A Review of Deterministic Anti-Collision Algorithm of Passive RFID Systems

Marizan Yaacob , Salwani Mohd Daud , Azizul Azizan

School of Professional and Continuing Education,
Universiti Teknologi Malaysia, Kuala Lumpur,
Malaysia

Razak Faculty of Technology and Informatics,
Universiti Teknologi Malaysia,
Kuala Lumpur, Malaysia

marizan.kl@utm.my
salwani.kl@utm.my
azizulazizan@utm.my

Article history

Received:
2 July 2019

Received in revised form:
25 August 2019

Accepted:
30 September 2019

Marizan Yaacob,
marizan.kl@utm.my

Abstract

Radio frequency identification (RFID) are used widely in many industries, manufacturing companies, material flow systems, etc. Collisions are divided into reader collisions and tag collisions. When the reader sends out a signal which supplies power and instructions to a tag, the tag transmits its ID to the reader and the reader link to an external database with the received ID recognizes the object. The tag collisions occur in the situation when more than one tag reflects their data at the same time, which make it difficult for the reader to search and identify all tags in interrogation zone. In Passive RFID system, tags' collision are the main issue of multi-tag identification. A various anti-collision techniques and mechanism had been researched for a better collision resolution hence improving system efficiency. Researchers had proposed and improved deterministic and probabilistic of TDMA based anti-collision algorithm. However the complexity of the algorithm is the issue in implementation for passive RFID systems. Limitation of the power and the computational capability of the tags are the constraint of passive RFID system in implementing the anti-collision algorithm. This paper provides a reviews of anti-collision algorithm, an insight of collision resolution techniques, a tag and a reader specification of deterministic algorithm of passive RFID systems. These can give an overview of anti-collision algorithm with system complexity tolerance.

Keywords: *RFID, Passive Tag, Anti-Collision Algorithm, Query Tree, Binary Search, Collision Tracking, Binary Tree Splitting.*

1. Introduction

Radio Frequency Identification (RFID) is an automatic object identification that use wireless communication system. RFID tag is a device attached to an object, which use a radio frequency (RF) to communicate. The reader queries tags by broadcasting an RF signal, and the tag in the interrogation zone response to the reader by backscattering the ID. The reader forward the tags response to the backend server. The server has a database of tags and can retrieve detailed information of the tags. RFID does not require line of sight communication, able to withstand the harsh physical environment, low cost and highly efficient operation in identify multiple tags at the same time. It has the advantages over a barcode technology.

The valid distance between RFID tags and a Reader is much longer than the barcode system. Most of RFID tags use silicon technology, providing a large memory for storage and having an ability to support security and privacy algorithms. In addition, barcode cannot change the stored information which is imprinted, but the RFID tags with write function can change the stored information. RFID has many application in access control, manufacturing automation, maintenance, supply chain management, parking garage management, automatic payment, tracking and inventory control.

In a passive RFID system, a tag does not have a battery. It harvest an energy from electromagnetic wave of the reader. It can be used over a long time period and can be produced at a favourable price but the read range is strongly limited by the overall efficiency of the system. RFID reader send an inquiry and the collisions occurred when multi tags respond to a reader simultaneously. An anticollision algorithm with a good resolution collision mechanism and technique would be able to resolve the collision with minimum communication overhead. Hence will improve the overall performance of the RFID systems.

2. RFID System

A typical RFID system consists of tags and readers. A block diagram of the system is shown in Figure 1. RFID reader consists of a transmitter-receiver module, a control unit and an antenna. A tag is made up of a microchip with an antenna. The reader sends electromagnetic signals and the tag receives these signal through antenna. The microchip modulates the signals and send it back to the reader. This information received from the tags is then processed by sending it to the host computer.

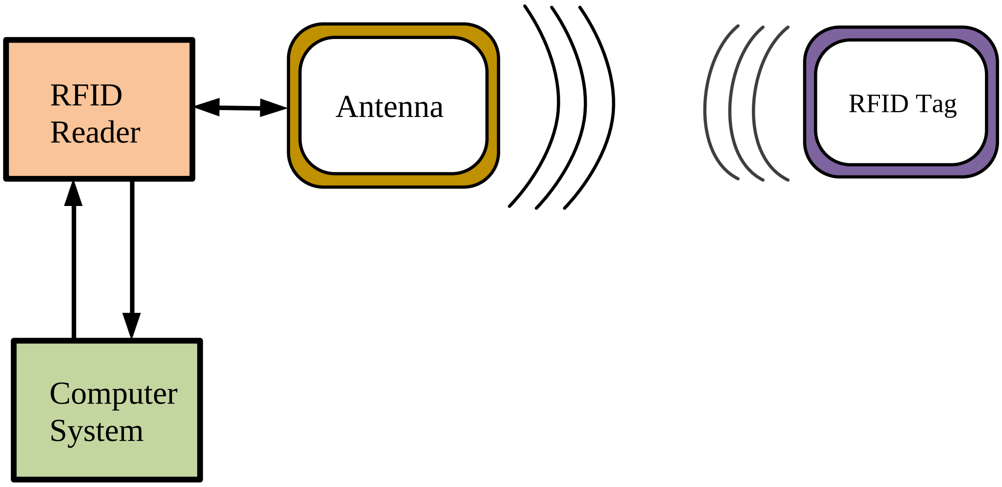
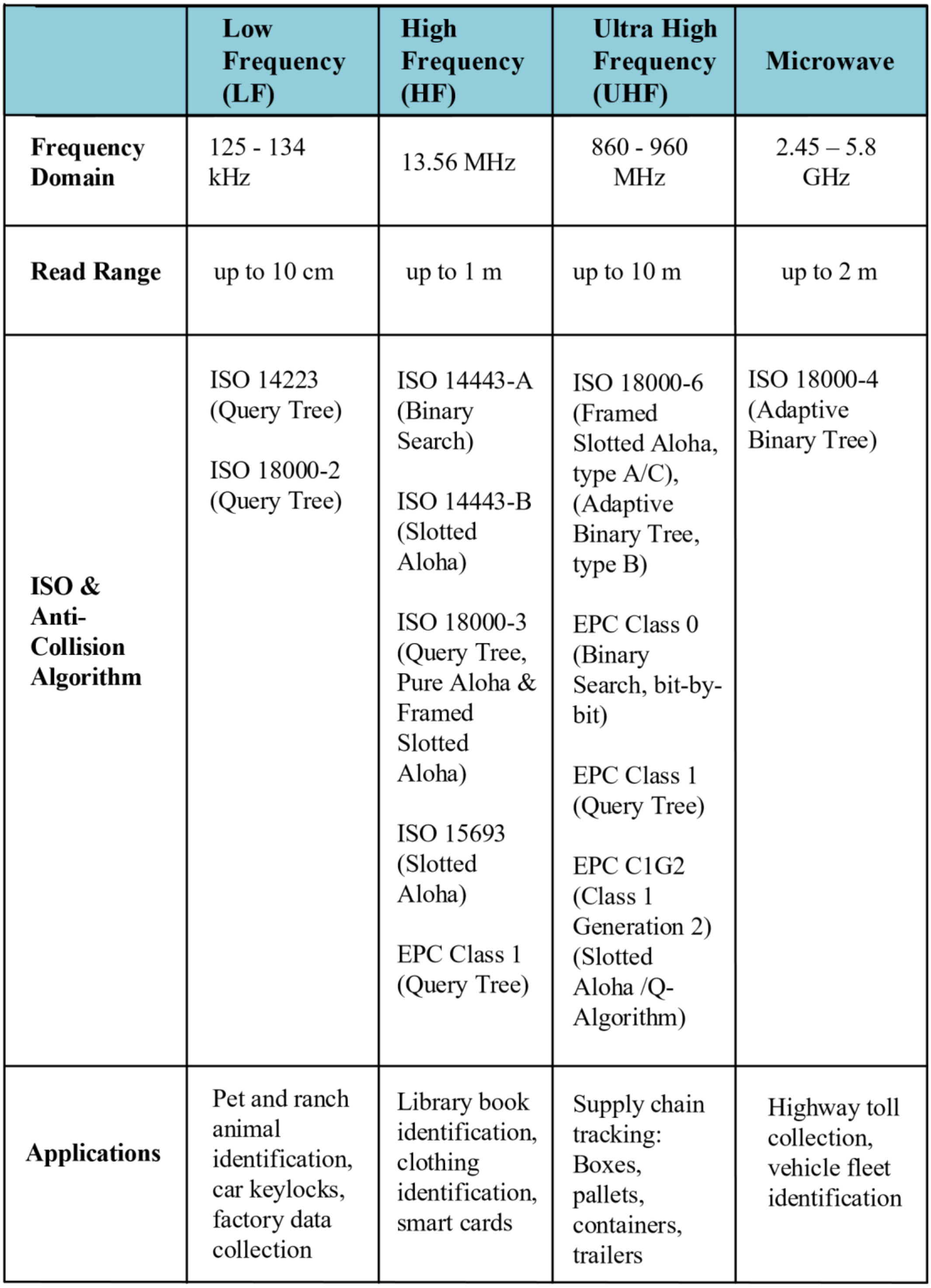


Figure 1. RFID System

RFID systems are operated in various frequency band. In the low frequency (LF) domain at around 125 kHz and in the frequency (HF) domain at 13.56 MHz passive tags are coupled with the reader via a dominantly magnetic field and transmit information to the reader by a load modulation technique. The ultra-high frequency (UHF) domain operates in the band of 860 – 960 MHz. Here, the RFID tags utilize backscatter modulation for communication [1]. Finally has been extended to the microwave band at 2.4 and 5.8 GHz [2]. In each of those many frequency bands several different standards have emerged.

Table 1. RFID systems classification [3] [4]



	Low Frequency (LF)	**High Frequency (HF)**	**Ultra High Frequency (UHF)**	**Microwave**
Frequency Domain	125 - 134 kHz	13.56 MHz	860 - 960 MHz	2.45 – 5.8 GHz
Read Range	up to 10 cm	up to 1 m	up to 10 m	up to 2 m
ISO & Anti-Collision Algorithm	ISO 14223 (Query Tree) ISO 18000-2 (Query Tree)	ISO 14443-A (Binary Search) ISO 14443-B (Slotted Aloha) ISO 18000-3 (Query Tree, Pure Aloha & Framed Slotted Aloha) ISO 15693 (Slotted Aloha) EPC Class 1 (Query Tree)	ISO 18000-6 (Framed Slotted Aloha, type A/C), (Adaptive Binary Tree, type B) EPC Class 0 (Binary Search, bit-by-bit) EPC Class 1 (Query Tree) EPC C1G2 (Class 1 Generation 2) (Slotted Aloha /Q-Algorithm)	ISO 18000-4 (Adaptive Binary Tree)
Applications	Pet and ranch animal identification, car keylocks, factory data collection	Library book identification, clothing identification, smart cards	Supply chain tracking: Boxes, pallets, containers, trailers	Highway toll collection, vehicle fleet identification

2.1. Classes of Tags

A tag, also called a transponder, normally consists of an RF- coupling element (antenna) and an integrated chip (IC). The antenna is for receiving and transmitting signals with reader; the IC is mainly for data processing, protocol control, and data storage. Data stored in a tag is usually the information of manufacturer, product category, serial number, shipment record, expiry date, and other useful information for object identification.

In terms of the availability of on-board power supply, RFID tags can be classified into three types: passive tag, active tag, and semi-passive tag. A passive tag does not have on-board power source so that its longevity is not limited by energy. It relies the continuous wave (CW) power emitted by reader to perform the identification process. The communication from passive tag to reader is usually backscatter (high frequency passive tags) or load modulation (low frequency passive tags). Because of power constraint, the communication range for passive tags is from 2.5 cm up to 10 m. In addition, it is the smallest in size and the cheapest in price. Passive tags typically operate at frequencies of 128 kHz, 13.5 MHz, 915 MHz, or 2.45 GHz.

An active tag has an on-board source, which mainly consists of batteries. It can transmit a stronger signal for a longer distance. The tag's command processing and data transmission depends on its power supply, so that its longevity is limited by the stored energy. The on-board power supply makes the circuit more complex, which making them larger and more expensive, so active RFID systems can support more complicated communication protocols and perform more complex tasks, such as tracking over long distance and sensing. Readers can communicate with active RFID tags across 20 to 100 meters. A semi-passive tag has an onboard power source for its command processing and data operation like an active tag. But it uses the CW power from reader for data communication like a passive tag. The communication range for semi-passive tags can be up to 30 meters.

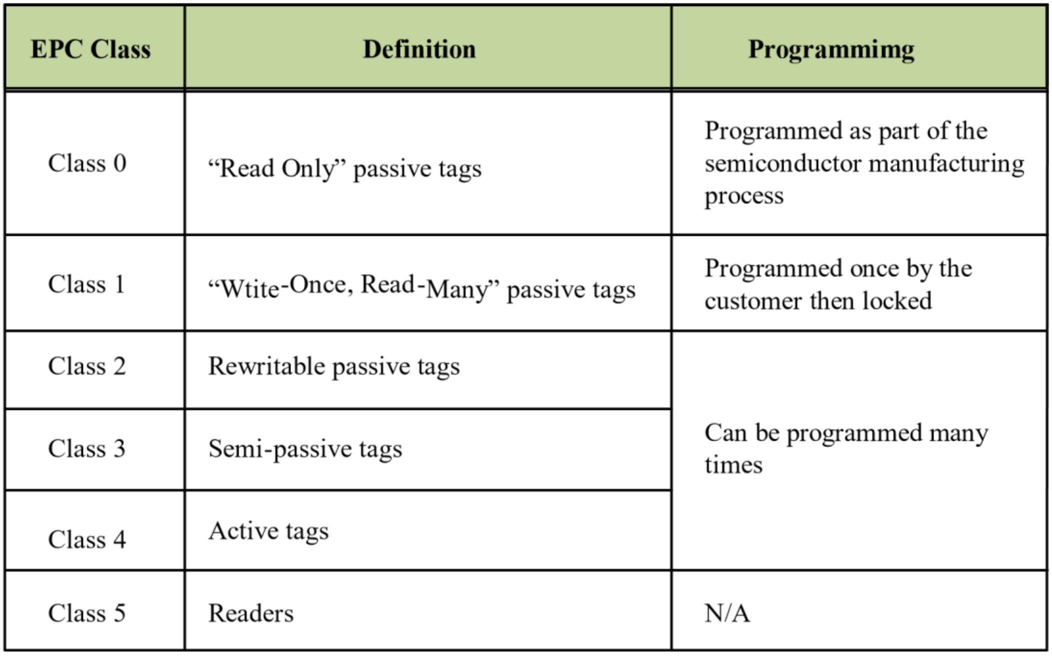
A passive tag has constraints in functionality, but it has the distinct feature to the active tag. It is enough small to attach to an object easily. There could be multiple tags a reader should identify. All the functionality tags should do is that they response with the data corresponding to the signal received from a reader. The communication between tags is impossible. Passive tags cannot make a decision of whether the channel is busy or not. A collision is occurred at the reader’s side when two more tags get transmitted simultaneously. Therefore, the arbitrational mechanism is required. The protocol aiming to avoid collisions between a reader and tags anti-collision protocol. The anti-collision protocol should have the following characteristics [5].

i. A reader should identify all the tags within its range.

ii. The anti-collision algorithm should have a mechanism which is capable of verifying that all the tags are identified.

iii. It should minimize the time elapsed for the identification of tags. It lies on same line as reducing collisions.

Table 2. EPCglobal, Classes of Tags



EPC Class	Definition	Programmimg
Class 0	“Read Only” passive tags	Programmed as part of the semiconductor manufacturing process
Class 1	“Wtite-Once, Read-Many” passive tags	Programmed once by the customer then locked
Class 2	Rewritable passive tags	Can be programmed many times
Class 3	Semi-passive tags	
Class 4	Active tags	
Class 5	Readers	N/A

2.2. Electronic Product Code (EPC)

The EPC protocols are developed by EPCglobal in cooperation with the AutoID laboratories. The code can have 64 bits or 96 bits with different functionalities. It contains a header, an EPC manager, an object class and a serial number. An example of a 96 bits EPC is shown in Figure 2. The next generation of EPC is the electronic product code information services (EPCIS), which control information contained by EPC (tag ID, date of manufacture, country of origin, production batch and shipment).

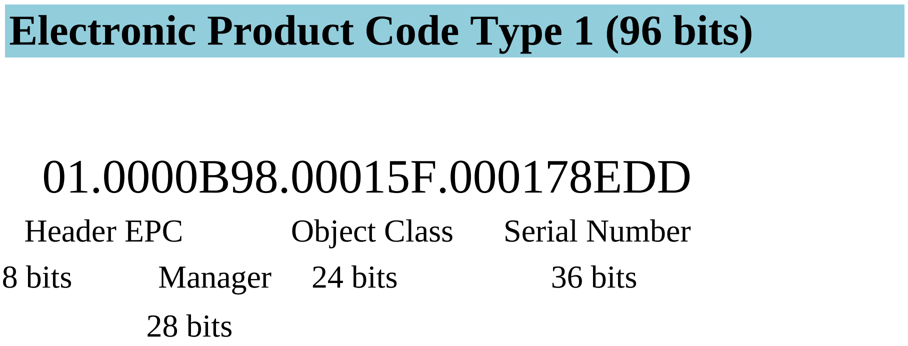


Figure 2. Electronic Product Code (EPC)

3. Anti-Collision Algorithm

Two major types of anti-collision algorithm in TDMA scheme are ALOHAbased and tree-based protocol. ALOHA-based is a type of probabilistic algorithm and Tree–based is a type of deterministic algorithm. A Tree-based methods can be classified into a Memory-based algorithm and Memoryless-based algorithm. In memory based algorithm, which can be grouped into a splitting tree and bit arbitration algorithm. Memoryless is dedicated to Query Tree (QT) algorithm. Query Tree (QT) [6], Tree Splitting (TS) [7], Binary Search (BS) [8] and Bitwise Arbitration (BTA) protocols [9] are considered the discovering of tree based algorithm of RFID system. Binary search algorithm (BS) is a founded of collision tracking mechanism by using Manchester coding technique.

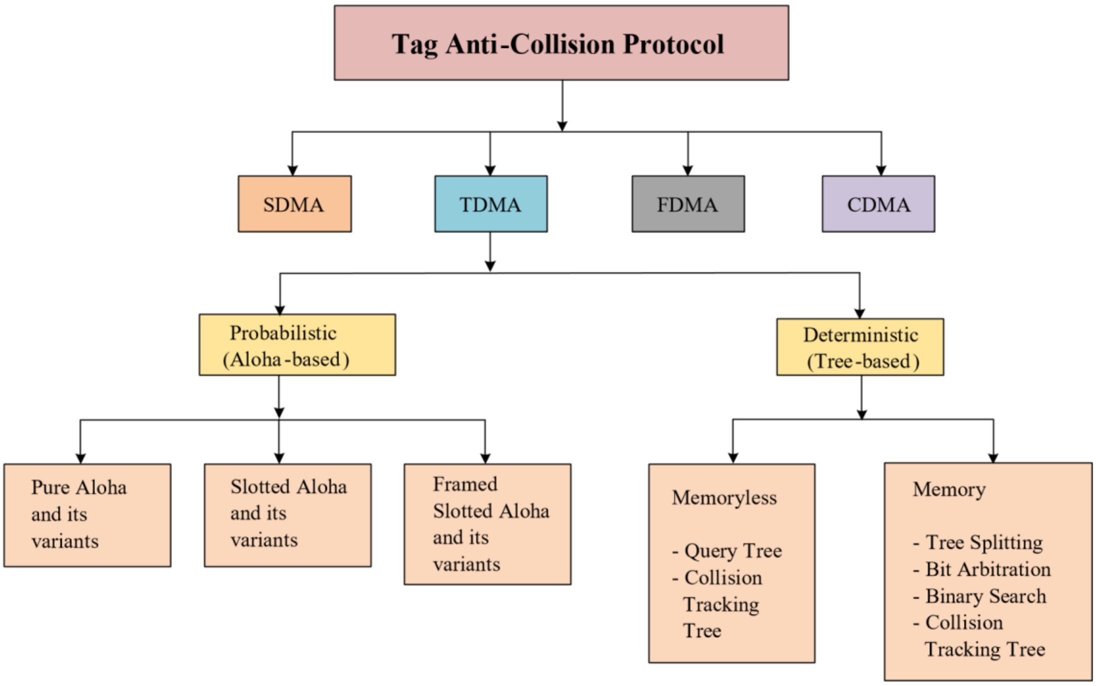


Figure 3. RFID Tags Anti-collision protocols

4. Memoryless Tree-based Algorithm

A Memoryless tree-based algorithm is considered of having a minimum requirement of tag’s circuit to store an ID and has a prefix matching circuit to match an ID with the reader queries/prefix. It doesn’t has a counter or additional circuitry to remember the state or record the path. Query Tree (QT) are considered as a memoryless algorithm based on tag’s circuit minimum specification. The variance of Query tree (QT) try to maintain the tag specification for memoryless, most of their improvement has been made on reader side by improving method of queries.

4.1 Query Tree

Query Tree (QT) [6] queries tags according to their ID. The reader interrogates tags sending a string, and only those tags IDs have a prefix matching that string respond to the query. At the beginning, the reader queries all tags by sending □ string. If a collision occurs, then the string length is increased of one bit until the collision is solved and a tag is identified. The reader then starts a new query with a different string. The QT system efficiency is 34% and time system efficiency is 40% [10]. The operation of Query Tree is as in Figure 4. Query Tree (QT) Protocol is a memoryless, requires very less tag circuitry. The tags do not need to remember their inquiring history.

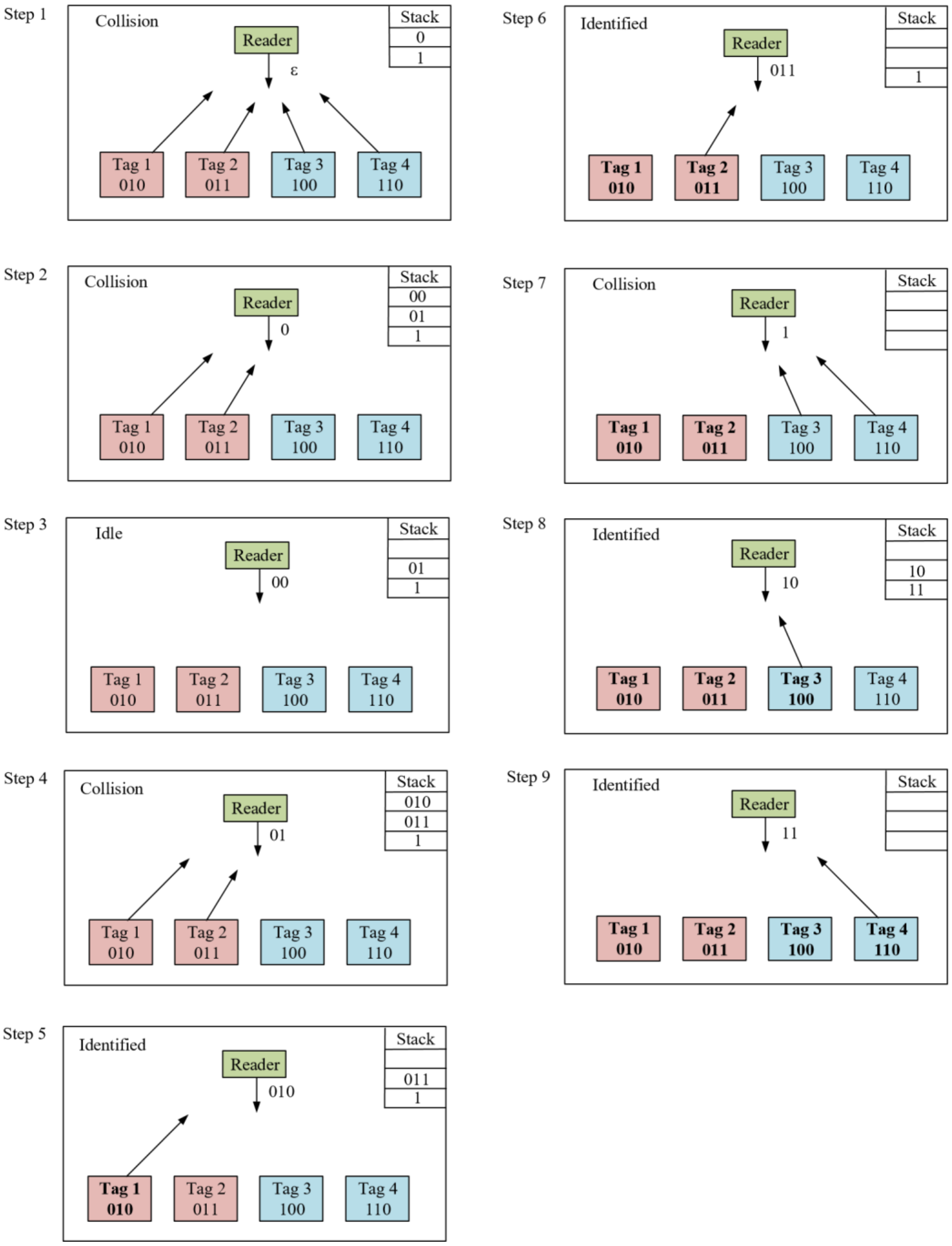


Figure 4. Query Tree

Query Tree Improved (QTI) is an extensions to the QT protocol [6], an enhancement of the QT protocol that optimizes the number of queries, avoiding those that will certainly produce collisions. This extension reduces QT's identification delay by removing redundant queries. As an example, consider the case in which prefix "p" produces a collision, while prefix "p0" results in no tag answers. In QTI the reader will then skip prefix "p1" that will certainly produce collision and query directly for "p10" and "p11". The expected number of queries EQ needed by QTI to identify all tags are estimated as $2.6607n \le EQ \le 2.665n - 1$ (11). Consequently, the QTI system efficiency is around 37% and time system efficiency is 41% [10].

Smart Trend Transversal (STT) [12] was a variant of Query Tree. A new query tree protocol to resolve the collision caused by tags whose binary identifiers (ID) are non-uniformly distributed. A Self-learned tag density and distribution, it proposes a combination of the QT protocol and the shortcutting method. When a query in a node results in successful, it is indicated that the protocol is located in the right level, the protocol moves horizontally and takes the binary string in the next node according to the breadth-first order as the next query string. When the query results in idle, it is indicated that the protocol is located in too low level and if current node is a right child node, the protocol should move up along the tree. Otherwise if the query result in collision, the protocol is in too high level and should move down along the tree. In a lot of cases, the STT protocol moves in the correct level of the binary query tree to solve tag collision resolution. System efficiency is 33.9% and time system efficiency is 54.9% [13].

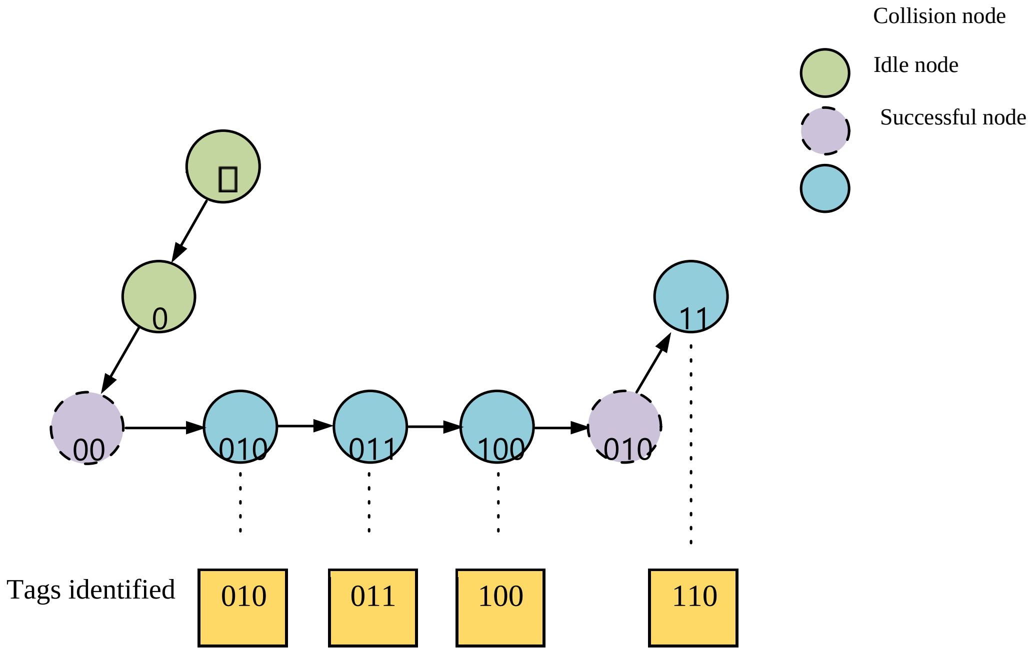
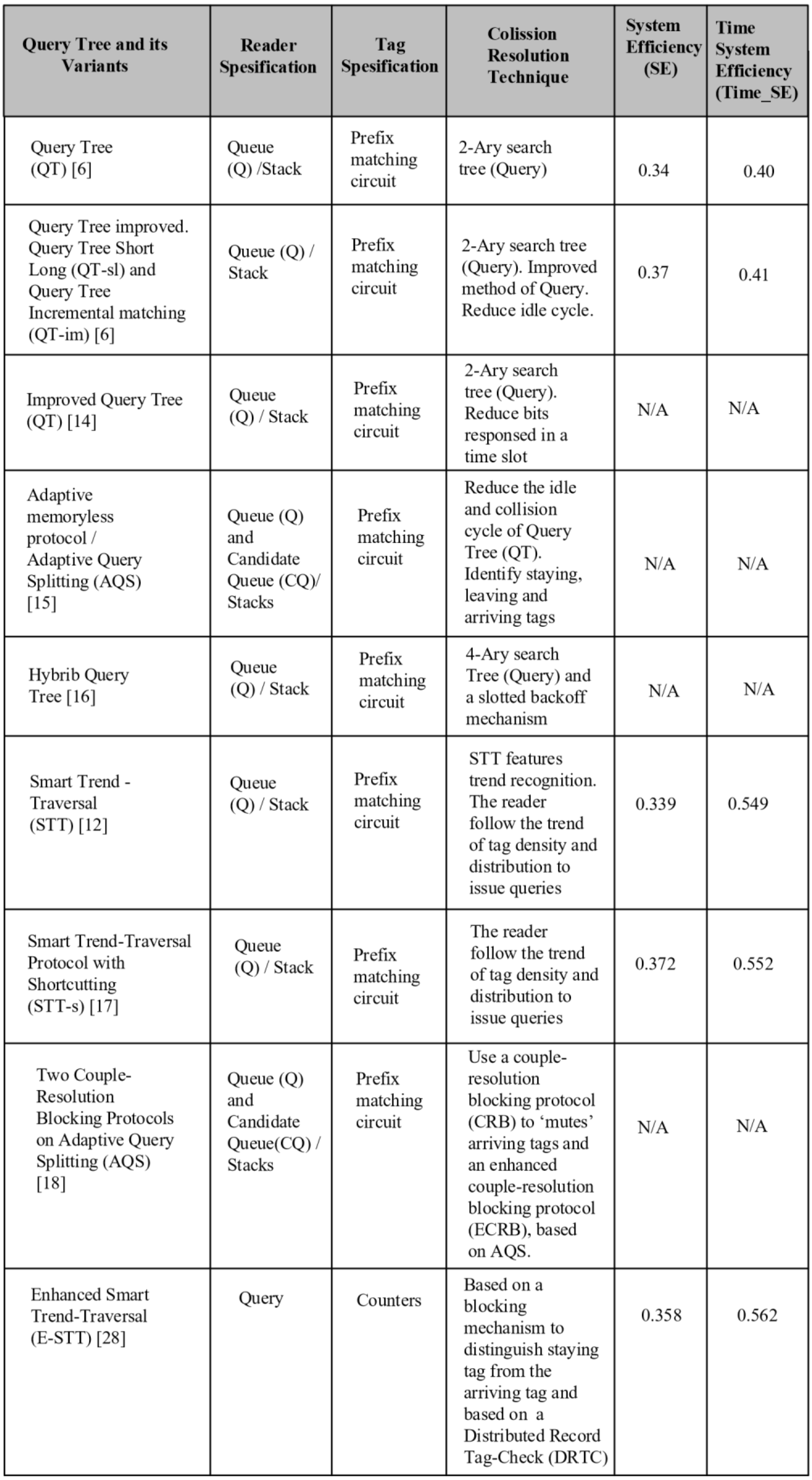


Figure 5. An example of STT protocol

Table 3. Query Tree and Variants



Query Tree and its Variants	Reader Spesification	Tag Spesification	Colission Resolution Technique	System Efficiency (SE)	Time System Efficiency (Time_SE)
Query Tree (QT) [6]	Queue (Q) /Stack	Prefix matching circuit	2-Ary search tree (Query)	0.34	0.40
Query Tree improved. Query Tree Short Long (QT-sl) and Query Tree Incremental matching (QT-im) [6]	Queue (Q) / Stack	Prefix matching circuit	2-Ary search tree (Query). Improved method of Query. Reduce idle cycle.	0.37	0.41
Improved Query Tree (QT) [14]	Queue (Q) / Stack	Prefix matching circuit	2-Ary search tree (Query). Reduce bits responsed in a time slot	N/A	N/A
Adaptive memoryless protocol / Adaptive Query Splitting (AQS) [15]	Queue (Q) and Candidate Queue (CQ)/ Stacks	Prefix matching circuit	Reduce the idle and collision cycle of Query Tree (QT). Identify staying, leaving and arriving tags	N/A	N/A
Hybrib Query Tree [16]	Queue (Q) / Stack	Prefix matching circuit	4-Ary search Tree (Query) and a slotted backoff mechanism	N/A	N/A
Smart Trend - Traversal (STT) [12]	Queue (Q) / Stack	Prefix matching circuit	STT features trend recognition. The reader follow the trend of tag density and distribution to issue queries	0.339	0.549
Smart Trend-Traversal Protocol with Shortcutting (STT-s) [17]	Queue (Q) / Stack	Prefix matching circuit	The reader follow the trend of tag density and distribution to issue queries	0.372	0.552
Two Couple-Resolution Blocking Protocols on Adaptive Query Splitting (AQS) [18]	Queue (Q) and Candidate Queue(CQ) / Stacks	Prefix matching circuit	Use a couple-resolution blocking protocol (CRB) to 'mutes' arriving tags and an enhanced couple-resolution blocking protocol (ECRB), based on AQS.	N/A	N/A
Enhanced Smart Trend-Traversal (E-STT) [28]	Query	Counters	Based on a blocking mechanism to distinguish staying tag from the arriving tag and based on a Distributed Record Tag-Check (DRTC)	0.358	0.562

4.2 Collision Tracking Tree (CT)

Collision tree protocol [19] [20] uses collided bit to split tags into two groups. A stack is used as a prefix pool to hold prefixes for next queries. Initially, an empty string ε is pushed into the stack. After that the reader repeats popping prefix, querying tags, and pushing new prefixes (if new prefixes are generated), until the stack is empty, which means that all tags are identified successfully. At the same time, in CT, the tags only transmit their IDs except the part which is the same as the received. The reader transmit a query including a bit string as prefix. The tags in the range of the reader compare the prefix with their own IDs. If the result of comparing is true, tags transmit their IDs except the part which is the same as the received prefix to the reader. If collision occurs in the response, the reader generates two new prefixes according to the first collided bit in the received bits string, and the tags having answered the query are divided into two subsets. The reader repeats to split the tags into two subsets until the number of tags in a subset is only one. When there is only one tag in a subset, the reader directly identifies the tag. Once a tag is identified, the reader starts a new round of queries with another prefix, until all tags are identified.

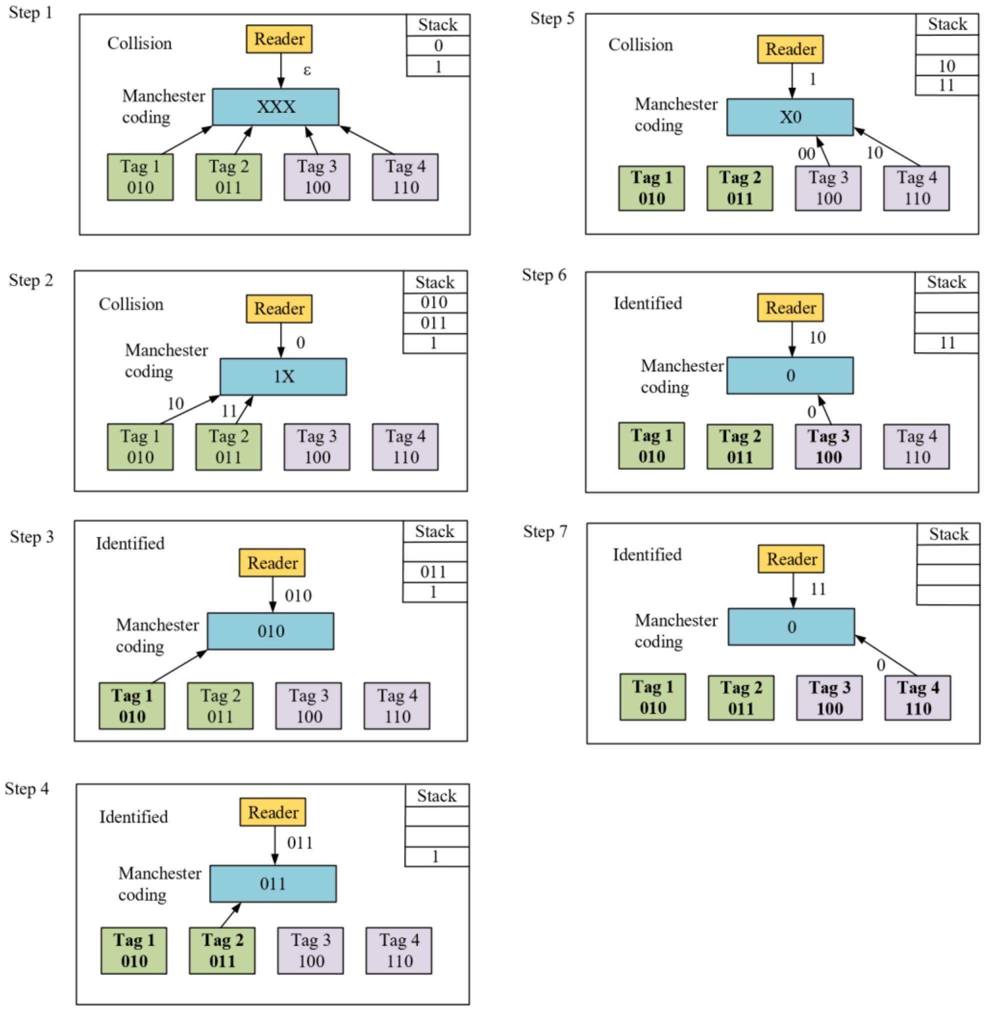


Figure 6. Collision Tracking Tree (CT)

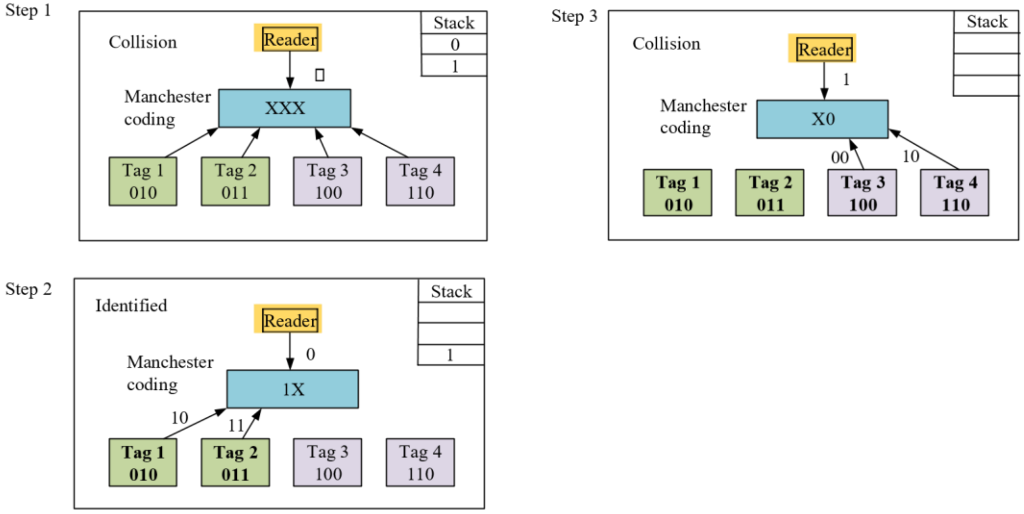


Figure 7. An Improve Collision Tracking Tree (ICT)

Binary Search Fast-2 (BSF2) [21] algorithm has a new approach for solving the collision, the tags are grouped in two groups, as for example of 3-bit ID, the first group has an ID range of 000 to 011 and second group from 100 to 111. This method is more effective, fast detection since tags are grouped in small number which reducing time to solve the collision. The approach is reducing the inquiry step and bit from the reader to tag. Instead of using request of full length of ID, the reader only send bit request as reflex by Manchester coding of collision bit. It will reduce the communication overhead between reader and tag. Also introduce a new approach of backtracking, if the tags are in the first group, the reader send request ‘0’, for backtracking to the first group. After all tags from the first group are selected, then it is searching and backtracking to the second group with reader send request ‘1’. BSF2 is as a robust anti-collision algorithm, the implementation covering small and large number of tags and detection can cope with any possibilities of tag ID arrangement.

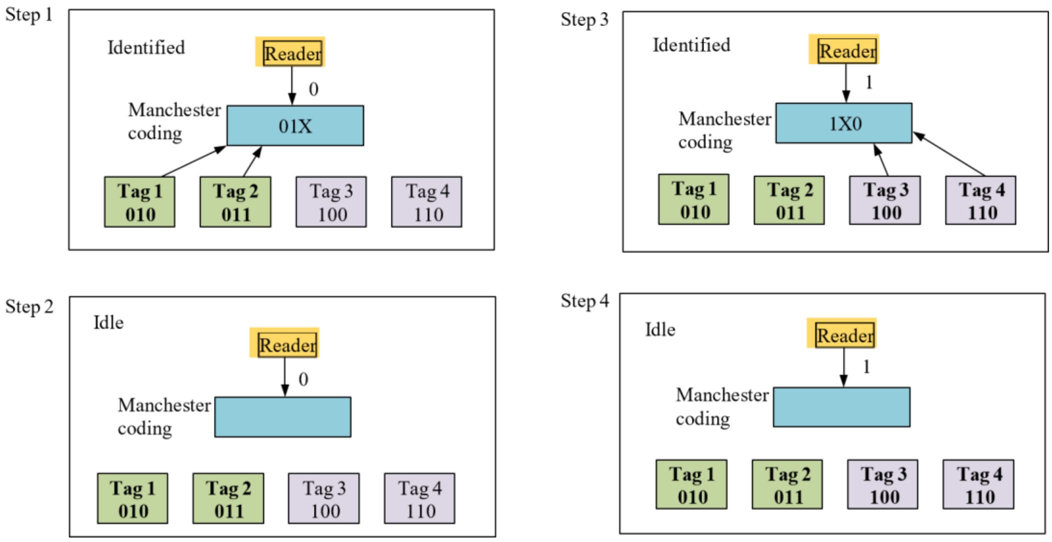


Figure 8. Binary Search Fast-2 (BSF2) [21] Table 4. Collision Tracking Tree and Variants

Collision Tracking Tree and its Variants	**Reader spesification**	**Tag spesification**	**Collision Resolution Techniuque**	**System Efficiency (SE)**	**Time System Efficiency (Time_SE)**
Collision Tree Protocol (CT) [19]	Queue (Q) / Stack, Bit Tracking Technology (Manchester Coding)	Prefix matching circuit	Use the collided bit to split the tag into two groups and generates the new prefix	0.36	N/A
Improved Collision Tree Protocol (ICT) [20]	Queue (Q) / Stack, Bit Tracking Technology (Manchester Coding)	Prefix matching circuit	Use the collided bit to split the tag into two groups and generates the new prefix, apply ‘duality’ to read two tags directly if only one collision bit	N/A	N/A

Binary Search Fast-2 (BSF2) [21]	Queue (Q) / Stack, Bit Tracking Technology (Manchester Coding)	Prefix matching circuit	Grouping the tags into two groups according to MSB bit. Solved the collision group by group separately, use the collided bit to split the tag and generates the new prefix, apply 'duality' to read two tags direcly if only one collision bit	N/A	N/A
Adaptive Collision Tree Protocol (ACT) [22]	Queue (Q) / Stack, Bit Tracking Technology (Manchester Coding)	Prefix matching circuit	Can use 2-Ary or 4-Ary (Query) depending on the calculation of collision factor (z), apply 'duality' to read two tags directly if only one collision bit	N/A	N/A
Optimal Query Tracking Tree Protocol (OQTT) [23]	Queue (Q) / Stack, Bit Tracking Technology (Manchester Coding)	Prefix matching Circuit	Implement bit estimation, optimal partition, and query tracking tree.	0.611 to 0.614	N/A

5. Memory Tree-based Algorithm

Memory based algorithm indication of tag that have components such as counter, generator and extra memory space. Binary Tree Splitting (BTS), Binary Search Tree (BS) and Bit Arbitration (BA) algorithm are categorized in memory based algorithm.

5.1. Binary Search Algorithm (BS)

Binary search algorithm [8] involves the reader transmitting a first string at highest binary bits equivalent to decimal value 2^n-1 to tags, n is the number of bit as following the EPC system's requirement. The tags then compare against its ID. Those tags respond, with ID equal to or lower than the requested string. The reader then monitors tags' response bit by bit using *Manchester coding* as in Figure 2.9 where the value of bit is define by the change in level (negative or positive transition) within a bit window. A logic zero is coded by a positive transition. If the bits of different value then the positive and negative transitions of the received bit cancel each other out, so that a subcarrier signal is received for that duration and it can be seen as *no transition state.* It is not permissible during data transmission and recognized as an error. Once a collision occur in BS, the reader splits tags into subset, based on collided bits. Binary search (BS) is a classical anti-collision algorithm in RFID. The content of this algorithm is simple, but it spends too many timeslots to identify tags, and the length of a timeslot is longer because it's proportional to the number of ID's bit. BS system efficiency is 34% and time system efficiency is 40% [13].

The enhanced version of the BS protocol is called "Dynamic Binary Search Algorithm" (DBSA) [3]. In DBSA, the reader and tags do not use the entire length of EPC and tags ID during the identification process. For example, if a reader receives the response 01X, tags only need to transmit the remaining part of their ID since the reader has already identified the prefix 01. This enhancement effectively reduces the amount of data sent by the reader to tags.

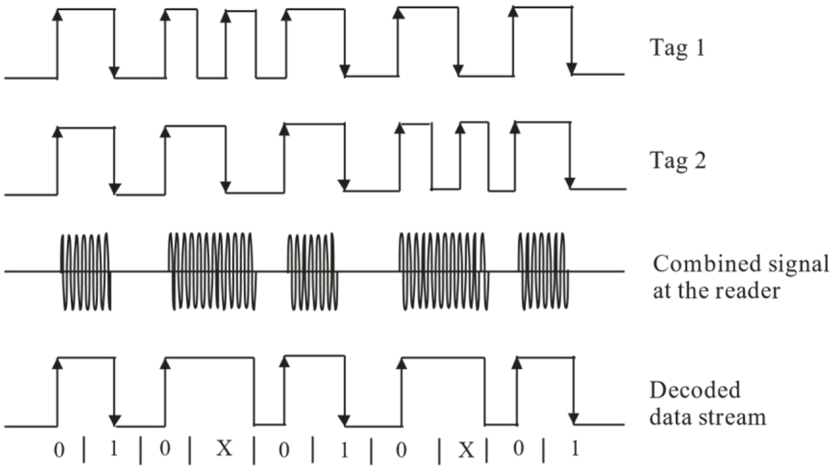


Figure 9. Manchester coding, X is a collision bit.

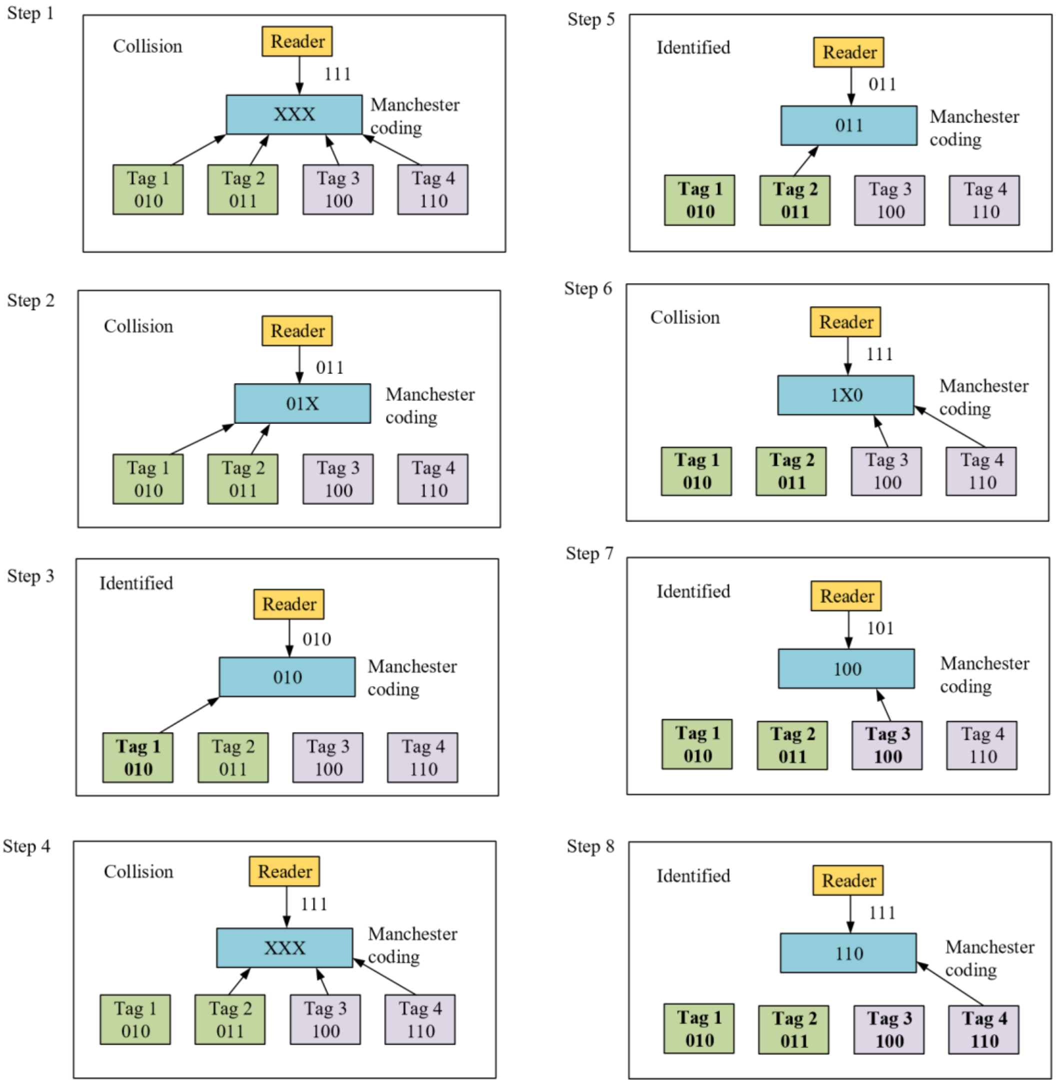


Figure 10. Binary Search Algorithm (BS)

5.2. Tree Splitting Algorithm (TS)

Tree Splitting (TS) protocols operate by splitting responding tags into multiple subsets using a random number generator. Binary Tree Splitting (BTS) [7], an algorithm that performs collision resolution by splitting collided tags into subsets. These subsets become increasingly smaller until they contain one tag. Identification is achieved in a sequence of timeslots. Each tag has a random binary number generator. Each tag maintains a counter to record its position in the resulting tree. Tags with a counter value of zero are considered to be in the transmit state, otherwise tags are in the wait or sleep state. After each timeslot, the reader informs tags whether the last timeslot resulted in a collision, single or no response. If there was a collision, each tag in the transmit state generates a random binary number and adds the number to its current counter value. On the other hand, tags in the wait state increment their counter by one. In the case of idle or single response, tags in the wait state decrement their counter by one. After identification, tags enter the sleep state. Thus, the Tree Splitting (TS) protocol minimizes the number of messages sent by an interrogator but its implementation has more complexity [17].

Adaptive Binary Splitting by [15] recursively splits tags into two sub-groups until obtaining single-tag groups. Each tag maintains a counter, the value of counter will change according to the status whether it is in collision, identification or no answer. If it is in collision, the splitting procedure is by adding a random binary number, 0 or 1 to the counter. Tags having '0' will transmit their ID to the Reader and then the tags having '1' transmit later. The staying tags which were not involved in the collisions increase their counters by one. In case of identification or no answer, all the tags decrease their counter by one, so that those with the counter at one will answer the next query. It is categorized under a memory algorithm, because of additional component in the tag's circuit. The expected number of queries (EQ) needed to identify all tags is $2.881n-1 \leq EQ \leq$
$2.887n-1$, and hence Adaptive Binary Splitting system efficiency is about 34%, n is the number of tags [10].

Table 5. Binary Tree Splitting and Variants

Binary Tree Splitting and its Variants	**Reader spesification**	**Tag spesification**	**Collision Resolution Techniuque**	**System Efficiency (SE)**	**Time System Efficiency (Time_SE)**

Binary Tree [7]	Uses a power interrupting to transmit a simple shutdown code to the tag	Random binary number generator, counter	Splitting responding tags into multiple subsets using a random number generator	N/A	N/A
Adaptive binary splitting (ABS) [15]	Reader's allocated slot counter (Rc)	Random binary number generator, Progressed-slot counter (Pc), Allocated-slot counter (Ac)	Tag splitting and responding to the reader's query according to its counter value. Reduce idle and collision cycle. Cover staying, leaving and arriving tags	0.34	0.40
Enhanced Anticollision Algorithm (EAA) [24]	Bit tracking technology (Manchester coding), reader's allocated slot counter (Rc)	Random binary number generator, Progressed-slot counter (Pc), Allocated-slot counter (Ac)	Based on ABS and a collision tracking (Manchester Coding)	N/A	N/A
New Enhanced Anti-Collision Algorithm (NEAA) [25]	Bit tracking technology (Manchester coding), use counter, stack and temp-string to record the paths	Random binary number generator, Progressed-slot counter (Pc), Allocated-slot counter (Ac)	Based on ABS, a collision tracking, identify multiple passive tags in a timeslot. Reduce the number of timeslots and the length of a timeslot.	N/A	N/A
Modified AntiBinary Collision Protocol (MABS) [26]	Reader's allocated slot counter (Rc)	Random binary number generator, Progressed-slot counter (Pc), Allocated-slot counter (Ac)	Based on ABS, adopt a blocking technique.	N/A	N/A
Optimal Binary Tracking Tree Protocol (OBTT) [27]	Bit Tracking Technology (Manchester Coding), Reader Counter	Counter	Implement bit estimation, optimal partition, and query tracking tree.	0.611 to 0.614	0.75

6. Conclusion

Memoryless based anti-collision algorithm is preferable implemented with passive RFID system with less complexity in tags and reader deployment. It is a 'reader-driven'. ID distributions, length and density of the tags are the factors that influence the performance of the system because these are the information that mapping the root of the tree that need to be solved by algorithm. Memory based algorithm is a 'Tag-driven'. Tag can remember its transmitting or waiting state. The reader send query and the tag on transmitting-state will response to the query. The capability of 'memory' make the tag independent and yet does not rely on length of ID. A new tags can be inserted in the interrogation zone and a staying tags also could leave. However the tag's density will degrade the performance because the tags need splitting to the singleton then it can be identified. This contribute to the delay of the system in which the System Efficiency (SE) and Time System Efficiency (TimeSE) has not much different with memoryless algorithm. This paper is a review of deterministic anti-collision algorithm and the improvement of resolution collision techniques in order to achieve a better efficiency with regards of the passive RFID system.

References

[1] D. M. Dobkin (2008). The RF in RFID: Passive UHF in Practice. Elsevier.

[2] ISO / IEC.ISO / IEC 18000. RFID for Item Management: Air Interface. Part 5: Parameters for Air Interface Communication at 5.8 GHz, Jan. 2000. [3] Kuo-Hui Yeh, Nai-Wei Lo, Enrico Winata. A Code Expansion Based Tag Identification Protocol for RFID Systems Department of Information Management.

[4] Shun Xue. MASTER'S THESIS. Multi-tag content access control in RFID system.

[5] Jihoon Choi and Wonjun Lee (2007). Comparative Evaluation of Probabilistic and Deterministic Tag Anti-collision Protocols for RFID Networks.

[6] C. Law, K. Lee, and K. Y. Siu (2000). Efficient memoryless protocol for tag identification. Workshop Discrete Algorithms Methods Mobile Computer Communication. pp. 75–84.

[7] Shun Chan, Harley Heinrich, Dilip Kandlur, Arvind Krishna (1996). Multiple item radio frequency tag identification protocol, U.S. Patent: US5550547.

[8] Klaus Finkenzeller (2003). RFID Handbook: Fundamentals and Application in Contactless Smart Cards and Identification. John Wiley & Sons, 2nd edition. [9] M. Jacomet, A. Ehrsam and U. Gehrig (1999). Contact-less identification device with anti-collision algorithm. In Proc. Conf. Circuits System Computer Communication.

[10] Giuseppe Bagnato, Gaia Maselli, Chiara Petrioli, Claudio Vicari (2009) Performance analysis of anti-collision protocols for RFID systems. IEEE.

[11] J. Massey (1981). Collision resolution algorithms and random-access communication. Multi-Users Communication Networks, CISM Courses and Lectures, no. 256, pp. 73–137.

[12] Lei Pan, Hongyi Wu (2011). Smart Trend-Traversal Protocol for RFID Tag Arbitration. *IEEE Transactions on Wireless Communication,* Vol 10. No. 11.

[13] Shuen Chih Tsai, Yu Min Hu, Chen Hsun Chai, Jun Shian Li (2016). Efficient tag reading protocol for Large-Scale RFID systems with pre-reading. ELSEVIER Computer Communication, pp 73-83.

[14] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min. (2004). Evaluating and optimizing power consumption of anticollision protocols for applications in RFID systems. In Proc. Int. Symp. Low Power Electron. Design, pp. 357–362.

[15] J. Myung and W. Lee (2006). Adaptive binary splitting: A rfid tag collision arbitration protocol for tag identification. *Mob. Netw. Appl.,* vol. 11, no. 5, pp. 711–722. [16] Jiho Ryu, Hojin Lee, Yongho Seok, Taekyoung Kwon and Yanghee Choi (2007). A Hybrid Query Tree Protocol for Tag Collision Arbitration in RFID systems.

[17] Xin-Qing Yan, Rui-Xia Zhang, Bin Li (2012). Smart Trend-Traversal Protocol with Shortcutting for Memory-less RFID Tag Collision Resolution. 9th International Conference on Ubiquitous and Computing and 9th International Conference on Automatic and Trusted Computing. Pp 857-861.

[18] Yuan-Cheng Lai and Chih-Chung Lin (2012). Two Couple-Resolution Blocking Protocols on Adaptive Query Splitting for RFID. IEEE Transactions on Mobile Computing, Vol. 11, No. 10.

[19] Xiaolin Jia, Quan Yuan Feng, Chengzhen Ma (2010). An efficient Anti-Collision Protocol for RFID Tag Identification. IEEE Communication Letters, Vol. 14, No.11. [20] Xiaolin Jia, Quan Yuan Feng (2015). An improved anti-collision protocol for radio frequency identification tag. International Journal of Communication System.

[21] Marizan Yaacob, Salwani Mohd Daud (2012). Novel Binary Search Algorithm for Fast Tags detection in Robust and Secure RFID systems. Communications in Computer and Information Science, pp 575-582.

[22] Liu Xiaohui, Qian Zhihong, Zhoa Yanhang, Guo Yuqi (2014). An Adaptive Tag Anti-Collision Protocol in RFID Wireless Systems.
[23] Yuan-Cheng Lai, Ling-Yen Hsiao, Hong-Jie Chen, Ching-Neng Lai, and Jian-Wei Lin (2013). A Novel Query Tree Protocol with Bit Tracking in RFID Tag Identification
[24] Wei-Chih Chen, Shi-Jinn Horng, Pingzhi Fan (2007). An enhanced anti-collision algorithm in RFID based on counter and stack. Proc. 2nd ICSNC, pp 21–24.
[25] Yuan-Hsin Chen, Shi-Jinn Horng, Ray-Shine Run, Jui-Lin Lai, Rong-Jian Chen, Wei-Chih Chen, Yi Pan, Terano Takao (2010). A Novel Anti-Collision Algorithm in RFID Systems for Identifying Passive Tags.
[26] Mohamed Mostafa, Abd Allah (2011). Modified Anti-Binary Collision Protocol (MABS) of Radio Frequency Identification (RFID) System. Journal of Wireless Networking and Communications
[27] Yuan-Cheng Lai, Ling-Yen Hsiao, and Bor-Shen Lin (2015). Optimal Slot Assignment for Binary Tracking Tree Protocol in RFID Tag Identification.
[28] Shuen-Chih Tsai, Yu-Min Hu, Chen-Hsun Chai and Jung-Shian Li (2016). Efficient tag reading protocol for largescale RFID systems with pre-reading, Computer Communications 88, pp 73–83.