Query Tree-Based Reservation for Efficient RFID Tag Anti-Collision

Ji Hwan Choi, Student Member, IEEE, Dongwook Lee, and Hyuckjae Lee, Member, IEEE

Abstract—Since the tree based RFID tag anti-collision protocol achieves 100% read rate, and the slotted ALOHA based tag anti-collision protocol allows simple implementation and good performance in small amount of tags, we consider how to take the advantages of each algorithm. This paper presents query tree based reservation for efficient RFID tag anti-collision. According to the simulation results, the proposed protocol requires less time consumption for tag identification than the present tag anti-collision protocols implemented in EPC Class 0, Class 1, and Class 1 Gen. 2.

Index Terms—Anti-collision protocols, distributed reservation-based protocols, tag collisions, RFID.

I. INTRODUCTION

RFID systems are coming the most promising technologies used for contactless object identification. One of the research areas in RFID systems is a tag anti-collision protocol; how to reduce identification time with a given number of tags in the field of an RFID reader [1]. For fast tag identification, anti-collision protocols, which reduce the frequency of collisions occurred and identify tags independent of occurring collisions, are required. Moreover, tag anti-collision protocols are more important than reader anti-collision protocols because of the incompetence of tags in low-cost passive RFID systems.

There are two types of tag anti-collision protocols for RFID systems: deterministic methods and probabilistic methods [1]. The deterministic methods are tree based protocols [1,2,3,6]. The tree based protocols keep splitting the group of colliding tags into two subgroups until all tags are identified. The probabilistic methods are based on slotted ALOHA [4]. The slotted ALOHA based protocols reduce the probability of occurring tag collisions how tags respond at the different time.

This paper considers how to improve the performance of the RFID tag anti-collision protocol by applying the characteristics of EPC Class 1 Gen. 2 protocol to the query tree algorithm. The crux of the proposed query tree based reservation is that the RFID systems do not have to use the whole length of tag IDs for resolving each tag in the field of a reader. Thus, the tags generate reservation sequences, which are 16-bit random numbers (RN16s), for assigning slots to transmit their IDs, and the reader identifies tags with the query tree algorithm using the RN16s. During the identification process, the reader calls a tag with the received RN16 for collecting the tag ID when a RN16 is identified by the reader.

II. QUERY TREE-BASED RESERVATION (RN16QTA)

Since the slotted ALOHA based algorithms cannot guarantee the 100% read rate, and experience significant performance degradation in the large amount of tags, this paper considers the performance enhancement of the tree based algorithm. For enhancing the performance of the tree based algorithm, we apply the characteristics of EPC Class 1 Gen. 2 protocol to the query tree algorithm as follows:

1) RN16 Generation: All tags in the field of a reader generate temporary IDs, 16-bit random numbers (RN16s), for giving the uniqueness to themselves [4]. This part is performed exactly like in standard EPC Class 1 Gen. 2.
   - Probability of a single RN16: $0.8/2^{16} < P(RN16 = j) < 1.25/2^{16}$, where $j$ is any possible number generated by the random number generator (RNG).
   - Probability of simultaneously identical sequences: For a tag population up to 1,000 tags, the probability of existing tags having the same RN16 is less than 0.1%.
   - Probability of predicting a RN16: A RN16 shall not be predictable with a probability greater than 0.025%.

2) Applying RN16s to the Query Tree Algorithm (QTA [3])
   - Request: The reader sends $n$-length inquiring bits (prefix) to tags.
   - Response: Tags send their RN16s from $n + 1_0$ bit to the end when the first $n$ bits of the tag IDs are the same as the prefix.
   - Decision: Depending on whether collisions have occurred or not, the reader decides on proceeding procedure with the following conditions.
     - If collisions occur, the reader saves two new prefixes at the last input first output (LIFO).
     * Two new prefixes: the connection of the prefix and either ‘0’ or ‘1’
     - If a collision occurs when the tags respond with only the last bits in the RN16s to the reader’s inquiring, the reader assumes that there are two tags because of the uniqueness of RN16s.

Manuscript received September 12, 2006. The associate editor coordinating the review of this letter and approving it for publication was Dr. Fabrizio Granelli. This work is supported by the RFID/USN standardization project of ETRI under the information communication standardization program of MIC (Ministry of Information and Communication), Korea.

J. H. Choi, D. Lee, and H. Lee are with the School of Engineering, Information and Communications University (ICU), Daejeon, Korea (e-mail: {jhwanchoi, dwook, hjlee}@icu.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2007.061471.
The performance of the RFID tag anti-collision protocols means how fast the identification process in the field of a reader is. In the tree based algorithms, the performance depends on the tree depth, which is equal to the length of the temporary IDs shortening the tree depth. After identifying a temporary ID, the reader calls a tag with the temporary ID for collecting the tag ID. With RN16QTA, consequently, we can reduce the identification time by the shortened tree depth.

III. SIMULATIONS AND RESULTS

This simulation is intended for the performance comparison between the present tree based RFID tag anti-collision protocols and RN16QTA in the algorithm point, and the performance comparison between RN16QTA and DFS-ALOHA in EPC Class 1 Gen. 2 protocol in the implementation point.

One important point of the observation is whether the simulated protocol is implementable to the low-cost passive RFID systems or not. QTA is already applied to the RFID systems with EPC Class 1 protocol. On the other hand, collision tracking tree algorithm (CTTA [3]) is not adopted to any RFID systems yet because tags cannot support CTTA with transmitting and receiving simultaneously. Since RN16QTA is the protocol applying the characteristics of DFS-ALOHA in EPC Class 1 Gen. 2 protocol to the query tree algorithm, the algorithm is easily implementable for the low-cost passive RFID systems.

A. Simulation environment

The simulation condition is as follows. There is only one reader. In the field of the reader, the number of tags increases from 2 to 512. The length of the tag IDs is 96 bits. Both tag-to-reader data rate and reader-to-tag data rate are chosen as 80 kbps. The reason is that the middle speed in EPC Class 1 Gen. 2 proposed by EPCglobal to ISO/IEC 18000-6 C is equal to the chosen data rate [4,5]. There is some iteration overhead because of propagation delay from the channel and latency from the signal processing. For the comparison of the algorithm point, the iteration overhead is not considered.

For the comparison of the implementation point, we set a simple EPC Class 1 Gen. 2 architecture without considering both the multi-session commands for the multi-reader environment and the probability of receiving an invalid tag ID. To calculate the average number of identified tags per second, the RFID systems choose 8 bits for request and 3 bits for detecting collision, no collision, or no response for substituting the iteration overhead. For solving error detecting problem, moreover, 16-bit CRC coding is required.

B. Results

1) The algorithm point: Above all else, Fig.2 shows the average inquiring bits and response bits for one-tag identification. In the RN16QTA, both the average inquiring bits and response bits for one-tag identification are between the case of QTA and the case of CTTA. The reason is that RN16QTA reduces the overhead from both the average inquiring bits and response bits by using the temporary IDs, but generates the overhead caused by collecting the tag IDs. The former one affects RN16QTA to get better performance than QTA, and the
The number of tags

Fig. 2. The average inquiring and response bits for one-tag identification.

The average required iterations for one-tag identification.

Fig. 3. The average required iterations for one-tag identification.

The number of tags

Fig. 4. The average number of identified tags per second.

Fig. 5. The average number of identified tags per second.

latter one influences RN16QTA to get worse performance than CTTA. Next, Fig.3 indicates the average required iterations for one-tag identification. RN16QTA requires one more iteration than QTA because of the step collecting the real tag IDs, but the overhead from each iteration is 1/6 of the others. Finally, Fig.4 shows the average number of identified tags per second in each algorithm. In conformity with the simulations, RN16QTA achieves faster identification than QTA, and approaches to the performance of CTTA.

2) The implementation point: Fig.5 indicates the average number of identified tags per second in each protocol. Depending on the default frame size (16 slots or 64 slots) in EPC Class 1 Gen. 2 protocol, the tendency of the performance is different in the small amount of tags. However, the performance of DFS-ALOHA in EPC Class 1 Gen. 2 protocol is always less than that of the proposed RN16QTA as shown in Fig.5. Consequently, the proposed protocol, RN16QTA, accomplishes faster tag identification than the implemented protocols, QTA in EPC Class 1 protocol and DFS-ALOHA in EPC Class 1 Gen. 2 protocol.

IV. CONCLUSION

This paper proposes the query tree based reservation, which reduces the tree depth on the identification process for decreasing the identification time. The query tree based reservation applies 16-bit random numbers to the query tree algorithm, which is for assigning slots to transmit tag IDs, instead of the real tag IDs. According to the simulation results, the query tree based reservation achieves less time consumption for the tag identification than the implemented tag anti-collision protocols in EPC Class 0, Class 1, and Class 1 Gen. 2.

REFERENCES