Gen2-Based Tag Anti-collision Algorithms Using Chebyshev's Inequality and Adjustable Frame Size

Xiao Fan, InChan Song, KyungHi Chang, Dong-Beom Shin,
Heyung-Sub Lee, Cheol-Sig Pyo, and Jong-Suk Chae

Arbitration of tag collision is a significant issue for fast tag identification in RFID systems. A good tag anti-collision algorithm can reduce collisions and increase the efficiency of tag identification. EPCglobal Generation-2 (Gen2) for passive RFID systems uses probabilistic slotted ALOHA with a Q algorithm, which is a kind of dynamic framed slotted ALOHA (DFSA), as the tag anti-collision algorithm. In this paper, we analyze the performance of the Q algorithm used in Gen2, and analyze the methods for estimating the number of slots and tags for DFSA. To increase the efficiency of tag identification, we propose new tag anti-collision algorithms, namely, Chebyshev's inequality, fixed adjustable framed Q, adaptive adjustable framed Q, and hybrid Q. The simulation results show that all the proposed algorithms outperform the conventional Q algorithm used in Gen2. Of all the proposed algorithms, AAFQ provides the best performance in terms of identification time and collision ratio and maximizes throughput and system efficiency. However, there is a tradeoff of complexity and performance between the CHI and AAFQ algorithms.

Keywords: RFID, Tag anti-collision, Gen2.

Manuscript received Jan. 22, 2008; revised June 13, 2008; accepted Aug. 20, 2008.

This work was supported by the IT R&D program of MIC/IITA, Rep of Korea [2005-S-106-03, Development of Sensor Tag and Sensor Node Technologies for RFID/USN].

Xiao Fan (email: fxismonk@gmail.com), InChan Song (email: ny10023@hanmail.net), and KyungHi Chang (phone: +82 32 860 8422, email: khchang@inha.ac.kr) are with the Graduate School of Information Technology & Telecommunications, Inha University, Incheon, Rep. of Korea.

Dong-Beom Shin (email: sdb@etri.re.kr), Heyung-Sub Lee (email: leehs@etri.re.kr), Cheol-Sig Pyo (email: cspyo@etri.re.kr), and Jong-Suk Chae (email: jschae@etri.re.kr) are with IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

I. Introduction

A radio frequency identification (RFID) system is a contactless automatic identification system, which comprises interrogators, also known as readers, and tags, also known as labels [1]. A reader can identify a tag by its unique ID number and obtain the information stored on the tag. When multiple tags respond to the reader at the same time, a tag collision occurs and the reader fails to identify any tag. A good tag anti-collision algorithm can reduce collisions so as to increase the efficiency of identification.

Two types of tag anti-collision algorithms widely used in RFID systems are the binary tree algorithm and the ALOHA algorithm [2]. A binary tree algorithm splits tags into two subsets when there is a collision, then divides and processes every subset separately. On the other hand, an ALOHA algorithm decreases the probability of collision by scheduling the responses of tags [3]. Both of them are based on time division multiple access scheduling.

There are several versions of the ALOHA algorithm. The simplest version is pure ALOHA. When a tag reaches the interrogation area of a reader, the tag transmits the data immediately. This algorithm has a high probability of collision [2], [4]. An improved algorithm is slotted ALOHA. In this algorithm, time is divided into slots, and tags can only respond at the beginning of a time slot. As a consequence, the rate of collision can be reduced by half [4], [5]. However, due to the limitation of the number of slots, this algorithm is usually used when there are a few tags in the interrogation zone. The framed slotted ALOHA (FSA) algorithm can solve this problem. In this algorithm, time is divided into frames, and every frame consists of several slots. However, this FSA uses a fixed frame

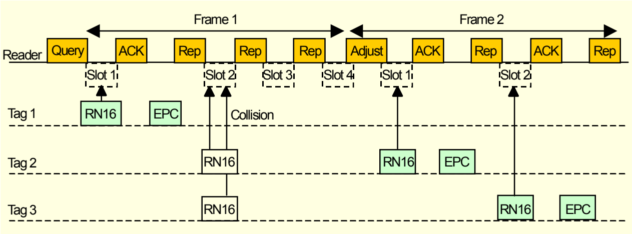


Fig. 1. Conceptual Gen2 anti-collision algorithm.

```
Reader sends Query.
for inventory procedure
      Every tag generates RN16 & slot counter.
      for current frame
            If slot counter == 0
                  Tag replies its RN16.
            end
            If a single tag replies
                  Reader sends ACK with RN16.
                  If RN16 received by tag == RN16 saved in tag
                        Tag sends EPC to reader.
                  end
                  Reader sends QueryRep.
            else if multiple tags reply
                  Reader sends QueryRep or QueryAdjust.
            else if no tag replies
                  Reader sends QueryRep or QueryAdjust.
            end
            If tag receives QueryRep
                  slot counter = slot counter – 1
            end
      end
      Reader sends QueryAdjust.
end
```

Fig. 2. Probabilistic slotted ALOHA in Gen2.

size and does not change the frame size during the process of tag identification. This is simple, but not efficient for tag identification [2]. The dynamic framed slotted ALOHA (DFSA) algorithm can change the frame size to increase the efficiency of tag identification, and there are several ways to modify the frame size [6]-[8].

EPCglobal Class-1 Generation-2 (Gen2) for passive RFID systems uses the probabilistic slotted ALOHA with a Q algorithm, which is a kind of DFSA, as a tag anti-collision algorithm [1]. In this paper, we propose four new tag anti-collision algorithms to increase the efficiency of tag identification. The first proposed algorithm is Chebyshev's inequality (CHI) algorithm, which is based on Chebyshev's inequality and can estimate the frame size more accurately than the Q algorithm in Gen2. The proposed fixed adjustable framed Q (FAFQ) algorithm and adaptive adjustable framed Q (AAFQ) algorithm implement QueryAdjust command during a frame to modify the current frame size according to the status of slot occupation. The last proposed hybrid Q algorithm is based on combining Chebyshev's inequality and the adjustment for the current frame size. That is, we use the method of Chebyshev's inequality to set the frame size at the

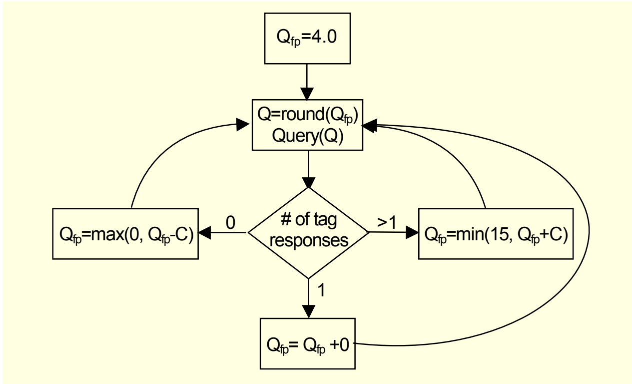


Fig. 3. Algorithm for choosing the slot-count parameter Q.

beginning of a frame and use a QueryAdjust command to adjust the current frame size during a frame.

The remaining part of this paper is organized as follows. Section II introduces and analyzes the Q algorithm used in Gen2. In section III, we estimate the number of slots and the number of tags for the DFSA algorithm. In section IV we describe our proposed new tag anti-collision algorithms, and we verify the performance of the Q algorithm and the proposed algorithms in section V. Finally, in section VI we draw conclusions.

II. Tag Anti-collision Algorithm in Gen2

1. Q Algorithm in Gen2

The tag anti-collision algorithm in Gen2 is probabilistic slotted ALOHA, which is a kind of DFSA [1]. Probabilistic slotted ALOHA defined by the EPCglobal Gen2 protocol is shown in Fig. 1 [9], which describes an inventory process for cases in which there are multiple tags. This algorithm is shown in Fig. 2.

Probabilistic slotted ALOHA in Gen2 changes the frame size by varying the Q value, which is used in Query and QueryAdjust commands. Figure 3 shows an exemplary algorithm called the Q algorithm for choosing Q given in Gen2 specification [1]. Here, Q_{fp} is a floating-point representation of Q. When a new Q value is required, the reader rounds Q_{fp} to the nearest integer and sets it as a value of Q. There are three cases to change Q_{fp}.

Case 1. If a single tag replies,

$$Q_{fp} = Q_{fp}.$$

Case 2. If no tag replies,

$$Q_{fp} = Q_{fp} - C \text{ due to the empty slot.}$$

Case 3. If multiple tags reply,

$$Q_{fp} = Q_{fp} + C \text{ due to the collided slot.}$$

The typical value for the constant C is between 0.1 and 0.5. When Q is large, the reader uses a small value of C, and larger value of C for small Q [1].

2. Analytical and Simulation Performance of Q Algorithm

As previously mentioned in the introduction, the probabilistic slotted ALOHA with Q algorithm in Gen2 is a kind of DFSA. Given N slots and n tags, r tags in one slot are binomially distributed with parameters n and $1/N$ [6]:

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

The value r is the number of tags in a particular slot, that is, the occupancy number of the slot. The expected value $a_r^{N,n}$ of the number of slots with occupancy number r is given in [6] as

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

Consequently, the expected value $a_0^{N,n}$, that is, the number of empty slots is given by

$$a_0^{N,n}=NB_{n,\frac{1}{N}}(0)=N\binom{n}{0}\left(\frac{1}{N}\right)^{0}\left(1-\frac{1}{N}\right)^{n}=N\left(1-\frac{1}{N}\right)^{n}, \quad (3)$$

and the expected value $a_1^{N,n}$, that is, the number of successful slots is given by

$$a_1^{N,n}=NB_{n,\frac{1}{N}}(1)=N\binom{n}{1}\left(\frac{1}{N}\right)^{1}\left(1-\frac{1}{N}\right)^{n-1}=n\left(1-\frac{1}{N}\right)^{n-1}. \quad (4)$$

Then, the expected value $a_k^{N,n}$, that is, the number of collided slots becomes

$$a_k^{N,n}=NB_{n,\frac{1}{N}}(k)=N-a_0^{N,n}-a_1^{N,n}, \quad k\geq 2. \quad (5)$$

According to (4), we calculate the theoretical number of successful slots when N is 16, 32, and 64 for various numbers of tags. Figure 4 shows good agreement between theoretical and simulated values.

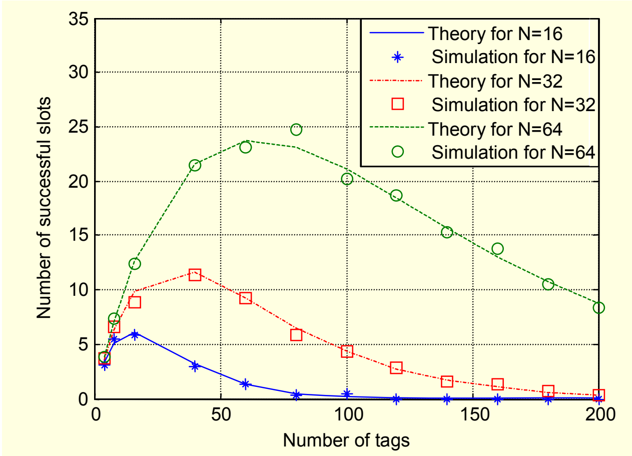


Fig. 4. Number of successful slots in a frame (theory vs. simulation).

3. Performance Index

In this paper, we propose four performance indices to examine the efficiency of the anti-collision algorithms.

A. Identification Time

Identification time is defined as the time to identify all the tags in the interrogation zone. Instead, we can use the total number of slots to reflect this performance.

Now we derive the theoretical value for the total number of slots. In the Gen2 scenario, the QueryAdjust command sets the number L_i of slots for the i-th frame to $2\cdot L_{i-1}$, $L_{i-1}/2$, or L_{i-1}. According to the Q algorithm used in Gen2, a variable I is used to decide the increment or decrement of the number L_i of slots:

$$I=round[(a_k^{N,n}-a_0^{N,n})\cdot C], \quad (6)$$

where $a_k^{N,n}$ and $a_0^{N,n}$ are the expected numbers of collided slots and empty slots, respectively, and are given by (5) and (3).

The number L_i of slots of the i-th frame is given by

$$L_i=\begin{cases}2\cdot L_{i-1} & \text{if } I\geq 1,\\ L_{i-1} & \text{if } I=0,\\ L_{i-1}/2 & \text{if } I\leq -1.\end{cases} \quad (7)$$

Here, the initial frame size L_1 is set to 16 [1]. Then we obtain the expected total number of time slots as

$$N_{Total}=\sum_i L_i\ . \quad (8)$$

The summation in (8) stops when all tags in the interrogation zone are identified.

B. Throughput

We define throughput for Gen2 as

$$\begin{aligned}\text{Throughput} &= \frac{\text{Total transmission bits}}{\text{Identification time}}\ \text{(bps)}\\ &= \frac{\sum_i(\text{Transmission bits in the } i\text{-th successful slot})}{\text{Identification time}}\\ &= \frac{\text{Data rate}\times\sum_i(i\text{-th Successful slot time})}{\text{Identification time}}\\ &= \text{Data rate}\\ &\quad\times\frac{\text{Total successful slot time (only considering traffic)}}{\text{Identification time}}.\end{aligned} \quad (9)$$

C. System Efficiency

There are two definitions for system efficiency. The conventional definition for system efficiency is given in [8] as

$$\text{System efficiency} = \frac{\text{Number of successful slots}}{\text{Total number of slots}} . \quad (10)$$

By using (4), (8), and (10), the expected system efficiency is derived as

$$\text{System efficiency} = \frac{\sum_i a_{1,i}}{N_{Total}} . \quad (11)$$

The slot time in Gen2 varies mainly due to pulse interval encoding (PIE). Therefore, we newly define system efficiency by using slot time instead of the number of slots:

$$\text{System efficiency} = \frac{\text{Total successful slot time}}{\text{Identification time}} . \quad (12)$$

D. Collision Ratio

A good anti-collision algorithm should provide a low collision ratio, which is defined as

$$C_{ratio} = \frac{\text{Total number of collided slots}}{\text{Total number of slots}} . \quad (13)$$

By using (5), (8), and (13), we can derive the theoretical value of the collision ratio as

$$C_{ratio} = \frac{\sum_i a_{k,i}}{N_{Total}} . \quad (14)$$

We can also derive the collision ratio by using (1) as

$$\begin{aligned} C_{ratio} &= 1 - P_{succ} - P_{empty} \\ &= 1 - B_{n,\frac{1}{N}}(1) - B_{n,\frac{1}{N}}(0) \\ &= 1 - \binom{n}{1}\left(\frac{1}{N}\right)^1\left(1-\frac{1}{N}\right)^{n-1} - \binom{n}{0}\left(\frac{1}{N}\right)^0\left(1-\frac{1}{N}\right)^{n-0} \\ &= 1 - \left(1-\frac{1}{N}\right)^n\left(1+\frac{n}{N-1}\right), \quad (15) \end{aligned}$$

where P_{succ} is the probability of successful slots and P_{empty} is the probability of empty slots.

III. Dynamic Framed Slotted ALOHA

DFSA changes the frame size dynamically. To set the appropriate frame length, slot estimation is required to estimate the optimal frame size. For slot estimation, we need to know the number of tags, so both slot and tag estimation methods are essential in DFSA.

1. Slot Estimation Methods

There are two methods to estimate the optimal number of slots. The first method is based on the minimization of identification time, and the second one considers maximizing the system throughput. Both methods draw the same conclusion that the optimal frame size is equal to the number of tags [7]:

$$L_{optimal} = n . \quad (16)$$

2. Tag Estimation Methods

According to (16), the number of slots is equal to the number of tags. Therefore, in order to estimate the number of slots, we should estimate the number of tags first, using one of the methods below.

A. Lower Bound

The first estimation method is obtained through the observation that a collision involves at least two different tags [6]. Therefore, a lower bound on n can be obtained by the following simple estimation:

$$n_{\text{Lower Bound}} = 2 \times (\text{Number of collided slots}) . \quad (17)$$

B. Maximum Throughput

The *a posteriori* expectation of the number of tags that choose one time slot simultaneously is equal to 2.39 [10]. Using this *a posteriori* expected value, a system can reach the maximum throughput [6]. Therefore, n can be calculated by

$$n_{\text{MaximumThroughput}} = 2.39 \times (\text{Number of collided slots}) . \quad (18)$$

C. Collision Ratio

After one frame, we have the frame size and collision ratio, so n can be calculated by using (15).

D. Chebyshev's Inequality

This method is based on the fact that the outcome of a random experiment is most likely somewhere near the expected value [6]. Thus an alternative estimation function uses the distance between the read result c and the expected value vector to determine the value of n for which the distance becomes minimal. We denote this estimation function by ξ as

$$\xi(N, c_0, c_1, c_k) = \min_n \left\| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_{\geq 2}^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right\| . \quad (19)$$

Now, we compare the performance of these four tag estimation methods in Figs. 5 and 6. Figure 5 shows the total number of slots for tag identification. In these four methods, Chebyshev's inequality using (19) costs the fewest slots. The

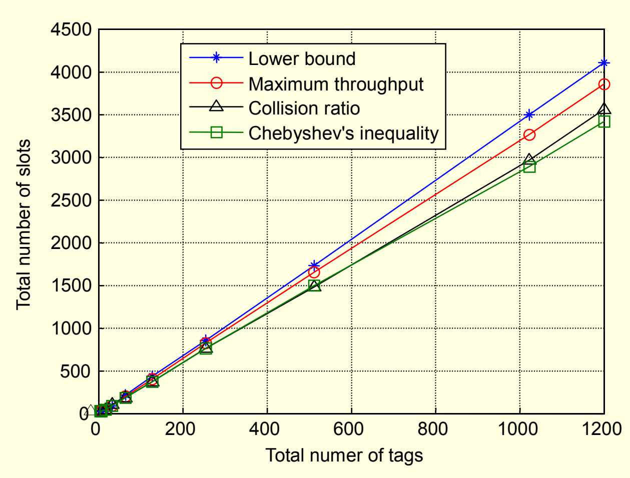


Fig. 5. Total number of slots for tag identification with four tag estimation methods.

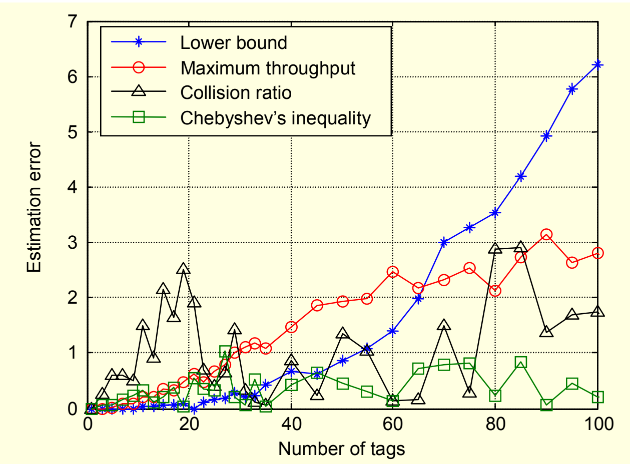


Fig. 6. Estimation error for the case of a 128-slot frame.

estimation error of these four methods, which is defined as the difference between the estimated number of tags and the actual number of tags, is given in Fig. 6. The method of Chebyshev's inequality has fewer estimation errors than the other three methods.

IV. Proposed Gen2-Based Tag Anti-collision Algorithms

Probabilistic slotted ALOHA with Q algorithm uses a Query command, which includes a parameter Q, to set the frame size. In the Gen2 Q algorithm, the frame size is equal to 2^Q. After a frame, a QueryAdjust command is transmitted from reader to tag to adjust the Q value by ±1. Consequently, the frame size is doubled or divided by 2. Because the frame size in the Q algorithm may not be optimal according to (16), we use other more accurate tag estimation methods instead of the Q algorithm in Gen2.

On the other hand, the QueryAdjust command allows us to increase or decrease the Q value and to change the next frame size at the end of the current frame. The QueryAdjust command can also be used to modify the current frame size during a frame. However, the exact procedure to perform such an adjustment using the QueryAdjust command during a frame is not specified in the Gen2 specification, so we have developed our own procedure and algorithm to be employed in the Gen2 scenario.

1. Chebyshev's Inequality Algorithm

Section III introduced four methods to estimate the number of tags, which are the methods of lower bound, maximum throughput, collision ratio, and Chebyshev's inequality, and we verified that Chebyshev's inequality (19) gives the most accurate estimation for the number of tags.

Now, we propose the CHI algorithm to estimate the optimal frame length, which is set to the size of the following frame, instead of the Q algorithm used in Gen2. Figure 7 shows the implementation of probabilistic slotted ALOHA with a Q algorithm in Gen2. At the end of a frame, a QueryAdjust command is transmitted to modify the frame size. In the proposed CHI algorithm, the QueryAdjust command is replaced by Chebyshev's inequality to set the size of the following frame as shown in Fig. 8, where the proposed part is represented by a shadowed block. Because the proposed CHI algorithm can estimate the number of tags accurately and obtain the optimal frame size, it can improve the performance of tag identification compared with the Q algorithm in Gen2.

2. Adjustable Framed Q Algorithm

In addition to setting the frame size at the beginning of a frame using Chebyshev's inequality, we can also adjust the frame size during the frame adaptively. We propose the adjustable framed Q (AFQ) algorithm to implement the QueryAdjust command during a frame and modify the current frame size dynamically. Compared with the CHI algorithm, the AFQ algorithm changes the frame size multiple times during a frame. The flow chart for the proposed AFQ algorithm is shown in Fig. 9, where the proposed part is represented by a shadowed block. Compared with the Q algorithm in Gen2, the proposed AFQ algorithm can change the current frame size dynamically and use the following criterion to judge whether the reader sends a QueryRep command or a QueryAdjust command at the beginning of the following slot. If it is not necessary to change the current frame size, the reader sends a QueryRep command. Otherwise, a QueryAdjust command is sent to modify the current frame size.

We provide two different criteria to implement the AFQ

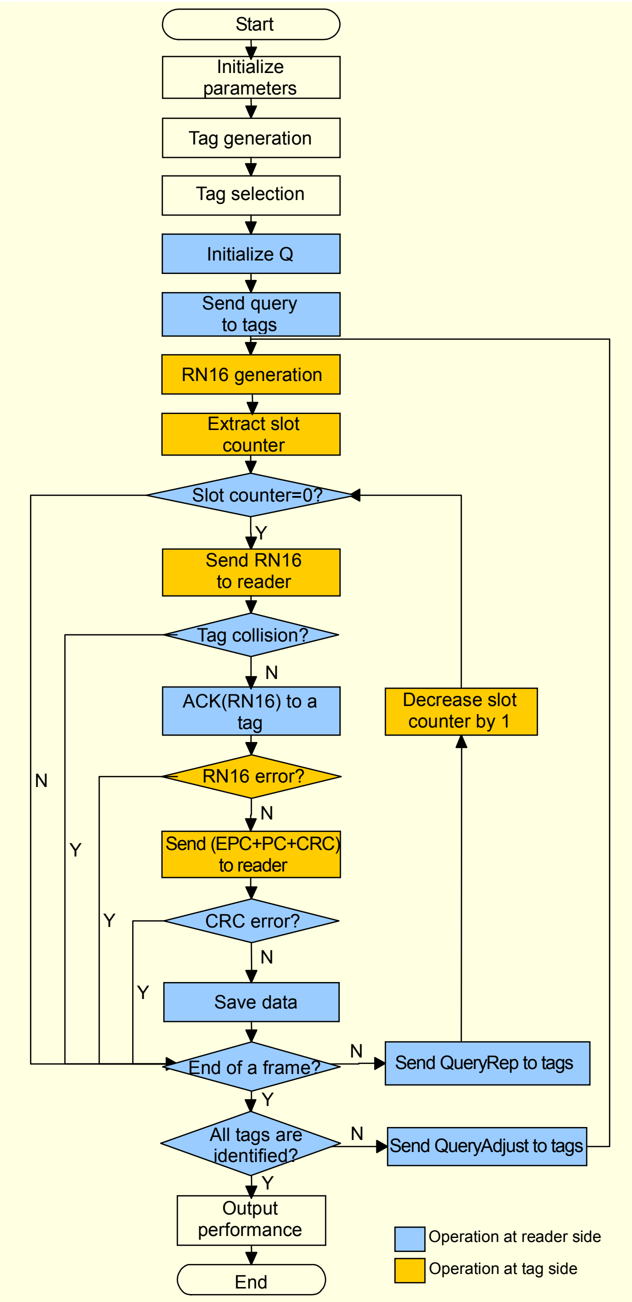


Fig. 7. Implementation of probabilistic slotted ALOHA in Gen2.

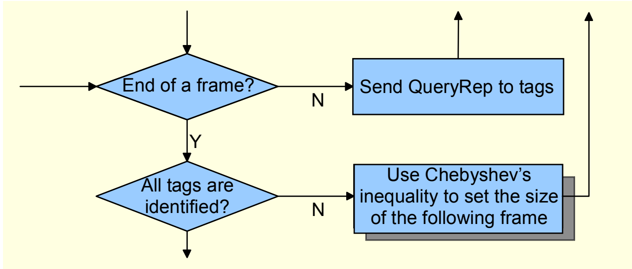


Fig. 8. Flow chart of the CHI algorithm.

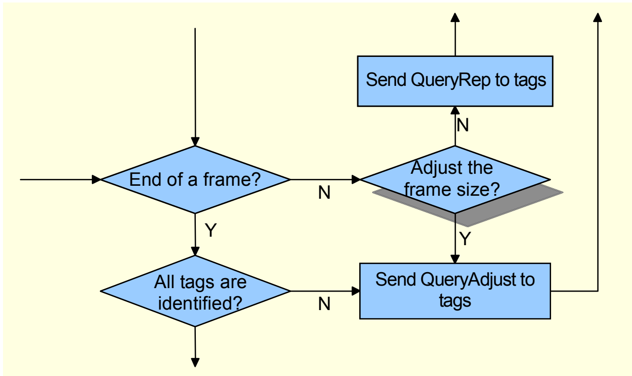


Fig. 9. Flow chart of the AFQ algorithm.

```
If number of continuous collided slots > Th_coll
    Reader sends QueryAdjust.
    Q = Q + 1
else if number of continuous empty slots > Th_emp
    Reader sends QueryAdjust.
    Q = Q - 1
else
    Reader sends QueryRep.
end
```

Fig. 10. FAFQ algorithm.

algorithm depending on the usage of fixed threshold values or adaptive threshold values.

A. Fixed Adjustable Framed Q Algorithm

When a QueryAdjust command is used during a frame to adjust the frame size dynamically, two threshold values are necessary to decide whether to increase or decrease the number of slots in a frame: Th_{emp} for continuous empty slots and Th_{coll} for continuous collided slots.

If the number of continuous collided slots is larger than the threshold value Th_{coll}, we use a QueryAdjust command to increase the frame size to prevent more collision. On the other hand, if the number of continuous empty slots exceeds the threshold value Th_{emp}, the frame size is decreased by using a QueryAdjust command to reduce the number of empty slots. Otherwise, the frame size is left unchanged. This FAFQ algorithm is given in Fig. 10.

B. Adaptive Adjustable Framed Q Algorithm

In the FAFQ algorithm, the threshold values of Th_{coll} and Th_{emp} are fixed during the whole inventory procedure. However, if the frame size is comparably small considering the large number of tags, there will be a greater chance of slots colliding; otherwise, there will be more chance of empty slots. By the adaptive adjustment of the values of Th_{coll} and Th_{emp}, we can minimize the frequency of transmitting QueryAdjust commands and so minimize the identification time.

```
Reader sends QueryAdjust.
If Q == Q + 1
      Th_emp = Th_emp + 1
      Th_coll = Th_coll + 1
else if Q == Q - 1
      Th_emp = Th_emp – 1
      Th_coll = Th_coll – 1
end
```

Fig. 11. AAFQ algorithm.

```
Reader sends Query.
for inventory procedure
      Every tag generates RN16 & slot counter.
      for current frame
            If slot counter == 0
                  Tag replies its RN16.
            end
            If a single tag replies
                  Reader sends ACK with RN16.
                  If RN16 received by tag == RN16 saved in tag
                        Tag sends EPC to reader.
                  end
                  Reader sends QueryRep.
            else if multiple tags reply
                  If number of continuous collided slots > Th_coll
                        Reader sends QueryAdjust.
                        Q = Q + 1
                        Th_emp = Th_emp + 1
                        Th_coll = Th_coll + 1
                  else
                        Reader sends QueryRep.
                  end
            else if no tag replies
                  If number of continuous empty slots > Th_emp
                        Reader sends QueryAdjust.
                        Q = Q – 1
                        Th_emp = Th_emp – 1
                        Th_coll = Th_coll – 1
                  else
                        Reader sends QueryRep.
                  end
            end
            If tag receives QueryRep
                  slot counter = slot counter – 1
            end
      end
      Use Chebyshev's inequality to set the size of the following frame.
end
```

Fig.12. Hybrid Q algorithm.

Therefore, we propose AAFQ algorithm, in which the threshold values of Th_{coll} and Th_{emp} are adaptively changed for different frame sizes as described in Fig. 11. The reader sends a QueryAdjust command to adjust the frame size by changing the Q value. If the value of Q is increased by 1, the frame size is doubled, which means there are too many tags, so it is necessary to increase the frame size. Accordingly, the threshold values of Th_{coll} and Th_{emp} are also increased by 1. Otherwise, if the Q value is decreased, we decrease the threshold values of Th_{coll} and Th_{emp} correspondingly.

3. Hybrid Q Algorithm

By combining the ideas of the previous two methods, we designed the hybrid Q algorithm, which uses Chebyshev's inequality to estimate and set the frame size at the beginning of a frame, while during a frame the threshold values of Th_{coll} and Th_{emp} are used to decide to increase or decrease the current frame size by comparing them with the continuous number of collided and empty slots, respectively. The hybrid Q algorithm is described in Fig. 12.

Table 1. Simulation parameters for the Gen2 scenario.

Parameters	Descriptions	Values in specification	Values in simulation
Tari	Reference time interval for a data-0 in interrogator-to-tag signaling	6.25 µs, 12.5 µs, or 25 µs	12.5 µs
DR	Divide ratio	64/3 or 8	8
RTcal	Interrogator-to-tag calibration	2.5 Tari ≤ RTcal ≤ 3.0 Tari	3 Tari = 37.5 µs
TRcal	Tag-to-interrogator calibration RTcal ≤ TRcal ≤ 3 RTcal	17.2 µs ≤ TRcal ≤ 200 µs, if DR = 8	2 RTcal = 75 µs
LF	Link frequency	LF = DR/TRcal	107 kHz
T_{pri}	Link pulse-repetition interval	T_{pri} = 1/LF	9.375 µs
T_1	Time from interrogator transmission to tag response	MAX (RTcal, 10T_{pri})	10 T_{pri} = 93.75 µs
T_2	Time from tag response to interrogator transmission	3.0 T_{pri} ≤ T2 ≤ 20.0 Tpri	10 T_{pri} = 93.75 µs
T_3	Time an interrogator waits after T1 before it issues another command	0.0 T_{pri}	0 µs
T_4	Minimum time between interrogator commands	2.0 RTcal	75 µs
T=>R Data rate	Tag-to-interrogator link data rate	LF, if FM0 modulation	LF = 107 kbps

V. Simulation Results and Performance Verification

According to the previous analysis and design, we carried out simulations for the Q algorithm and all the proposed algorithms, including CHI, FAFQ, AAFQ, and hybrid Q. We compare them in terms of identification time, throughput, system efficiency, and collision ratio.

The simulation parameters for the Gen2 scenario are shown in Table 1, which were chosen based on the Gen2 specification. In the FAFQ algorithm, both Th_{coll} and Th_{emp} were set to 10 in our simulation. To adaptively adjust the threshold values of Th_{coll} and Th_{emp} according to the varying frame size, both Th_{coll} and Th_{emp} were set to be equal to the Q value in the AAFQ algorithm.

Comparison of algorithms for each performance index is shown in Figs. 13 to 16. Figure 13 shows the identification time of the Q algorithm and our proposed algorithms. All the

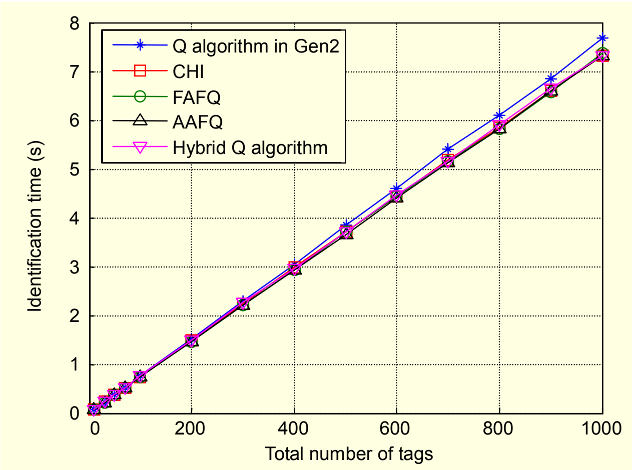


Fig. 13. Identification time of Q algorithm and proposed algorithms.

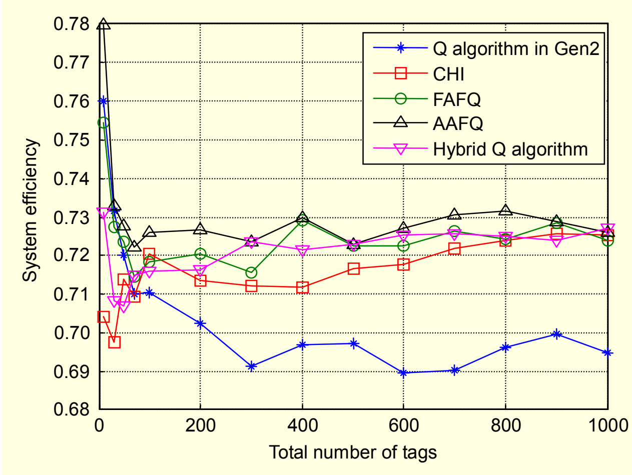


Fig. 15. System efficiency of Q algorithm and proposed algorithms.

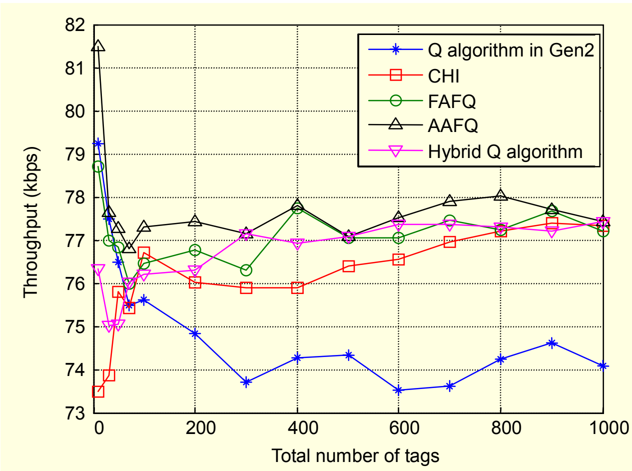


Fig. 14. Throughput of Q algorithm and proposed algorithms.

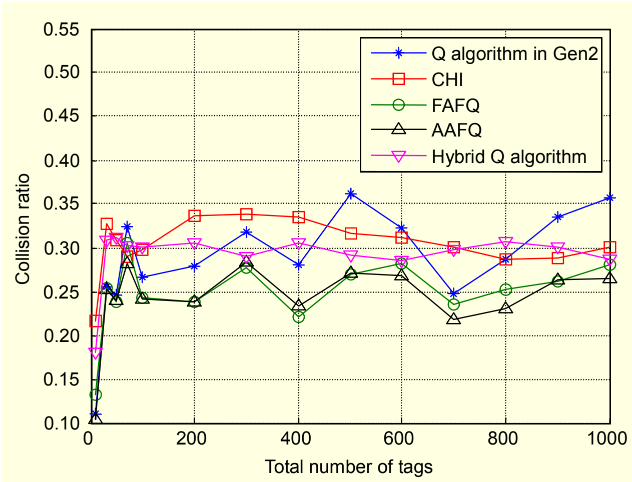


Fig. 16. Collision ratio of Q algorithm and proposed algorithms.

proposed algorithms reduce the identification time compared with the conventional Q algorithm used in Gen2. Figures 14 and 15 show the results for throughput and system efficiency. Again, all the proposed algorithms outperform the Q algorithm, and the AAFQ algorithm shows the best throughput and system efficiency. Finally, as seen in Fig. 16, the CHI and hybrid Q algorithms show more consistent collision ratios in relation to the total number of tags, than that of Q algorithm. The collision ratios of FAFQ and AAFQ are better than the other three algorithms.

Based on the above discussion, we conclude that AAFQ shows the best performance among all the conventional and proposed anti-collision algorithms. The AAFQ algorithm is superior in terms of identification time and collision ratio and maximizes throughput and system efficiency. However, the AAFQ algorithm changes the frame size multiple times during a frame, which increases the computational complexity compared with the CHI algorithm, which only calculates and sets the frame size once per frame. Therefore, there is a tradeoff of complexity and performance between the CHI and AAFQ algorithms.

VI. Conclusion

In this paper, we introduced and analyzed the performance of the Q algorithm in EPCglobal Gen2 for passive RFID systems. We compared the tag estimation methods for DFSA, and demonstrated that Chebyshev's inequality is the best estimation method to obtain the optimal frame size. We also propose new tag anti-collision algorithms, namely, CHI, FAFQ, AAFQ, and hybrid Q. We verified the performance of each algorithm in terms of identification time, throughput, system efficiency, and collision ratio. These performance indices were improved by accurate estimation of the number of tags for tag identification in the CHI and hybrid Q algorithms, and by dynamical adjustment of the current frame size in the FAFQ, AAFQ, and hybrid Q algorithms. The simulation results demonstrate that the AAFQ algorithm achieves the best performance in terms of identification time and collision ratio and maximizes throughput and system efficiency. However, compared with the CHI algorithm, AAFQ has more computational complexity due to frequent changes of frame size. Therefore, there is a tradeoff between the AAFQ algorithm and CHI algorithm, and they can be used in different scenarios.

References

[1] EPCglobal, *Radio-Frequency Identity Protocols Class-1 Generation-2 UHF Protocol for Communications at 860 MHz-960MHz Version 1.1.0*, Dec. 17, 2005.

[2] T. Cheng and J. Li, "Analysis and Simulation of RFID Anti-collision Algorithms," *Proc. 9th Int. Conf. on Advanced Communication Technology*, vol. 1, Feb. 2007, pp. 697-701.

[3] J. Wang, D. Wang, and Y. Zhao, "A Novel Anti-collision Algorithm with Dynamic Tag Number Estimation for RFID Systems," *Proc. Int. Conf. on Communication Technology,* Nov. 2006, pp. 1-4.

[4] K. Finkenzeller, *RFID Handbook - Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd ed., Wiley, 2003.

[5] H. Harada and R. Prasad, *Simulation and Software Radio for Mobile Communications*, Artech House, 2002.

[6] Harald Vogt, "Efficient Object Identification with Passive RFID Tags," *Proc. Int. Conf. on Pervasive Computing,* Zurich, vol. 2414, 2002, pp. 98-113.

[7] J.R. Cha and J.H. Kim, "Dynamic Framed Slotted ALOHA Algorithms Using Fast Tag Estimation Method for RFID System," *Proc. Consumer Communications and Networking Conf. 2006*, vol. 2, Jan. 2006, pp. 768-772.

[8] C.H. Quan et al., "A Study on Anti-collision Algorithm in Gen2 Protocol Based RFID Systems," *KICS Journal*, vol. 31, no. 6, June 2006, pp. 561-571.

[9] Y. Kawakita and J. Mitsugi, "Anti-collision Performance of Gen2 Air Protocol in Random Error Communication Link," *Proc. Int. Symposium on Applications and the Internet Workshops*, Jan. 2006, pp. 68-71.

[10] Frits Schoute, "Dynamic Frame Length ALOHA," *IEEE Trans. on Communications*, vol. 31, Apr. 1983, pp. 565-568.

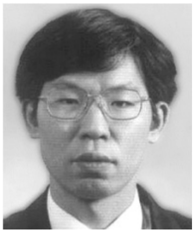
Xiao Fan received the BS degree in clinical medicine from China Medical University, China, in 2003, and a second BS degree in computer software from Tsinghua University, China, in 2005. He is currently with the Graduate School of Information Technology and Telecommunications, Inha University, where he has been an MS student since 2007. His research interests include RFID/USN, reader/tag anti-collision MAC protocols, and algorithm optimization.

InChan Song received the BS degree in electronics engineering from Incheon University, Incheon, Korea, in 2007. He is currently with the Graduate School of Information Technology and Telecommunications, Inha University, where he has been an MS student since 2007. His research interests include RFID/USN, reader/tag anti-collision MAC protocols, and algorithm optimization.

KyungHi Chang (M'93–SM'98) received the BS and MS degrees in electronics engineering from Yonsei University, Seoul, Korea, in 1985 and 1987, respectively. He received his PhD degree in electrical engineering from Texas A&M University, College Station, Texas, in 1992. From 1989 to 1990, he was with the Samsung Advanced Institute of Technology (SAIT) as a member of research staff and was involved in digital signal processing system design. From 1992 to 2003, he was with ETRI as a principal member of technical staff. During this period, he led the design teams working on WCDMA UE modem and 4G radio transmission technology. He is currently with the Graduate School of Information Technology and Telecommunications, Inha University, where he has been a professor since 2003. His current research interests include RTT design for IMT-Advanced and 3GPP LTE systems, WMAN system design, cognitive radio, cross-layer design, cooperative relaying systems, and RFID/USN. Dr. Chang has served as a senior member of IEEE since 1998. Currently, he is editor-in-chief of the *Korean Institute of Communication Sciences (KICS) Proceedings*. He has also served as an editor of ITU-R TG8/1 IMT.MOD, and he is currently an international IT standardization expert of the Telecommunications Technology Association (TTA).

Dong-Beom Shin received the BS and MS degrees in electronics engineering from Chungnam National University, Daejeon, Korea, in 1991 and 1993, respectively. From 1993 to 2000, he was with the Agency for Defense Development, where he was engaged in military related testing and evaluation research. Since May 2000, he has been with the Electronics and Telecommunications Research Institute (ETRI), where he is engaged in the research and development of high-speed router systems, EPON/GPON OLT systems, and RFID sensor tag reader systems. His research interests include design and implementation of RFID systems.

Heyung-Sub Lee received the BS, MS, and PhD degrees in electronics engineering from Chungnam National University, Daejeon, Korea, in 1985, 1994, and 2002, respectively. From 1985 to 1990, he was with Samsung Electronics, where he was engaged in the research and development of SRAM. Since September 1990, he has worked with the Electronics and Telecommunications Research Institute (ETRI), where he is engaged in the research and development of SDH-based transmission systems, high-speed router systems, optical premise network systems, and so on. He is currently a principal member of engineering staff in the RFID/USN division of ETRI and is the team leader of the RFID basic technology team. His research interests include technology development and service implementation of RFID systems.



Cheol-Sig Pyo received the BS degree in electronic engineering from Yonsei University, Seoul, Rep. of Korea, in 1991, and the MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 1999. Currently, he is a principal engineer with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea, where he is a team leader with the Radio Frequency Identification (RFID)/Ubiquitous Sensor Network (USN) Service Research Team. Since 1991, he has worked on satellite communication systems, antennas, RFID, and wireless sensor networks at ETRI. His current research interests include RFID, USN, and convergence technology.

Jong-Suk Chae received the BS degree in electronic engineering from National Aviation Collage, Korea, in 1977; and the MS and PhD degrees in electronic engineering from Yonsei University, Korea, in 1979 and 1989, respectively. Since 1985, he has been with the Electronics and Telecommunications Research Institute (ETRI), where he has worked on many R&D projects including IMT2000 system development and earth station development. He is currently the executive director of the RFID/USN research division and is a permanent member of the IEEK and KEES, Korea. His research interests include new RFID/USN systems.