MRCT: An Efficient Tag Identification Protocol in RFID Systems with Capture Effect

Sunwoong Choi[1], Jaehyuk Choi[2] and Joon Yoo[2]
[1] School of Electrical Engineering, Kookmin University
Seoul 136-702 Korea
[e-mail: schoi@kookmin.ac.kr]
[2] Department of Software Design & Management, Gachon University
Seongnam 461-701 Korea
[e-mail: {jchoi, joon.yoo}@gachon.ac.kr]
*Corresponding author: Joon Yoo

Received April 13, 2013; accepted July 10, 2013; published July 30, 2013

Abstract

In RFID systems, one important issue is how to effectively address tag collision, which occurs when multiple tags reply simultaneously to a reader, so that all the tags are correctly identified. However, most existing anti-collision protocols assume isotropic collisions where a reader cannot detect any of the tags from the collided signals. In practice, this assumption turns out to be too pessimistic since the capture effect may take place, in which the reader considers the strongest signal as a successful transmission and the others as interference. In this case, the reader disregards the other collided tags, and in turn, fails to read the tag(s) with weaker signal(s). In this paper, we propose a capture effect-aware anti-collision protocol, called Multi-Round Collision Tree (MRCT) protocol, which efficiently identifies the tags in real RFID environments. MRCT deals with the capture effect as well as channel error by employing a multi-round based identification algorithm. We also analyze the performance of MRCT in terms of the number of slots required for identifying all tags. The simulation results show that MRCT significantly outperforms the existing protocol especially in a practical environment where the capture effect occurs.

Keywords: Anti-collision algorithm, capture effect, collision tree, RFID, tag identification

A preliminary version of this paper appeared in ICUFN 2012, July 4-6, Phuket, Tailand. This version includes a concrete mathematical analysis and supporting simulation results on the performance of MRCT protocol.
This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2011-0023856), and in part by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (2012R1A1A1014755 and NRF-2013R1A1A1006823), and in part by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

http://dx.doi.org/10.3837/tiis.2013.07.006

1. Introduction

Radio Frequency Identification (RFID) is a technology which uses radio waves to identify tagged objects. An RFID system consists of a reader and multiple tags. The reader broadcasts the query messages and identifies the tags based on the reply messages from the tags. Since the tags typically reply over the shared wireless medium and multiple tags can reply simultaneously to the reader, tag collision may occur at the reader. As a result, the reader experiences low tag reading performance, leading to a low system reliability, where the system reliability is a major challenge in RFID systems.

To resolve this collision problem and to successfully identify all the tags in RFID systems, many tag anti-collision protocols have been proposed. Generally, the anti-collision protocols are categorized into two classes: ALOHA-based and tree-based protocols. In ALOHA-based protocols, such as Dynamic framed-slotted ALOHA (DFSA) [1], Enhanced DFSA (EDFSA) [2], and MFML-DFSA [11], each tag defers for some random time before replying. This reduces the probability of reply packet collision. On the other hand, tree-based protocols continuously split the set of tags into two subsets each time a collision occurs. For the splitting, the binary tree (BT) protocol [3] uses a random number while the query tree (QT) protocol [4] uses tag IDs. Collision tree (CT) protocol [5] enhances QT by using the Manchester code and enables the reader to detect the first collided bit [1]. In [12], the authors propose a new query tracking tree protocol that splits the tags into smaller sets and reduces collisions based on the locations of collided bits.

Unfortunately, these solutions may not work properly in real wireless communication environments since they do not consider the *capture effect*. Recent measurement studies [6][7] in RFID systems have shown that the collided frames are not necessarily corrupted. One of the received frames can be decoded successfully at the reader if the signal to interference and noise ratio (SINR) of the frame is sufficiently high enough. Under this capture effect, the reader that employs the anti-collision protocol captures a single tag ID in the midst of a collision. In turn, the reader mistakenly regards this as a successful slot and leaves the other collided tags unidentified [8].

Some approaches have been recently proposed [8][9] to deal with this problem caused by the capture effect. The Generalized Query Tree (GQT) protocol [8] broadcasts the query message after each success slot to check whether there are some unidentified tags due to the capture effect. The General Binary Tree (GBT) protocol [9] improves GQT by separating the overall identification process into several cycles. Although these solutions provide useful insights on the proper implementation of anti-collision protocols toward the capture effect problem, none of these algorithms gives a complete solution. Since the query message is sent after every success slot, GQT obviously causes a large number of idle slots, especially when the frequency of the capture effect is low. In GBT, the multiple cycle operation reduces the waste of idle slots. However, since GBT uses a binary random number which is generated by each tag to avoid collisions in the next responses, the tree splitting occasionally fails and then GBT can also suffer from the waste of slots.

In this paper, we propose an efficient tag anti-collision protocol, called Multi-Round Collision Tree (MRCT). MRCT works efficiently even under the influence of capture effect and takes less overhead in the tag identification compared to the state-of-the-art protocol [9]. We analyze the number of slots needed for identifying tags and then evaluate the performance of MRCT and CT protocols through simulations. We consider a practical environment where the channel is imperfect and the capture effect occurs. The analytic and simulation results show that MRCT outperforms the CT and GBT protocols in various scenarios.

The remainder of the paper is organized as follows. In Section 2, we review the previous work. In Section 3, we explain the CT protocol and its limitations. We propose a new RFID anti-collision algorithm, MRCT in Section 4. Section 5 analyzes the performance of the CT protocol and the MRCT protocol. In Section 6, we provide the results obtained from the computer simulations. We conclude with a discussion of our results.

2. Related Work

The RFID reader may use the Manchester code to identify the bit where the collision occurred. In Manchester code, a bit is represented by the transition of the level; 0 is represented by a positive transition and 1 by a negative transition. When the positive transition and negative transitions are received at the same time, the receiver can identify the collided bits, as shown in **Fig. 1**.

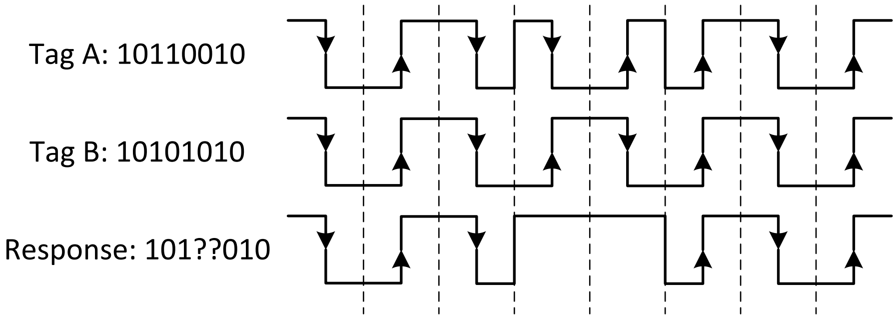


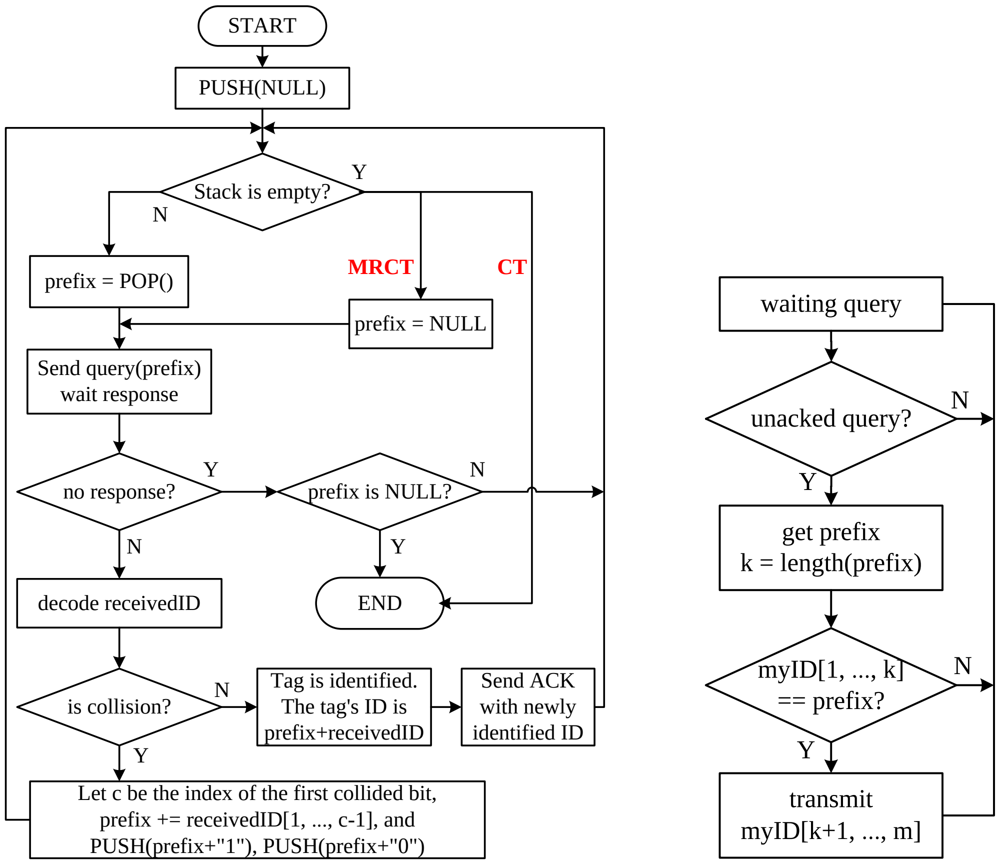
Fig. 1. Collision behaviors for Manchester code.

The Collision Tree (CT) protocol [5] improves the Query Tree (QT) protocol [4] by using the Manchester code. The QT protocol uses the prefix of the tag ID as a query message. When a collision occurs, the length of the prefix is increased by one bit. The CT protocol is similar to QT, but it increases the length of the prefix up to the first collided bit, not just one bit. Hence, CT reduces the number queries needed for identification.

However, CT does not consider the capture effect like most previous RFID anti-collision protocols such as QT or BT. Recently, some protocols such as the General Query Tree (GQT) [8] and the General Binary Tree (GBT) [9] consider the capture effect. GQT broadcasts the query message after each success slot to find the unidentified tags due to the capture effect. Since the query message is sent after every success slot, it obviously causes a large overhead. In particular, many query messages are sent in vain when the frequency of the capture effect is low. The GBT enhances the BT protocol to take capture effect into account. GBT separates the overall identification process into several cycles, as opposed to BT's single cycle procedure. GBT is shown to outperform GQT, since the multiple cycle operation reduces the waste of idle slots. However, the main drawback of GBT is its inefficiency of splitting when a collision occurs. In more specific, GBT uses a binary random number which is generated by each tag to avoid collisions in the next responses, so that the tree splitting occasionally fails.

3. Collision Tree Protocol and its Limitation

The CT protocol enhances the QT protocol by using the Manchester code which is used to detect the collided bit. **Fig. 2** shows the flow diagram of the CT protocol. For the query $q_1q_2\ldots q_k$, suppose that two or more tags respond with either the prefix $p_1p_2\ldots p_{c\text{-}1}1$ or $p_1p_2\ldots p_{c\text{-}1}0$, where p_i, q_i $\in\{0, 1\}$. Then, by using the Manchester code, the reader can distinguish the first collided bit, i.e., the c-th bit is identified as the first collided bit, and the reader uses $q_1q_2\ldots q_kp_1p_2\ldots p_{c\text{-}1}0$ and $q_1q_2\ldots q_kp_1p_2\ldots p_{c\text{-}1}1$ as new prefixes of next queries. That is, the CT protocol reader extends its prefix by c-bits. On the other hand, the reader of QT always queries for 1-bit longer prefix. That is the reason why CT protocol shows better performance than QT protocol.



(a) Reader (b) Tag

Fig. 2. Flow diagram of CT and MRCT

However, the CT protocol still may not identify some tags when the capture effect occurs. **Fig. 3** shows an example of the identifying process using CT protocol to identify five tags, which have 0001, 0010, 0011, 1110, 1111 as tag IDs. We assume that the two tags, 0010 and 1111, are located relatively close to the reader. Therefore, they are likely to be decoded even when a collision occurs, since the capture effect occurs at the reader. For example, when the two tags 0010 and 0011 collide, the response signal from 0010 captures the response signal from 0011. So the reader regards this as a sucessful query/reply and leaves 0011 undetected. This is the same for collision of tags 1111 and 1110. In summary, 3 tags are identified with 5 queries and the two tags 0011 and 1110 are left unidentified.

It is worthwhile noting that if the capture effect did *not* occur, then the reader should have detected the collision, and splitted the collision tree further more. In turn, the reader correctly detects all the tags.

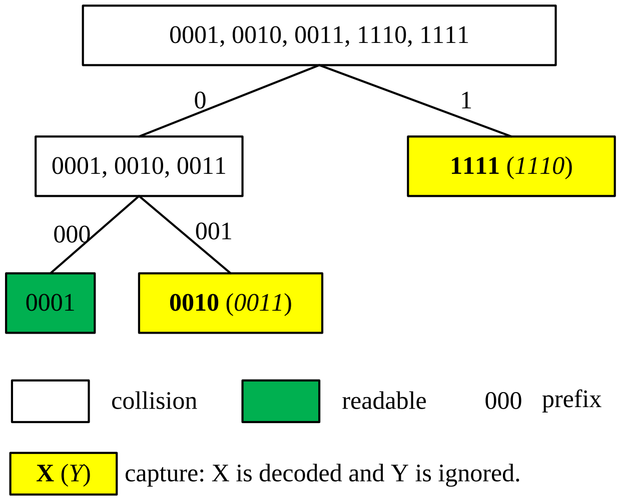


Fig. 3. Identifying process example for CT

4. Multi-round Collision Tree Protocol

In CT, when the reader captures a signal, it regards this as a successful query instead of a collision, leaving some of the tags unidentified. The MRCT protocol takes multiple rounds to discover these disregarded weak-signal tags. The first round of MRCT is very similar to CT. However, when the stack, i.e., a prefix pool of the next query, of the reader becomes empty, MRCT starts a new identification round with a NULL prefix, whereas CT just finishes its operation. The reason why we start a new identification round is to check whether there are unidentified tags due to the capture effect. If there are no responses from the tags in the new round, MRCT finalizes the tag identification process.

Note that the already identified tags do not transmit any response messages to the query. Once the MRCT reader receives a reply from a tag successfully, it transmits an ACK message to the identified tag. The tag that receives the ACK does not respond to the following queries until all the tags are identified. **Fig. 2** shows the flow diagram of the proposed MRCT protocol.

In addition, multi-round identification makes the protocol robust to the channel errors. When the wireless channel is imperfect, the queries from the reader may fail to reach some tags or the responds from the tags may fail to reach the reader. For the tree-based protocol, when these occur, the failed tag may not have the chance to respond again. MRCT provides those tags additional chances to respond.

Fig. 4 shows an example of the identifying process using MRCT protocol to identify five tags, which have 0001, 0010, 0011, 1110, 1111 as tag IDs. We assume that the two tags, 0010 and 1111, are located close to the reader. Therefore, they are likely to be decoded even when a collision occurs because of the capture effect at the reader. In this scenario, it takes 3 rounds with 9 queries to identify all 5 tags. In the first round, 3 tags are identified with 5 queries. Two tags 0011 and 1110 are not identified due to collisions. This part is similar to the CT protocol.

However, those unidentified tags from the first round are identified in the second round with 3 queries. In the third round, one query is used to ascertain that there are no unidentified tags and finally to terminate the identification. On the other hand, CT finishes its tag identification after the first round and thus 2 tags (0011 and 1110) are left unidentified which are hidden by the capture effect.

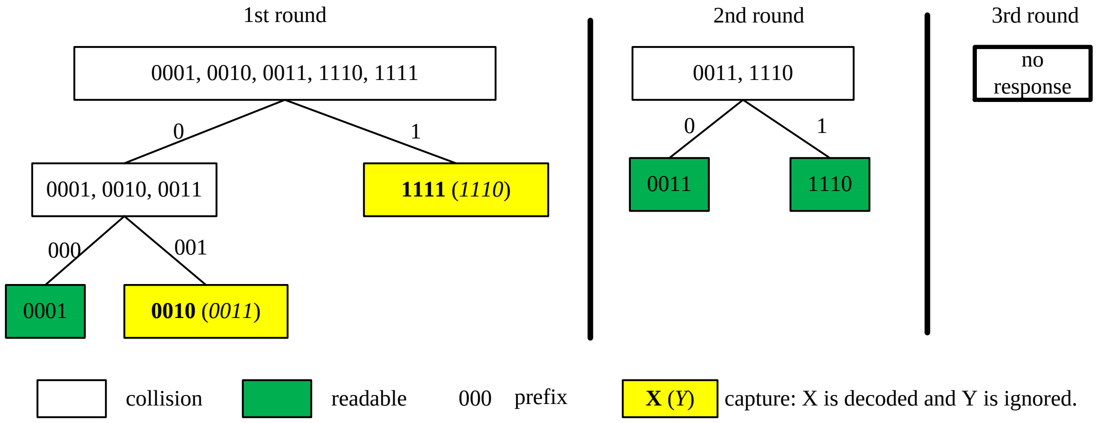


Fig. 4. Identifying process example for MRCT

5. Mathematical Analysis

In this section, we analyze the MRCT protocol in terms of the number of slots for identifying the tags under the channel condition in which the capture effect may occur.

5.1 Collision Tree Protocol under Capture Effect

First, we analyze the CT protocol under the channel condition in which the capture effect may occur. It is shown that the CT protocol requires (2n-1) slots for identifying n tags if the capture effect does not occur [5]. However, the CT protocol suffers from the capture effect, i.e., the CT protocol does not identify some tags when the capture effect occurs. We assume that the capture effect occurs at the receiver with a probability α, where α is constant and independent of the number of tags [9]. Here, we assume that the channel error does not occur to focus on the capture effect.

A collision tree of the CT protocol, as presented in **Fig. 3**, is a finite set of nodes that is either empty or consists of a root and two disjoint collision trees called the left and right subtrees. The subtree is still a collision tree. It is easy that a collision tree is a full binary tree, in which each node has either no children or two children. This can be explained in a recursive way. All n tags will respond to the first query with a NULL prefix. If there is no collision, i.e., only one tag responds to the query or the capture effect occurs, one tag is identified and then the identification process finishes since the stack for the next query becomes empty; if there is collision between responses, the root node has two children (or subtrees). Once detecting collision, the CT protocol splits tags into two groups. All internal nodes in the collision tree are corresponding to collision nodes, as shown in **Fig. 3**.

It is important that there is at least one tag in every subtree because the CT protocol uses tag IDs for splitting. This is different from the BT protocol [3] which uses a randomly generated bit value for splitting tags, and it can produce an empty subgroup. As a result, there are no idle nodes in a collision tree and all leaf nodes correspond to readable or capture nodes. Exactly one tag is identified in each leaf node, regardless of whether it is a readable or a capture node. Therefore, the number of leaf nodes is equivalent to the number of tags that the CT protocol identifies. If k is the number of leaf nodes and N is the total number of nodes, then $N = 2k - 1$ [5]. Let $N_{CT}(k\,|\,n)$ denote the number of slots required for identifying k out of n tags with the CT protocol where $1 \le k \le n$ .Then, we have,

$$N_{CT}(k\,|\,n) = 2k - 1. \tag{1}$$

We observe that $N_{CT}(3\,|\,5) = 5$ in **Fig. 3** where three out of five tags are identified and it takes five slots.

Let $P_{CT}(k\,|\,n)$ denote the probability that k out of n tags are identified with the CT protocol where $1 \le k \le n$. It is clear that $P_{CT}(1\,|\,1) = 1$ since there are no contending nodes. For $n \ge 2$, if the first query with a NULL prefix succeeds due to the capture effect, one tag is identified and the identification process finishes. If the first query fails due to collision, the collision tree will have at least two leaf nodes and hence more than one tag shall be identified. Therefore, we have $P_{CT}(1\,|\,n) = \alpha \ (n \ge 2)$.

Suppose that after the first query fails, n tags are divided into two subsets: n_1 and $(n\text{-}n_1)$ tags where $1 \le n_1 \le n$. Let $P_{sp}(n, n_1)$ denote the probability that n tags are divided into two subsets of n_1 tags and $(n\text{-}n_1)$ tags. We assume that tag IDs are uniformly distributed and then we have,

$$P_{sp}(n,n_1) = \binom{n}{n_1} \Big/ \sum_{i=1}^{n-1}\binom{n}{i} = \binom{n}{n_1} \Big/ \left(2^n - 2\right). \tag{2}$$

Because the root node has two subtrees and they are disjoint, the total number of leaf nodes can be obtained from the sum of the number of leaf nodes of two subtrees. Therefore, $P_{CT}(k\,|\,n)$ $(2 \le k \le n)$ can be obtained with the recursive equation:

$$P_{CT}(k\,|\,n) = (1-\alpha)\sum_{n_1=1}^{n-1} P_{sp}(n,n_1) \sum_{k_1+k_2=k} P_{CT}(k_1\,|\,n_1) P_{CT}(k_2\,|\,n-n_1) \quad (k \ge 2). \tag{3}$$

$P_{CT}(k\,|\,n)$ is recursively derived. It is trivial that,

$$\sum_{k=1}^{n} P_{CT}(k\,|\,n) = 1. \tag{4}$$

From $P_{CT}(k\,|\,n)$, we compute how many tags are unidentified with CT protocol. Let $R_{CT}(n)$ be the average tag identification ratio with CT protocol when there are n tags to be identified. Then, we have,

$$R_{CT}(n) = \frac{1}{n} \cdot \sum_{k=1}^{n} k \cdot P_{CT}(k\,|\,n). \tag{5}$$

There are two extreme cases: (i) when the capture effect never occurs (i.e., α=0), we have $P_{CT}(n\,|\,n) = 1$ and $R_{CT}(n) = 1$; (ii) when the capture effect always occurs (i.e., α=1), we have $P_{CT}(1|n)=1$ and $R_{CT}(n)=\frac{1}{n}$ since the first query with NULL prefix must be successful due to the capture effect.

5.2 MRCT Protocol under Capture Effect

The behavior of MRCT protocol in each round is same with that of the CT protocol. Let $N_{MRCT}(n)$ denote the average number of slots for identifying *n* tags with the MRCT protocol. If *k* out of *n* tags are identified in the first round, the remaining (*n*-*k*) tags will be identified in next rounds. Thus, the average number of slots for identifying *n* tags, $N_{MRCT}(n)$, can be written by Eq. (6).

$$N_{MRCT}(n)=\sum_{k=0}^{n}P_{CT}(k\,|\,n)\left(N_{CT}(k\,|\,n)+N_{MRCT}(n-k)\right). \tag{6}$$

We note that $N_{MRCT}(0)=1$ since one query with a NULL prefix is needed to ascertain that there are no unidentified tags at the final round.

Finally, applying Eq. (1) and Eq. (3) to Eq. (6) we obtain the average number of slots for identifying *n* tags with MRCT protocol, $N_{MRCT}(n)$.

There are two extreme cases: (i) when the capture effect never occurs (i.e., α=0), we have $P_{CT}(n|n)=1$ and $N_{MRCT}(n)=N_{CT}(n)+1$; (ii) when the capture effect always occurs (i.e., α=1), we have $P_{CT}(1|n)=1$ and $N_{MRCT}(n)=n+1$.

Also, we compute $N_{MRCT}(n)$ in terms of the number of rounds. Let *r* denote the number of rounds for MRCT protocol to finish the identification process. Let N_i and k_i denote the number of slots and the number of identified tags in the *i*-th round, respectively. Since one slot is needed in the final round, we have

$$\begin{aligned}N_{MRCT}(n)&=\sum_{i=1}^{r}N_i=\sum_{i=1}^{r-1}(2k_i-1)+1\\&=2\sum_{i=1}^{r-1}k_i-(r-1)+1=2n-r+2\end{aligned}. \tag{7}$$

MRCT protocol starts a new identification round with a NULL prefix to check whether there are unidentified tags due to the capture effect. MRCT protocol repeats the identification rounds until there are no responses from the tags in the new round. Hence if the channel error does not occur, MRCT protocol identifies all the tags and we have,

$$R_{MRCT}(n)=1. \tag{8}$$

We observe that the five tags are completely identified in three rounds in **Fig. 4**. A total of nine slots are used, that is, five slots in the first round, three slots in the second round, and one slot in the final round. It is well matched with Eq. (7).

6. Simulation Results

We compare the performance of MRCT and the state-of-the-art capture-aware protocol, GBT [9] for various channel conditions with different capture probabilities. To make a fair comparison with GBT, we give the same assumption that the capture effect occurs at the receiver with a probability α, where α is constant and independent of the number of tags.

The simulation setup is as follows. There are one reader and *n* tags. Each tag has a 96-bit ID, which is uniquely and randomly chosen from a uniform distribution. We run the simulation for each case 100 times and use their average results.

We first examine the average number of unidentified tags by the CT protocol, the GBT protocol, and the proposed MRCT protocol as a function of α. **Fig. 5** shows the results obtained from both the simulation and analysis. We observe that the MRCT protocol and the GBT protocol successfuly identify all the tags, but the CT protocol does not. MRCT and GBT take additional cycles to make sure there are no unidentfied tags due to capture effect.

The analytic results from Eq. (5) closely match with the simulation results, showing that our analytical model is quite accurate. We observe that the CT protocol identifies all the tags when the capture effect does not occur, that is, when the probability of the capture effect, α, is 0. However, the tag identification ratio decreases as α increases. At the extreme case where the capture effect always occurs (i.e., $\alpha = 1$), only one tag is identified by the CT protocol.

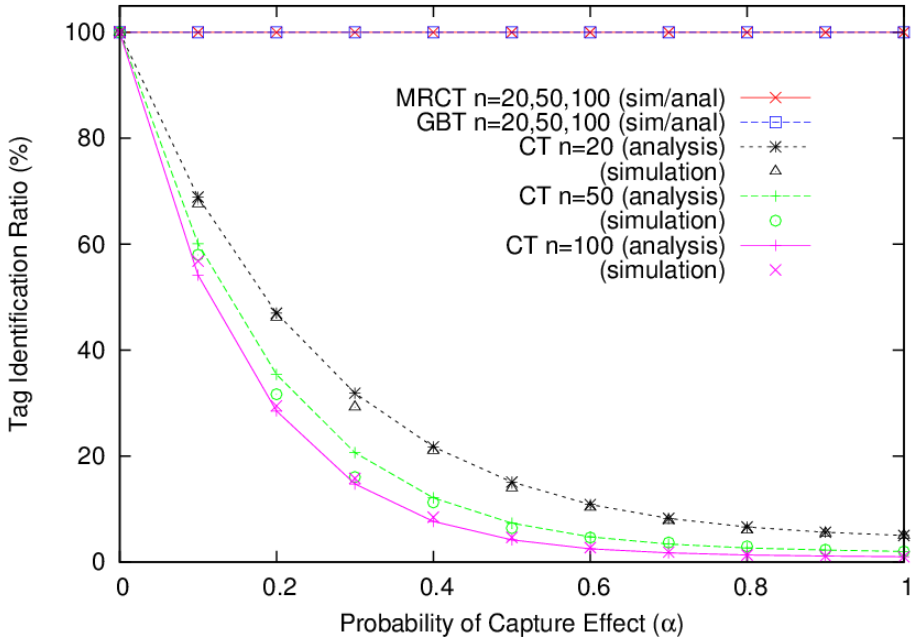


Fig. 5. Average tag identification ratio under the capture effect

Next, we examine the average number of slots taken for each protocol to identify 100 tags (**Fig. 6**). The results validate that the analytic model from Eq. (6) is very accurate. We observe that the number of slots for the MRCT protocol and the GBT protocol decreases as α increases. This implies that both protocols become effective as the capture probability increases. Although the CT protocol takes a smaller number of slots, it fails to identify some of the tags when the capture effect occurs as shown in **Fig. 5**.

Fig. 7 shows the identification efficiency which is the ratio of the number of tags over the number of the slots used [10]. We observe that the MRCT protocol and the CT protocol show similar efficiency, but the GBT protocol is less efficient when the probability of capture effect is low.

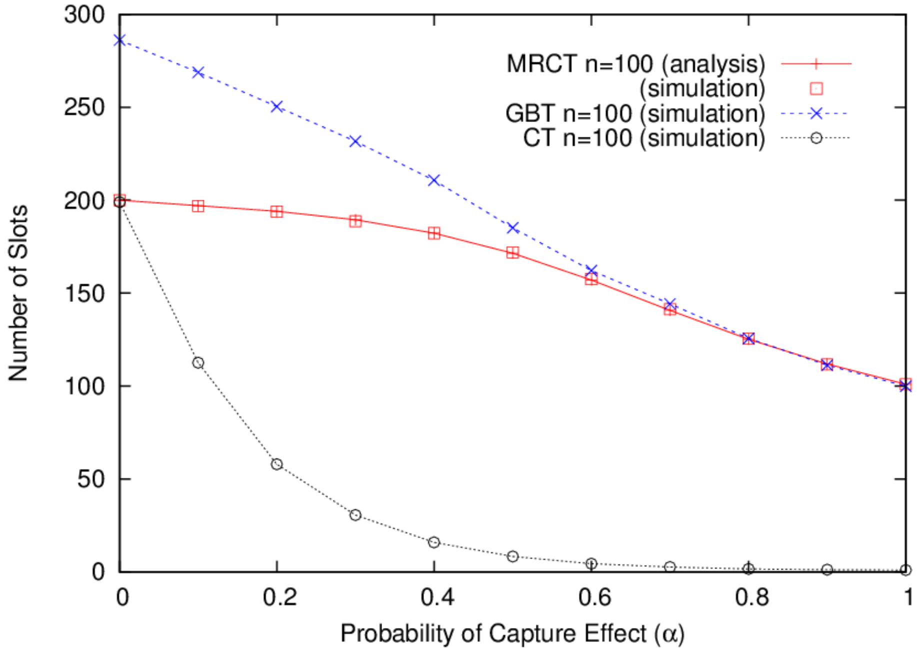


Fig. 6. Average number of slots under the capture effect (n = 100)

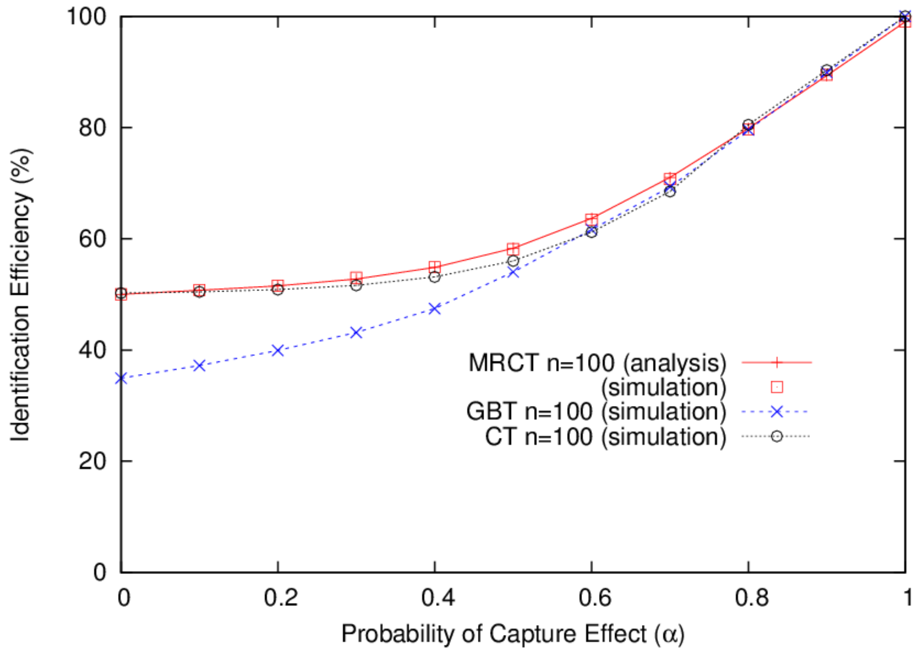


Fig. 7. Identification efficiency under the capture effect (n = 100)

Note that MRCT significantly outperforms GBT, especially when the probability of capture effect is relatively low. When a collision occurs, each subset after a split has at least one tag in MRCT. On the other hand, GBT may have a subset with no tags because it splits the tags by using a random number generation. Then, each subset with no tags will result in an empty (no response) slot and be wasted for GBT while MRCT has only one empty slot in the final round. The performance enhancement of MRCT over GBT results from this property. Therefore, the performance gap between MRCT and GBT is significant when the probability of collision in GBT is high, that is, when the probability of capture effect is low.

Next, we examine the effect of the channel error. When the wireless channel is imperfect, the query from the reader may fail to reach some of the tags or vice versa. For the tree-based protocol, when a channel error occurs, the failed tag may be able to respond again. **Fig. 8** shows the results obtained from the simulation with two different capture effect probabilities, α = 0 and 0.1. We assume that the packet error probability is independent and constant. We observe that the identification efficiency clearly decreases as the channel error increases. In particular, the CT protocol suffers from much significant degradation than the MRCT protocol and the GBT protocol. The reason is that multi-round identification protocol such as MRCT and GBT provides the failed tags additional chances to respond, while CT does not. It makes the MRCT and GBT protocol more robust to the channel errors.

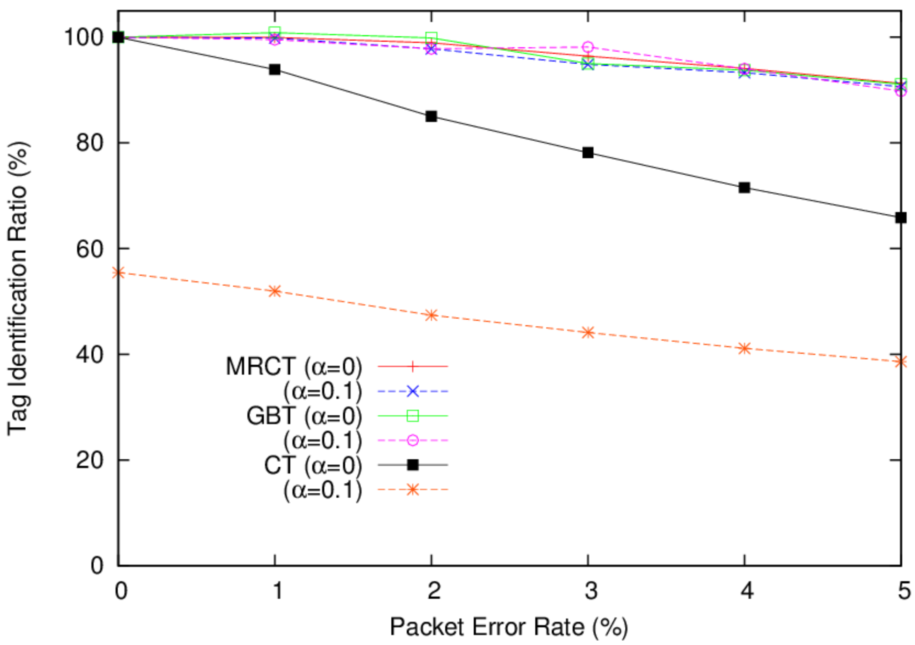


Fig. 8. Average tag identification ratio versus packet error rate (n = 100)

Until now, we have assumed that the capture effect occurs at a constant probability α for simplicity and fair comparison. We next examine the performance for the different number of tags in a more realistic scenario. We place a reader at the center and *n* tags in a circular disk of radius 4m, with their locations following a uniform distribution. We assume a two-ray ground propagation model where the received power falls off with distance raised to the fourth power. We model the capture effect using a parameterized signal-to-interference ratio (SIR) threshold, γ. That is, if the SIR is larger than γ then the capture effect occurs. The channel error does not occur.

Fig. 9 and **Fig. 10** show the results obtained from the simulation with various SIR thresholds. We observe that the MRCT protocol and the GBT protocol successfuly identify all the tags, but the CT protocol does not in **Fig. 9**. **Fig. 10** shows that MRCT requires much smaller number of slots compared with GBT. The CT protocol is ignored since it does not identify some tags when the capture effect occurs. We observe that about 25 to 30 % of slots can be reduced with MRCT from GBT in a random topology. This improvement gets higher as SIR threshold increases since the high SIR threshold lessens the capture effect. As mentioned before, GBT works inefficiently when the probability of the capture effect is low.

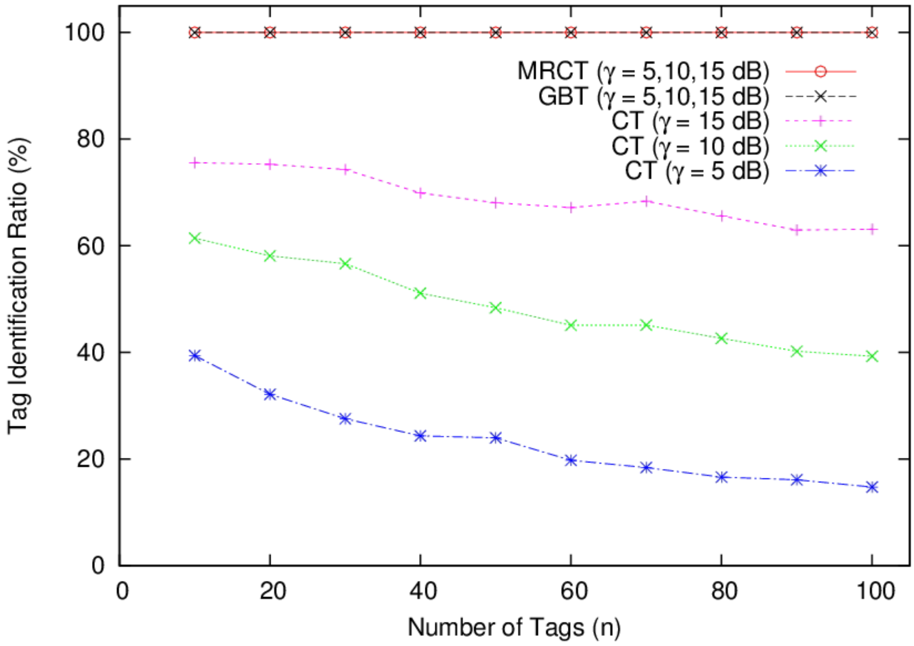


Fig. 9. Average tag identification ratio in a random topology

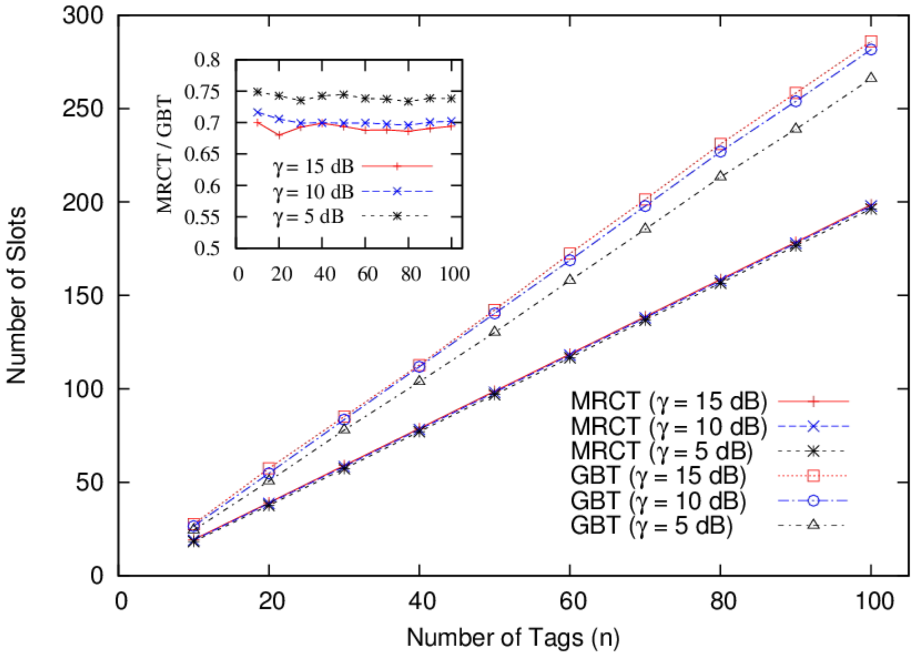


Fig. 10. Average number of slots for MRCT and GBT in a random topology

7. Conclusion

In this paper, we propose a new RFID tag identification protocol, called MRCT that efficiently copes with the capture effect problem. We analyze the number of slots required for identifying all the tags with MRCT. Analytic and simulation results show that MRCT significantly outperforms the existing protocol, GBT, especially in a practical environment where the probability of the capture effect is fairly low.

References

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, pp. 206–219. John Wiley & Sons, 2003. Article (CrossRef Link)

[2] S. Lee, S. Joo, and C. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification," in *Proc. of International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp. 166–172, 2005. Article (CrossRef Link)

[3] D. R. Hush and C. Wood, "Analysis of Tree Algorithms for RFID arbitration," in *Proc. of IEEE International Symposium on Information Theory*, pp. 107–107, 1998. Article (CrossRef Link)

[4] C. Law, K. Lee, and K. Siu, "Efficient Memoryless Protocol for Tag Identification," in *Proc. of Fourth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM'00)*, pp. 75-84, 2000. Article (CrossRef Link)

[5] X. Jia, Q. Feng, and C. Ma, "An Efficient Anti-Collision Protocol for RFID Tag Identification," *IEEE Communications Letters*, vol.14, no.11, pp. 1014-1016, 2010. Article (CrossRef Link)

[6] Y. Maguire and R. Pappu, "An optimal Q-Algorithm for the ISO 18000-6C RFID Protocol," *IEEE Trans. Automation Science and Engineering*, vol. 6, no. 1, pp. 16-24, Jan. 2009. Article (CrossRef Link)

[7] C. Floerkemeier and M. Lampe, "Issues with RFID usage in ubiquitous computing applications," *Lecture Notes in Computer Science*, vol. 3001, pp. 183-193, 2004. Article (CrossRef Link)

[8] V. Wu and R. Campbell, "Using Generalized Query Tree to cope with the Capture Effect in RFID Singulation," in *Proc. IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 1-5, Jan. 2009. Article (CrossRef Link)

[9] Y. Lai and L. Hsiao, "General Binary Tree Protocol for Coping with the Capture Effect in RFID Tag Identification," *IEEE Communications Letters*, vol.14, no.3, pp. 208-210, 2010. Article (CrossRef Link)

[10] G. Bagnato, G. Maselli, C. Etrioli, and C. Vicari, "Performance analysis of anti-collision protocols for RFID systems," in *Proc. VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, Barcelona, Spain, pp. 1-5, Apr. 2009. Article (CrossRef Link)

[11] J. Vales-Alonso, V. Bueno-Delgado, E. Egea-Lopez, F. Gonzalez-Castano, and J. J. Alcaraz, "Multiframe Maximum-Likelihood Tag Estimation for RFID Anticollision Protocols," *IEEE Trans. Industrial Informatics*, vol. 7, no. 3, pp. 487–496, Aug. 2011. Article (CrossRef Link)

[12] Y. Lai, "A Novel Query Tree Protocol with Bit Tracking in RFID Tag Identification," published online in *IEEE Trans. Mobile Computing*, Article (CrossRef Link)

Sunwoong Choi received the BS and MS degrees in computer science and the PhD degree in electrical engineering and computer science from Seoul National University in 1998, 2000, and 2005, respectively. From 2005 to 2007, he was with Samsung Electronics Company, Ltd., Kyungki-do, Korea, where he contributed to the design and development of Mobile WiMAX system. Since February 2007, he has been with the School of Electrical Engineering, Kookmin University, Seoul, where he is currently an Associate Professor. His research interests include wireless networks, resource management, and performance evaluation.

Jaehyuk Choi received the PhD degree in electrical engineering and computer science from Seoul National University, Korea, in 2008. He is currently an assistant professor in the Department of Software Design & Management at Gachon University, Seongnam, Korea. From 2009 to 2011, he was a postdoctoral research fellow in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. He was a postdoctoral fellow in Brain Korea 21 at Seoul National University in 2008. His current research interests are in the areas of wireless/mobile networks with emphasis on wireless LAN/MAN/PAN, network management, next-generation mobile networks, cognitive radios, and wireless security.

Joon Yoo received his B.S. in Mechanical Engineering from KAIST, and Ph.D in Computer Science and Engineering from Seoul National University in 1997 and 2009, respectively. He worked as a research assistant professor at University of Seoul in 2009, and from 2009 to 2010, he worked as a postdoctoral researcher at the University of California, Los Angeles. From 2010 to 2012, he worked at Bell Labs, Alcatel-Lucent, Seoul, Korea as a Member of Technical Staff. He has been with the Department of Software Design & Management, Gachon University, Korea, as an assistant professor since 2012. His research interests include vehicular ad hoc networks, mesh networks, cognitive radio and IEEE 802.11 and 802.16.