

The Bit Query for Labels in a Binary Tree-Based Anti-Collision Recognition Algorithm

Chenyao Sun

Department of Computer Systems Engineering, University of Glasgow, United Kingdom

E-mail: suncypro@163.com

(Received 10 November 2023; Revised 24 November 2023, Accepted 28 November 2023; Available online 4 December 2023)

Abstract - With its benefits including recognition distance, penetration ability, and multi-object recognition, radio frequency identification technology is a non-contact automatic identification method that is currently being used extensively in a variety of industries, including document anti-counterfeiting, automation, transportation, control management, and product services. An efficient anti-collision mechanism, such as anti-collision algorithms or anti-collision protocols, must be established in order to coordinate communication between the tag and the reader in the event that an RFID system experiences signal interference in the wireless channel, conflict, or collision, which will result in tag recognition or data collection failure. In response to tag conflicts in radio frequency identification (RFID) systems, a novel hybrid blocking query tree algorithm is suggested that leverages blocking prediction and child node hedging techniques to overcome the issues of lengthy time intervals and high communication complexity. The technique uses quadruple and bifurcated query trees as the foundation for its conflict avoidance strategy. The approach minimizes the conflict time and prevents the construction of idle subunits by forcing the reader to create new query prefixes via prediction, which is accomplished by employing locking commands to extract and forecast information about the conflicting bits. According to simulation data, this strategy performs better in terms of lowering the overall number of time slots and communication complexity as well as enhancing identification and labelling abilities than the current adaptive multi-tree search (RLAMS) and improved hybrid query tree (IHQT) techniques. It greatly increases the effectiveness of label identification while lowering the overall amount of time slots and communication complexity.

Keywords: Radio Frequency Identification, Bit Blocking, Query Trees, Conflict Avoidance Algorithms, Predictive Controls

I. INTRODUCTION

Radio frequency identification technology has advanced and grown in recent years due to the rapid development of wireless communications, microelectronics, embedded systems, signal processing, and other technologies. The size and power consumption of RFID systems are decreasing, as is their cost, and their functionality is becoming more and more flawless. The automation of social production is rapidly utilizing RFID technology, which is a type of technology that can automatically finish information collecting and processing. Utilizing RFID technology can improve supply chain management, lower expenses associated with production management, automate business processes, increase revenues for businesses, and improve people's

quality of life. RFID technology will be widely used and vigorously promoted once the cost of RFID system components decreases and a worldwide unified standard is achieved. Research on RFID technology has been relatively late in China; applications of RFID technology are still in the early stages of development; the market outlook is very broad; and the study of RFID technology has significant social and economic benefits.

The read-write, electronic label, and computer system components make up the three main components of a radio frequency recognition system (Su, 2020). The read-write and electronic label emit radio frequency signals from their respective antennas, and spatial coupling facilitates the transmission of data information to complete the non-contact automatic diagnosis. The early information systems, which relied on human labor to enter data, had several drawbacks, including sluggish data entry speed, a high rate of data error codes, and a high labor intensity from manual labor (Tang, 2019). These were overcome by radio frequency recognition technology. Furthermore, RFID technology has been widely applied in a variety of fields, including environmental monitoring, target detection and tracking, supply chain management, business automation, transportation control management, electronic access control, and other fields (Mu, 2021). It also boasts a large communication range, large data capacity, high accuracy, data security, and reliability, and does not require line-of-sight communication. Additionally, RFID technology can be used for multi-target identification. A reader and many inexpensive tags that are affixed to objects for tracking and communication make up a standard RFID system (Peng, 2023). The reader can recognise the tags in the RF field it creates; however, when two or more tags communicate the reader tag ID information at the same time, the reader will not be able to correctly identify each individual ID; this is known as a collision, which has a significant negative impact on recognition efficiency (Yaacob, 2019). Consequently, it is imperative that an ideal anti-collision algorithm be used to RFID systems (Kumar, 2021).

The three primary types of collision avoidance algorithms currently in use are hybrid, deterministic, and randomised algorithms. Random methods such as dynamic frame time slot are based on the ALOHA protocol, frame time slot ALOHA (Rasina Begum, 2023), and time slot ALOHA

(Munir, 2019). The basic idea is as follows: The reader counts the number of intervals inside the frame, and random methods are used to make the tags select the time slots as distributed as possible to send data to the reader. This makes the idea simple to implement but may result in long-term tag famine. Tree structure based algorithms (Besta, 2023) that avoid the tag hunger problem are examples of deterministic algorithms. These include the Q-ary search scheme (QAS) algorithm (Khalil, 2019), the Query tree (QT) algorithm (Alimohammadi, 2019), and Binary search (BS) (Gul, 2022). Although this class of algorithms is capable of tracking the precise site of a collision, there are issues with the lengthy identification time and high label energy usage. Hybrid algorithms, which include deterministic and stochastic algorithms, typically perform better, but they also require more sophisticated reader and tag designs and result in higher system energy loss (Akavia, 2022).

The QT algorithm (Ai, 2022) is the basis for this paper's two-way matching and segmented query algorithm, which matches from the high and low bits of the ID based on the prefix and suffix in the query command, respectively. Each reader query is directed toward two target tag groups, and up to four tags can be recognized in a single time slot. In addition, the idea of a segmentation point is put forth, which divides the length of the data that the tag sends in response to a query into two segments. The first and second segments of the query each return a portion of the data other than the prefix and suffix, resulting in a significant reduction in communication transmission and an improvement in performance. The results of matlab simulation tests confirm that, in comparison to standard anti-collision algorithms, the proposed Bi-direction segmented anti-collision algorithm (BDS) greatly decreases the number of queries, increases system throughput, and reduces data transmission.

II. LITERATURE REVIEW

RFID systems employ four primary techniques for preventing collisions: time division multiplexing (TDMA), frequency division multiplexing (FDMA), space division multiplexing (SDMA), and code division multiplexing (CDM). While the binary tree search algorithm (also known as binary tree-scanning) and the ALOHA algorithm are the most often used algorithms in RFID systems, the time division multiplexing approach is the one that is employed in RFID systems the most.

Binary tree-based collision avoidance systems (Wang, 2023) work on the basis of dividing colliding tags into subsets according to tag ID or randomly generated binary integers. This process is repeated until each set has exactly one tag that can be correctly identified. Binary Search Tree (BS) (Umelo, 2022), Dynamic Binary Search Tree (DBS) (Zhang, 2022), Regressive Binary Search Tree (RBS) [18], and so forth are examples of tree-based algorithms. A fast tree-based collision avoidance method (SDT) was proposed in study (Lai, 2022). SDT delivers only the known portion of the sequence number, finds the label, and returns to the parent node rather

than the root node. An enhanced dynamic binary tree search technique was presented by Research (Umelo, 2023), in which the reader/writer creates a new query command by setting all collision bits before the lowest collision bit to '0'. The reader-writer detects two tags at once if the supplied data contains just one collision bit. In order to increase the efficiency of tag identification, research (Baghdad, 2022) introduced the parallel response query tree method (PRQT), which is based on the concept of quadrature trees and tags employing dual-carrier frequency modulation signals. Data is sent to the reader in parallel.

Research on anti-collision algorithms and RFID technologies has also been emerging domestically in China in recent years. The RFID system's actual application efficiency was substantially enhanced by the efforts of numerous scholars and specialists in anti-collision algorithms based on environmental changes. The application of the matrix search algorithm, an efficient enhancement of the anti-collision algorithm in the manufacturing field, is documented in the literature (Hu, 2023) by researchers. With the ability to respond to information data formed by multiple labels' data requests simultaneously, this algorithm can prevent the formation of an information collision matrix, improving the binary tree search algorithm's bit-by-bit search method and overall work efficiency. Additionally, this algorithm - which is built on the idea of a collision stack - can dynamically and spontaneously adapt to the search path, effectively reducing the amount of label collisions and idle time slots during transmission and significantly increasing search efficiency (Sulaiman, 2022). Researchers suggested lowering the overhead of the empty time slots in the anti-collision method application in the literature (Zhou, 2023), and they created a unique technique in response. This algorithm can dynamically modify the length of the command frame to minimize the empty time slot time wasted in order to improve the efficiency of the system read. It can also command the position of the empty time slot in the frame in the recognition of the first scanning and the number of tags to estimate.

The new improved binary search tree anti-collision technique (EBS) that is suggested in this work organizes all of the tags in the reader region. The tags in the group give the reader data in a query round in a sequential manner based on the collision bit. In addition to lowering the amount of inquiries and increasing recognition rate, this technique can also decrease collisions.

III. NLHQT ALGORITHM ANALYSIS

Presuming that the reader's query range contains four tags, each with an 8-bit ID and an associated tag code: Tag A (01000001), Tag B (01101101), Tag C (01110110), and Tag D (11001101) (Guo, 2022). The reader expects that each time it sends 1, the NLHQT algorithm may be deemed to detect the tag's free child node since the locking instruction lowers the superfluous free time slots. The tag is identified as shown in Table 1 by the algorithm at $t = 1$. Figure 1 shows how the IHQT algorithm compares at $t = 1$ and the NLHQT method

at $n = 1$ (Seo, 2023). Figure 1 illustrates how the methods employed in this work are successful in preventing conflicting time slots from forming, enhancing recognition

performance, and guaranteeing that there are no idle time slots.

TABLE I RECOGNITION PROCESS OF NLHQT ALGORITHM FOR $n = 1$

Queries	1	2	3	4	5	6
Query Order	Ask for (1111111)	Ask for (UID,1)	-	Ask for (UID,1)	Ask for (UID,1)	-
Readers receive information	01000001, 01101101, 01110110, 11001101	000011, 011011, 011010, 101011	-	00011, 11110, 11001	10,01	-
Prediction (R for T)	-	1	-	1	1	-
Prediction (T for R)	-	01/10	-	01/01	10/01	-
Query Header	-	Ask for (0,1)	Ask for (1,0)	Ask for (0,0)	Ask for (0,1)	Ask for (1,0)
Label A	A	A		A		
Tag B	B	B			B	
Tag C	C	C				C
Tag D	D		D			
Label Status	Collision	Collision	Recognition	Recognition	Recognition	Recognition
Stack Contents	-	000011, 011011, 011010, 101011	000011, 011011, 011010	11110,11001	01	-

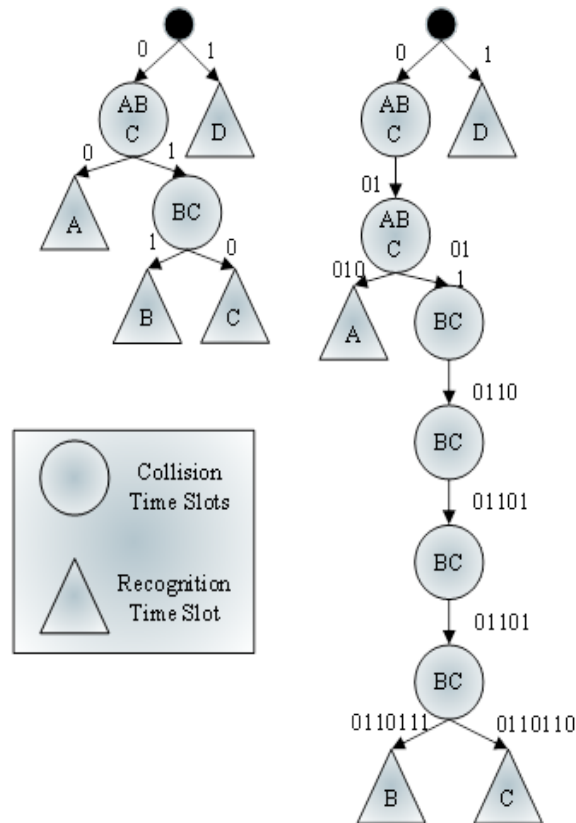


Fig. 1 For $n=1$, the NLHQT algorithm and the IHQT algorithm flow

Table II shows the NLHQT algorithm at $n = 2$. By sending “11” each time during the prediction phase, the bit-locking instruction, like $n = 1$, allows the reader to accurately determine whether idle child nodes exist. A comparison of the IHQT approach and the NLHQT algorithm is shown in Figure 2

with $n = 2$. The NLHQT method outperforms the IHQT technique in terms of recognition efficiency when the tags cause continuous collisions, as a result producing fewer collision time slots without producing any idle time slots.

TABLE II RECOGNITION PROCESS OF NLHQT ALGORITHM WHEN $n = 2$

Queries	1	2	3	4	5
Query Order	Ask for (11111111)	Ask for (UID,1)	-	Ask for (UID,1)	-
Readers receive information	01000001, 01101101, 01110110, 11001101	000011, 011011, 011010, 101011	-	10,01	-
Prediction (R for T)	-	11	-	1	-
Prediction (T for R)	-	1000/0100/0001	-	0011/0100	-
Query Prefix	-	Request(01,00)	Request(00,01)	Request(10,10)	Request(00,10)
Label A	A	A			
Tag B	B			B	
Tag C	C				C
Tag D	D		D		
Label Status	Collision	Collision	Recognition	Recognition	Recognition
Stack Contents	-	011011, 011010, 101011	011011, 011010	01	-

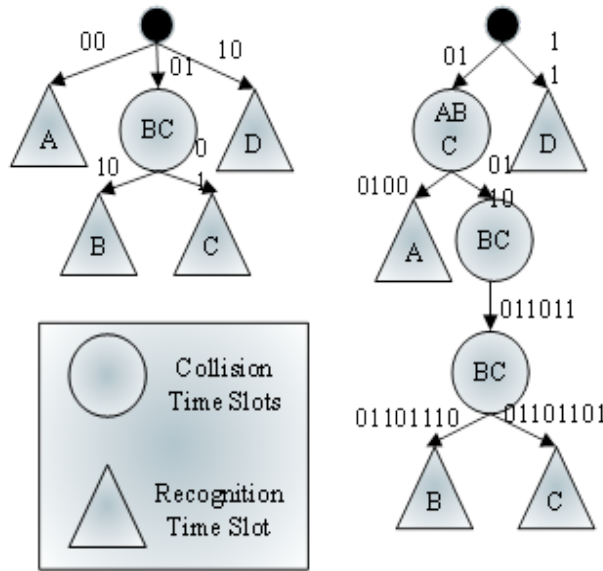


Fig. 2 The algorithm flow for NLHQT and IHQT for $n = 2$ and $n = 2$, respectively

A. Guidelines for the NLHQT Algorithm

1. Request (UID, 1) lock command: the conflict bit position is “1” and the non-conflict bit position is “0”; the tag locates the precise tag collision by receiving instructions, tags, and data from the reader. Once the tag has received the command, it compares its tag with the reader’s information and sets the reader’s bit position to “1”, creating a new tag sequence.
2. Request(s, m, n), where the modified request prefix is indicated by s ; the tag to be compared to the largest bit

request prefix (i.e., the largest bit found in the conflict) is indicated by m ; n is applied to ascertain which tag should be compared to the second-highest reader’s request prefix (i.e., the second highest reader with detected conflict, $n = 1$, no such element), and so on (encoded labels are made up of p - bits) from 0 to p , both top to bottom and left to right.

B. Steps in the NLHQT Algorithm

The NLHQT algorithm consists of the following steps.

Once the initialization stack is empty, all tags in the reader’s recognition range reply when the reader sends a request (1111111).

Based on the responses from the tags, the reader determines the status of the time slot. In case of multiple tags responding simultaneously, the reader will switch to “request to send command” to indicate a conflict in the time slot; If only one tag answers, it will correctly identify the tag and go straight to “end”.

The conflict bit “1” will be set to “1” when the reader submits a lock command request (UID, 1), in accordance with the Manchester encoding principle, which stipulates that when a tag bit conflicts, the conflict position is “1” and the non-conflict position is “0”. In order for the reader to receive a fresh tag sequence, it must transmit a request (UID, 1) to lock the tag sequence with the conflict bit set to “1”.

The guidelines for prediction readers are required to complete Prognosis(1) or Prognosis(11).

The tag transforms bit 0 or bits 0 and 1 to the equivalent decimal value a by utilising the Prognosis instruction. After then, it provides the reader with a response of length 2 bits or 4 bits, where “0” represents the amount of bits that are still missing and “1” represents the entire number of bits in the response.

After receiving all tag replies, the reader categorises the child nodes that are useful at the proper points in the record sequence as well as the other child nodes that are idle. Subsequently, the query prefixes required for the query command will be constructed from useful child nodes. The label is identified by the query prefix.

Sees if the tag has collided; if not, the tag has been correctly detected. If it has, then sends the request command again. Determines whether the stack is empty. Should that be the case, everything is identified accurately; if not, “Request to send command” is returned and the stack is removed from the request prefix. As the flow of the algorithm is shown in Figure 3.

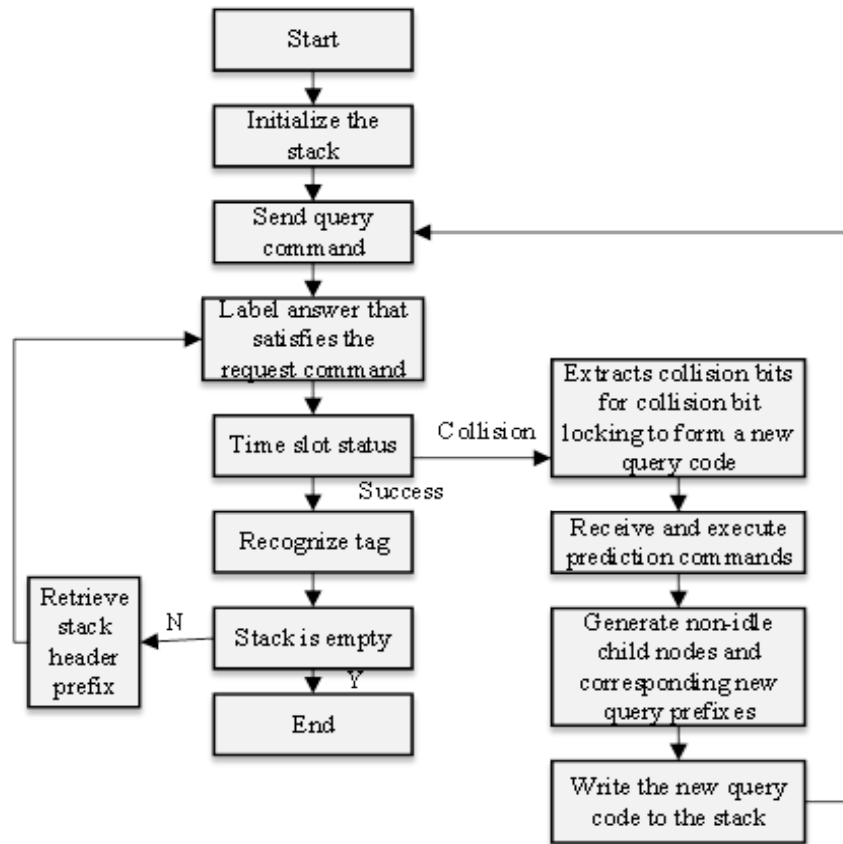


Fig. 3 Flowchart of an algorithm

IV. SIMULATION EXPERIMENT

A series of tests is established with respect to the selection of segmentation point length: the prefix and suffix segmentation point lengths are simultaneously raised from 3 to 96, with an increment of 1. In addition to counting the change in the average transmission data volume, there are now 1,000 tags

to identify instead of just 200. The experimental results are displayed in Figure 4, where a minimum value is reached at a segmentation point length of approximately 12. The average transmission data volume of various numbers of tags to be recognized first lowers and then grows with the segmentation point length.

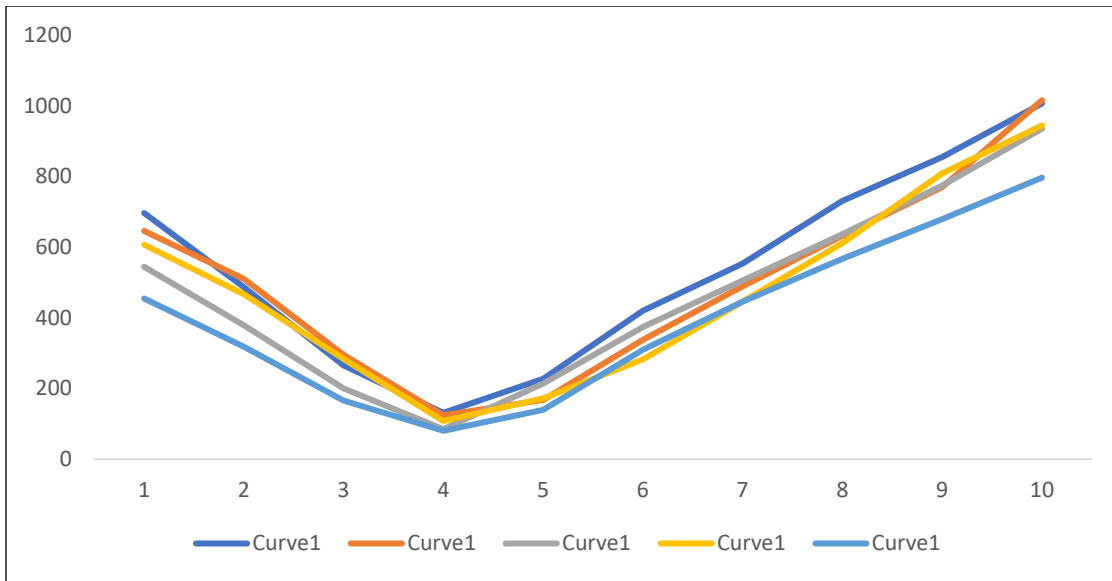


Fig. 4 Length of the segmentation points affecting the average data volume transmitted by the BDS algorithm

Figure 5 illustrates how the average amount of data transmitted by the bidirectional segmentation algorithm changes as the number of tags to be recognized increases. Here, m represents the positions of the first sixteenth, twelfth, eighth, sixth, fifth, fourth, third, and second equidistant points of the 96-bit ID, or 6, 8, 12, 16, 19, 24, 32, and 48, respectively. Starting at 100, the number of tags to be recognised is increased in increments of 100 to 1000. It is

evident that, in the majority of situations, the performance at segmentation point length of 12 necessitates the least amount of data to be provided in order to identify a unit tag. When the number of tags rises from 100 to 1000, the average amount of data sent varies from 169 to 197. This variation is more consistent than the more notable change in segmentation lengths of 6 bits, 8 bits, and 48 bits.

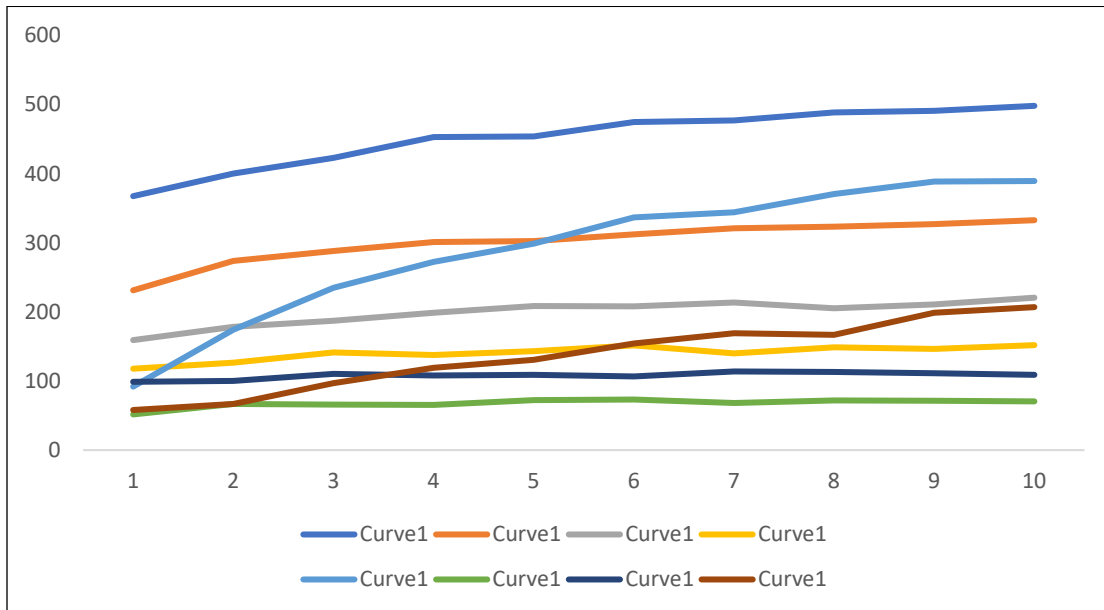


Fig. 5 Average transmission data volume of BDS algorithm with different segmentation point lengths

Using the Monte Carlo approach, the paper simulates and examines the proposed bidirectional segmentation (BDS) collision avoidance algorithm. The algorithm's performance is evaluated using three indexes: system throughput rate, average transmission data volume, and average query cycle

time. The Q-advanced search method (QAS) algorithm, the optimal query tracking tree model (OQTT) algorithm.

The 96-bit tag IDs are produced at random in accordance with the EPC worldwide 96-bit coding standard, and the number

of tags rises in increments of 100 from 100 to 1000. (12, 84) is the BDS segmentation point.

The throughput rates that each method achieves are displayed in Figure 6. The results indicate that the BDS algorithm outperforms the other algorithms in terms of throughput because it eliminates idle time slots, while the QT algorithm

has the lowest throughput due to its lack of use of collision information. The BDS algorithm also tends to approach $\frac{1}{1 + \alpha / 2}, \alpha \in (0,1)$ depending on the length of the segmentation points, in terms of throughput as the number of tags rises. The CT algorithm has a stable throughput of 0.5.

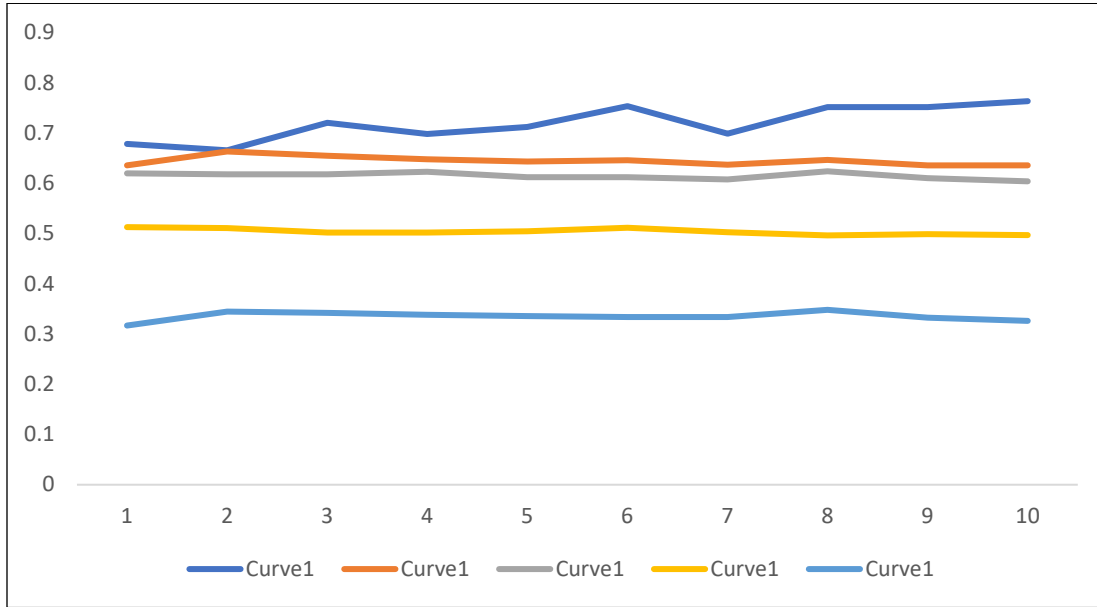


Fig. 6 The efficiency of various algorithms

As seen in Figure 7, the BDS algorithm sends around 100 bits less data than the nearest algorithm, the QAS algorithm, and substantially less data overall than the QT algorithm. A maximum of 197 bits are needed to identify 1000 tags. Bi-

directional segmentation has advanced to the point where it can reduce the quantity of information carried while increasing capacity (Zhou, 2023) (Wu2023) (Zhou, 2023).

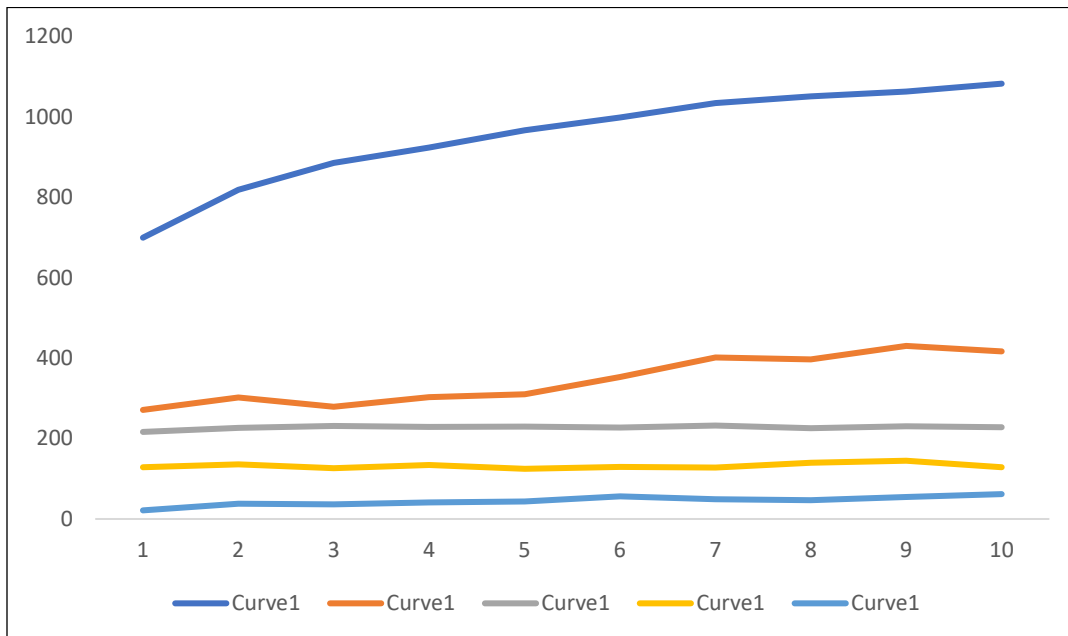


Fig. 7 Average amount of data transferred by different algorithms

V. CONCLUSION

A novel hybrid bit-blocking query tree method is presented in this paper. The fundamentals of the bit-blocking method and idle time elimination for this algorithm are presented in this work. It provides a detailed explanation of the algorithm's identification process while reducing the algorithm's complexity without sacrificing its usefulness. NLHQT algorithm is done by setting the algorithm's parameter in the cases of $n=1$ and $n=2$, i.e., optimizing the algorithm based on binary and quadtree networks. This significantly simplifies the realization of the algorithm. It is evident from a careful examination and comparison of the throughput, conflicting interval count, and performance characteristics of various methods, total interval count, regarding communication complexity, the technique presented in this research successfully lowers the quantity of intervals that conflict, eliminates idle intervals, reduces communication complexity, and increases throughput - all of which improve recognition efficiency and overall performance. This was accomplished by using Matlab simulation software. Both binary tree and quadtree enquiry techniques, which have a broad range of applications and are simple to build with minimal complexity, can benefit from the methodology presented in this study. More research is needed to determine how to determine the optimal fork tree for the first search based on the predicted number of labels, as most of the time the number of labels can be predicted in practice. However, the algorithm presented in this paper has a greater practical value.

REFERENCES

- [1] Ai, Y., Bai, T., Xu, Y., & Zhang, W. (2022). Anti-collision algorithm based on slotted random regressive-style binary search tree in RFID technology. *IET Communications*, 16(10), 1200-1208.
- [2] Akavia, A., Leibovich, M., Resheff, Y. S., Ron, R., Shahar, M., & Vald, M. (2022). Privacy-preserving decision trees training and prediction. *ACM Transactions on Privacy and Security*, 25(3), 1-30.
- [3] Alimohammadi, H., & Ahmadi, M. (2019). Clustering-based many-field packet classification in software-defined networking. *Journal of Network and Computer Applications*, 147, 102428.
- [4] Baghdad, A. (2022). An improved RFID anti-collision protocol (IMRAP) with low energy consumption and high throughput. *Scientific African*, 16, e01209.
- [5] Besta, M., Gerstenberger, R., Peter, E., Fischer, M., Podstawski, M., Barthels, C., & Hoefler, T. (2023). Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries. *ACM Computing Surveys*, 56(2), 1-40.
- [6] Gul, F., Mir, I., Alarabiat, D., Alabool, H. M., Abualigah, L., & Mir, S. (2022). Implementation of bio-inspired hybrid algorithm with mutation operator for robotic path planning. *Journal of Parallel and Distributed Computing*, 169, 171-184.
- [7] Guo, K., Xie, X., Chen, S., Qi, H., & Li, K. (2022). Efficient collision-slot utilization for missing tags identification in RFID system. *Computer Communications*, 195, 61-72.
- [8] Hu, J. G., Mei, W. Z., Wu, J., Li, J. W., & Wang, D. M. (2023). A Fully Integrated RFID Reader SoC. *Micromachines*, 14(9), 1691.
- [9] Khalil, G., Doss, R., & Chowdhury, M. (2019). A comparison survey study on RFID-based anti-counterfeiting systems. *Journal of Sensor and Actuator Networks*, 8(3), 37.
- [10] Kumar, A., Aggarwal, A., & Gopal, K. (2021). A novel and efficient reader-to-reader and tag-to-tag anti-collision protocol. *IETE Journal of Research*, 67(3), 301-312.
- [11] Lai, Y. C., Chen, S. Y., Hailemariam, Z. L., & Lin, C. C. (2022). A bit-tracking knowledge-based query tree for RFID tag identification in IoT systems. *Sensors*, 22(9), 3323.
- [12] Mu, Y., Ni, R., Sun, Y., Zhang, T., Li, J., Hu, T., & Tyas, T. L. (2021). A novel hybrid tag identification protocol for large-scale RFID systems. *Computers, Materials & Continua*, 68, 2516-2526.
- [13] Munir, A., Laskar, M. T. R., Hossen, M. S., & Choudhury, S. (2019). A localized fault-tolerant load balancing algorithm for RFID systems. *Journal of Ambient Intelligence and Humanized Computing*, 10, 4305-4317.
- [14] Peng, J., Zhang, L., Fan, M., Zhao, N., Lei, L., He, Q., & Xia, J. (2023). An Admission-Control-Based Dynamic Query Tree Protocol for Fast Moving RFID Tag Identification. *Applied Sciences*, 13(4), 2228.
- [15] Rasina Begum, B., & Chitra, P. (2023). SEEDDUP: a three-tier SEcurE data DedUPlication architecture-based storage and retrieval for cross-domains over cloud. *IETE Journal of Research*, 69(4), 2224-2241.
- [16] Seo, C., Noh, Y., Abebe, M., Kang, Y. J., Park, S., & Kwon, C. (2023). Ship collision avoidance route planning using CRI-based A* algorithm. *International Journal of Naval Architecture and Ocean Engineering*, 100551.
- [17] Su, J., Chen, Y., Sheng, Z., Huang, Z., & Liu, A. X. (2020). From M-ary query to bit query: a new strategy for efficient large-scale RFID identification. *IEEE Transactions on Communications*, 68(4), 2381-2393.
- [18] Sulaiman, S., & Sudheer, A. P. (2022). Modeling of a wheeled humanoid robot and hybrid algorithm-based path planning of wheelbase for dynamic obstacles avoidance. *Industrial Robot: The International Journal of Robotics Research and Application*, 49(6), 1058-1076.
- [19] Tang, Z. J., Guo, Y., & Liu, Q. (2019). Research on an improved fusion RFID collision avoidance algorithm. *Journal of Communications Technology, Electronics and Computer Science*, 22, 6-19.
- [20] Umelo, N. H., Noordin, N. K., Rasid, M. F. A., Geok, T. K., & Hashim, F. (2022). Grouping based radio frequency identification anti-collision protocols for dense internet of things application. *International Journal of Electrical and Computer Engineering*, 12(6), 5848.
- [21] Umelo, N. H., Noordin, N. K., Rasid, M. F. A., Tan, K. G., & Hashim, F. (2023). Efficient Tag Grouping RFID Anti-Collision Algorithm for Internet of Things Applications Based on Improved K-Means Clustering. *IEEE Access*, 11, 11102-11117.
- [22] Wang, L., Luo, Z., Guo, R., & Li, Y. (2023). A Review of Tags Anti-Collision Identification Methods Used in RFID Technology. *Electronics*, 12(17), 3644.
- [23] Wu, J., Chen, X., Bie, Y., & Zhou, W. (2023). A co-evolutionary lane-changing trajectory planning method for automated vehicles based on the instantaneous risk identification. *Accident Analysis & Prevention*, 180, 106907.
- [24] Yaacob, M., Daud, S. M., & Azizan, A. (2019). A review of deterministic anti-collision algorithm of passive RFID systems. *Open International Journal of Informatics*, 7(1), 8-25.
- [25] Zhang, H., Gao, L., Luo, H. G., & Zhai, Y. (2022). Research on the RFID anticollision strategy based on decision tree. *Wireless Communications and Mobile Computing*, 2022, 1-7.
- [26] Zhou, H., Feng, P., & Chou, W. (2023). A hybrid obstacle avoidance method for mobile robot navigation in unstructured environment. *Industrial Robot: The International Journal of Robotics Research and Application*, 50(1), 94-106.
- [27] Zhou, J., Liu, Q., Jiang, Q., Ren, W., Lam, K.-M., & Zhang, W. (2023). Underwater image restoration via adaptive dark pixel prior and color correction. *International Journal of Computer Vision*. DOI: 10.1007/s11263-023-01853-3.
- [28] Zhou, J., Zhang, D., & Zhang, W. (2023). Cross-view enhancement network for underwater images. *Engineering Applications of Artificial Intelligence*, 121, 105952.