Optimal Parameter Selection of Q-Algorithm in EPCglobal Gen-2 RFID System

Intaek Lim, *Member, KIMICS*

Abstract—Q-algorithm is proposed at EPCglobal Class-1 Generation-2 RFID systems to determine the frame size of next query round. In Q-algorithm, the reader calculates the frame size without estimating the number of tags. But, it uses only the slot conditions: empty, success, or collision. Therefore, it wastes less computational cost and is simpler than other algorithms. However, the constant parameter *C* value, which is used for calculating the next frame size, is not optimized. In this paper, we propose the optimized *C* values of Q-algorithm according to the number of tags within the identification range of reader through a lot of computer simulations.

Index Terms—RFID, Anti-collision algorithm, FSA algorithm, EPCglobal Gen-2, Q-algorithm

I. INTRODUCTION

RFID (Radio Frequency Identification) systems allow contactless identification of objects using radio frequency. In recent years, RFID technology has been widely adapted in a variety of applications such as production processes, distribution, animal identification, etc., which allow collecting automatically information about goods, people, and animals [1][2]. Unlike the traditional bar code system, RFID has advantages of high storage capacity and simultaneous identification. Another benefit is that the information stored in the tags can be easily analyzed and transmitted through a computer network, and is useful to manufacturers, retailers, and consumers. In the near future, RFID system will become ubiquitous when the cost of tags reduces to an acceptable level.

In RFID system, tag identification is performed by the reader's query to a tag and the tag's backscattering its identifier as its response. But if there are multiple tags within the identification range of a reader, some of them may send data simultaneously which may lead to mutual interference. This causes data loss and is referred to as a collision which increases identification time. A technical scheme that handles multiple-access without any interference is called as an anti-collision algorithm [3][4]. The primary concern in anti-collision algorithm is how to read multiple tags as fast and reliably as possible.

In RFID systems, there are two main types of anti-collision algorithms: deterministic and probabilistic algorithms [5][6]. The first ones are ALOHA-based protocols; the last ones are tree-based protocols. In the deterministic algorithm, a reader queries all the tags within the reader's identification range for the next bit of their identifier [7]. On detecting collision, the reader mutes subsets of tags that are involved in a collision and queries again until there is only one tag response. The main advantage of deterministic algorithm is its effectiveness, which means all tags can be successively identified even when the number of tags is large. However, it requires a lot of query cycles to identify all tags. Also, the tags may need complex circuit and memory.

The probabilistic algorithm is based on an ALOHA-like protocol that provides slots for the tags to send their data. Almost all the probabilistic algorithms use framed slot ALOHA (FSA) [8][9]. A frame is a time interval between queries of a reader and consists of a number of slots. Each tag transmits its identifier to the reader in a slot of a frame, and the reader identifies the tag when a time slot is used by only one tag. EPCglobal Class-1 Generation-2 and ISO/IEC 18000-6 Type C standards use the probabilistic approach. The probabilistic algorithm can accomplish fast identification, but there is still a probability of failing to identify all tags in a limited time period. When the number of tags is not very large, the probabilistic algorithm is more efficient than the deterministic algorithm.

A lot of researches have been performed to enhance the performance of FSA algorithm. Among those algorithms, DFSA (Dynamic Framed Slot ALOHA) dynamically allocates the frame length based on the number of tags in the reader's

---

Manuscript received August 11, 2009 ; revised August 24, 2009.

Intaek Lim is with the Department of Embedded IT, Pusan University of Foreign Studies, Busan, 608-738, Korea (Tel: +82-51-640-3423, Fax: +82-51-640-3038, Email: itlim@pufs.ac.kr)

identification range. There are two main research areas in DFSA algorithm: i) tag number estimation scheme, and ii) dynamic frame size allocation scheme. Almost all the researches in DFSA algorithm have been carried out to estimate exactly the number of tags. But, due to the estimation errors, there are some problems that the performance degrades in spite of optimal frame size allocation.

EPCglobal Class-1 Generation-2 standard proposed Q-algorithm to determine the frame size for the next query round [8]. Q-algorithm calculates the frame size without conducting a tag number estimation. Therefore, it wastes less computational cost and is simpler than other DFSA algorithms. But the constant parameter C value, which is used for calculating the next frame size, is not optimized. Therefore, in this paper, we propose optimized parameter C values to enhance the performance of Q-algorithm.

This paper is organized as follows. In Section II, we describe EPCglobal Class-1 Generation-2 anti-collision algorithm, and Section III describes Q-algorithm. In Section IV, we analyze the performance of Q-algorithm through various computer simulations and propose the optimized parameter C values. Section V concludes the paper.

II. EPCglobal Class-1 GEN-2 ALGORITHM

EPCglobal Inc. proposed Class-1 Generation-2 RFID protocol operating at the 860MHz – 960MHz UHF band. EPCglobal Class-1 Generation-2 uses a frame-based slot ALOHA as an anti-collision algorithm for identifying tags within the reader's identification range.

The reader begins a query round by transmitting a Query command. After issuing a Query command to initiate a query round, the reader transmits one or more QueryRep commands to detect each slot during a query round. Fig.1 shows the simplified tag state transition diagram of Generation-2 anti-collision algorithm. Upon entering an energizing RF field, a tag that is not killed enters Ready state and remains in Ready state until it receives a Query command. When tags receive a Query command, they draw a Q-bit number from their random number generator, load this number into their slot counter, and transition to the Arbitrate state if the number is nonzero, or to the Reply state if the number is zero.

A tag in Arbitrate state decrements its slot counter every time it receives a QueryRep command, and it transitions to the Reply state and backscatters an RN16 when its slot counter reaches zero. If it receives a QueryAdjust command, it adjusts its Q value according to UpDn field of QueryAdjust command, select a new random number, and load this number into its slot counter. A tag that reinitialized its slot counter remains in the Arbitrate state if the slot counter is nonzero, or transitions to the Reply state.

Upon entering Reply state, a tag backscatters an RN16. If the tag receives a valid acknowledgement, it transmits its PC, EPC, and CRC-16, and enters the Acknowledged state. If the tag fails to receive an ACK command within the specified time, or receives an invalid ACK or an erroneous RN16, it returns to the Arbitrate state. After acknowledging a tag, a reader can choose to access it.

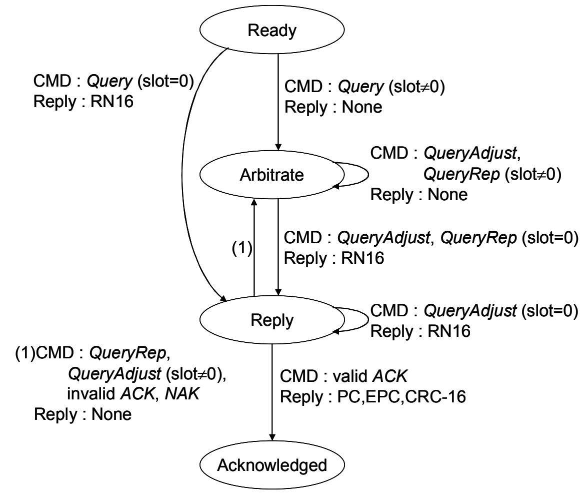

Fig. 1. Simplified tag state diagram.

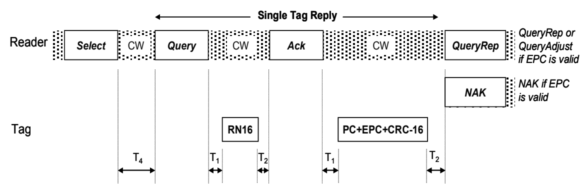

Fig. 2. Single tag reply.

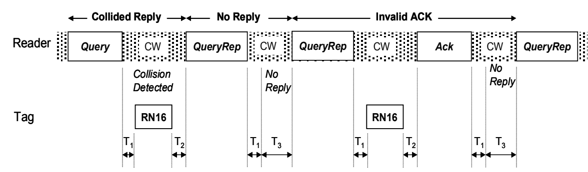

Fig. 3. No tags reply or multiple tags reply.

If there is only one tag reply for a Query command, it is a successful query round. However, if there is no tags reply or multiple tags reply, we consider it as a failure. Fig.2 and Fig.3 illustrate the timing diagrams for the case of single tag reply and collision or no tags reply, respectively. As shown in the figures, if only one tag transmits its RN16 for a Query command, the reader successfully receives without collisions, and then transmits an ACK command. If the tag receives the ACK command with a correct RN16, it backscatters it PC, EPC, and CRC-16.

Fig.4 shows the operation for anti-collision algorithm that is proposed by EPCglobal Class-1 Generation-2. At first, the reader selects a tag population for a query process by transmitting a Select command. And it transmits a Query command to decide which tags participate in the query round. Query command contains a slot-count parameter Q, and the initial value of Q is 4.

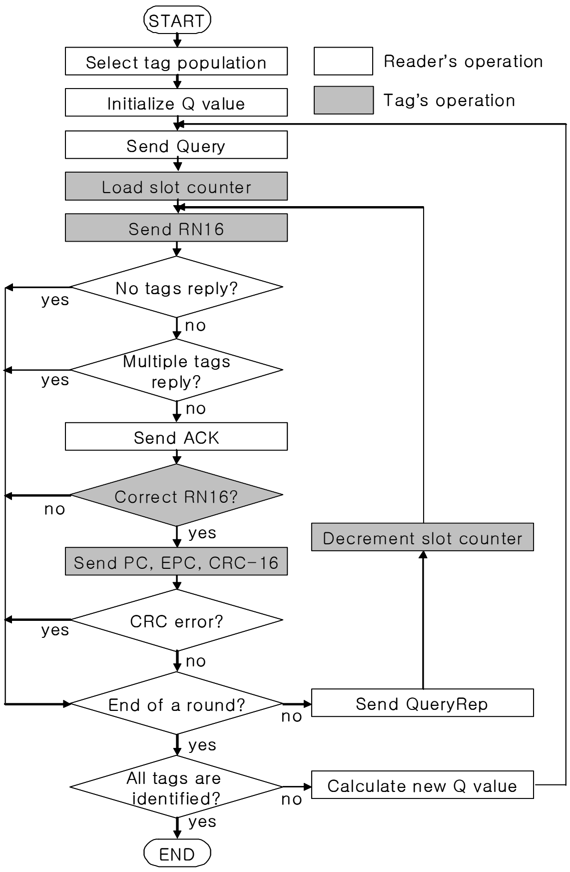

Fig. 4. Flowchart for Gen-2 anti-collision algorithm.

Upon receiving a Query command, participating tags pick a random value in the range (0, 2^Q-1), inclusive, and load this value into their slot counter. Tags that pick a zero backscatter an RN16, which is a 16-bit random number. If the reader receivers an RN16 without collision, it acknowledges the tag with an ACK command containing this same RN16. If the tag receives the ACK command with a correct RN16, it transmits its PC, EPC, and CRC-16. The reader checks CRC errors with the received CRC-16 value. If no CRC errors are found, the reader assumes that this tag is successfully identified. After that, it reads from or writes to the identified tag.

In the EPCglobal Class-1 Generation-2 algorithm, the number of slots in the frame of query round is 2^Q slots. If a query round is not expired, the reader continues an identification process for the next slot by transmitting a QueryRep command. On the other hand, if unidentified tags still remain though a query round is terminated, the reader issues a new Query command to initiate another query round. The new Query command also contains a slot-count parameter Q, which is calculated through Q-algorithm described later.

III. Q-ALGORITHM

EPCglobal Class-1 Generation-2 proposed Q-algorithm to determine the number of slots in the next query round. Q-algorithm basically calculates the slot-count parameter Q based on the slot status that tags are responded. The slot status is classified into three categories: success, collision, and empty slot.

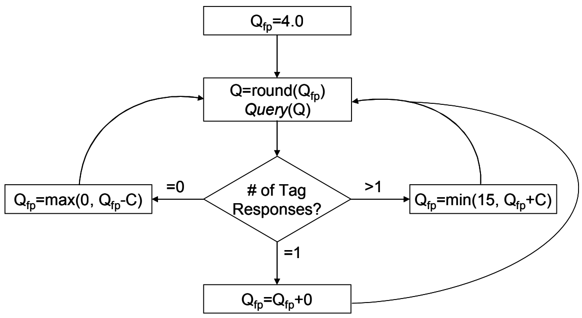

Fig. 5. Q-algorithm.

Fig.5 shows an algorithm that the reader might use for setting the slot-count parameter Q in a query round. In the figure, Q_{fp} is a floating-point representation of Q. As shown in the figure, the reader updates Q_{fp} in accordance with the slot status at every slot. When a collision occurs, it adds the constant C value to the previous Q_{fp}, because it means the frame length is smaller than the number of tags. If the slot is empty, which means that there are no tag responses in the slot, the reader subtracts the constant C value from the previous Q_{fp}, because the frame length is larger than the ideal one. When a new query round begins, the reader rounds Q_{fp} to an integer value Q in the Query command. Typical values for the parameter C are $0.1<C<0.5$. EPCglobal Class-1 Generation-2 standard suggests that the reader typically uses small values of C when Q is large and large values of C when Q is small. However, the performance of Gen-2 anti-collision algorithm, which uses the frame-based slot ALOHA, is dependent on the number of tags in the reader's identification range and frame length. Therefore, the reader must choose a constant parameter *C* value according to the number of tags in a query round.

IV. SIMULATION RESULTS AND PARAMETER OPTIMIZATION

In this paper, computer simulations are performed to evaluate the performance of Gen-2 anti-collision algorithm and derive the optimum constant parameter *C* values of Q-algorithm. In Q-algorithm of Gen-2 anti-collision algorithm, the slot-count parameter for the next query round is incremented or decremented by the constant parameter *C* value according to the slot status. Therefore, it is anticipated that the performance of Gen-2 anti-collision algorithm will mainly depend on the constant parameter *C* value and the number of tags within the identification range of reader. The system parameters for simulation are shown in Table 1, which are based on the EPCglobal Class-1 Gen-2 specification.

Fig.6 shows the total identification time of Gen-2 algorithm according to the constant parameter *C* value and number of tags. The total identification time means the number of all consumed slots for identifying all tags. As depicted in the figure, the identification times of Gen-2 algorithm are various with the constant parameter *C* and number of tags. If *C* is 0.5, the reader can identify faster than other cases when the number of tags increase. In the case of constant parameter *C* value with 0.2, 0.3, and 0.4, the identification time increases sharply when the number of tags is 200, 300, and 350, respectively. Moreover, when the number of tags is over 400, the identification time is similar to each other with the *C* value between 0.2 and 0.4. According to the results of Fig.6, it is necessary to analyze more precisely the identification time when the number of tags are below 200, between 200 and 400, and above 400.

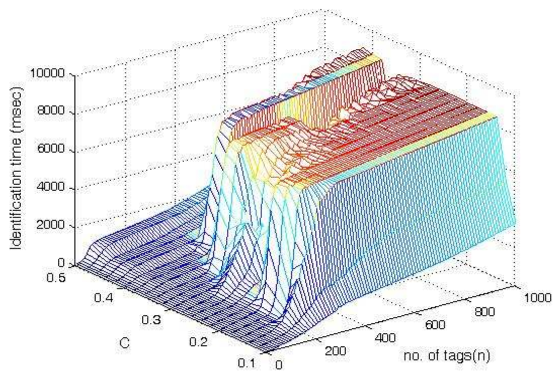

Fig. 6. Identification time according to *C* value.

Table 1. Simulation parameters.

Parameter	Description	Value
T_{ari}	Reference time interval for a data-0 in Interrogator-to-Tag signaling	12.5μs
0_{length}	Time interval for a data-0 in Interrogator-to-Tag signaling	12.5μs
1_{length}	Time interval for a data-1 in Interrogator-to-Tag signaling	18.75μs
RT_{cal}	Interrogator-to-Tag calibration symbol	31.25μs
TR_{cal}	Tag-to-Interrogator calibration symbol	64μs
DR	Divide ratio	8
LF	Link frequency	125KHz
T_{pri}	Link pulse repletion interval	8μs
RT_{rate}	Interrogator-to-Tag data rate	64Kbps
M	Number of subcarrier cycles per symbol	1
TR_{rate}	Tag-to-Interrogator data rate	125Kbps
T_1	Time from interrogator transmission to tag response	80μs
T_2	Time from tag response to interrogator transmission	80μs
T_3	Time an interrogator waits, after T1, before it issues another command	0μs
T_4	Minimum time between interrogator commands	62.5μs
R=>T preamble	Preamble for R=>T signaling. Used for a Query command	120.25μs
R=>T frame sync	Frame synchronization for R=>T signaling. Used for all commands except Query	56.25μs
T=>R preamble	Preamble for T=>R	48μs

The identification time when the number of tags is below 200 is shown at Fig.7. As depicted in the figure, the identification time is nearly same when *C* is 0.1 and 0.3. When *C* is 0.5, the identification time sharply increases at $n \geq 40$. When the number of tags are above 130, the identification time with *C*=0.3 is faster than one with *C*=0.1.

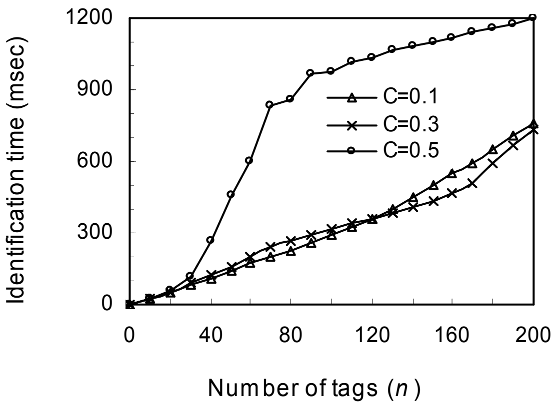

Fig.7. Identification time when $n \leq 200$.

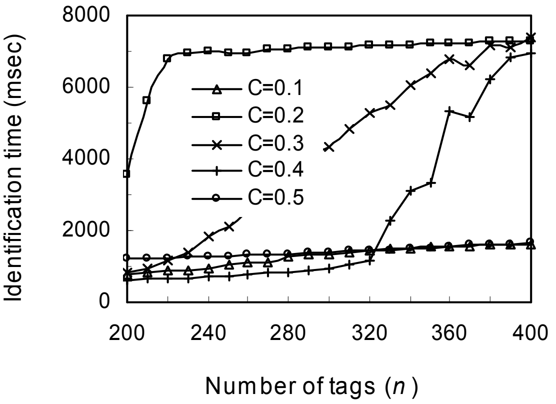

Fig.8. Identification time when $200 \leq n \leq 400$.

Fig.8 shows the identification time when the number of tags is between 200 and 400. If we let C be 0.2 and 0.4, the identification time sharply increases when $n \geq 220$ and $n \geq 340$, respectively. When C is 0.3, the identification time increases linearly at $n \geq 200$. On the other hand, when C is 0.1 and 0.5, the identification time is nearly same regardless of the number tags.

Fig.9 depicts the identification time according to the number of tags. As shown in the figure, when the number of tags is small (in the case of n=100), the identification time is rarely influenced by the constant parameter C values. But, in this case, the identification time is as fast as C is small. On the other hand, when the number of tags is large, the reader can identify fast if we use very large or very small C values. In the case of n=500, the identification time increases sharply at C=0.14 and decreases sharply at C=0.42. Also, when the number of tags is 1000, the identification time increases sharply at C=0.14 along with n=500, but it decreases sharply at C=0.44.

Throughout the above simulation results, the performance of EPCglobal Class-1 Generation-2 anti-collision algorithm is influenced by the constant parameter C values of Q-algorithm and number of tags within the identification range of reader. Fig.10 shows the optimal identification time according to the number of tags. And Table 2 shows the optimal C values to obtain the optimal identification time with regard to the number of tags. For maintaining the optimal performance of EPCglobal Class-1 Generation-2 anti-collision algorithm, it is necessary to choose the optimized constant parameter C values as shown in Table 2.

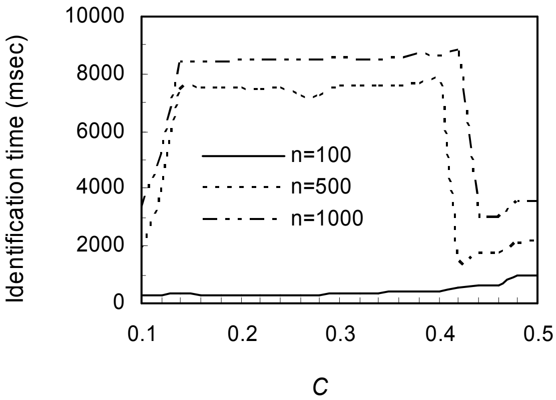

Fig.9. Identification time vs. the number of tags.

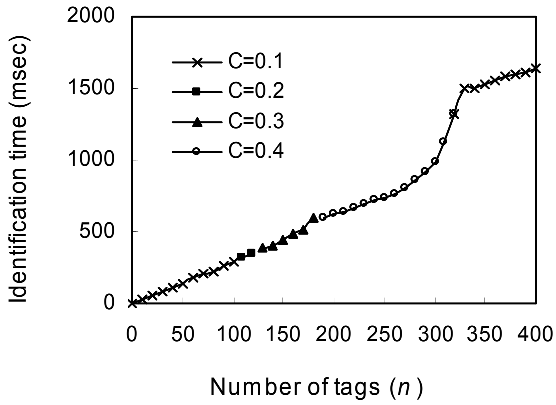

Fig.10. Optimal identification time.

Table 2. Optimal C values vs. the number of tags.

Number of tags (n)	C
$n \leq 107$	0.1
$108 \leq n \leq 126$	0.2
$127 \leq n \leq 183$	0.3
$184 \leq n \leq 321$	0.4
$322 \leq n \leq \infty$	0.1

V. CONCLUSIONS

EPCglobal Class-1 Generation-2 standard proposed Q-algorithm to determine the frame size for the next query round. In Q-algorithm, the reader calculates the frame size without conducting a tag number estimation. In calculating the slot-count parameter Q in a query round, the reader updates Q_{fp}, which is a floating-point representation of Q, by a constant parameter C ($0.1<C<0.5$) depending on the slot status. The constant parameter C value is not defined in the standard but opened in implementing the algorithm. In this paper, we proposed the optimized C values of Q-algorithm according to the number of tags within the identification range of reader through a lot of computer simulations.

REFERENCES

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, Carl Hanser Verlag GmbH & Co., 2002.

[2] W. Chen, and G. Lin, "An Efficient Anti-Collision Method for tag Identification in a RFID System," *IEICE Trans. Commun*., vol.E89-B, no.12, pp.3386-3392, Dec. 2006.

[3] H. Vogt, "Efficient Object Identification with Passive RFID Tags," *First International Conf. on Pervasive Computing*, LNCS, vol.2414, pp.99-113, Springer-Verlag, 2002.

[4] M. A. Bonucelli, F. Lonetti, and F. Martelli, "Instant Collision Resolution for Tag Identification in RFID networks," *Ad Hoc Networks*, vol.5, pp.1120-1232, 2007.

[5] Z. Wang, et al., "Anti-collision Scheme Analysis of RFID System," *Auto-ID Labs White Paper*, WP-HARDWARE-045, Oct. 2007.

[6] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for Multiple RFID Objects Identification," *IEICE Trans. Commun*., vol.E88-B, no.3, pp.991-999, Mar. 2005.

[7] C. Law, L. Lee, and K. Y. Siu, "Efficient Memoryless Protocol for Tag Identification," Auto-ID Center, *MIT-AUTOID-TR-003*, Oct. 2000.

[8] EPCglobal, "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocols for Communication at 860MHz-960MHz, Ver.1.2.0," *EPCglobal Inc*., Oct. 2008.

[9] ISO/IEC, "Information Technology – Radio Frequency Identification for Item Management – Part 6: Parameters for Air Interface Communication at 860-960 MHz, 19000-6," *ISO/IEC*, 2006.

Intaek Lim 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 IT. From July 2006 to June 2007, He was a visiting scholar at Cleveland State University in USA. His research interests include the MAC protocol design, ad-hoc network, RFID, and mobile computing.