Prefix-Randomized Query-Tree Protocol for RFID Systems*

Kong Wa Chiang, Cunqing Hua and Tak-Shing Peter Yum

Department of Information Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong
E-mails: {kwchian4, chua0, yum}@ie.cuhk.edu.hk

Abstract— **In this paper we present a new tree search-based protocol for the anti-collision problem of RFID systems. This protocol builds a binary search tree according to the prefixes chosen randomly by tags rather than using their ID-based prefixes. Therefore, the tag identification time of the proposed protocol is no longer limited by the tag ID distribution and ID length as the conventional tree search protocol. The time complexity of the protocol is derived and shown that it can identify tags faster than the Query-Tree protocol.**

I. INTRODUCTION

The Radio Frequency Identification (RFID) system consists of a number of tags with unique IDs, a reader to obtain information from the tags and a data processing subsystem. A vast majority of current RFID systems work in a Reader-Talk-First mode, where a reader issues query commands first, and those tags that are within the reading range of the reader will respond with the stored information with internal energy (for an Active Tag) or external energy powered by the reader (for a Passive Tag). Since all tags have to share the common broadcast channel to communicate with the reader, this will lead to collision as multiple tags transmit simultaneously.

The anti-collision problem of RFID is similar to the classical multi-access communication systems with solutions such as Tree protocol, Aloha (slotted-Aloha, framed-Aloha) and Carrier Sense Multiple Access (CSMA). However, anti-collision protocols of RFID systems are constrained by low computational capability and small memory. In addition, they must be optimized for low power operation to increase the communication range in the case of passive tags. The limited power supply, memory and computing capability of low-cost RFID tags rule out the use of complicated anti-collision algorithms. In addition, low-cost tags are not able to sense the medium, so the use of CSMA is also not possible. The Query-Tree protocol is simple, but it has scalability problem because its worst-case time complexity is on the order of $n(k+2-\log_2 n)$ [2], where n is the number of tags and k is the length of ID string.

The majority of RFID anti-collision protocols are time-domain based of either deterministic type or stochastic type. For deterministic schemes, the reader broadcasts a command requesting certain tags, based on their IDs, to respond. It then either polls a list of tags' IDs or performs some variations of binary search algorithm. Typical polling schemes can be time exhaustive if there are many tags under the reader's interrogation area. Moreover, the length and distribution of tag IDs can affect the identification time significantly. In [1], a reader queries all tags for the next bit of their IDs. When collision occurs, the reader splits the queries until there is only one tag respond. An efficient memoryless scheme called Query-Tree protocol is proposed in [2]. In this protocol, a reader sends out a prefix in each communication round and tags simply respond with their IDs if the prefix is match with their IDs. If there is a collision for a particular prefix, the reader ignores the response and polls a one-bit-longer prefix later. The polling efficiency for this protocol is low when the tag population size is large or the ID address distribution is sparse.

For stochastic schemes, tags respond to reader's interrogation at randomly chosen time slots. There are a number of variations of these schemes based on the Aloha protocol family for many commercial RFID systems [3]-[4]. Among this protocol family, framed-Aloha is a favorite choice from both theoretical and practical consideration. This protocol is an extension of slotted-Aloha by grouping several slots into one frame. Tags are required to send their IDs in a randomly chosen time slot of each frame. The frame size is determined by the reader. In [5], the author built a Markov model for the

* The work was supported in part by the Hong Kong Research Grants Council under Grant CUHK 4220/03E.

1-4244-0355-3/06/$20.00 (c) 2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

dynamic framed-Aloha protocol for passive tags identification. Optimal parameters such as frame size and number of communication rounds can be derived based on the estimation of tag set size. In [6], the identification time of passive tags was derived for framed-Aloha under a given missing tag probability. The wireless channel models and capture effect are also taken into consideration in this paper. The performance of stochastic scheme is not limited by the length and distribution of tag IDs, however, it cannot guarantee that all tags are identified within a certain time interval if the number of tags is not known in advance.

In this paper, we introduce a new protocol called Prefix-Randomized Query-Tree (PRQT) protocol for tag identification. PRQT protocol differs from the Query-Tree (QT) protocol [2] in that it uses prefixes chosen randomly by tags (rather than using their ID-based prefixes). Therefore, the identification time of PRQT is no longer affected by the length and distribution of tag IDs as does for QT. We compare PRQT and QT by analysis and computer simulation.

II. TAG IDENTIFICATION

In this section, we introduce the Prefix-Randomized Query-Tree (PRQT) protocol. PRQT is designed under the assumptions: (i) the tag set size is fixed during the identification process; (ii) tags can randomly generate a prefix with a prescribed length, and (iii) tags cannot communicate with each other and they may choose the same prefix.

A. Prefix-Randomized Query-Tree Protocol

The PRQT algorithm consists of rounds of "queries from the reader" and "responses from tags". In the initialization round, the reader broadcasts a command with an initial prefix length l ($l \le L$, L is the length of the ID string) which is determined from the tag set size n. After receiving this initial command, each tag generates an l bits random binary prefix. The reader then polls each of these 2^l prefixes sequentially. In each round, tags with prefix matches respond with their IDs. Since tags cannot coordinate the prefix choices, multiple tags may choose the same prefix. Therefore tags with the same prefix will respond to the reader at the same time and cause a collision. Suppose a collision is detected when polling the initial prefix f_i. The reader will broadcast a command asking this group of collided tags matching this prefix to expand their prefixes by one bit randomly drawn from '0' or '1' and polls the extended prefixes f_i0 and f_i1. If collisions occur in these polls, the same procedure is repeated. In essence, PRQT grows a binary query tree from the collided prefix until all tags choosing this prefix are identified. After that, the algorithm returns and continues polling the rest of 2^l initial prefixes. The same procedure is repeated until all initial prefixes have been polled.

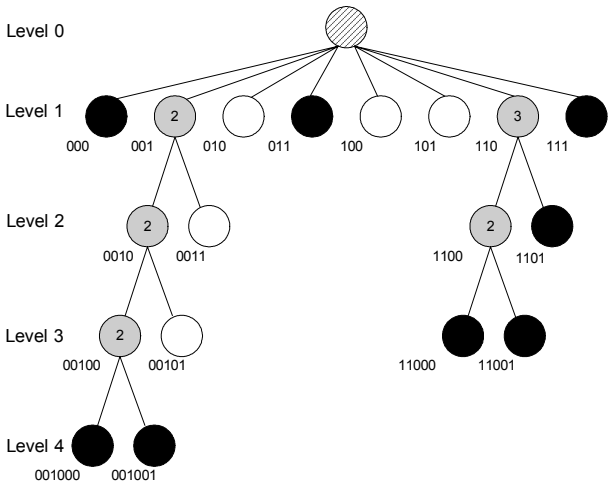

Fig.1 PRQT example for identifying eight tags.

Step	Query	Response
1	$L = 3$	No response
2	000	Success
3	001	Collision
4	0010	Collision
5	0011	No response
6	00100	Collision
7	00101	No response
8	001000	Success
9	001001	Success
10	010	No response
…	…	…
19	111	Success

Fig. 2. Communication between reader and tags.

In Fig. 1 we show a query tree of PRQT for identifying eight tags. The shaded root node indicates the broadcasting of the initial prefix length by the reader. A dark node indicates a prefix polled by the reader and is chosen by a single tag (which corresponds to a success response). A gray node indicates a prefix polled by the reader and is chosen by multiple tags (the number inside the circle indicates the number of collided tags). A white node indicates a prefix polled by the reader and is

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

not chosen by any tag (no response). In this example, there are eight level-1 nodes corresponding to the eight initial prefixes. The initial prefix '001' is chosen by two tags and '110' is chosen by three tags. So a binary tree is grown from each of these two initial prefixes. The left and right child nodes of a collided parent node for prefix f_i represent prefixes f_i0 and f_i1 respectively. The communication between the reader and tags for this example are shown in Fig. 2 where PRQT uses a total of 19 steps to identify all eight tags.

B. Time Complexity Analysis

In the PRQT protocol, tags may choose the same prefix round after round. But this probability drops geometrically as the prefix length gets longer. The classical analysis methods for tree search algorithm [7]-[9] cannot be used here as tags are not guaranteed to generate unique prefixes.

To find the average tree size given the number of tags and the initial prefix length, let us assume the tag set size is n, the initial prefix length is l and there are $N = 2^l$ nodes in the first level of the query tree. In this level, some nodes (or initial prefixes) may be chosen by multiple tags as shown in Fig. 1, which leads to collision when the reader polls these initial prefixes. Let p_k denote the probability of k tags choosing the same initial prefix in a level-1 node. Then p_k follows a binominal distribution

$$p_k = \binom{n}{k}\left(\frac{1}{N}\right)^k\left(1-\frac{1}{N}\right)^{n-k} \qquad (1)$$

Every collision from level-1 node onward causes a split of the query tree. Since each node (excluding the root node) on the query tree represents a round of polling operation by the reader, the expected number of polling rounds needed to completely identify all tags, W, is numerically equal to the size of the query tree excluding the root node. Assuming the amount of time for each polling-response round is fixed, then W is also identical to the expected tag *identification time*.

Let t_k denote the average size of a sub-tree with k tags in its root node, then W equals the summation of all sub-tree sizes

$$W = N\sum_{k=0}^{n} p_k t_k \qquad (2)$$

Obviously $t_0 = t_1 = 1$ because there are no collision in these cases. To calculate t_k for k from 2 to n, we proceed as follows. We condition on the first split (level-2) of the collided level-1 node. Suppose i out of k ($i \in [0,k]$) tags choose to expand the prefix by "*0*" and the rest k-i tags choose to expand the prefix by "*1*", which leads to two sub-trees each with i and k-i nodes respectively. Therefore, $t_k = 1 + t_i + t_{k-i}$ and this probability is given by

$$p_k(i,k-i) = \binom{k}{i} \Big/ 2^k, \; i = 0,1,\cdots k \qquad (3)$$

Summing over all $i \in [0,k]$ we obtain

$$t_k = 1 + \sum_{i=0}^{k} p_k(i,k-i)(t_i + t_{k-i}) \qquad (4)$$

Substituting (3) into (4) and rearranging the term t_k to the left-hand side, we can obtain the result as follows.

$$t_k = \frac{1}{2^{k-1}-1}\left[2^{k-1} + \sum_{i=0}^{k-1}\binom{k}{i} t_i\right] \qquad (5)$$

where $t_0 = t_1 = 1$ and $k = 2, 3, \ldots, n$.

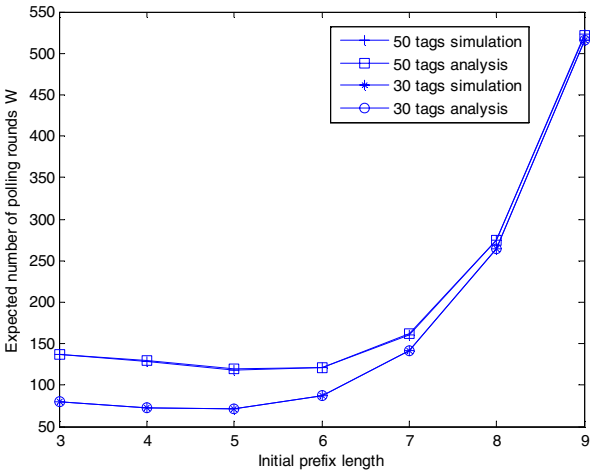

Fig. 3 Analytical and simulation results of PRQT protocol.

Fig. 3 shows the close matching of the analytical and simulation results of W under different settings of the initial prefix length. Each simulation point represents the average value of 1000 trials.

C. Optimal Initial Prefix Length

In this section, we derive the optimal initial prefix length for a given tag set size. From (1), (2) and (5) we can see that given n, the expected number of polling rounds W is a function of N only. We therefore can minimize W with respect to N. As stated before, the initial prefix length is $l = \log_2 N$.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

However, W is not differentiable because N is an integer. To cope with this problem, we relax N to be a real number and introduce a variable $q = 1/N$. Substituting q into (1) and (2) we have

$$W(q) = \frac{1}{q}\sum_{k=0}^{n}\binom{n}{k}q^k(1-q)^{n-k}t_k \qquad (6)$$

Differentiating $W(q)$ with respect to q we have

$$\frac{\partial W}{\partial q} = \sum_{k=0}^{n}\binom{n}{k}(k-1+q-nq)q^{k-2}(1-q)^{n-k-1}t_k \quad (7)$$

Let $\partial W / \partial q = 0$, q^* that minimizes W can be found numerically. Since q^* is a real number, the optimal initial prefix length will be either $l^+ = \lceil \log_2(1/q^*) \rceil$ or $l^- = \lfloor \log_2(1/q^*) \rfloor$. Putting them together, the optimal prefix length is given by

$$l^* = \arg\min\{W(l^+), W(l^-)\} \qquad (8)$$

Table I shows the values of l^* as a function of n, for $30 \le n \le 500$.

TABLE I OPTIMAL INITIAL PREFIX LENGTH FOR DIFFERENT TAG SET SIZE

Tag set size n	30 - 53	54 - 107	108 - 215	216 - 429	430 - 500
Optimal initial prefix length l^*	5	6	7	8	9

III. PERFORMANCE EVALUATION

We now compare the performance of PRQT with the Query-Tree protocol. QT is a deterministic collision resolution scheme with an average time complexity of ($2.881n - 1$, $2.887n - 1$) for uniformly distributed set with n tags. However, its worst-case time complexity is on the order of $n(k + 2 - \log_2 n)$ [2].

Firstly, we investigate the effect of tag ID distribution on the tag identification time of QT by varying the probability of '0's occurring in the ID string from 0.05 to 0.95. The ID length is set to be 64 bits [10]. Each data point shown in Fig. 4 is the average value of 1000 trials. We can see that the tag identification time of the QT protocol is heavily dependent on the uniformity of the tag distribution, particularly when the tag set size is large. The performance of PRQT, however, is totally independent of ID distribution and ID length.

In Fig. 5, we compare the expected identification time for three different schemes:

1. PRQT with optimal initial prefix length (analytical result).
2. QT with uniform tag ID distribution (lower bound of its average time complexity $2.881n$-1).
3. QT with non-uniform tag ID distribution where Prob('0') = 0.3.

It can be seen from Fig.5 that PRQT has better performance than the other two schemes for all tag set size. The expected identification time of PRQT increases linearly with n with a slope of 2.36. So the average time complexity of PRQT is $O(2.36n)$.

In Fig. 6, we compare the worst-case identification time for PRQT and QT for tag set size equal to 50, 100, and 150 respectively assuming uniform ID distribution and 64 bit ID length. This worst-case is the worst of the 1000 trials in each case. The results show that PRQT gives increasingly better worst case performance than QT as the tag set size increases.

Fig. 7 shows the cumulative distributions of polling rounds for PRQT and QT to identify all tags. These distributions are obtained by 1000 trials each of PRQT and QT for tag set sizes equal to 50, 100, and 150 assuming uniform ID distribution and 64 bit ID length. It is noted that PRQT always performs better than QT for the same tag set size and performs increasingly better with the increasing tag set size. As an example, with a population of 150 tags, the probability of identifying all tags by no more than 400 polling rounds is 0.988 for PRQT and 0.158 for QT.

IV. CONCLUSION

In this paper we propose a Prefix-Randomized Query-Tree (PRQT) protocol for the anti-collision problem of RFID systems. We study the relation of the initial prefix length and the tag identification time of PRQT and summarize the optimal initial prefix length for different tag set size in Table I. Through theoretical analysis and simulation studies, we show that PRQT performs better than the Query-Tree protocol in terms of both average and worst-case time complexity. The design of a light-weight tag set size estimation algorithm will be part of the future work.

REFERENCES

[1] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," *IEEE Inter. Sym. On Information Theory*, 1998.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

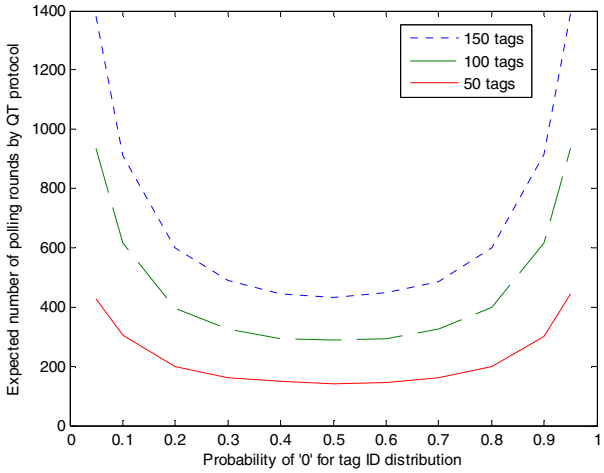

Fig. 4. Effect of the tag ID distribution on the identification time of QT protocol.

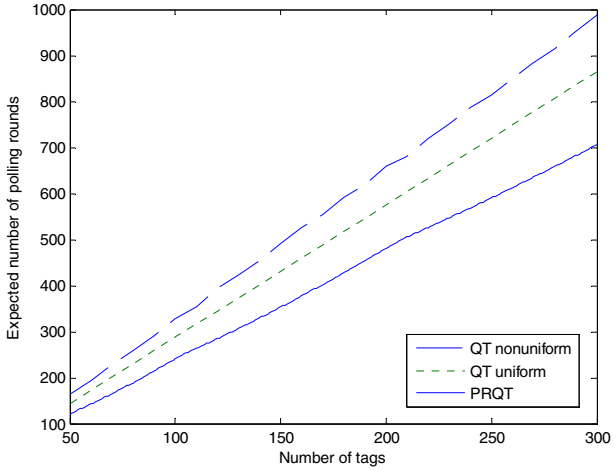

Fig. 5 Comparison of expected identification time

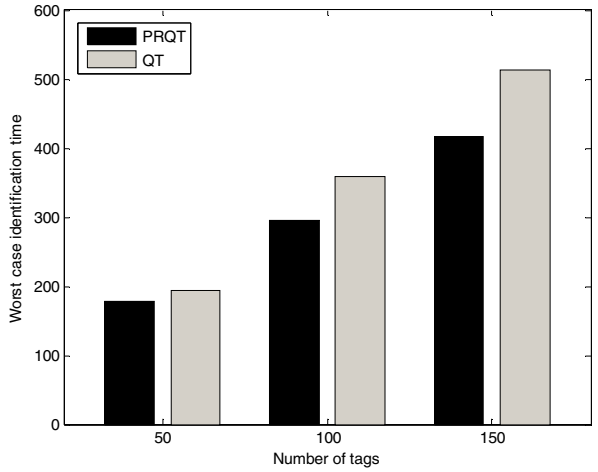

Fig. 6 Worst-case tag identification time among 1000 trials for different tag set size

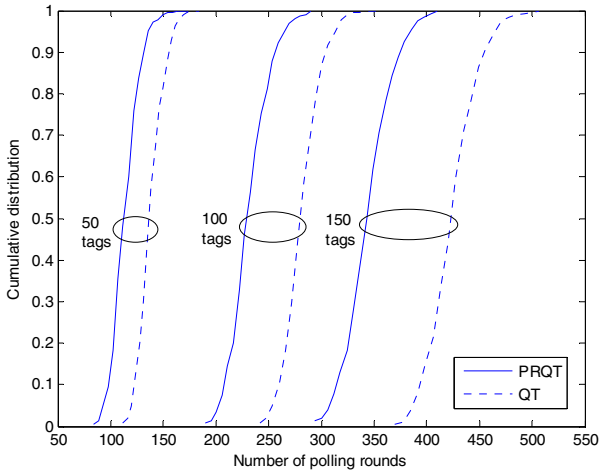

Fig. 7 Cumulative distribution for number of polling rounds of PRQT and QT for different tag set size

[2] C. Law, K. Lee, and K. Siu, "Efficient memoryless protocol for tag identification," In *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, August 2000.

[3] Philips I*Code1 System Design Guide – "Application Note AN00025", May 2002, Philips Semiconductors. http://www.semiconductors.philips.com/markets/identification/products/icode/.

[4] ISO/IEC 18000-7 draft, "RFID for item management-Air interface, Part 7-Parameters for an active RFID interface communications at 433MHz," (684_18000-7_FCD.doc at http://www.autoid.org)

[5] H. Vogt, "Efficient object identification with passive RFID tags," *Inter. Conf. on Pervasive Computing*, LNCS, pp. 98-113, 2002.

[6] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed Aloha for Multiple RFID Objects Identification," *IEICE Trans. Commun.*, vol. E88-B, No. 3, March 2005.

[7] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," IEEE Transactions on Information Theory, IT-25(5): 505-515, 1979.

[8] M. A. Kaplan and E. Gulko, "Analytic properties of multiple-access trees," *IEEE Transactions on Information Theory*, IT-31(2): 255-263, 1985.

[9] J. l. Massey, "Collision resolution algorithms and random-access communications," In G. Longa, editor, *Multi-User Communication Systems*, pages 73-137. Springer Verlag, New York, 1981.

[10] EPC™ Generation 1 Tag Data Standards Version 1.1 Rev.1.27, EPCglobal. http://www.epcglobalinc.org/standards_technology/specifications.html.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.