International Journal of Recent Trends in Engineering & Research
ISSN (ONLINE) : 2455 – 1457
IMPACT FACTOR : 4.101

Memoryless Query Tree Algorithm for Identifying Tags in Passive RFID Systems

Intaek Lim
Department of Embedded Software, Busan University of Foreign Studies

Abstract— This paper proposes a revised query tree algorithm in RFID systems. In the proposed algorithm, the tag will send the remaining bits of their identification codes when the query string matches the first bits of their identification codes. When the reader receives all the responses of the tags, it knows which bit is collided. If the collision occurs in the last bit, the reader can identify two tags simultaneously without further query. While the tags are sending their identification codes, if the reader detects a collision bit, it will send a signal to the tags to stop sending. According to the simulation results, the proposed algorithm outperforms the QT algorithm in terms of the number of queries and the number of response bits.

Keywords—RFID, Anti-collision algorithm, QT protocol, Memoryless protocol

I. INTRODUCTION

In recent years, radio frequency identification (RFID) technology has been widely adapted in a variety of application fields such as public transportation, production control, animal identification, and object tracking [1]. An RFID system consists of radio frequency tags attached to objects that need to be identified and one or more electromagnetic readers. Unlike the traditional bar code system, the great benefit of RFID technology is that it allows information to be stored and read without requiring either contact or line of sight between the tag and the reader. For this contact-less feature, RFID technology in the near future will become ubiquitous and an attractive alternative to bar code in many application fields.

In RFID systems, tags may either be actively or passively powered [2][3]. Active tags are provided with their own power sources for computing and transmitting data. Due to the complexity and cost of mounting a power source onto a tag, active tags are not practical for the consumer products. On the other hand, passive tags do not need their own power source. They instead rely on RF energy induced by the electromagnetic waves emitted by the reader. When a passive tag enters into the RF field of the reader, it receives energy from the field. Due to more advanced technology for the protocol and circuit design, the reliability and the read range of passive RFID networks continue to improve, and the cost continues to decrease, which lead to an increase of passive tag applications.

When there is more than one tag in the identification range of a reader, all or some tags may send their response back to the reader at the same time. If only one tag answers, the reader receives just one message which is correctly decoded. If two or more tags answer, their messages will collide on the RF channel and cannot be correctly received by the reader. This may lead to mutual interference, which is referred to as a collision. A technical scheme that handles multiple-access without any interference is called as an anti-collision algorithm. In RFID systems, an anti-collision algorithm must be carefully designed for conserving low power consumption and fast identification of multiple tagged objects simultaneously.

There are two types of anti-collision algorithms: deterministic and probabilistic algorithm [4][5]. The deterministic algorithm resolves collisions by muting subsets of tags that are involved in a collision. By successively muting larger subsets, only one tag will be left and finally led to successful transmission. Binary tree and query tree algorithms are the two main methods of the deterministic algorithm. The probabilistic algorithm is based on an ALOHA-like protocol that provides slots for the tags to send their data. Whenever a collision has occurred, another frame of slots is provided, and the tags that are involved in collisions will choose different slots in the next read cycle.

The query tree (QT) algorithm is the deterministic algorithm designed by AutoID center [6]. In the QT algorithm, the reader broadcasts a prefix and tags having their IDs matching with the prefix respond with their entire IDs. If there is a collision, the reader appends symbol 0 or 1 to the prefix, and continues to query for longer prefixes until no collision occurs. Once a tag is identified, the reader starts a new round of queries with another prefix. The main feature of the QT algorithm is that each tag is memory-less, i.e., the current response of each tag depends only on the current query of the reader but not on the past history of the reader's queries. The QT algorithm has two problems in the process of tag identification. At first, tags will send their entire Ids including the prefix when the prefix matches the first bits of their IDs. Secondly, while the tags are sending their IDs, the tags cannot know the collision status of their transmissions because the reader does not notify the tags of the collision even though a collision occurs. These problems will make the number of tag response bits and read cycles increase and eventually lead to relatively long identification delay.

In this paper, we present a new anti-collision algorithm based on the QT protocol. In the proposed protocol, the tags that match the prefix send only the remaining bits of their IDs. While the tags are sending their responses, if the reader detects a collision bit, it sends a signal to the tags to stop their transmissions.

II. PROPOSED ALGORITHM

In the proposed algorithm, named as QT_ecfi (QT for energy conserving and fast identification), it is assumed that the reader can detect the exact position of the collision bit in the tag's responses. Like the QT algorithm, the reader broadcasts a prefix as a query string, which may have the length of 1 to (k-1) bits. When the prefix matches the first bits of their IDs, the tags will send the remaining bits of their IDs instead of the entire IDs and wait the next query string. The remaining bits are the bits of ID except that matches the prefix.

The collision occurs when two or more tags respond to the query string. When the reader detects a collision, it can tell from the tags' response at which bit the collision occurred. If a collision occurs, it makes a new query string. The new query string is made by extending the prefix to the non-collision bits and then appending a symbol '0' or '1'. While the tags are sending their responses, it the reader detects a collision bit, it will broadcast a signal to the tags to stop sending their responses. If a collision occurs at the last bit of tags' response, the reader can make sure that two tags having different IDs only at the last bit exist in the identification range of a reader. Therefore it can simultaneously identify two tags. If a collision is not occurred, the reader can identify a tag and starts a new round of queries with another prefix.

The operation of QT_ecfi algorithm is as follows. Let q be a query string and $|q|$ be the length of string q. And let a binary string w be the tag's ID with the length k. If we let r be a tag's response string, then the string r is the tag's ID except the query string. Therefore r can be defined as follows.

$$r = w_{|q|+1} w_{|q|+2} \cdots w_k \quad (1)$$

The following is the operational sequence for the reader.

1) Let the queue $Q =< q_1, q_2, \cdots, q_l >$
2) Delete a query q_1 from the queue and broadcast the query to the tags.
3) Update the queue to be $Q =< q_2, \cdots, q_l >$
4) According to the responses from the tags:
   ① Let z be the non-collision bits string in the tag's response string, and $|z|$ be the length of string z.
   ② If $|z|=k-|q|$, i.e. a collision is not detected, then set the tag ID $q_1 r$ to be identified and insert it into the memory M.
   ③ If $|z|=k-|q|-1$, i.e. a collision is detected at the last bit of tag's response, then set the tag's IDs $q_1 w_{|q|+1} w_{|q|+2} \cdots w_{k-1} 0$ and $q_1 w_{|q|+1} w_{|q|+2} \cdots w_{k-1} 1$ to be identified and insert them into the memory M.
   ④ If a collision is detected at any bit except the last bit, then broadcast s stop signal to the tags and insert the string "$q_1 z 0$" and "$q_1 z 1$" into the queue, i.e. set the queue $Q =< q_2, \cdots, q_l, q_1 z 0, q_1 z 1 >$.
5) Repeat the above sequences until the queue is empty.

The following is the operational sequence for the tag.

1) Let the tag's ID $w = w_1 w_2 \cdots w_k$.
2) Let q be the query string received from the reader, and $|q|$ be the length of string q.
3) If $q=\varepsilon$ or $q = w_1 w_2 \cdots w_{|q|}$, then the tag sends the string $r = w_{|q|+1} w_{|q|+2} \cdots w_k$ to the reader. This means that the tag sends the remaining bits of its ID as a response when the query matches the first bits of ID.
4) If the tag receives a stop signal while sending response, it stops the transmission.

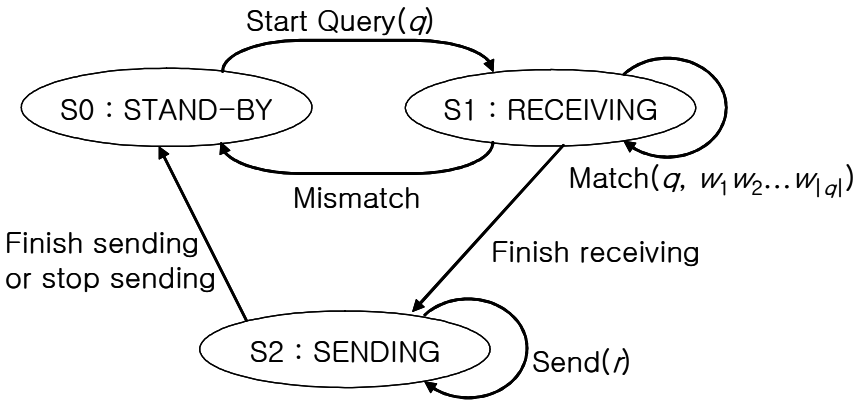

Figure1. State transition diagram for QT_ecfi algorithm.

Figure 1 shows the state transition diagram of the tag's state machine for the QT_ecfi algorithm. When the tag in the STAND_BY state receives a query string q that matches the first bit of its ID, it changes its state into the RECEIVING state for receiving all the query string. If the query string matches, the tag enters into the SENDING state. The tag in the SENDING state sends the remaining bits of ID except the query string. If it detects a stop signal while sending, it stops the transmission immediately and changes into the STAND_BY state. Otherwise, it continues the transmission and changes into the STAND_BY state after completing the transmission.

Query string	Tag answers	Answered tags	Updated Queue
ε	X	all	<0,1>
0	0X	0001, 0010	<1,000,001>
1	01X	1010, 1011	<000,001>
000	1	0001	<001>
001	0	0010	<ε>

Figure 2. An example of QT_ecfi algorithm.

Figure 2 illustrates an example of the communication between the reader and the tags. The tag's IDs used in the example are <0001, 0010, 1010, 1011>. As shown in the figure, the total number of query and response rounds is 5 for identifying all 4 tags. In this case, the total number of bits as query strings that the reader broadcasts is 8, and all the tag answers are 16 bits.

III. SIMULATION RESULTS

In this paper, the computer simulations are performed to compare the performance of QT_ecfi and QT algorithms. It is assumed that the length of tag ID is 64 bits. The performance measurement parameters to be interested are the total number of reader's query bits and the average number of a tag's response bits. We used the random number and sequential number as the tag ID for simulation. In the case of sequential tag ID, it is assumed that the tag ID increases sequentially from the least significant bit.

Figure 3 and 4 show the total number of reader's query bits in the case of random ID and sequential ID, respectively, As shown in Figure 3, when the tag IDs are random, the reader in QT algorithm has to broadcast 2,090 bits for identifying 100 tags, but only 1,323 bits are broadcasted in QT_ecfi algorithm. Therefore, it is manifest that the performance of the proposed QT_ecfi algorithm outperforms the QT algorithm. It is because in the QT_ecfi algorithm the reader tells the exact position of collision bit. Also for identifying 100 tags with the sequential ID, the QT algorithm needs 3.2 times more query bits than the QT_ecfi algorithm.

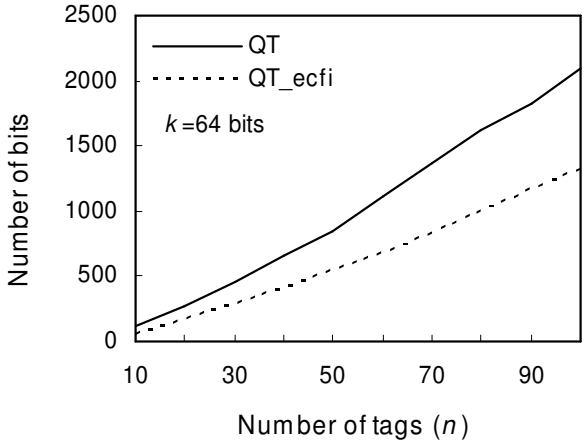

Figure 3. Total number of query bits(randon ID).

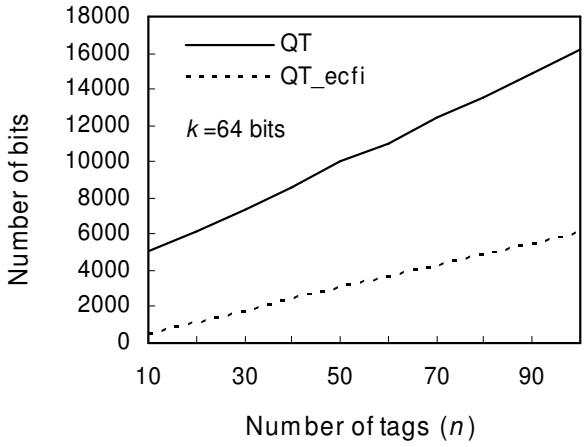

Figure 4. Total number of query bits(sequential ID).

Figure 5 and 6 show the average number of a tag's response bits. As shown in the figures, the average number of a tag's response bits of QT_ecfi algorithm is less than that of QT algorithm regardless of ID. This is because in QT_ecfi algorithm the tag sends the remaining bits except the query prefix and stops sending the response when it receives the stop signal from the reader. The tags in QT algorithm need 7.9 and 59 times more response bits than the QT_ecfi algorithm with respect to the random and sequential ID, respectively. Therefore, the QT_ecfi algorithm can save the identification time and conserve the tag's energy.

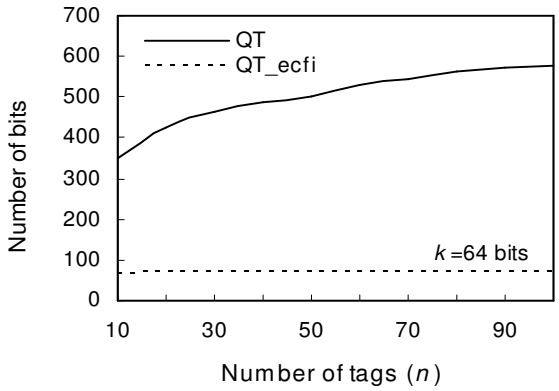

Figure 5. Average number of response bits(random ID).

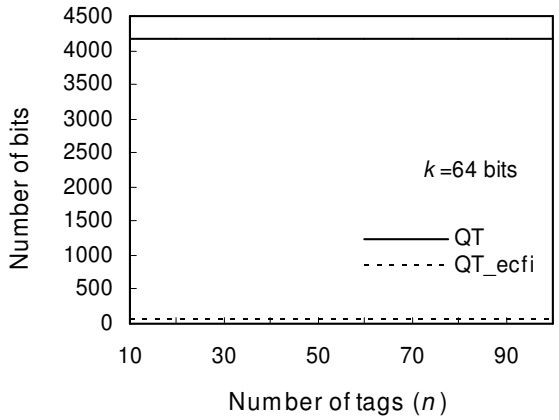

Figure 6. Average number of response bits(sequential ID.

IV. CONCLUSIONS

In the QT_ecfi algorithm, when the query string received from the reader matches with the first bits of ID, the tags send the remaining bits of their ID except the matched prefix. Also the reader broadcasts a stop signal when it detects a collision. When the tag receives the stop signal while sending, it stops sending its response immediately. If the position of collision bit is the last bit, the reader can identify two tags simultaneously. The simulation results showed that the total number of query bits and the average number of tag's response bits are smaller than QT algorithm.

REFERENCES

I. 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.
II. K. Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, Carl Hanser Verlag GmbH & Co., 2002.
III. H. Vogt, "Efficient Object Identification with Passive RFID Tags," First International Conf. on Pervasive Computing, LNCS, vol.2414, pp.99-113, Springer-Verlag, 2002.
IV. F. Zhou, C, Chen, D, Jin, C, Huang, and H. Min, "Evaluating and Optimizing Power Consumption of Anti-Collision Protocols for Applications in RFID Systems," Proc. ISLPED'04, New York, USA, pp.357-362, 2004.
V. S. E. Sarma, S. A. Weis, and D. W. Engels, "RFID Systems and Security and Privacy Implications," Proc. CHES2002, LNCS, vol.2523, pp.454-469, 2003.
VI. C. Law, L. Lee, and K. Y. Siu, "Efficient Memoryless Protocol for Tag Identification," Auto-ID Center, MIT-AUTOID-TR-003, Oct. 2000.
VII. Auto-ID Center, "860MHz-930MHz Class 0 Radio Frequency Identification Tag Protocol Specification Candidate Recommendation, Version 1.0.0," June 2003.

