

An Enhanced Progressive Scanning Algorithm for Improving Tag Identification Performance

Intaek Lim*

**Department of Embedded Software, Busan University of Foreign Studies
itlim@bufs.ac.kr

Abstract

The PS algorithm divides the tags within the reader's identification range into smaller groups by increasing the transmission power incrementally and identifies them. This algorithm uses the fixed frame size at every scan. Therefore, it has problems that the performance can be variously shown according to the number of tags, frame size, and power level increase. In this paper, we propose an EPS algorithm that allocates the optimal frame size by estimating the number of tags at each scan. The simulation results showed that the identification delay of EPS algorithm could be improved 70% compared with PS algorithm. It also provided a stable identification delay regardless of power level increase.

Keywords: *RFID, Anti-collision Algorithm, Multiple Tag Identification, PS Algorithm*

1. Introduction

Unlike conventional barcode systems, radio frequency identification (RFID) is a wireless technology that uses electromagnetic or magnetic response exchange to identify objects at a distance without direct line of sight [1]. Recently, the RFID technique attracts a lot of attention due to its automatic identification capability for the identity information of an object. Also, it is an emerging technology that has been applied to various object identification and tracking over the past decade. RFID is a generic term for technologies which is used to identify objects embedded with tags by wireless channel without manual intervention. It has been used in various real-life applications such as supply chain management, transportation, logistics, and medical management [2]. It also provides an efficient and inexpensive mechanism for automatically collecting the identity information.

An RFID system consists of a reader and one or more tags. The tag has an identification code stored in its memory that is represented by a bit string. The reader can read the identification code of tags within its range by an anticollision algorithm over the wireless channel. RFID systems have many advantages over barcode systems. RFID systems use radio frequency as a method of data transmission. Thus, unlike a barcode label, a tag does not need to be placed in a line of sight position from the reader in order to be successfully identified [3].

In the RFID system, tag identification process is performed by the reader's query to tags and then tag's backscattering its identifier as its response. But, if there are multiple tags within the identification range of the reader, some of them might respond simultaneously and leads to collisions which decrease the performance. When multiple tags respond to the reader simultaneously, tag collisions occur, and no tag can be identified by the reader successfully. Therefore, the system requires a multiple-access scheme that allows the reader to read data from the individual tags. The technical scheme that handles multiple-access is called an anticollision algorithm [4, 5].

There are two types of anticollision algorithms: deterministic and probabilistic algorithm [6]. The deterministic algorithm resolves collisions by muting subsets of tags that are involved in a collision. By successively muting larger subsets, only one tag will be left and finally lead to successful transmission. Binary tree and query tree algorithms are the two common methods of the deterministic algorithm. The probabilistic algorithms are based on ALOHA-like protocol that provides slots for the tags to send their data. Almost all the probabilistic algorithms use framed slot ALOHA (FSA), which has been advanced in function by adding slotting and framing on ALOHA. The tags send their identifiers at a randomly selected slot. When collisions occur, the tags that are involved in collisions retransmit their identification codes in the next query round. The probabilistic algorithms may have limitations on the completeness of tag identification because there is a probability of failing to be identified in a limited time period.

In the probabilistic algorithms, when the number of tags is small, a probability of tag collision is low, so the identification delay is relatively short. On the other hand, as the number of tags increases, the probability of tag collision becomes higher and the time needed to identify all tags increases rapidly. Therefore, the identification performances are largely dependent on the probability of tag collision. There are EDFSA (Enhanced Dynamic FSA) and PS (Progressive Scanning) algorithms to reduce the probability of tag collision [7, 8]. Those algorithms limit the number of responding tags by grouping the tags within the identification range of the reader, and thus can reduce the probability of tag collision. Especially, in the PS algorithm, the reader takes advantages of the identification range difference between the tags and the reader's antenna. Tags that are near the reader receive more power from the reader than those that are further away. The reader starts transmitting from the minimum power level until the maximum power level that is permitted by regulations. Tags that are further away from the reader do not receive enough power to energize its circuits and cannot transmit their replies. In the PS algorithm, whenever the reader makes a new scan with the increased power level, it uses a fixed frame size regardless of the number of tags. Since the frame size is fixed, we are able to implement it easily, but the identification performances change variously according to the frame size and the number of tags. To solve this problem, we propose an EPS (Enhanced PS) algorithm that uses a dynamic frame size.

This paper is organized as follows. In Section 2, we introduce the PS algorithm and analyze the performances of PS algorithm. In Section 3, we propose an EPS algorithm. Section 4 shows the simulation results, and Section 5 concludes the paper.

2. Related works

In the PS algorithm, the reader transmits a command from a minimum power level $P_{r,min}$ until the power level reaches the maximum $P_{r,max}$ by increasing the transmitted power by the power increment k . In the identification process, the scanning is defined as the process that the reader transmits a command with a certain power level and identifies tags within the identification range of reader. The cycle is defined as the whole scanning process from the minimum power level to the maximum power level.

The PS algorithm divides the number of tags into smaller groups by controlling the transmission power level of the reader, and performs the scanning process for each group. Therefore, this algorithm has the benefits of taking faster identification delay than the FSA algorithm since the number of tags is small for each scanning process [8]. But it can be anticipated that the performance of this algorithm may change variously according to the frame size and number of tags since the frame size is fixed. The

number of tags in each scanning process is dependent on the power increment k . The number of tags in the scanning process decreases as the increment k decreases and vice versa. Therefore, we can also anticipate that the performances will change variously according to the power increment k .

In this section, we analyze the performances of PS algorithm through the computer simulations. It is assumed that the minimum power level $P_{r,min}$ and the maximum power level $P_{r,max}$ are 0.4 Watt and 4.0 Watt, respectively as in the reference [8].

The total identification delay of PS algorithm according to the frame size is depicted in Figure 1. In this figure, we assumed that the power increment is 0.2 Watt. The total identification delay means the number of slots that are consumed until all the tags within the reader's identification range are recognized by the reader. As shown in the figure, when a large frame size is used by the reader in spite of the small number of tags, the identification delay is rather longer than when the frame size is small. If we let the frame size be large, the reader can identify a lot of tags at the initial cycles. Therefore, there are not many unidentified tags. Regardless of the small number of unidentified tags, the reader must repeat cycles with the same large frame size until all tags are identified. This may cause a lot of idle slots and make the identification delay long. On the other hand, if the frame size is too small, the tag collisions probability will increase when the number of tags increases. This also lets the identification delay be long.

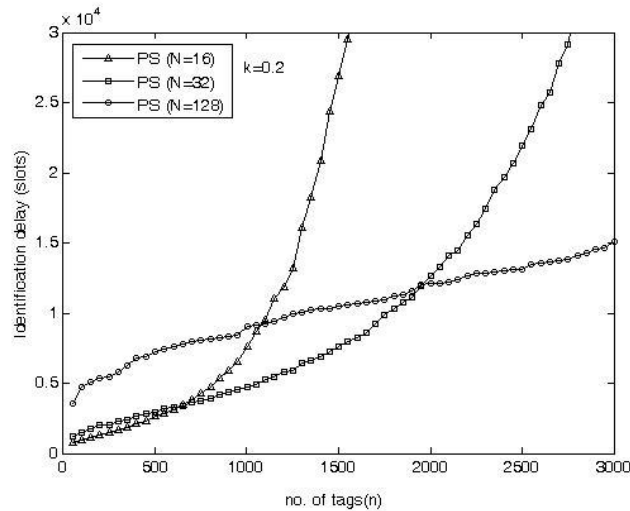


Figure 1. Identification Delay of PS Algorithm According to the Frame Size

Figure 2 and 3 show the identification delay according to the power increment when the frame size is 64 and 128 slots, respectively. As the power increment increases, so the number of scans in a cycle decreases. This may cause the number of tags in each scan to increase. The identification delay increases rapidly due to a lot of collisions when the number of tags within the identification ranges of reader increases. In spite of the small power increment, if the frame size is large, a lot of slots will be idle until all tags are identified. This also makes the identification delay increasing. If the number of tags is small, many tags will be identified at the initial cycle. Therefore, the larger the power increment is, the smaller the identification delay is. As shown in the simulation results, the identification delay of PS algorithm is dependent mainly on the frame size and the power increment.

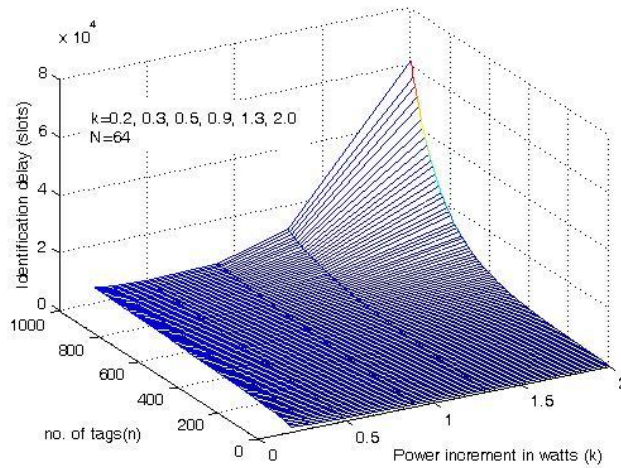


Figure 2. Identification Delay of PS Algorithm According to the Power Increment ($N=64$)

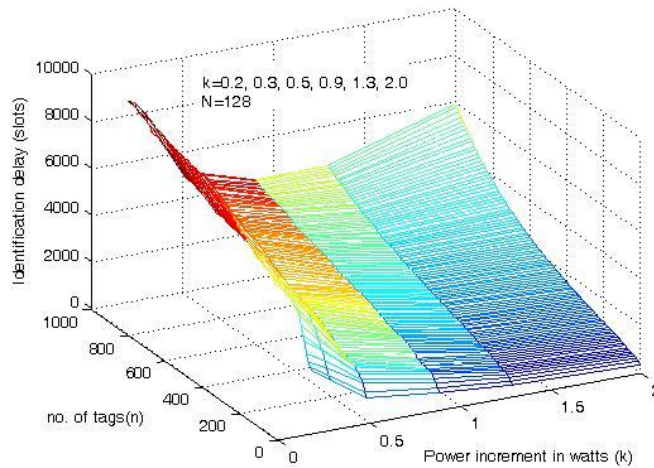


Figure 3. Identification Delay of PS Algorithm According to the Power Increment ($N=128$)

3. EPS Algorithm

3.1. Reader's Operation

As described in Section 2, the PS algorithm shows a variety of identification performances according to the number of tags, the frame size, and the power increment because it uses a fixed frame size. In this paper, we propose an EPS (Enhanced Progressive Scanning) algorithm, which gives the stable identification performance regardless of the frame size and the power increment.

Figure 4 depicts the reader's operation for the proposed EPS algorithm. All the operations for tags are same as the original PS algorithm. Henceforth, the proposed scheme does not impose any modification on tags.

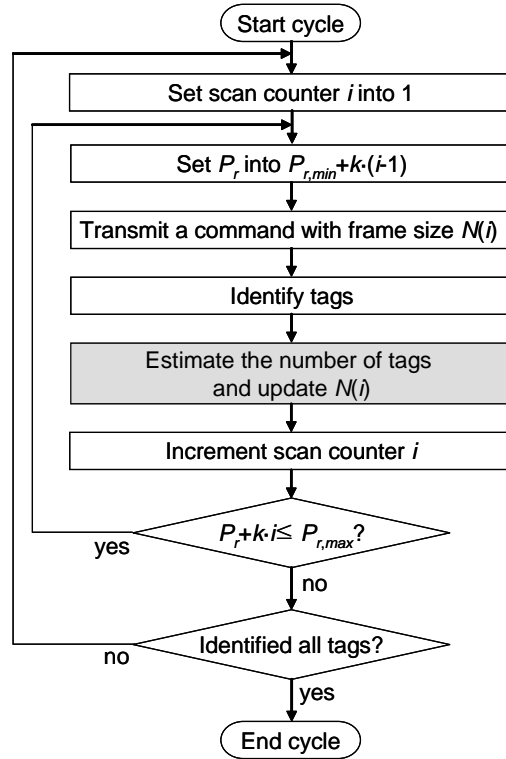


Figure 4. Reader's Operation for EPS Algorithm

A detailed description of the PS algorithm is as follows:

- ① At the first scan, the reader transmits a command with $P_r = P_{r,min}$ and frame size $N(1)$. The tags that are within the range of power P_r become energized, and reply using the FSA algorithm.
- ② The reader identifies tags and estimates the number of unidentified tags. The estimated number of tags is used for updating the frame size of the next cycle.
- ③ The reader increases the power level by the power increment k and repeats the scanning process with the transmitting power $P_r = P_{r,min} + k$. All the tags that are entered into the identification range of power P_r reply, but tags that replied in the previous scanning do not respond to the reader's command.
- ④ The reader repeats the above mentioned procedures with $P_r = P_{r,min} + k \cdot (i-1)$ ($i=2, 3, 4, \dots$). In the final scanning process, the reader transmits with $P_r = P_{r,max}$. Those are the scanning processes for the first cycle. A cycle in the PS algorithm consists of $\left\lfloor \frac{P_{r,max} - P_{r,min}}{k} \right\rfloor + 1$ scanning processes.
- ⑤ After a cycle, a new cycle begins if there are unidentified tags in the identification range of reader.

At every scan, the reader determines the frame size after estimating the number of tags. When the scan counter is i , the reader sets the frame size as $N(i)$ and transmits a command with the transmission power $P_r = P_{r,min} + k \cdot (i-1)$. After the reader identifies tags, it estimates the number of unidentified tags for the scan of the next cycle.

3.2. Tag Number Estimation

The reader estimates the number of tags by combining the information obtained after a scan with the result of probabilistic analysis. It is assumed that a frame of i -th scan consists of $N(i)$ slots and there are $n(i)$ tags in the reader's identification range. Let $N_e(i)$ and $N_s(i)$ be the number of empty slots and successful slots observed after the i -th scan, respectively. If we assume that the number of empty slots and successful slots observed after the i -th scan is equal to their expected values, respectively [9], then $N_e(i)$ and $N_s(i)$ can be given as follows.

$$N_e(i) = N(i) \left(1 - \frac{1}{N(i)}\right)^{n(i)} \quad (1)$$

$$N_s(i) = n(i) \left(1 - \frac{1}{N(i)}\right)^{n(i)-1} \quad (2)$$

By solving Eq.(1) and (2) for the number of tags $n(i)$, we can obtain

$$n(i) = (N(i) - 1) \frac{N_s(i)}{N_e(i)} \quad (3)$$

The number of tags that will be involved in the next cycle is equal to the number of colliding tags $n_c(i)$. Therefore, the number of colliding tags can be estimated by subtracting $N_s(i)$ from Eq.(3) as follows.

$$n_c(i) = (N(i) - N_e(i) - 1) \frac{N_s(i)}{N_e(i)} \quad (4)$$

Because the reader can obtain the number of successful slots and empty slots after the scan, it also can estimate the number of unidentified tags from Eq.(4). However, as shown in Eq.(4), if there are neither successful slots nor empty slots, we cannot use the above equation. Therefore, the tag number estimation scheme supplements the result of probabilistic analysis for the exact estimation.

If we let α be the average number of tags in each collision slot, α can be given by

$$\alpha = \frac{\sum_{r=2}^{n(i)} r N(i) \binom{n(i)}{r} \left(\frac{1}{N(i)}\right)^r \left(1 - \frac{1}{N(i)}\right)^{n(i)-r}}{N(i) - N(i) \left(1 - \frac{1}{N(i)}\right)^{n(i)} - n(i) \left(1 - \frac{1}{N(i)}\right)^{n(i)-1}} \quad (5)$$

In FSA algorithm, the optimal performance for the system efficiency and identification delay can be obtained when the frame size is equal to the number of tags. Therefore, the average number of tags in each collision slot is 2.4 [9]. If we let $N_c(i)$ be the number of collision slots, the number of unread tags will be given by

$$n_c(i) = 2.4 N_c(i) \quad (6)$$

By combining the observed result in Eq.(4) with the probabilistic result in Eq.(6), we can estimate the number of unread tags as following.

$$n_c(i) = \begin{cases} (N(i) - N_e(i) - 1) \frac{N_s(i)}{N_e(i)} & , \text{if } N_e(i) \neq N_s(i) \neq 0 \\ 2.4N_c(i) & , \text{otherwise} \end{cases} \quad (7)$$

The frame size for maximizing the performances of FSA algorithm is same as the number of tags within the reader's identification range. Therefore, the frame size $N(i)$ for the i -th scan of the next cycle is as follows:

$$N(i) = 2^{\lceil \log_2 n_c(i) \rceil} \quad (8)$$

4. Simulation Results

In this paper, we evaluate the performance for the proposed algorithm through the computer simulations. The simulation system was developed with SMPL libraries [10] and MS Visual C++ 6.0. The system parameters for simulations are same as the values for simulating PS algorithm in Section 2. We compare the proposed EPS algorithm with the original PS algorithm. All the results of simulations were averaged after iterating 100 times. For the simulation of PS algorithm, it is assumed that the frame size is 128 slots.

Figure 5 and 6 show the collision ratio and idle slot ratio according to the number of tags, respectively. If we let R_c and R_i be the collision ratio and idle slot ratio, R_c and R_i are defined as the ratio of the number of collision slots and idle slots to the total number of slots, respectively. As shown in the figures, in the PS algorithm, when the collision ratio is increasing, the idle slot ratio inversely is decreasing. When the frame size is small, the collision ratio is large, and the idle slot ratio is small. Also, when the frame size is large, there are many idle slots. On the other hand, the collision ratio and idle slot ratio of EPS algorithm are nearly constant when the number of tags is over 500 tags. The identification failure rate can be defined as the sum of the collision ratio and idle slot ratio. The identification failure rate of PS algorithm fluctuates according to the number of tags and frame size. On the other hand, the identification failure rate of EPS algorithm is stable.

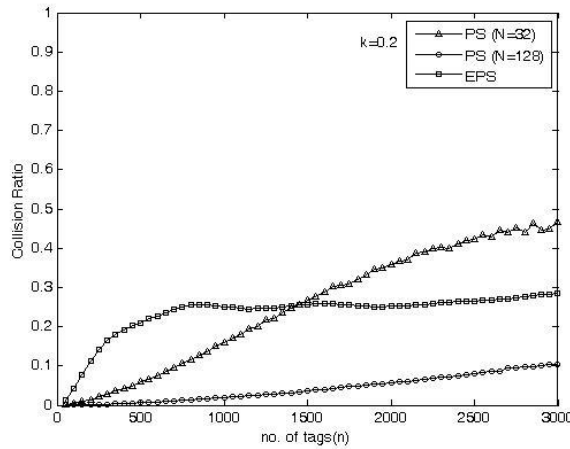


Figure 5. Collision Ratio According to Number of Tags

Figure 7 and 8 show the identification delay and system efficiency according to the number of tags, respectively. As shown in the figures, the identification delay for both of algorithms increases linearly as the number of tags increases. If we define the identification speed as the number of slots consumed for identifying one tag, the identification speed for PS algorithm is 10.7 slots. On the other hand, the identification speed for EPS algorithm is about 3.2 slots. Therefore, the proposed EPS algorithm can identify 70% faster than PS algorithm.

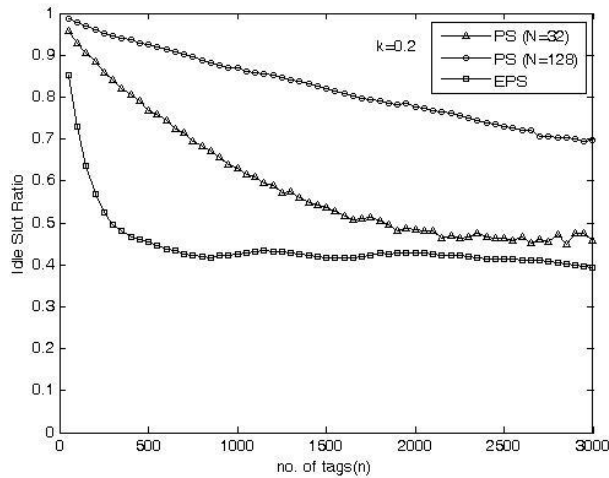


Figure 6. Idle Slot Ratio According to Number of Tags

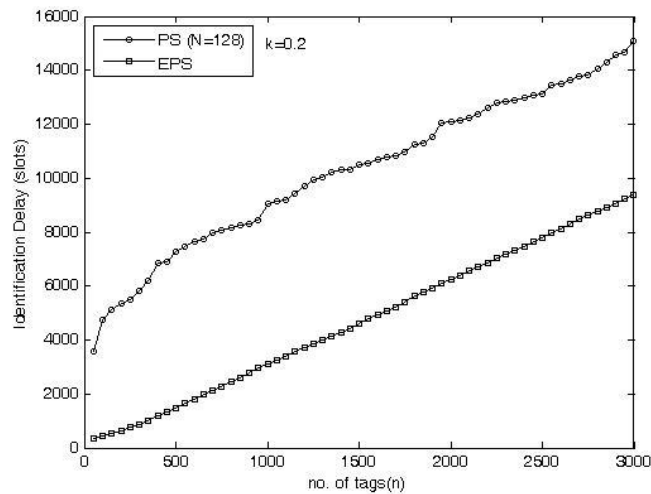


Figure 7. Identification Delay According to Number of Tags

As shown in Figure 8, the efficiency of EPS algorithm is 1.3 times superior to the PS algorithm. In the PS algorithm, the reader uses a fixed frame size. When the frame size is small, a lot of collisions will occur if there are many tags in the reader’s identification range. Also, if the reader uses a large frame size although the number of tags is small, there are many idle slots. This is the main reason why the PS algorithm is inefficient. On the other hand, in the EPS algorithm, the reader at each scan allocates a frame based on the number of

tags that are estimated in the previous cycle. Therefore, the identification delay and efficiency of EPS algorithm are better than those of PS algorithm.

Figure 9 shows the identification delay of EPS algorithm according to the various power increments. This can be compared with Figure 2 and 3 in Section 2. As shown in Figure 2 and 3, the identification delay of PS algorithm changes variously according to the power increment. However, EPS algorithm shows a stable identification delay independent on the power increment. This is because the EPS algorithm uses an optimal frame size by estimating the number of tags.

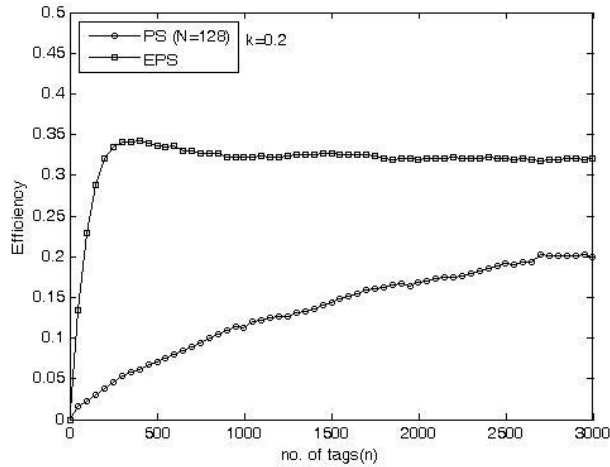


Figure 8. System Efficiency According to Number of Tags

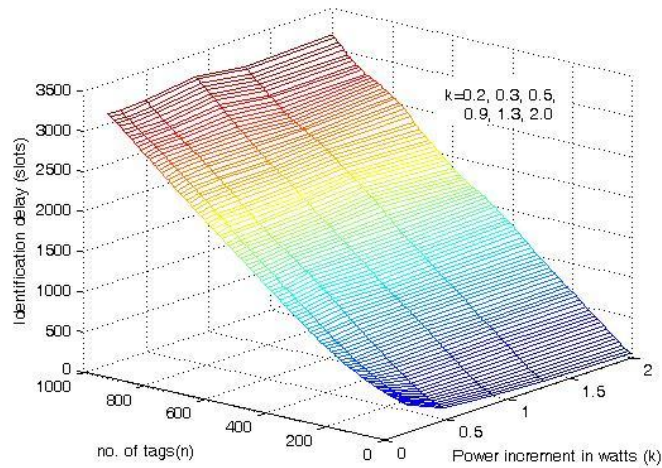


Figure 9. Identificaion Dely of EPS Algorithm According to the Power Increment

5. Conclusions

In RFID system, the identification performances are closely related to the tag collisions. The tag collisions occur when multiple tags reply at the same time. The PS algorithm is one of algorithms for reducing the tag collision probability. It divides the tags within the reader's identification range into small groups by controlling the reader's transmission power. In the PS algorithm, the reader allocates a fixed frame size at every scan. Because of the fixed frame size, the PS algorithm shows unstable identification performances according to the number of tags, frame size, and power increments.

In this paper, we proposed an EPS algorithm for improving the performance of PS algorithm. In the proposed algorithm, the reader estimates the number of tags at every scan and allocates an optimal frame size based on the estimated number of tags in the previous cycle. The simulation results demonstrated that the identification failure rate of proposed algorithm was less than PS algorithm. Therefore, the proposed algorithm can achieve better performances than PS algorithm.

Acknowledgements

This work was supported by the research grant of the Busan University of Foreign Studies in 2014.

References

- [1] D. Klair, K. Chin, and R. Raad, "On the Energy Consumption of Pure and Slotted Aloha based RFID Anti-collision Protocols", *Computer Communications*, Elsevier, vol. 32, Issue 5, (2009), pp. 961-973.
- [2] C. Wang, C. Lee and M. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Anti-Collision Method for Mobile RFID Tag Identification", *Journal of Convergence Information Technology*, AICIT, vol. 6, no. 4, (2011), pp. 340-351.
- [3] M. Djeddou, R. Khelladi and M. Benssalah, "Improved RFID Anti-collision Algorithm", *International Journal of Electronics and Communications*, Elsevier, vol. 67, Issue 3, (2013), pp. 256-262.
- [4] W. Chen and G. Lin, "An Efficient Anti-Collision Method for tag Identification in a RFID System", *IEICE Transactions on Communications*, IEICE, vol. E89-B, no. 12, (2006), pp. 3386-3392.
- [5] Y. Park and Y. Kim, "Accelerating RFID Tag Identification Processes with Frame Size Constraint Relaxation", *Journal of Information and Communication Convergence Engineering*, KIICE, vol. 10, no. 3, (2012), pp. 242-247.
- [6] K. Finkenzeller, "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification", Carl Hanser Verlag GmbH and Co., Germany, (2002).
- [7] S. Lee, S. Joo and C. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification", In *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous2005)*, (2005), pp. 166-172.
- [8] W. Su, N. V. Alchazidis and T. T. Ha, "Multiple RFID Tags Access Algorithm", *IEEE Transactions on Mobile Computing*, IEEE, vol. 9, no. 2, (2010), pp. 174-187.
- [9] I. Lim, "A Scheme for Estimating Number of Tags in FSA-based RFID Systems", *International Journal of MICS, KIMICS*, vol. 7, no. 2, (2009), pp. 164-169.
- [10] M. H. MacDougall, "Simulating Computer Systems: Techniques and Tools", MIT Press, USA, (1987).

Author



Intaek Lim, He received the B.S. degree in computer science from University of Ulsan, Ulsan, Korea, in 1984, and the M.S. degree in computer science and statistics from Seoul National University, Seoul, Korea, in 1986. He received the Ph. D. degree in computer engineering from University of Ulsan, Ulsan, Korea, in 1998. From 1986 to 1993, he was a Senior Researcher at Samsung Electronics Co., Ltd. In 1998, he joined the faculty at Pusan University of Foreign Studies, Busan, Korea, where he is currently a Professor in the Department of Embedded Software. His research interests include the MAC protocol design, ad-hoc network, RFID, and mobile computing.

