Prefix-Length Adaptation for PRQT Protocol in 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
Email: {kwchian4, chua0, yum}@ie.cuhk.edu.hk

***Abstract*— Prefix-Randomized Query-Tree (PRQT) protocol has been proposed for multiple tag identification in RFID systems. The optimal performance of PRQT can be achieved with a proper choice of the initial prefix length according to the tag set size. In this paper, we propose an initial prefix length adaptation algorithm for PRQT protocol when the tag set size is unknown before identification. The algorithm starts with the setting of a small initial prefix length l followed by the polling of all 2^l prefixes. The initial prefix length is then increased repeatedly until the collision ratio satisfies a prescribed condition. We derive the optimal increment step size and the respective sequence of decision thresholds. Simulation results show that PRQT with initial prefix length adaptation can significantly reduce the expected tag read time for all range of tag set size when compared to the use of Query-Tree protocol.**

***Index Terms*— RFID, anti-collision, prefix.**

I. Introduction

Radio Frequency Identification (RFID) technology is an important kind of Automatic Identification and Data Capture (AIDC) technology and is getting increasingly popular in many application domains. In the form of smart labels, they are used to improve supply chain management and manufacturing logistics. In the form of contactless smart cards, they are used for speeding up transportation ticketing and toll collection. In the form of smart tags, they are used to improve security with car immobilization and remote keyless entry. RFID technology enables efficient wireless object identification which is envisioned to bridge the physical world and virtual world [1]-[2]. A typical RFID system consists of: (i) tags with unique IDs tagging on different items and storing various item information, (ii) readers to collect information from tags, (iii) a data processing system, including edge interface, middleware and enterprise backend, to aggregate and extract meaningful information to advance business processes, (iv) the communication infrastructure providing the set of wired and wireless connections for the above components.

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4220/03E)

The biggest advantage of RFID over other data collection techniques is the ability to identify multiple tags simultaneously without physical contact and orientation requirements. However, since all tags share the broadcast channel to communicate with the reader, multiple access techniques should be designed for RFID systems. Tags usually are simple tiny devices with limited computation and communication capability. In the case of low-cost passive RFID systems, which are envisioned to realize item-level tagging and have very high economic potential, tags may not be able to sense the shared medium to know whether their transmission is successful or not. Therefore, anti-collision protocols of RFID systems should be optimized for tags with low computational capability, small memory size and limited power supply.

For deterministic anti-collision protocols, the reader broadcasts a query command for all tags and then polls each tag based on its unique ID. QT [3] is a simple deterministic algorithm designed for RFID systems. In this protocol, a reader announces a prefix to all tags, and tags respond with their IDs if they have a prefix match. When a collision occurs for the prefix, the reader polls a one-bit-longer prefix later. The polling efficiency of QT is low when the tag set size is large or the ID distribution is sparse. Its worst-case identification time for n tags is $n(k+2-\log_2 n)$, where k is the length of ID string. For stochastic anti-collision protocols, tags transmit their IDs to the reader in randomly chosen time slots. The framed-Aloha protocol used by passive RFID systems [4]-[5] groups several time slots into a frame. Each tag sends its ID in a randomly chosen time slot once per frame. The identification time of stochastic protocol is not affected by the length and distribution of tag IDs, however, it cannot identify all tags with complete certainty if the tag set size is unknown before identification.

PRQT protocol for multiple tag identification is proposed in [6]. PRQT differs from QT in that it uses prefixes chosen randomly by tags rather than using their ID-based prefixes. Therefore, its performance is independent of the tag ID distribution and ID length. Moreover, by using polling, PRQT is capable of identifying all passive tags with complete certainty even if the tag set size is unknown. PRQT is shown to have better expected tag read time than QT. The optimal

1-4244-0357-X/06/$20.00 ©2006 Crown Copyright
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

performance of PRQT requires the use of the optimal initial prefix length, which is a function of tag set size.

In this paper, we propose an initial prefix length adaptation algorithm for PRQT in applications where the tag set size is unknown before identification. This algorithm starts with a small initial prefix length l followed by the polling of all 2^l prefixes. The collision ratio is computed from the 2^l responses from tags. The initial prefix length is increased repeatedly until a decision threshold on collision ratio is satisfied. We study the relation of decision thresholds and increment step size for this algorithm and derive the optimal increment step size. The performance of PRQT with initial prefix length adaptation is compared with that of using QT through simulations.

II. Prefix-Randomized Query-Tree Protocol

PRQT [6] requires each tag to randomly generate a binary prefix with a prescribed length. After the reader broadcasts a query command with an initial prefix length l, each tag randomly generates an l-bit binary prefix. The reader then polls each of these 2^l prefixes sequentially. In each polling round, tags with prefix matches respond with their IDs. Those tags generating the same prefix will respond to the reader at the same time and cause a collision. After the reader polls all 2^l prefixes, it knows the set of collided prefixes. For the collided prefix f_i, the reader broadcasts a command asking those tags matching this prefix to expand f_i 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 f_i until all tags choosing this prefix are identified. After that, the algorithm returns and continues polling the rest of collided prefixes. The same procedure is repeated until all collided prefixes have been resolved. This process differs slightly from the original PRQT [6] in that the collided subtrees are queried after all level-l nodes have been polled. This revision does not affect the performance of original PRQT and is necessary for incorporating with the initial prefix length adaptation algorithm as will be presented in the next section. Fig. 1 is an example of the query tree of PRQT for identifying eight tags with shaded node indicating the broadcasting command, dark node indicating prefix with single response, white node indicating prefix with no response, and grey node with a number x indicating prefix with x responses.

In [6], the expected number of polling rounds needed to completely identify all tags, W, for PRQT is derived. Assume the amount of time for each polling-response round is fixed, W is identical to the expected tag read time. Given the tag set size n and the the initial prefix length l, W can be obtained by

$$W = 2^l \sum_{k=0}^{n} p_k t_k \tag{1}$$

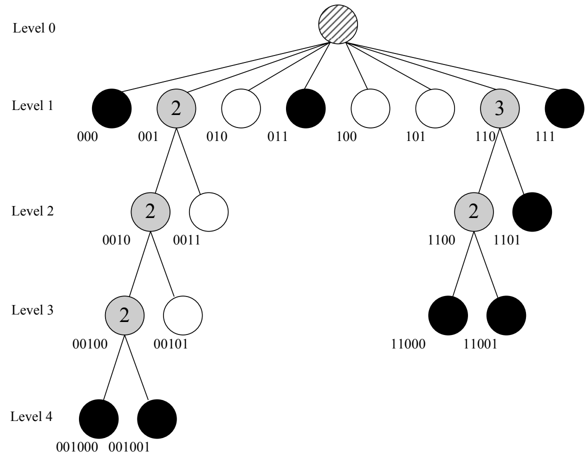

Fig. 1. PRQT example for identifying eight tags.

Tag set size n	Optimal initial prefix length l^*
1 - 3	1
3 - 6	2
7 - 14	3
15 - 26	4
27 - 53	5
54 - 107	6
108 - 215	7
216 - 429	8
430 - 858	9

TABLE I
OPTIMAL INITIAL PREFIX LENGTH FOR DIFFERENT TAG SET SIZE

where

$$p_k = \binom{n}{k} \left(\frac{1}{2^l}\right)^k \left(1 - \frac{1}{2^l}\right)^{n-k} \tag{2}$$

and

$$t_k = \frac{1}{2^{k-1}-1} \left[2^{k-1} + \sum_{i=0}^{k-1} \binom{k}{i} t_i \right] \tag{3}$$

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

The optimal initial prefix length l^* for a given tag set size n is also derived in [6]. Table I shows the values of l^* as a function of n, for $1 \le n \le 858$.

III. Tag Identification with Unknown Tag Set Size

PRQT protocol described in the previous section requires the knowledge of tag set size for setting the optimal initial prefix length l^*. In this section, we propose an initial prefix length adaptation algorithm to be used with the PRQT protocol when the tag set size is unknown before identification.

A. Initial Prefix Length Adaptation Algorithm

To begin, let the reader choose a small enough initial prefix length l, announce it to all tags and poll all 2^l

prefixes. From responses of the tags, the reader knows the number of prefixes with single response N_1, the number of prefixes with no response N_0 and the number of collided prefixes N_c. The reader decides whether the current l value should be increased or not by checking the collision ratio r ($r = N_c/2^l$) against two predefined thresholds $r_2^*(l)$ and $r_3^*(l)$. Specifically, l is increased by a step size $\Delta = 3$ if $r \geq r_3^*(l)$ or increased by a step size $\Delta = 2$ if $r_2^*(l) \leq r < r_3^*(l)$. The updated l is broadcasted to all tags again for faster identification time. This procedure is repeated until $r < r_2^*(l)$ and the PRQT protocol is proceeded with the final initial prefix length to resolve all collided prefixes. Using this method, all previous polling efforts for initial prefix length adaptation are wasted. But being able to use the best initial prefix length (i.e. matches the estimated tag set size) can shorten the overall tag read time. Methods for computing thresholds $r_2^*(l)$ and $r_3^*(l)$ and choosing the optimal step size Δ are presented in the following sections. The following is the initial prefix length adaptation algorithm.

Step 1: $l = 1$.
Step 2: Broadcast l to all tags and poll all 2^l initial prefixes.
Step 3: Compute the collision ratio r.
Step 4: If $r \geq r_3^*(l)$
$l \leftarrow l + 3$, go to **Step 2**.
Else if $r_2^*(l) \leq r < r_3^*(l)$
$l \leftarrow l + 2$, go to **Step 2**.
Else if $r < r_2^*(l)$
Proceed PRQT with l.
Step 5: End.

B. Computing $r_\Delta^*(l)$

Let W_l be the expected tag read time with initial prefix length l and $n_\Delta^*(l)$ be the minimum tag set size for which increasing l by Δ leads to a shorter expected tag read time. After the polling of 2^l prefixes, if we increase l by Δ , the total expected tag read time is $2^l + W_{l+\Delta}$. Therefore $n_\Delta^*(l)$ can be obtained as

$$n_\Delta^*(l) = \min\left\{n | W_{l+\Delta} + 2^l < W_l\right\} \tag{4}$$

Using W_l from (1), we obtain

$$\begin{aligned} n_\Delta^*(l) &= \min\Bigg\{n | 2^{l+\Delta} \sum_{k=0}^{n} \binom{n}{k} (\frac{1}{2^{l+\Delta}})^k (1 - \frac{1}{2^{l+\Delta}})^{n-k} t_k \\ &\qquad + 2^l < 2^l \sum_{k=0}^{n} \binom{n}{k} (\frac{1}{2^l})^k (1 - \frac{1}{2^l})^{n-k} t_k \Bigg\} \\ &= \min\Bigg\{n | \sum_{k=0}^{n} \binom{n}{k} ((\frac{1}{2^l})^k (1 - \frac{1}{2^l})^{n-k} \\ &\qquad - 2^\Delta (\frac{1}{2^{l+\Delta}})^k (1 - \frac{1}{2^{l+\Delta}})^{n-k}) t_k > 1 \Bigg\} \end{aligned}$$

For $n_\Delta^*(l)$ tags, the collision probability $r_\Delta^*(l)$ is

$$r_\Delta^*(l) = 1 - (1 - \frac{1}{2^l})^{n_\Delta^*(l)} - (\frac{n_\Delta^*(l)}{2^l})(1 - \frac{1}{2^l})^{n_\Delta^*(l)-1} \tag{5}$$

In our algorithm, we check the collision ratio r against the collision probability $r_\Delta^*(l)$ to decide if l should be increased by Δ. In other words, if $r \geq r_\Delta^*(l)$, the unknown tag set size is probably larger than $n_\Delta^*(l)$ and hence increasing l by Δ will lead to shorter expected tag read time. Table II shows sequences of $n_\Delta^*(l)$ and $r_\Delta^*(l)$ for Δ from 2 to 4 and l from 1 to 5.

	l	1	2	3	4	5
$\Delta = 2$	$n_2^*(l)$	7	15	30	60	119
	$r_2^*(l)$	0.9375	0.9198	0.9038	0.8954	0.8894
$\Delta = 3$	$n_3^*(l)$	11	21	43	85	171
	$r_3^*(l)$	0.9941	0.9810	0.9771	0.9724	0.9714
$\Delta = 4$	$n_4^*(l)$	18	36	72	144	289
	$r_4^*(l)$	0.9999	0.9996	0.9992	0.9990	0.9989

TABLE II
SEQUENCE OF $n_\Delta^*(l)$ AND $r_\Delta^*(l)$ FOR Δ = 2, 3 AND 4

C. Optimal Choice of Step Size Δ

We now proceed to a three-part argument leading to the conclusion that Δ = 2 or 3 are desirable choices.

Part 1: $\Delta = 1$ is a bad choice.

Let $\Delta = 1$ in (4), and compare $2^l + W_{l+1}$ with W_l. Since W_l is at most $2^l + W_{l+1}$ (occurs when all level-1 nodes of the query tree are collided), the condition in (4) cannot be satisfied and no value of tag set size will lead to a shorter expected tag read time by increasing l to $l + 1$. Therefore, $\Delta = 1$ is a bad choice.

Part 2: $\Delta \geq 4$ are bad choices.

From Table II, we can see that larger Δ results in $r_\Delta^*(l)$ closer to 1. When $\Delta \geq 4$, $r_\Delta^*(l)$ are all very close to 1. This leads to decision ambiguity when they are used to compare with the collision ratio r. Specifically, if the collision ratio $r = 1$, many values of Δ are suitable. But to avoid the high cost of overshooting l, $\Delta \geq 4$ should not be chosen.

Part 3: Δ = 2 or 3 depending on r.

From the last two parts, Δ = 2 or 3 are the only feasible choices for the algorithm. Its choice depends on the value of collision ratio r. If $r \geq r_3^*(l)$, l is increased by $l + 3$ ($\Delta = 3$). If $r_2^*(l) \leq r < r_3^*(l)$, l is increased to $l + 2$ ($\Delta = 2$). Table III shows the sequence of $r_2^*(l)$ and $r_3^*(l)$. Note that since we start at $l = 1$ and Δ = 2 or 3, an initial prefix length of $l = 2$ is not possible. Therefore the decision thresholds for $l = 2$ is absent in Table III.

l	1	3	4	5	6	7
$r_2^*(l)$	0.9375	0.9038	0.8954	0.8894	0.8888	0.8872
$r_3^*(l)$	0.9941	0.9771	0.9724	0.9714	0.9706	0.9699

TABLE III

DECISION THRESHOLDS FOR INITIAL PREFIX LENGTH ADAPTATION ALGORITHM

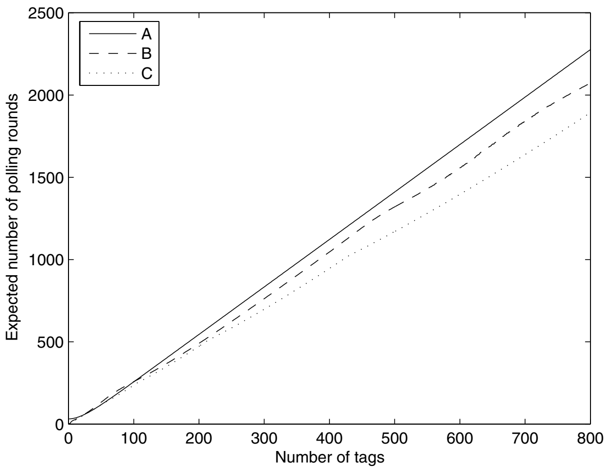

Fig. 2. Expected number of polling rounds by (1) PRQT (unknown tag set size) with fixed l = 5 (curve A), (2) PRQT (unknown tag set size) with the initial prefix length adaptation algorithm (curve B) and (3) PRQT (known tag set size) with l^* (curve C).

In Fig. 2 we compare the expected tag read time for 1 to 800 tags for two methods: (i) PRQT (unknown tag set size) with fixed l = 5 (analytical result), (ii) PRQT (unknown tag set size) with the initial prefix length adaptation algorithm (average result of 1000 trials of simulation). Also shown is the result of PRQT (known tag set size) with l^* in Table I (analytical result). This is the performance lower bound. These results show that when the tag set size is unknown, the initial prefix length adaptation algorithm can reduce the expected tag read time and the reduction is larger when n increases.

IV. PERFORMANCE EVALUATION

For simplicity, we denote PRQT with initial prefix length adaptation algorithm the *Adaptive-PRQT*. The expected and worst-case identification time of PRQT, Adaptive-PRQT, and QT (with uniform and nonuniform tag ID distribution) are compared in this section. QT is a deterministic anti-collision protocol with an expected identification time in the range of $(2.881n - 1, 2.887n - 1)$ for n tags with uniformly distributed IDs. However, its worst-case identification time for identifying n tags is $n(k + 2 - \log_2 n)$ where k is the ID length [3]. In [6], the effect of tag ID distribution on identification time of QT is investigated by simulation. By varying the probability of ‘0’ in the ID, it is noted that the tag identification time of QT is heavily dependent on the uniformity of the tag ID distribution. This performance degradation is more serious for larger tag set size. The identification time of PRQT protocol, however, is totally independent of ID distribution and ID length of tags.

In Fig. 3, we compare the expected identification time for four different schemes:

1) PRQT with optimal initial prefix length l^* of Table I (analytical result).
2) Adaptive-PRQT (average result of 1000 trials of simulation).
3) QT with uniform tag ID distribution (analytical result using the lower bound of its expected identification time $2.881n - 1$).
4) QT with non-uniform tag ID distribution where Prob(‘0’) = 0.3 (average result of 1000 trials of simulation).

It can be seen from Fig. 3 that the two PRQT schemes have better performance than the two QT 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 $O(2.36n)$. The ID length is set to 64 bits for QT protocol.

In Fig. 4, we compare the worst-case identification time for PRQT, Adaptive-PRQT, and QT with uniform and nonuniform (Prob(‘0’) = 0.3) ID distribution, for tag set size equal to 50, 100, and 150 respectively assuming 64 bit ID length. This worst-case result is the worst of 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. 5 shows the cumulative distributions of polling rounds to identify 50 and 150 tags respectively. These distributions are obtained by 1000 trials each of PRQT, Adaptive-PRQT, and QT with uniform and nonuniform (Prob(‘0’) = 0.3) ID distribution assuming 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 set size of 150 tags, the probability of identifying all tags by no more than 400 polling rounds is 0.988 for PRQT, 0.958 for Adaptive-PRQT, 0.158 for QT with uniform ID distribution and 0 for QT with nonuniform ID distribution.

V. CONCLUSION

In this paper we propose an initial prefix length adaptation algorithm for PRQT protocol in applications where the tag set size is unknown before identification. This algorithm adaptively increases the initial prefix length based on the collision ratio such that an initial prefix length, which matches the estimated tag set size, can be found for proceeding PRQT protocol with better performance. We derive the sequence

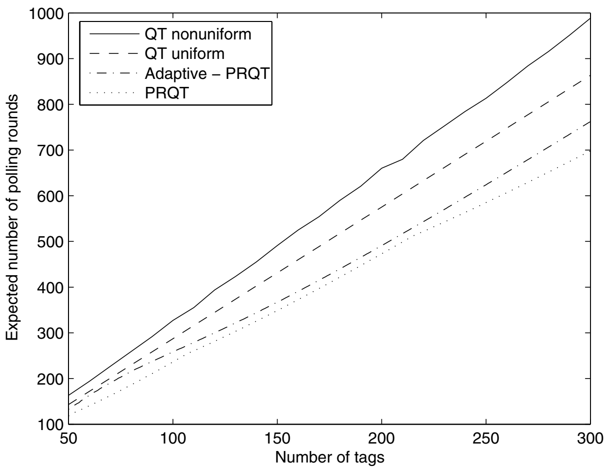

Fig. 3. Comparison of expected identification time.

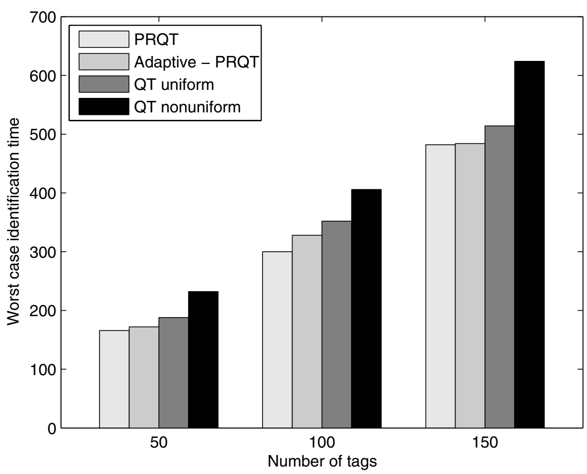

Fig. 4. Worst-case tag identification time among 1000 trials for different tag set size.

of decision thresholds $r^*_\Delta(l)$ and the optimal initial prefix length increment Δ. Through simulation studies, we show that PRQT with initial prefix length adaptation algorithm performs better than the Query-Tree protocol in terms of both average and worst-case time complexity.

References

[1] R. Want. Enabling ubiquitous sensing with RFID. *Computer*, 37(4):84–86, 2004.

[2] V. Stanford. Pervasive computing goes the last hundred feet with RFID systems. *Pervasive Computing, IEEE*, 2(2):9–14, 2003.

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

[4] H. Vogt. Efficient object identification with passive RFID tags. In *Inter. Conf. on Pervasive Computing, LNCS*, pages 98–113, 2002.

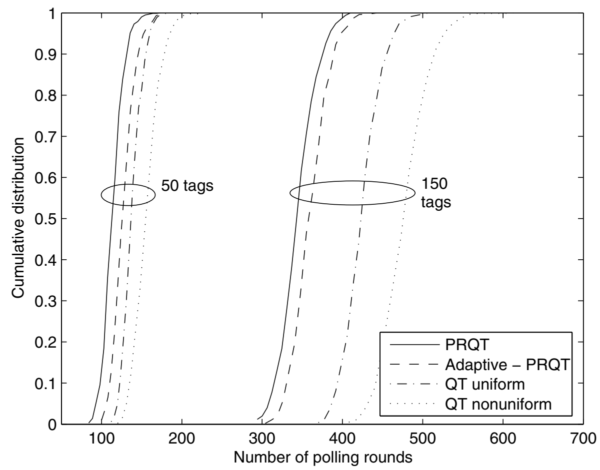

Fig. 5. Cumulative distribution for number of polling rounds of PRQT and QT.

[5] M. Kobayashi B. Zhen and M. Shimizu. Framed Aloha for Multiple RFID Objects Identification. *IEICE Trans. Commun.*, E88-B, March 2005.

[6] K. W. Chiang C. Q. Hua and T. S. Yum. Prefix-Randomized Query-Tree Protocol for RFID systems. In *Proceedings of the Inter. Conf. on Commun. 2006*, June 2006. Preprint appears in http://personal.ie.cuhk.edu.hk/∼kwchian4/ICC2006-PRQT.pdf.

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