)

Thus, the upper threshold of probability of collision \( P_{up} \) can be set to 0.5 and the lower threshold of probability of collision \( P_{low} \) can be set to 0.1. Figure 1 shows the relationship between the probability of success (Eq. (13)) and the probability of collision (Eq. (16)). The probability of success is a convex function and the probability of collision is an increasing function of the number of tags. If we define a required probability of success, we can get the corresponding range of the probability of collision.

Figure 2 shows an example of the proposed DFSA-L. Let’s assume that the initial frame size \( K_0 \) is 64 and the size of the test slots \( k \) is 10. In the first round (frame size \( K_0 \)), the probability of collision \( (\hat{P}_{c,0}) \) in the test slots is 0.6 and it is greater than the upper threshold \( P_{up}(= 0.5) \), which implies that with the current frame size \( K_0 = 64 \), tags experience many collisions. So tag identification process is immediately suspended. And, the reader calculates the next frame size \( K_1 \) using the number of collided slots \( (\hat{m}_{c,0}) \), the ratio \( (K_0/k) \) of the frame size and the size of the test slots, \([K_1 = (\hat{m}_{c,0} \times 2.39) \times (K_0/k)]\), where \([x]\) is the smallest integer which is greater than or equal to \( x \). Similarly, in the second round, the reader measures the probability of collision \( (\hat{P}_{c,1}) \) during the \( k \) test slots of the frame with size \( K_1 \). The probability of collision \( (\hat{P}_{c,1}) \) is 0.1 and it is less than or equal to the lower threshold \( P_{low}(= 0.1) \). Thus, the reader stops the operation in the current frame right after the test slots and calculates the next frame size \( K_2 \) using the number of collided slots \( (\hat{m}_{c,1}) \) and the ratio \( (K_1/k) \), \( K_2 = [(\hat{m}_{c,1} \times 2.39) \times (K_1/k)]\). In the third round,
the reader measure the probability of collision during $k$ in $K_2$. The probability of collision $\hat{P}_{c,2}$ is 0.3 and it is within the threshold range ($P_{low} < \hat{P}_{c,2} < P_{up}$). Now the frame size $K_2$ is appropriate based on the observation during the $k$ test slots. So the identification process continues until the end of the frame $K_2$. Then, the frame size ($K_3$) is calculated $K_3 = [(\hat{m}_c,2 \times 2.39)]$. Similar steps continue afterwards. The pseudo code for the proposed DFSA-L protocol is given as follows.

### 3. Performance Analysis

We analyze the total number of slots used for tag identification in the existing DFSA methods and the proposed DFSA-L.

#### 3.1 Optimal DFSA

It is proved that the highest efficiency can be achieved if the frame size is equal to the number of tags (the optimal DFSA) [13]. Total number of time slots used in the optimal DFSA is

$$T_{DFSA} = \sum_{i=0}^{\infty} L_i.$$ 

where $L_i$ is frame size at the $i$th frame and is the same as the number of tags $n_i$ in the $i$th frame, i.e., $L_i = n_i$. The number of unread tags $n_i$ in the $i$th frame is

$$n_i = n_{i-1} - m_{s,i-1}, \quad i \geq 1,$$

$$n_0 = n$$

where $m_{s,i-1}$ is the number of success slots in the $(i-1)$th frame. The number of success slots $m_{s,i}$ in the $i$th frame can be calculated

$$m_{s,i} = L_i \times P_{s,i} = L_i \times \left( n_i \cdot \frac{1}{L_i} \left( 1 - \frac{1}{L_i} \right)^{n_i-1} \right), \quad i \geq 0.$$ 

And, $P_{d,i}$, $P_{s,i}$, and $P_{c,i}$ are the success, idle and collision probabilities, respectively, at the $i$th frame.

$$P_{d,i} = \left(1 - \frac{1}{L_i}\right)^{n_i},$$

$$P_{s,i} = n_i \cdot \frac{1}{L_i} \left( 1 - \frac{1}{L_i} \right)^{n_i-1},$$

$$P_{c,i} = 1 - P_{d,i} - P_{s,i}.$$ 

#### 3.2 Schoute, Lower Bound, Vogt and C-ratio Mechanism

The frame size of the $i$th frame in the Schoute method can be found by (see Eq. (4))

$$L_i = m_{c,i-1} \times 2.39 = (L_{i-1} \times P_{c,i-1}) \times 2.39,$$

where $m_{c,i-1}$ is the number of collision slots in the $(i-1)$th frame. The Lower bound is based on the principle that for a collision, at least two tags are involved, hence the frame size at the $i$th frame can be found by (see Eq. (5))

$$L_i = m_{c,i-1} \times 2 = (L_{i-1} \times P_{c,i-1}) \times 2.$$ 

The frame size of the Vogt method can be calculated by subtracting the number of successful time slots $m_{s,i-1}$ from the number of estimated tags (frame size) $e_{ad}$ at the $(i-1)$th frame (see Eq. (10))

$$L_i = e_{ad}(L_{i-1}, m_{d,i-1}, m_{s,i-1}, m_{e,i-1}) - m_{s,i-1}.$$ 

The frame size of the C-ratio method can be found by (see Eq. (12))

$$L_i = n_{i-1} - m_{s,i-1} - n_{i-1} - (L_{i-1} \times P_{s,i-1}).$$

The identification times of Schoute, Lower bound, Vogt, and C-ratio can be obtained by using Eq. (17), with Eqs. (23) ~ (26), respectively.

#### 3.3 Proposed DFSA-L

We can obtain the total number of time slots in DFSA-L as

$$T_{DFSA-L} = \sum_{i=0}^{\infty} L_{i,L},$$

where $L_{i,L}$ is frame size at the $i$th frame and is the same as the number of tags $n_{i,L}$ in the $i$th frame, i.e., $L_{i,L} = n_{i,L}$. The number of unread tags $n_{i,L}$ in the $i$th frame is

$$n_{i,L} = n_{i-1} - m_{s_{i-1,L}}, \quad i \geq 1,$$

$$n_0 = n$$

where $m_{s_{i-1,L}}$ is the number of success slots in the $(i-1)$th frame. The number of success slots $m_{s_{i,L}}$ in the $i$th frame can be calculated

$$m_{s_{i,L}} = L_{i,L} \times P_{s_{i,L}} = L_{i,L} \times \left( n_{i,L} \cdot \frac{1}{L_{i,L}} \left( 1 - \frac{1}{L_{i,L}} \right)^{n_{i,L}-1} \right), \quad i \geq 0.$$ 

And, $P_{d_{i,L}}$, $P_{s_{i,L}}$, and $P_{c_{i,L}}$ are the success, idle and collision probabilities, respectively, at the $i$th frame.
\[ T_{DFSA-L} = \sum_{i=0}^{\infty} L_i. \]  
(27)

If the probability of collision \( p_{c,i} \) in the test slots of the \( i \)th frame is within the threshold range, i.e., \( P_{\text{low}} < p_{c,i} < P_{\text{up}} \), the number of used time slots \( L_i \) is equal to the frame size \( K_i \) in the \( i \)th frame.

\[ L_i = K_i, \quad i \geq 0. \]  
(28)

Then, the next frame size \( K_{i+1} \) is determined by the number of collision slots during the \( i \)th frame.

\[ K_{i+1} = [m_{c,i} \times 2.39], \quad i \geq 0. \]  
(29)

where the number of collision slots \( m_{c,i} \) in the \( i \)th frame is

\[ m_{c,i} = K_i \times p_{c,i}, \quad i \geq 0. \]  
(30)

The probability of collision at the \( i \)th frame is

\[ P_{c,i} = 1 - \left( 1 - \frac{1}{K_i} \right)^n_i - n_i \cdot \frac{1}{K_i} \left( 1 - \frac{1}{K_i} \right)^{n_i-1}, \quad i \geq 0. \]  
(31)

On the other hand, if the probability of collision is not within the threshold range, i.e., \( p_{c,i} \leq P_{\text{low}} \) or \( p_{c,i} \geq P_{\text{up}} \), the number of used time slots \( L_i \) is equal to the size of the test slots \( k \) in the \( i \)th frame.

\[ L_i = k, \quad i \geq 0. \]  
(32)

The next frame size \( K_{i+1} \) is determined by the number of collision slots during \( k \) test slots, the size of the test slots \( k \) and the frame size \( K_i \) in the \( i \)th frame.

\[ K_{i+1} = \left[ (m_{c,i} \times 2.39) \times \left( \frac{K_i}{k} \right) \right], \quad i \geq 0. \]  
(33)

The number of collision slots \( m_{c,i} \) in the test slots is

\[ m_{c,i} = k \times p_{c,i}, \quad i \geq 0. \]  
(34)

And the probability of collision during the test slots in \( i \)th frame is

\[ p_{c,i} = 1 - \left( 1 - \frac{1}{K_i} \right)^{n_i} \]

\[ - n_i \left( \frac{k}{K_i} \right) \cdot \frac{1}{k} \left( 1 - \frac{1}{K_i} \right)^{n_i-1}, \quad i \geq 0. \]  
(35)

The number of tags \( n_i \) in the \( i \)th frame

\[ n_i = n_{i-1} - m_{s,i-1}, \quad i \geq 0, \]

\[ n_0 = n. \]  
(36)

Note that \( n_0 = n \) and \( K_0 \) are the initial number of tags and the initial frame size, respectively. The number of successful slots in the \( i \)th round is

\[ m_{s,i} = \begin{cases} 
K_i \times p_{s,i}, & P_{\text{low}} < p_{c,i} < P_{\text{up}} \\
k \times p_{s,i}, & p_{c,i} \leq P_{\text{low}}, \ p_{c,i} \geq P_{\text{up}} 
\end{cases} \]

where

\[ P_{s,i} = n_i \cdot \frac{1}{K_i} \left( 1 - \frac{1}{K_i} \right)^{n_i-1}, \quad i \geq 0, \]  
(37)

\[ p_{s,i} = n_i \left( \frac{k}{K_i} \right) \cdot \frac{1}{k} \left( 1 - \frac{1}{K_i} \right)^{-1}, \quad i \geq 0. \]  
(38)

4. Simulation Result

In this section, we present performance evaluation of the proposed DFSA-L scheme and the existing schemes via matlab simulations and verify the analysis of Sect. 3. Figure 3 shows the difference \( (D_i = P_{s,i} - P_{c,i}) \) between the probability of collision \( P_{c,i} \) for the frame size \( K_i \) and the probability of collision \( p_{c,i} \) for the \( k \) test slots with varying size of test slots, frame size, and the number of tags. We must decide an appropriate number of test slots for DFSA-L. An appropriate number of test slots must be small enough to reduce the identification time and large enough to represent the whole frame size. When the frame size is 64 \((K_i = 64)\) and 128 \((K_i = 128)\), as the size of test slots increases, the difference between \( P_{s,i} \) and \( P_{c,i} \) becomes very small after about 5 test slots. So one can use 5 ~ 10 litmus (test) slots to investigate the characteristics of the whole frame.

Figure 4 shows the total number of time slots with varying size of frame and test slots. We set the initial frame size to 64 or 128 and vary the number of tags from 100 to 300 and the size of test slots from 5 to 60. When the number of tags is 100, the difference between total number of slots for \( K_0 = 64 \) and that for \( K_0 = 128 \) is small because the frame size is around the number of tags, i.e., the probability of collision in the test slots is within the threshold range and the size of the frame \((K_0 = 64, 128)\) is appropriate for 100 tags.

Also, when the number of tags is 200, total number of slots for \( K_0 = 128 \) is smaller than that for \( K_0 = 64 \). Therefore,
when the frame size approaches the number of tags, total number of time slots becomes small. In addition, when the test slots is in the range of 10 \sim 20, the total number of time slots is the smallest, regardless of the initial frame size. The number of test slots that is either too small (below 10) or too large (above 20) is not appropriate.

Figure 5 shows total number of time slots with varying initial frame size and the number of tags. We set the number of test slots to 10 and vary the initial frame size from 100 to 500 and the number of tags from 100 to 500. The more the number of tags increases, the more the total number of time slots is used. Total number of time slots is minimum when the number of tags is equal to the frame size. As shown in Fig. 5, the initial frame size is likely to have little overall impact on total number of time slots. If the initial frame size is smaller than the number of tags, the probability of collision becomes high. And, if the initial frame size is larger than the number of tags, the probability of collision becomes low. Then, the litmus test from the test slots quickly adjust the next frame size, so that the next frame size can be larger or smaller than the current frame size. Thus, although the initial frame size may not be appropriate for a given number of tags, the appropriate frame size is determined by test slots at every round.

In Fig. 6, we compare the total identification time of the proposed DFSA-L protocol with those of the conventional methods. The solid lines represent the performances by the analysis of the methods. We set the initial frame size to 64 and the size of test slots to 10 and vary the number of tags from 100 to 1000. The lower bound method consumes more time slots than the Schoute, Vogt and C-ratio methods, and the performance of the Vogt method is similar to the that of C-ratio method. The proposed DFSA-L protocol is superior to other existing methods and shows similar performance compared to the optimal DFSA which has the best performance. And the simulation results are closely matched with the analysis in Sect. 3.

Figure 7 shows the number of rounds that only use test slots not a whole frame for tag identification. When the size of tests slots is set to 10 and the number of tags is 50 \sim 500, and as we vary the initial frame size from 32 to 256, the number of rounds with only test slots is less than 12. When the number of tags is 100 and the initial frame size is 32, the probability of collision is rather high, and when the initial frame size is 256, the probability of collision is rather low. So, the number of rounds using test slots increases. When the number of tags is similar to the frame size, the number of rounds using only test slots is small. The appropriate frame size is quickly reached using small number of test slots in our DFSA-L, while it is found using a whole frame in the existing DFSA methods. Therefore, the required total number of slots for identification is reduced compared to the existing methods.

Figure 8 shows total round for tag identification. We set the initial frame size to 64 and the size of test slots to 10. In our DFSA-L, the number of rounds for tag identification
increases due to test slots compared to other existing methods. When the number of tags increases, the number of use for test slots increases because the initial frame size is very small compared to the number of tags, and the number of rounds increases. However, the total slots are not affected by the number of rounds because the size of test slots is small. As shown Fig. 8, the average number of rounds in DFSA-L and lower bound method are about 20 and the average numbers of rounds of the existing methods are about 15.

Figure 9 shows the cumulative number of used slots at each round when the number of tags is 500 and the initial frame size is 64. In DFSA-L, the number of cumulative slots is small due to use of small-size test slots. The number of used slots for the initial round corresponds to the test slots because the initial frame size is very small compared to the number of tags. As the round evolves, the number of cumulative slots increases slightly and the used total number of slots is small compared with the others. As a result, the performance for DFSA-L shows similar performance as the optimal DFSA (see Fig. 6).

Figure 10 shows the number of time slots used to identify all tags under the error-prone channel. We have used the path-loss model in [22], and the Bit Error Rate (BER) vs. Signal to Noise Ratio (SNR) relationship in [23]. The system bandwidth is 500 kHz, the center frequency is 900 MHz, and the thermal noise density is $-174$ dBm/Hz. The size of a tag ID is 96 bits, and the reader’s identification range is 10 m. The transmission power of a reader is 10 dBm, the antenna gain of a reader is 0 dBi, and the antenna gain of a tag is $-6$ dBi. The performance of the optimal DFSA is shown to be the best, since the optimal DFSA estimates the number of tags accurately regardless of channel conditions. Proposed DFSA-L and the Schoute methods may not distinguish collision slots with frame error slots. Thus the proposed DFSA-L and the Schoute methods may over-estimate the number of tags, which may cause performance degradation. Since the proposed method estimates the number
of tags only when the probability of success slots is greater than $P_{\text{req}}$, the influence of channel condition is not much. So the performance of the proposed method is better than that of the Vogt method and the Schoute method. The Vogt method uses probabilities of idle slots and success slots as well as the probability of collision slots, so the effect of channel error is relatively smaller than the Schoute method.

5. Conclusion

We have proposed new frame size estimation and tag identification mechanism based on DFSA. The proposed DFSA-L estimates an appropriate frame size using the collision probability of the small limutus (test) slots. The proposed DFSA-L saves time slots by reducing unnecessary waste of idle slots when the number of tags is very small compared to the frame size and by decreasing the collision probability when the number of tags is very large compared to the frame size by using small number of test slots. We have provided analytical models of DFSA-L and existing anti-collision protocols. Performance analysis and simulation results indicate that the proposed protocol consumes less time slots than other DFSA methods and the performance of DFSA-L approaches the performance of the optimal DFSA.

Acknowledgments

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-(C1090-1211-0005)).

References


Soon-Bin Yim received the B.S. degree in electrical engineering from Hanseo University, Korea, in 1998 and the M.S. degree in electrical and computer engineering from Sungkyunkwan University, Korea, in 2004. She finished the Ph.D. coursework in electrical and computer engineering from Sungkyunkwan University, Korea, in 2007. Her research interests include Radio Frequency IDentification (RFID), wireless communication networks and optical networks.
Jongho Park received the B.S. and M.S. degrees in electrical and computer engineering from Sungkyunkwan University, Korea, in 2004 and 2006, respectively. He is currently pursuing his Ph.D. degree in department of electrical and computer engineering at Sungkyunkwan University since March 2006. His research interests include Radio Frequency IDentification (RFID), wireless communication networks, Wireless LAN, Wireless PAN, and Ad-hoc network.

Tae-Jin Lee received the B.S. and M.S. degrees in electronics engineering from Yonsei University, Seoul, Korea, in 1989 and 1991, respectively, the M.S.E. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 1995. He received the Ph.D. degree in electrical and computer engineering from the University of Texas, Austin, in 1999. In 1999, he joined the Corporate R&D Center, Samsung Electronics, where he was a Senior Engineer. Since 2001, he has been an Associate Professor with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea. His research interests include performance evaluation, resource allocation, medium-access control, and design of communication networks and systems, wireless LAN, wireless PAN, RFID, and ad hoc/sensor networks, home networks, next-generation wireless communication systems, and optical networks. Dr. Lee is a member of IEEE, KICS, KIIEE and KIPS, and has been a voting member of IEEE 802.11 WLAN.