

Research Article

A Fast Hybrid Strategy-Based RFID Tag Identification Protocol

Xinyan Wang 

Intelligent Science and Information Engineering College, Xi'an Peihua University, Xi'an 710125, China

Correspondence should be addressed to Xinyan Wang; 36404843@qq.com

Received 22 December 2020; Revised 9 January 2021; Accepted 19 January 2021; Published 3 February 2021

Academic Editor: Yuapeng Zhang

Copyright © 2021 Xinyan Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Tag collision is one of the critical problems in radiofrequency identification (RFID) technology which can be widely used to identify objects using tag attachment automatically. Through the transmission and reflection of wireless radiofrequency signals, noncontact identification is realized. However, when multiple tags respond to the reader simultaneously, a collision occurs, significantly degrading the identification performance of RFID systems. To tackle the tag collisions, we propose a fast hybrid strategy-based RFID anticollision (FHS-RAC) protocol. Based on the conventional query tree algorithm, the proposed FHS-RAC makes full use of collision bits and the total response bits to achieve the faster tag identification. Extensive simulations and experiments verify the feasibility and effectiveness of our proposed scheme.

1. Introduction

As the development of radiofrequency identification (RFID) technology, the RFID tag has become more and more inexpensive, facilitating a wide deployment of RFID systems. Meanwhile, the advent of low-cost RFID reader modules has made feasible the integration of RFID readers and mobile devices [1]. The direct adaptability of RFID to the demands of supply chains [2–4], object tracking and monitoring [5–7], traceability of patients, or the management of medication for elderly people are a number of applications that have, in recent years, increased the popularity of this technology. In these broader range of RFID-enabled applications, tag identification problem [8] is one of crucial issues, which is to allow the unread tagged objects appear in the interrogation area to be identified as fast as possible. Tag identification problem happens when unidentified tagged objects are moved in or the tagged objects are misplaced in the region of other readers. For example, in a shopping mall with thousands of tagged objects, a lot of new tagged commodities are transported into the warehouse or placed on the shelves every day. Often, RFID systems suffer from collisions when a reader identifies multiple tags or multiple readers contact with a tag simultaneously. As a result, tag reading failure occurs and tag reading efficiency degrades significantly. In order to solve the tag identification problem in a dense envi-

ronment, an anticollision mechanism is needed to coordinate the communication between the reader and tags, which is named the tag identification or anticollision algorithm.

To address tag collisions, a large number of anticollision protocols have been proposed, which can be divided into probabilistic [9–14], deterministic [15–18], and hybrid algorithms. The probabilistic algorithms are essentially based on the ALOHA protocol. The basic idea behind this kind of algorithm is that the reader estimates the number of tags according to the collision, success, and idle time slots first and then adjusts the frame length dynamically based on the estimation results. When the number of tags is equal to the number of time slots, the ALOHA-based algorithm can achieve the highest throughput rate of 36.8%. Ease of implementation is an advantage of the algorithm based on ALOHA. The randomness of the identification process and the starvation problem are the disadvantages of such algorithms. Deterministic algorithms are studied in articles [15–18]. The basic idea behind this algorithm is that first using bit identification and tracking technology to identify the collision bit accurately. The tags are then grouped according to the collision bits until there is only one tag in the group. One disadvantage of these protocols is that the length of probe command of such algorithm is not fixed; it will change with the change of the query prefix length during the tag identification process. In addition, query tree-based algorithm consumes more transmitted bits.

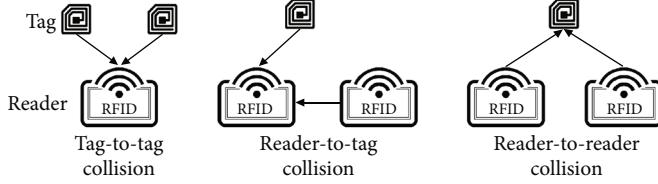


FIGURE 1: Three types of RFID collisions.

Combining the advantages of the ALOHA-based algorithm and the deterministic algorithm, the literatures [19, 20] proposed the hybrid-based algorithm. In the paper [20], based on the dynamic frame slot ALOHA algorithm, the binary tree algorithm is used to identify the collision slot. Compared with the ALOHA algorithm, the hybrid-based algorithm has a more considerable improvement in the throughput rate. However, the starvation problem is still to be solved, and its implementation complexity is higher than the ALOHA-based algorithm.

Considering the stable nature of deterministic algorithm, we propose a fast hybrid strategy-based RFID anticollision (FHS-RAC) with adaptive variable fork number. The proposed algorithm increases the number of probe forks based on the common query tree protocol, so that it can speed up the tag identification process. To avoid the problem that the reduced efficiency caused by the extra empty slots, the collision bits and the total response bits are used to estimate the tag cardinality, and the number of probe forks is adaptively adjusted according to the continuous situation of the highest collision bit. The proposed FHS-RAC can reduce the empty slots and quicken the convergence of query progress. Moreover, its efficiency will not be affected by the number of tags and tag ID distribution.

The remainder of this paper is organized as follows. In Section 2, we reviewed the collision types and the corresponding anticollision algorithms. Section 3 presents the system model and our proposed hybrid algorithm. In Section 4, mathematical analysis and performance evaluation through simulations is conducted. Section 5 concludes the whole paper.

2. Related Works

2.1. Static Tag Identification. To improve tag reading efficiency, many efforts have been made to solve RFID collision problems, which can be generally classified into three categories: tag-to-tag collisions, reader-to-tag collisions, and reader-to-reader collisions, as illustrated in Figure 1. Specifically, when one reader attempts to identify multiple tags, the tag responses reach the reader simultaneously and collide with each other (i.e., tag-to-tag collisions); when multiple readers try to read a tag, the reading request sent by one reader collide with the tag response to another reader (i.e., reader-to-tag collisions) or the signals transmitted by multiple readers may collide at the tag (i.e., reader-to-reader collisions).

Numerous tag anticollision algorithms have been proposed to reduce tag-to-tag collisions. Generally, there are two types of tag anticollision algorithms: ALOHA-based

algorithms and tree-based algorithms. The basic idea of ALOHA-based algorithms is that a tag waits for a random period of time and then retransmits data when collisions are detected. MAP [10] is an improved ALOHA-based algorithm for reducing tag reading collisions in the Internet of Things. Such an algorithm performs well, especially when the number of tags is small. In [14], an enhanced ALOHA-based algorithm has been presented to handle tag-to-tag collisions by querying the collided slots and then using different methods to adjust the frame length for collision resolution. However, because ALOHA-based algorithms randomly schedule data transmissions, some tags cannot be identified for a long time, causing the well-known tag starvation problem [16]. To address this problem, tree-based algorithms have been proposed and extensively studied [8, 15–17]. The principle of tree-based algorithms is to recursively divide tags into smaller subsets until each subset has only one or zero tags. In [8], an efficient collision arbitration algorithm, based on the binary tree algorithm, was designed to improve the performance of RFID systems through empty slot skipping. GBSA [17] is a tree-based collision arbitration strategy that integrates an efficient tag cardinality estimation method, an optimal grouping strategy, and a modified binary splitting. Although tree-based algorithms can eliminate tag starvation in dense tag scenarios, they often require lots of interactive processes, resulting in low reading efficiency [18].

To minimize reader-to-reader and reader-to-tag collisions, many reader anticollision protocols have been designed. Basically, they can be classified into two categories: distributed protocols and centralized protocols. As a distributed protocol, DCS [21] employs time division multiple access (TDMA) and divides the time into frames, each containing several time slots (colors) for tag readings. On top of this framework, an algorithm was designed to ensure different readers to choose different colors for collision avoidance. PDCS [22] is an enhanced version of DCS by enabling two collided readers to reselect their colors with a certain probability if they happen to choose the same color. EDCS [23], another enhancement of DCS, can increase the number of contemporary readers by allowing collided readers to complete tag identification in prior time slots. The above algorithms assume static or indoor scenarios and do not consider readers' mobility. Thus, they are not suitable for mobile tag identification.

2.2. Mobile Tag Identification. Recently, some research has been conducted to ensure tag reading performance in mobile environments. In [20], the authors proposed an anticollision protocol, called dynamic collision tree protocol, to identify many RFID tags that dynamically move into and leave the

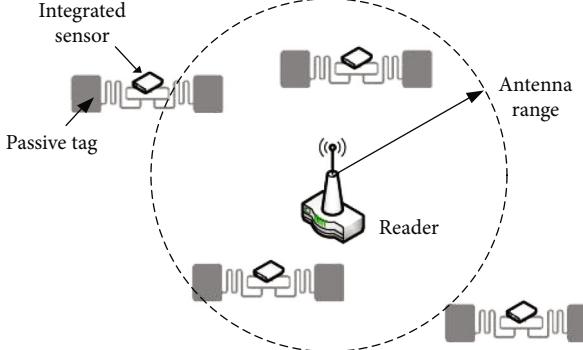


FIGURE 2: A scenario of RFID sensor network.

reading area in mobile RFID systems. MTIP [24] presents a feasible mobile tag estimation model, in which the superposition principle of Poisson process is used to solve the tag estimation problem. Moreover, an anticollision algorithm was proposed for identifying mobile RFID tags. To accelerate the reading process of large-scale mobile RFID systems, EBD [25] was designed to rapidly and efficiently distinguish and identify the unknown tags (i.e., not previously recognized tags) from the known tags while they are passing a reader's range. All these schemes focus on the situations in which many tags move through the reading area of a reader; thus, they can only handle tag-to-tag collisions. In contrast, this paper targets road environments where multiple vehicles (readers) pass a series of nonoverlapping tags sequentially. Hence, reader-to-tag and reader-to-reader collisions are the major issues to be addressed. Also, onboard tag readings are more challenging due to higher mobility and lower reliability.

3. System Model and Algorithm Description

3.1. System Model. A standard UHF RFID system, as shown in Figure 2, consists of at least one interrogator (reader) and a minimum of one transponder (tag). Such platforms communicate in the industrial, scientific, and medical (ISM) band around 900 MHz using the EPC Class 1 Generation 2 (EPC-C1G2) communication standard. Depending on the region of operation, the frequency bands and regulations differ (e.g., EU...865-868 MHz; NA...902-928 MHz) [2]. UHF RFID tags can be grouped depending on the technology incorporated. In this paper, we consider a large-scale RFID system with a single reader and a large number of EPC-C1G2 tags. To read a tag, the reader first transmits a reading request (RREQ). Then, the tag within the reading range collects and stores the energy carried by the signals. Once the energy reaches a certain threshold, the tag sends out a reading reply (RREP). As mentioned in Section 2.1, such communication process may suffer from three types of collisions. In this paper, we exclusively focus on the tag-to-tag collisions.

3.2. Communication Slots. The identification of FHS-RAC algorithm general includes three types of time slots: (1) singleton slot: that is picked by exactly one tag; (2) collision slot:

that is picked by multiple tags; and (3) empty slot: that is picked by none of tags. The total number of slots required for a multifork tree is

$$\left\{ \begin{array}{l} t(m) = 1 + M \sum_{L=0}^{\infty} M^L K, \\ K = 1 - (1 - M^{-L})^m - m M^{-L} (1 - M^{-L})^{m-1}, \end{array} \right. \quad (1)$$

where M is the number of branches of the tree, m is the number of tags, and L is the current search depth. We know that when M takes different values, the total number of time slots consumed by the FHS-RAC is different. For example, when M is 3, the total number of time slots is the smallest, that is, the tag identification using 3-ary is the most efficient. However, it is impossible to construct a 3-ary in the practical identification process. In the actual process, the most common is to build 2-ary, 4-ary, or 8-ary traversal tree. Some algorithms use a 4-ary or 8-ary to reduce the collisions. However, the multtree will introduce a lot of extra empty time slots, it is necessary to prune the branch of multtree and thus to reduce the empty slots.

3.3. Algorithm Description. In this paper, the workflow of our proposed FHS-RAC algorithm can be described as follows. In FHS-RAC, the reader will calculate the collision factor and determine the current search depth and then adaptively adjust a 2-ary tree to a 4-ary tree or 8-ary tree to avoid the generation of overmuch empty slots, where the search depth is defined as the number of branches from the root to the leaf during the traversal tree, which is also the number of layer of the multifork tree. After dynamically selecting the appropriate M -ary tree, the collision bits are extracted to form a new query prefix. When receiving the query prefix, the tag compares its own ID with it and locks its own bits to reduce the amount of messages to be sent. In our proposed FHS-RAC, it is not necessary to increase the number of fork tree in order to reduce the collisions, because the number of empty slots also needs to be considered at the same time. Obviously, the higher the fork tree, the fewer collision slots and the more empty slots. Therefore, we need to adaptively adjust the fork tree to reduce collisions while limiting the increase of empty slots. In what follows, we describe the adjustment mechanism. In this paper, we define a variable called collision factor, which is expressed as

$$\mu = \frac{\alpha}{\beta}, \quad (2)$$

where α is the number of collision bits and β is the length of tag response bits in a collision slot. Suppose there are m tags to be identified in the RFID system that meet the query conditions. The probability that any bit does not collide is $1/2^{m-1}$. Then, the collision factor μ can be reexpressed as

$$\mu = \frac{1 - (1/2)^{m-1}}{\beta} = 1 - \left(\frac{1}{2}\right)^{m-1}. \quad (3)$$

From Equation (3), we know that the larger the number of concurrent tags, the higher the collision factor, and vice versa. Suppose the number of multtree assigned by the system is L , when the search depth is 1, the identification probability of a tag is

$$P_1 = \left(1 - \frac{1}{L}\right)^{m-1}, \quad (4)$$

when the search depth is k , the corresponding identification probability is

$$P_k = P_1(1 - P_1)^{k-1}. \quad (5)$$

The expected value of the search depth can be calculated as

$$E(k) = \sum_{k=1}^{\infty} k \times P_k = \sum_{k=1}^{\infty} k \times P_1 \times (1 - P_1)^{k-1}. \quad (6)$$

The above formula can be simplified as

$$E(k) = \frac{1}{(1 - (1/L))^{m-1}}. \quad (7)$$

Thus, the average number of time slots of a 2-ary tree is

$$T_2 = \frac{2}{(1 - 1/2)^{m-1}}. \quad (8)$$

Similarly, the average number of time slots of a 4-ary tree is

$$T_4 = \frac{4}{(1 - 1/4)^{m-1}}. \quad (9)$$

Comparing (2) to (4), we can get the following judgment conditions:

$$\begin{cases} T_2 \geq T_4, m \geq 3, \\ T_2 < T_4, m < 3. \end{cases} \quad (10)$$

Thus, by judging the threshold of the collision factor, the reader can adaptively adjust the multtree to resolve the collided tags. When the value of μ is equal to or greater than 0.75, a 4-ary tree is used. And when the value of μ is less than 0.75, a 2-ary tree is used. The collision factor is an indicator that reflects the number of tags in the collision slot. However, such indicator is not completely reliable because it is easily affected by the distribution of tag IDs. Therefore, we use the collision factor as the threshost and the highest collision bit to decide whether to adjust the fork tree. The workflow of the proposed FHS-RAC is described as follows:

- (S1) The reader maintains an empty query stack $\$S\$$. First, the reader broadcasts a request, and the tags

within the reader's coverage receive the request command and make the corresponding operations

- (S2) The reader determines the real-time status of slots according to the tag response. If only one tag replies, the reader successfully identifies the tag and jumps to S5; if no tag replies, the search is stopped in the branch, jump to S5; if more than one tag reply, the reader determines the tags collide and jump to S3
- (S3) The reader determines whether the highest collision bit is continuous. If the highest collision bit is not continuous, the reader directly selects the 2-ary tree. If the highest collision bit is continuous, the reader skips to S4
- (S4) The reader calculates the collision factor μ . If $\mu \leq 0.75$, the reader selects a 2-ary tree. If $\mu > 0.75$, the reader selects a 4-ary tree
- (S5) Judging whether the stack is empty, if not, the reader extracts the first prefix of the query stack to probe tags, and returns to S3; otherwise, the entire identification process ends

4. Performance Analysis and Numerical Results

4.1. Performance Analysis. We theoretically analyze the total number of slots required for our proposed FHS-RAC algorithm and then deduce the system efficiency. Specifically, the total number of slots can be obtained by summing the number of slots consumed by multtree. The average number of tags contained in each intermediate node is 3 in k th search depth. In the tag identification process, if the required traversal depth is higher than k , the reader should use an 2-ary tree; otherwise, it should use a 4-ary tree. The total number of slots required by FHS-RAC can be expressed as

$$T_{\text{total}} = T_4 + T_2, \quad (11)$$

where T_4 means the total number of slots taken by a full 4-ary tree, which is calculated as

$$T_4 = \sum_{i=0}^k 4^i = \sum_{i=0}^{\lceil \log_4^{M/3} \rceil - 1} 4^i. \quad (12)$$

Referring to the analysis in [18], the number of total slots ($T_{4-\text{total}}$), empty slots ($T_{4-\text{idle}}$), and collision slots ($T_{4-\text{coll}}$) can be calculated as

$$\begin{aligned} T_{4-\text{total}}(m) &= 1 + 4 \sum_{j=0}^{\infty} 4^j \left[1 - (1 - 4^{-j})^m - \frac{m}{4^j} (1 - 4^{-j})^{m-1} \right], \\ T_{4-\text{coll}} &= \frac{1}{4} (T_{4-\text{total}} - 1), \\ T_{4-\text{idle}} &= T_{4-\text{total}} - T_{4-\text{coll}} = \frac{3}{4} T_{4-\text{total}} - m + \frac{1}{4}, \end{aligned} \quad (13)$$

in which m denotes the number of tags involved in current slot and j represents the traversal depth. Referring to the existing literature [16], empty slots can be eliminated by the introduction of *QueryRP* command. Therefore, the total number of slots taken by FHS-RAC to identify n tags is expressed as

$$\begin{aligned} T_{\text{FHS-RAC}}(n) &= T_{4-\text{total}} + T_{2-\text{total}} - T_{4-\text{idle}} + T_{4-\text{coll}} \\ &\approx \sum_{j=0}^{\lceil \log_4^{n-s} \rceil - 1} 4^j + \frac{5}{3}(n-s) - 1.16(n-s) \\ &\quad + 0.72(n-s). \end{aligned} \quad (14)$$

Then, the system efficiency of FHS-RAC can be calculated as $n/T_{\text{FHS-RAC}}$.

4.2. Numerical Results. In this section, we evaluate the tag identification performance of our proposed algorithm and the existing algorithms DBSA, DFSA [10], and IACA [15]. We implement the proposed algorithm in MATLAB on a ThinkPad X1 Carbon desktop with an Intel 2.4 GHz CPU. Our simulation setting follows the specifications of the EPC C1 Gen2 standard. The number of tags is from 20 to 500 in step of 20. The reader-to-tag transmission rate and the tag-to-reader data rate are not symmetric, which depends on specific physical implementations and practical environments. We analyze the performance of algorithms by using three metrics: the average request cycle and the total length of transmitted data between the reader and all the tags and the energy efficiency of the tags.

During our simulations, we randomly generated 16-bit ID tags and the number of tags is set between 5 and 500 tags. The average request cycle can be computed as

$$N_{\text{average}} = \frac{N_{\text{total}}}{n}, \quad (15)$$

herein, N_{total} is the total number of the request cycles which is the sum of the number collision, identification, and empty cycles. We denote E_{tr} and E_{tt} as the power consumption of tag receiving queries and transmitting responses, respectively. Thus, the energy efficiency can be written as

$$\eta = \frac{nE_u}{QE_{tr} + RE_{tt}}, \quad (16)$$

where Q is the number of the queries and R is the total number of tag responses.

Figure 3 shows the average cycles for one-tag identification as the number of tags is increased. Our proposed scheme takes 2.25 cycles, whereas DBSA and IACA take 4.90 and 2.98 cycles, respectively. Note that Figure 4 depicts the total length of transmitted data of DBSA, IACA, DFSA, and our proposed algorithm. The length of transmitted data of our proposed algorithm is significantly less than that in the DBSA and IACA algorithms. As the number of tags is increasing, the advantages of the proposed scheme become more noticeable, especially in the case of a very large number of tags. For example, when the number of tags is 405, the DBSA and IACA need to send a total length about 253756

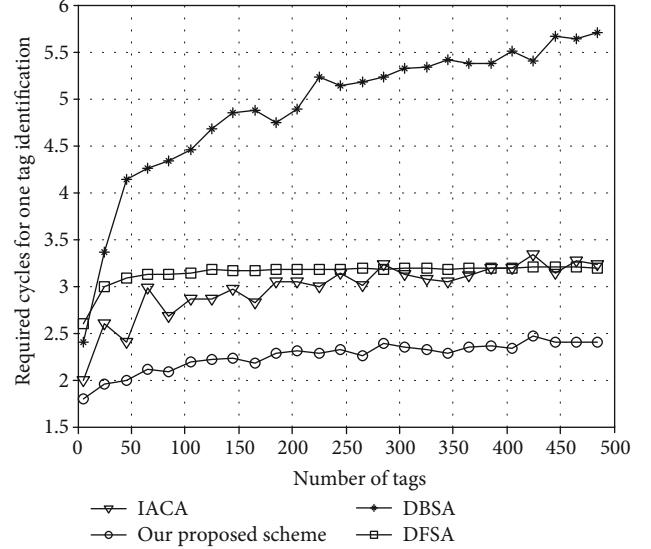


FIGURE 3: Required cycles for one tag identification.

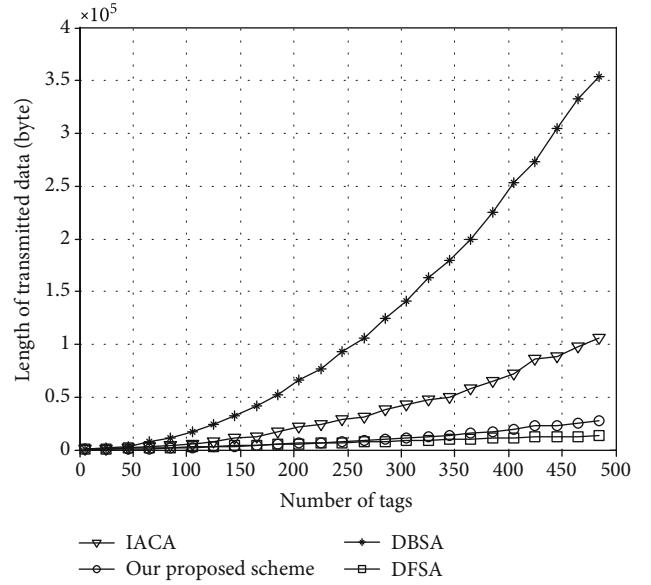


FIGURE 4: The length of transmitted data for all tag identifications.

and 77650 bytes, but our proposed algorithm needs to send only 19732 bytes. We found, in the results, the ALOHA-based algorithm outperforms tree-based algorithms in terms of the length of transmitted data.

Figure 5 shows the channel saving provided by our proposed algorithm compared to the DBSA and IACA, which are calculated by the following equation:

$$\begin{aligned} P_{\text{ChannelSaving}}^1 &= \left(1 - \frac{N_{\text{bits}_{\text{IACA}}}}{N_{\text{bits}_{\text{DBSA}}}} \right), \\ P_{\text{ChannelSaving}}^2 &= \left(1 - \frac{N_{\text{bits}_{\text{OurScheme}}}}{N_{\text{bits}_{\text{DBSA}}}} \right), \\ P_{\text{ChannelSaving}}^3 &= \left(1 - \frac{N_{\text{bits}_{\text{OurScheme}}}}{N_{\text{bits}_{\text{IACA}}}} \right). \end{aligned} \quad (17)$$

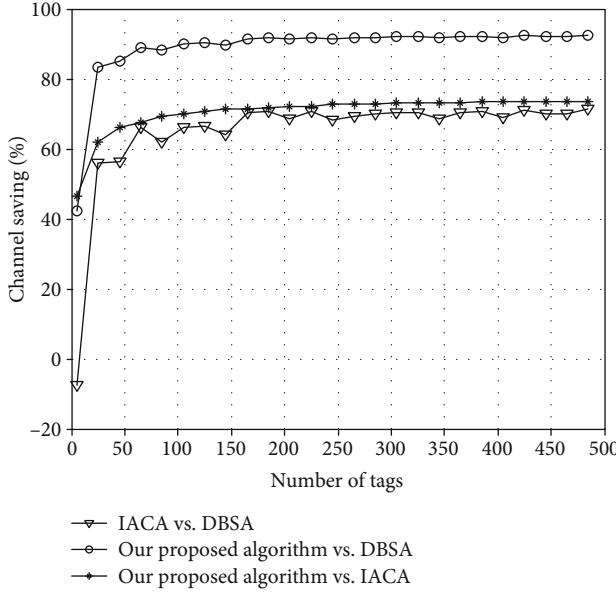


FIGURE 5: Channel saving under different algorithms.

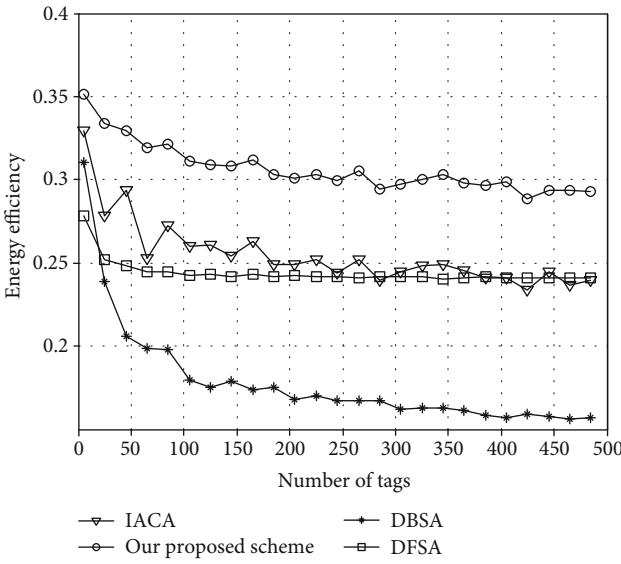


FIGURE 6: Comparison in energy efficiency.

From Figure 5, the results show that almost 70% channels are saved by our proposed algorithm than the IACA due to the use of additional request command's building and the mapping of first and second collision bits.

The comparison of energy efficiency is plotted in Figure 6. It is obvious that our proposed algorithm provides much higher energy efficiency than DBSA, IACA and DFSA.

5. Conclusion

This paper investigates the problem of tag collisions in the large RFID systems. We propose an efficient and complete a new tag identification algorithm, which is able to fast identify unread tags in a highly efficient and complete way. With

our algorithm, the request cycles and the length of transmitted data between the reader and tags are significantly reduced. The performance is achieved by the building of request command and the mapping of collision bits. Through simulation results, we conclude that our algorithm reduces time complexity, energy cost, and communication overhead in the identification process of multiple tags.

Data Availability

To be frank, I derived the writing material from different journals as provided in the references. A MATLAB tool has been utilized to simulate our concept.

Conflicts of Interest

The author declares that he has no conflicts of interest.

References

- [1] S. S. Park, "An IoT application service using mobile RFID technology," in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–4, Honolulu, HI, USA, January 2018.
- [2] C.-H. Lee and C.-W. Chung, "RFID data processing in supply chain management using a path encoding scheme," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 5, pp. 742–758, 2011.
- [3] J. Su, R. Xu, S. Yu, B. Wang, and J. Wang, "Idle slots skipped mechanism based tag identification algorithm with enhanced collision detection," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2294–2309, 2020.
- [4] J. Su, R. Xu, S. Yu, B. Wang, and J. Wang, "Redundant rule detection for software-defined networking," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 6, pp. 2735–2751, 2020.
- [5] L. Shangguan, Z. Zhouy, X. Zheng, L. Yang, Y. Liu, and J. Han, "Shopminer: mining customer shopping behavior in physical clothing stores with COTS RFID devices," in *Sensys '15: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 113–126, New York, NY, USA, November 2015.
- [6] L. Yang, J. Cao, W. Zhu, and S. Tang, "Accurate and efficient object tracking based on passive RFID," *IEEE Transactions on Mobile Computing*, vol. 14, no. 11, pp. 2188–2200, 2015.
- [7] H. Cai, L. D. Xu, B. Xu, C. Xie, S. Qin, and L. Jiang, "IoT-based configurable information service platform for product lifecycle management," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1558–1567, 2014.
- [8] J. Su, Z. Sheng, L. Xie, and G. Wen, "Idle-slots elimination based binary splitting anti-collision algorithm for RFID," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2394–2397, 2016.
- [9] L. Zhu and T. S. Yum, "A critical survey and analysis of RFID anti-collision mechanisms," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 214–221, 2011.
- [10] W. T. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 9–15, 2009.

- [11] J. Su, A. X. Liu, Z. Sheng, and Y. Chen, "A partitioning approach to RFID identification," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2160–2173, 2020.
- [12] J. Su, Z. Sheng, A. X. Liu, Z. Fu, and Y. Chen, "A time and energy saving based frame adjustment strategy (TES-FAS) tag identification algorithm for UHF RFID systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 2974–2986, 2020.
- [13] B. Li and J. Y. Wang, "Efficient anti-collision algorithm utilizing the capture effect for ISO 18000-6C RFID protocol," *IEEE Communications Letters*, vol. 15, no. 3, pp. 352–354, 2011.
- [14] H. Chen, K. Liu, C. Ma, Y. Han, and J. Su, "A novel time-aware frame adjustment strategy for RFID anti-collision," *Computers, Materials & Continua*, vol. 57, no. 2, pp. 195–204, 2018.
- [15] M. Djeddou, R. Khelladi, and M. Bensalah, "Improved RFID anti-collision algorithm," *AEU-International Journal of Electronics and Communications*, vol. 67, no. 3, pp. 256–262, 2013.
- [16] J. Su, Y. Chen, Z. Sheng, Z. Huang, and A. X. Liu, "From M-ary query to bit query: a new strategy for efficient large-scale RFID identification," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2381–2393, 2020.
- [17] J. Su, Z. Sheng, A. X. Liu, Y. Han, and Y. Chen, "A group-based binary splitting algorithm for UHF RFID anti-collision systems," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 998–1012, 2020.
- [18] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," in *Proceedings 1998 IEEE International Symposium on Information Theory (Cat. No.98CH36252)*, pp. 107–116, Cambridge, MA, USA, August 1998.
- [19] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary tree slotted ALOHA for passive RFID tag anti-collision," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 19–31, 2012.
- [20] X. Jia, M. Bolic, Y. Feng, and Y. Gu, "An efficient dynamic anti-collision protocol for mobile RFID tags identification," *IEEE Communications Letters*, vol. 23, no. 4, pp. 620–623, 2019.
- [21] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: a MAC for RFID reader networks," in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003*, pp. 1701–1704, New Orleans, LA, USA, March 2003.
- [22] F. Gandino, R. Ferrero, B. Montruccio, and M. Rebaudengo, "Introducing probability in RFID reader-to-reader anti-collision," in *2009 Eighth IEEE International Symposium on Network Computing and Applications*, pp. 250–257, Cambridge, MA, USA, July 2009.
- [23] N. Wang, G. Xie, and Y. Liu, "EDCS: an enhanced distributed color selection algorithm for anti-collision in RFID system," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1031–1035, Chengdu, China, December 2017.
- [24] Y. Chen and Q. Feng, "A collision avoidance identification algorithm for mobile RFID device," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 4, pp. 493–501, 2019.
- [25] L. Zhang, W. Xiang, and X. Tang, "An efficient bit-detecting protocol for continuous tag recognition in mobile RFID systems," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 503–516, 2018.