Global Scientific JOURNALS

GSJ: Volume 11, Issue 9, September 2023, Online: ISSN 2320-9186

www.globalscientificjournal.com

AN IMPROVED RFID IDENTIFICATION SYSTEM USING HYBRID ANTI-COLLISION ALGORITHMS

Orji Mary[1], Dr. John Abiodun Oladunjoye[2], Dr. Agu Edward O[3], Dr. Hambali M.A[4]

Department of Computer Science, School of Basic and Applied Sciences,Taraba State Polytechnic Suntai, Jalingo Campus Taraba State, Nigeria[1].

Department of computer science, faculty of computing and information sciences Federal University Wukari, Taraba State, Nigeria[2,3,4].

Marybless200@yahoo.com

Abstract

The Radio Frequency Identification (RFID) is a wireless communication technology that enables automatic identification, tracking and data collection from any tagged object. A simple RFID system uses radio signals to transmit data using a tiny portable device called tag or card, which is read by an RFID reader and processed by sending and receiving signals. One of the important performance issues in this system is to resolve RFID tags collision. Tags collision happens when two or more tags reflects-back their individual identification radio signals to the reader at the same time that is confusing the reader identification process. Different Anti-Collision Algorithm solutions on tag collision has been achieved though depending on the type of collisions they used. There are generally two main categories of anti-collision problems: ALOHA-based solutions and tree based solutions. However, ALOHA-based Algorithms are of different categories likewise the Tree-based Algorithms. In this research, two Anti-Collision Algorithms (Hybrid) is used combining the characteristics of these two categories (the framed slotted ALOHA algorithm, a typical ALOHA based algorithm, and the query tree algorithm, a typical tree based algorithm) are presented, a better performances of the two algorithms are simulated when compared to other existing Algorithms in RFID system.

Keywords: RFID, Tags, ALOHA-based Solution, TREE- based Solution, Hybrid Anti – Collision Algorithms.

1. INTRODUCTION

Due to security threats and terrorism that are currently plaguing the State and the Nation as a whole, identification of persons and vehicles are always important in places like Airports, Railway Stations, Schools and Companies etc. Identification can be made automatic using Auto-identification; different types of auto-identification method has been in used, some of them are Bar-code System, Optical Character Recognition, Biometrics, Smart cards and RFIDs, of which RFID Technology is a revolution.

What is RFID? Radio Frequency Identification stores and remotely retrieves information or data on/from its tags using an RFID reader/writer and a backend database. Unique identification numbers of objects are stored on tags and are retrieved remotely. RFID technology depends on the communication between the RF ID tags and RFID readers (Ashane, Shah and Kingston, 2017). RFID is a generic term that uses radio waves to automatically identify people or objects (Modin, 2016).

Like all other systems, the operations of RFID systems are based on certain principles. Manti, Seong and Kim (2016) says many types of RFID exist, but at the highest level, RFID devices/tags can be divided into two classes: Active and Passive. Active tags require a power source that is, they are either connected to a powered infrastructure or use energy stored in an integrated battery. In the latter case, a tag's lifetime is limited by the stored energy, balanced against the number of read operations the device must undergo, while Passive Tags does not have an internal power source, they utilized the electromagnetic waves received from a reader. Once a reader transmits to the tag, and an antenna inside the device creates a magnetic field. The tag circuit uses the power generated to transmit data back to the reader.

However, Morocco (2019) suggests that there are three categories of RFID devices. These devices can be passive, drawing power from the reader, semi-active, with a battery only for powering sensors, or fully active, with a local source providing direct energy to both the microcontroller and the radio transmitter. Each tag has its own memory that can be read-only or rewriteable, depending on the type and purpose. This memory is often used to store information about the product, such as a unique ID and manufacturing date. The RFID reader generates magnetic fields that enable the RFID system to locate objects (via the tags) that are within its range. The high-frequency electromagnetic energy and query signal generated by the reader triggers the tags to reply to the query; the query frequency could be up to 50 times per second. As a result, communication between the main components of the system that is tags and reader is established (Ashane et al., 2017).

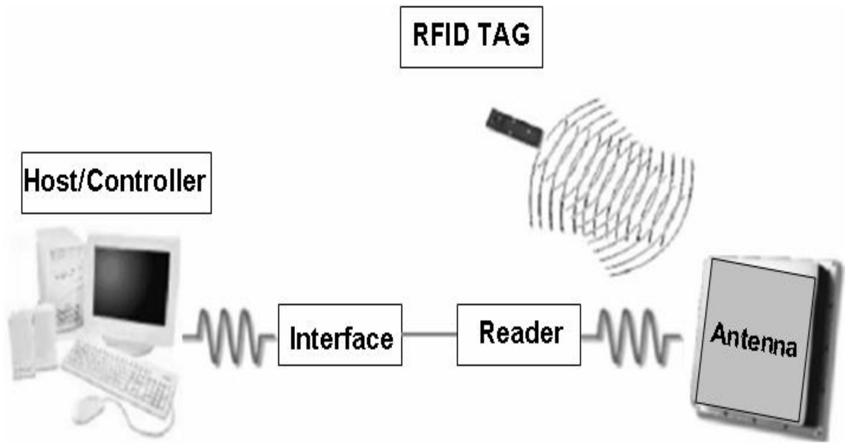


Figure 1.1: Configuration of RFID system **(Source:** Indiamart.com)

RFID consists of three basic components such as transponder (tag), interrogator (reader) and antenna. In a typical communication sequence, RFID system performs a number of Functionalities between reader and tag. RFID reader emits a continuous RF carrier sine wave. When a tag enters the RF field of the reader, the tag receives energy from the field. Further, receiving sufficient energy, it begins to modulate the carrier signal to the data storage on the tag. The modulating carrier signal is resonated from the tag to the reader. The reader detects the modulating signal from the tag, and decodes signal in order to retrieve the data from the tag. However, the information relays to the host computer where more manipulation data will be stored and finally will be displayed to the user.

Since the communication between the readers and the tags shares wireless channels, there exist collisions. The collisions can be divided into; Reader and Reader, Reader and Tag as well as Tag and Tag collision. The Reader collision occurs when the multiple readers send request signal to one tag, and the tag receives the wrong request signal due to signals interference between readers. (Myung and Lee, 2017) and the tag collision occurs when more than two tags simultaneously response to one reader and the reader cannot identify any tags.

This kind of collision makes the reader take long time to identify tags within the reader's identification range and impossible to identify even one tag. (Engels and Sarma, 2017).

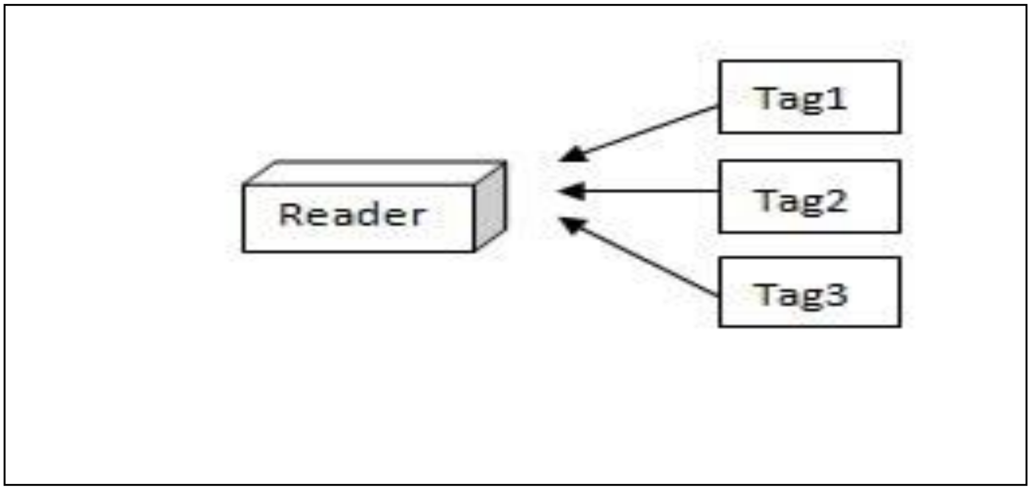


Fig 1.2: Tag collision problem in RFID system
(Source: Indiamart.com)

In general, the tag collision techniques can be classified into two categories: Aloha-Based and Tree-Based protocols. Aloha-Based approach uses time slot to reduce collision probability, such as Aloha, Slotted Aloha, framed-slotted Aloha algorithm and Dynamic Framed slotted Aloha, (Vogt and Park, 2019), which reduce the probability of the occurrence of tag collision since tags try to answer at distinct time. The Tree-based schemes use a data structure similar to a binary search algorithm, such as binary tree splitting protocol. (Myung, Lee and Srivastava, 2019), Query Tree (QT) algorithm, and tree working algorithm. (Lee, Joosd and Lee, 2020).

Tags randomly selects a particular slot in the time frame and transmit its identification to the reader, once the transmission is collided, tags will repeatedly send its ID in next interval of time to make sure its ID is successfully recognized. Therefore, when the number of RFID tags increases, the tag collision rate will increased as well: these may result to a low tag recognition rate.

So many studies to resolve this problem have been carried out as well as are still ongoing. Therefore, in this research work two Algorithms combination will be used to solve this problem: Hybrid tags Anti-collision Algorithm that is the framed slotted Aloha Algorithm and the Query Tree Algorithm. The performances of the two Algorithms are compared through simulations with an existing other Anti-collision Algorithms used in RFID systems.

2. Related work

RFID technology is an automatic identification technology of contactless method that identifies electronic tags attached to goods Ammiu J. Singh, E. Olsen, K. Vorst, and K. Tripp, (2017) For this RFID technology to be widely used, multiple tag identification problem must be solved in the first place. This problem is defined as a one-to-many communication problem between tags, this has to be identify by receiving the information transmitted from tags without collision in case a multiple number of tags exist within the identification area of a reader.

ALOHA based algorithms usually refer to the slotted ALOHA algorithm, an algorithm in which only one tag is make to respond in a slot, in the response of tags, by dividing a time into slot units. On the other hand, tree based algorithms make trees while performing the tag identification procedure using a unique ID of each tag. Hybrid tags Anti-collision Algorithm that is the framed slotted Aloha Algorithm and the Query Tree Algorithm is used in the research.

2.1. Framed Slotted ALOHA Algorithm

FS-ALOHA algorithm Schoute F.C (2018), is the most well-known of anti-collision algorithms used for solving the collision of tags in RFID systems. In FS-ALOHA algorithm, when a reader requests tags to transmit their ID, it also transmits a frame size (FS). On receiving ID transmission request from a reader, a tag randomly decides its own transmission slot within the frame size, and then transmits its ID after waiting for its turn. On the reader side, three kinds of cases can occur. To begin with, there is a case receiving no response to the slot. This is referred to as "no response", and the number of no responses in a frame is expressed as C_0.

The second one is a case only one tag has responded. This is referred to as "identification", and the number of identifications in a frame is expressed as C_1. Lastly, there is a case two tags or more attempt to transmit in the same slot. So, a collision takes place and the data transmitted by tags is lost. This is referred to as "collision", and the number of collisions in a frame is expressed as C_k.

Table 1.3 illustrates the operation of FS-Aloha algorithm using four tags. The reader requests tags to transmit their ID along with sending 4 as a frame size, and then each tag selects its own slot and attempts to transmit its ID.

In Slot1 and Slot4, only one tag attempted to transmit and thus the reader successfully identifies Tag2 and Tag3, then send the tags an Ack command informing it has identified, in order to keep tags from responding in the next frame. In Slot3, however, as there is no tag, it becomes no response. While in Slot2, a collision occurs since Tag1 and Tag4 sent their ID at the same time. Finishing a frame, the reader requests the retransmission of ID to the remaining tags. At this point, the number of remaining tags is estimated using the C_0, C_1, and C_k, and then the next frame begins by changing to a frame size suitable to the number of remaining tags, Vogt .H.(2019).

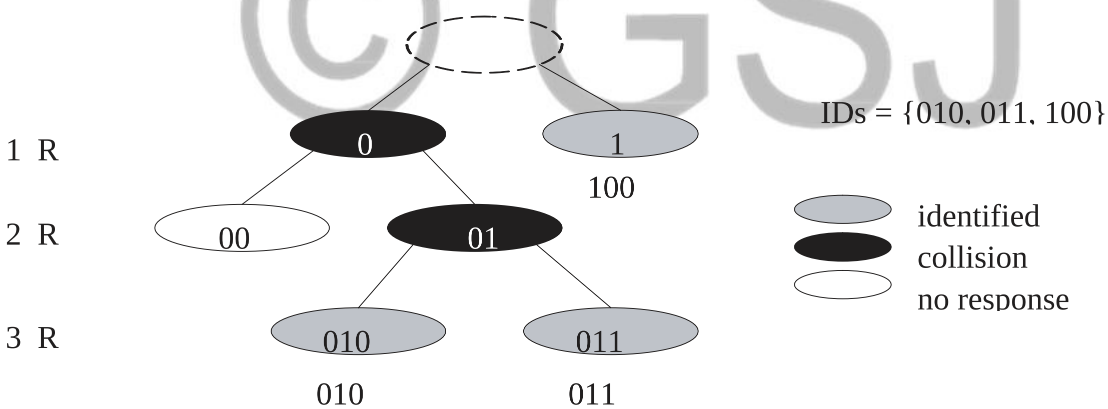
Forward link	Request	Slot1	Ack 1011	Slot2	Slot3	Slot4	Ack 0111	Request
Return link		1011		Collision		0111		
Tag1				0010				
Tag2						0111		
Tag3		1011						
Tag4				1110				

Frame size = 4

Table.2.1. An example of tag identification process in FS-ALOHA algorithm

2.2. Query Tree Algorithm

QT Algorithm Law et, al (2017), is a typical one of the tree based algorithms. When requesting tags to transmit their ID, a reader sends a prefix P_k of k bits together.



Query	0	1	00	01	010	011
Response	collision	Identified	no response	collision	identified	identified

Table 2.2. An example of identification process in QT

Then each tag confirms whether it is the same as the beginning part of its own ID, and responds its own ID to the reader if it is the same. Likewise as in FS-ALOHA algorithm, at this point, the three cases, “no response”, “identification”, and ”collision”, can occur. Here, when a collision takes place, the reader knows there are many tags with the same prefix. In that case, two new prefix P_{k+1} of $k + 1$ bits, which ”0” and ”1” are added at the end of the prefix that has just been transmitted, are made and placed in the queue. The prefix placed in the queue is queried again later. The initial value of the queue is”0” and”1”.

Table 1.2 is an example of having executed QT algorithm on the assumption of three tags whose identifying IDs are ”010”, ”011”, and ”100”, respectively. Prefix”0” and”1” are set up in the initial queue, and the reader queries tags by taking out prefixes in the queue. For a start, when ”0” is queried, Tags ”010” and ”011” respond at the same time because the prefix is the same as their ID. Again, the reader judges when there are two tags or more starting with ”0” and enters ”00” and ”01” in the queue. Thereafter, prefix ”1” is taken out from the queue and queried. Since there is only one tag, ”100”, it is identified normally to the reader. The first round ends by the method like this. And another round begins for ”00” and ”01” which has been previously placed in the queue. This algorithm ends when the queue is empty.

3. The Proposed System

3.1. Hybrid Anti-collision Algorithms

In FS-ALOHA and QT algorithm explained above, when there are many tags a reader wants to identify, there exists many tags that will respond at the same time and thus many collisions take place. This ultimately makes the time of identifying tags longer. Framed query tree (FQT) algorithm and query tree ALOHA (QT-ALOHA) algorithm presented in this research work are hybrid forms of FS-ALOHA algorithm and QT algorithm.

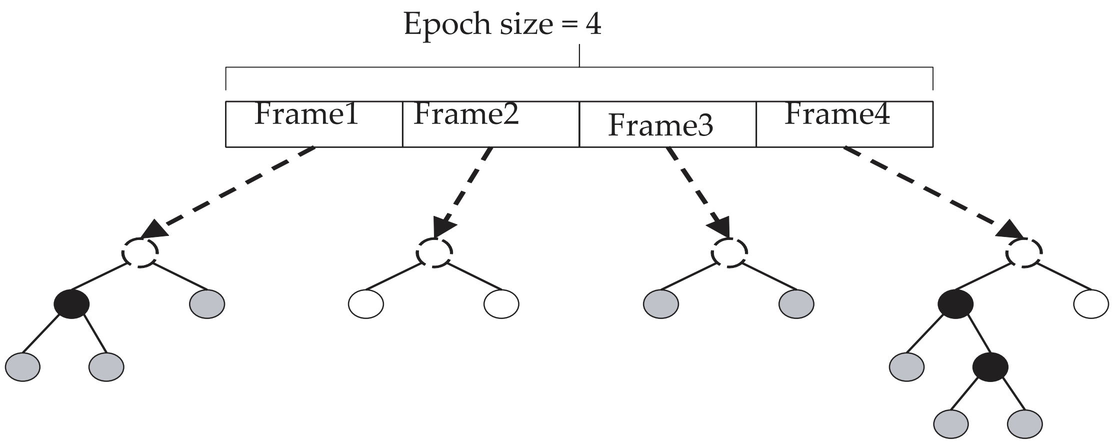


Fig. 3.1. An example of identification process in FQT Algorithm

3.2. Framed Query Tree Algorithm

FQT algorithm divides tags randomly into frame units. And within this unit, tags are identified using QT algorithm for them. The actual operation is as follows:

When requesting tags to transmit their ID, a reader also send an epoch size, the number of the total frames. Then each tag decides its own participating frame randomly and responds only when the reader queries its own frame. The reader executes its identification process within each frame using QT algorithm. However, the reader transmit to tags the number of frame as well as the prefix of ID.

Each tag confirms whether the number of frame is the same as its own selected number of frame, and if it is the same, transmits its own ID when looking the prefix and it is consistent, as in the existing QT algorithm. After identifying all tags within a frame through QT algorithm identification process in the frame, the reader proceeds to the next frame. This process is carried out repetitively for every frame.

The example of Fig.3.1 above shows the identification process of FQT algorithm when the number of tags to be identified is 8 and the epoch size is 4. To begin with, a reader transmits the epoch size to tags and then each tag selects its own frame randomly. For three tags which selected Frame1, QT algorithm is used for identifying tags. Thereafter, tags are identified while proceeding to the next frames by the same method.

To determine the most appropriate epoch size is very important in improving performance. By intuition, an epoch size which a tree depth will not exceed 2 when executing QT algorithm exhibits the best performance. The reason is that when executing ”0”, ”1” in the initial queue of QT algorithm, the case that two tags in total, one tag beginning with ”0” and another tag beginning with ”1”, are identified is the most ideal. Frame3 is exactly the best case. Assuming the number of tags to be identified is N and the epoch size is *ES*, the most ideal *ES* can be expressed as follow:

$N = 2 * ES$

This can be actually verified through the simulation in section 4.1. But there is a big problem with this case. For it is difficult to determine an suitable epoch size from the beginning since the tag identification procedure is initiated under the condition not knowing *N*, in other words, how many tags are to be identified. From that reason, the final FQT algorithm uses FFT (First Frame Test). The FFT begins from a small epoch size and stops its identification process when the first frame has collisions exceeding a collision threshold, and then resumes its identification of tags by increasing the epoch size. As it is assumed that all tags are randomly divided in frames, if many collisions occur in the first frame, remaining frames are more likely to have such a trend.

As shown in the above best case, Frame3 in Fig.3, a tree can have the best performance when its tags are two with the depth being 1. Thus a threshold is needed in order to prevent the tree from becoming deeper than this. The collision threshold is a constant based on the concept that the more collisions happen, the deeper the tree becomes.

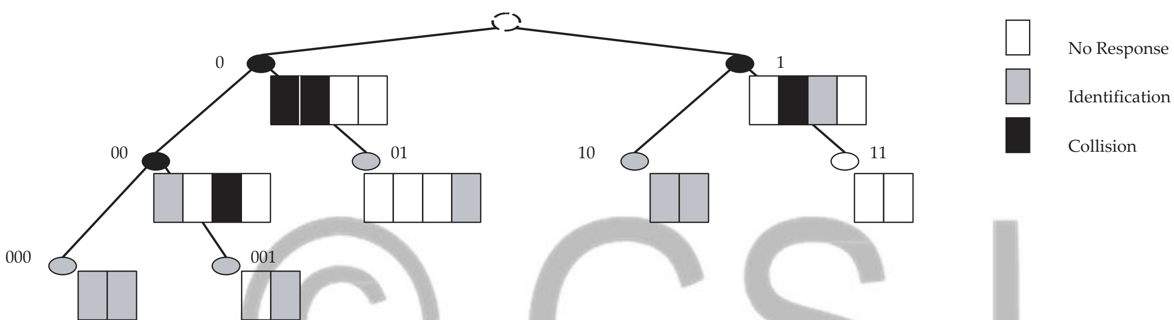
As a result of simulations conducted many times, the researcher have verified that an appropriate epoch size is approached faster than any other cases when this collision threshold is set at 3. In case it is smaller than 3, even when the epoch size is adequate, the epoch size becomes big and thus may be passed over. On the contrary, when it is bigger than 3, the epoch size gets too big and thus the speed to increase to a suitable epoch size becomes too slow. So the collision threshold is assumed to be 3 in this research work.

3.3. Query Tree ALOHA Algorithm

QT-ALOHA algorithm is another hybrid form of FS-ALOHA and QT algorithm. FQT algorithm basically implements FS-ALOHA algorithm and uses QT algorithm as actual tag identification process. On the other hand, in QT-ALOHA algorithm, QT algorithm is a big picture, while actual tag identification process progresses with FS-ALOHA algorithm. This operation is as follows:

On requesting tags to transmit their ID, a reader sends a prefix and a frame size together. Then only tags which are consistent with their own prefix proceed to FS-ALOHA algorithm with the transmitted frame size. And in the progress of FS-ALOHA algorithm, if a collision takes place even in a single slot, it is interpreted as a collision of QT algorithm, and then a new prefix is made and entered in the queue. At this point, a difference from QT algorithm is to calculate a frame size Vogt .H. (2019), to be transmitted next and also place this in the queue.

Fig.3.2. is an example of QT-ALOHA algorithm. It is assumed that the number of tags to be identified is 8 and the initial frame size begins from 4. In the first round, the reader transmits to tags a prefix ”0” and a frame size 4. And collisions have occurred in the frame. Then, ”00” and ”01” are entered in the queue and the frame size is determined as 4 through the calculation process of frame size. And then ”1” is taken out from the queue. When the queue becomes empty, the algorithm ends.



Query	0	1	00	01	10	11	000	001
Frame Size	4	4	4	4	2	2	2	2
Response	collision	collision	collision	identified	identified	no res.	identified	identified

Fig.3.2. An example of identification process in QT-ALOHA algorithm

4. Flow Chart

The reader sends query signal to all the tags. For example if 0001 is the data related to the tag1, the data is encoded and send to all the tags present in the readers interrogation area. All tags will decode the data and checks whether the data and address are correct or not. If match occurs then VT (valid transmission) of the corresponding tag will be high which makes the TE (transmission Enable) low of the encoder for sending acknowledgement signal to reader. once the acknowledgement signal is received then it send data serially to PC for display purpose and again start querying for next tag and continue the process until all tags are identified as flowchart.

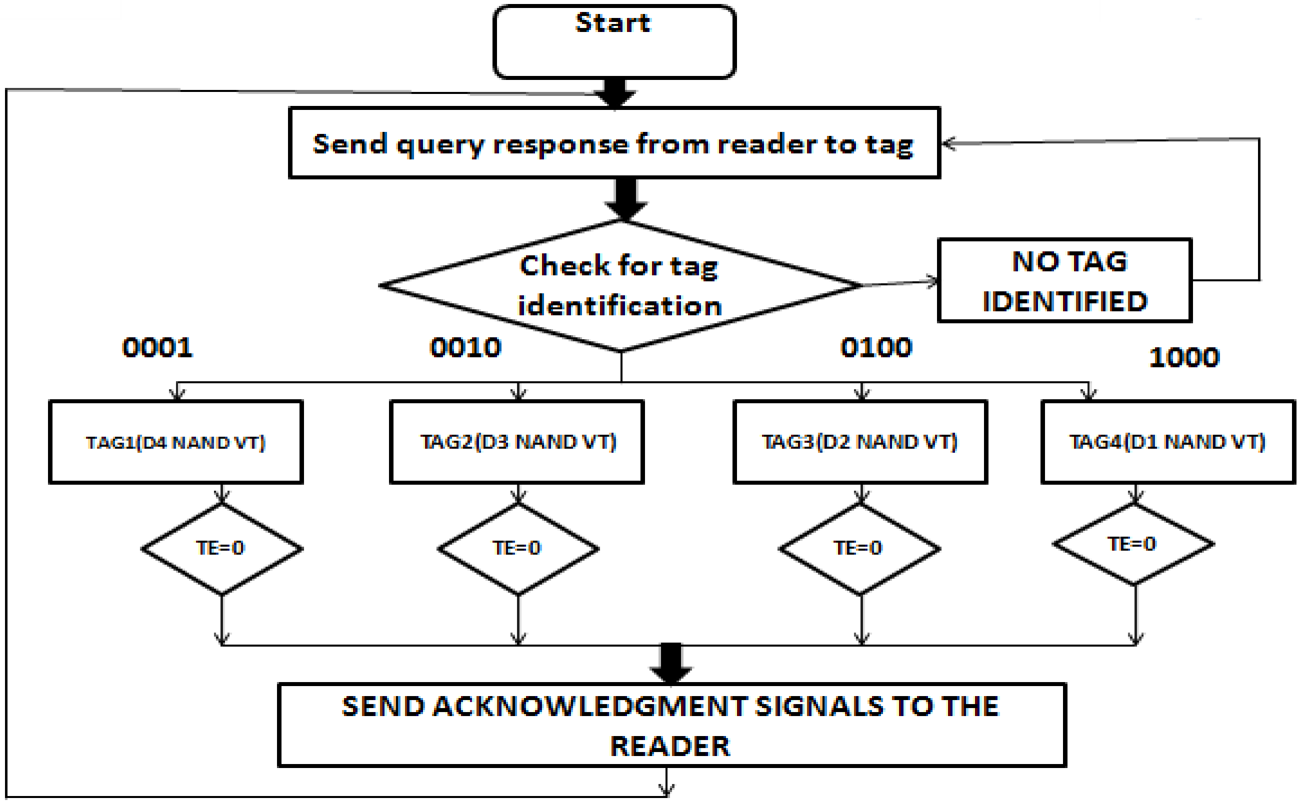


Fig.4.1. Flow Chart of RFID

5. Algorithm for Multiple Tag Identification

Step1: Reader initialization

Step2: Sends the query to all the active tags

Step3: Decoder of each tag checks for corresponding query

Step4: If match occurs then (valid transmission) VT=1

Step5: If VT=1 then (transmission enable) TE=0 for transmitter to send acknowledgement

Step6: Then VT bit of reader will be active high and data will be sent

Step7: Goto step2

6. Simulations

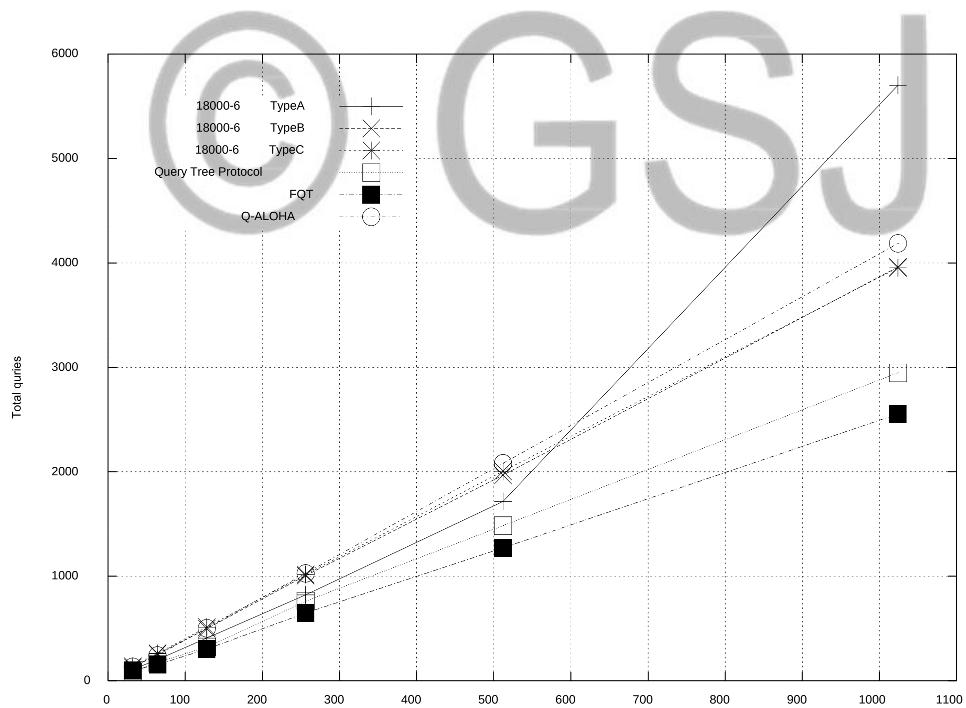
The ID of tags used in simulations was set at a size of 64 bits according to the international standard and was created using random number generator. And for raising the reliability of tag simulation results, 100-time simulations were conducted for every same environment and average it.

7. Epoch Size of FQT Algorithm

The first simulation was conducted with the epoch size being changing under the condition of 100 tags. Reviewing this results, they can be divided into three cases according to the epoch size. The first case is at 32 and 64, close to about 50, an ideal epoch size. This case demonstrates the best performance as mentioned in section 3.2. The second is a case smaller than the ideal epoch size. But in this case, an optimum epoch size is found by the first frame test. And the last is a case bigger than the ideal epoch size. As there is no operation to reduce the epoch size in FQT algorithm and thus the initial epoch have to be entirely executed.

Table 7. Performance comparison according to Epoch Size (*ES*)

Frame Size	Query	C_0	C_1	C_k
16	**246.6**	**55.1**	**100.0**	**81.2**
32	**238.2**	**56.9**	**100.0**	**78.2**
64	**241.3**	**85.9**	**100.0**	**54.1**
128	**327.2**	**193.6**	**100.0**	**32.4**
256	**550.2**	**430.6**	**100.0**	**18.6**



The number of tags

Fig.7. Comparison of query-response number

7.1. Performance Comparison between Presented Algorithms and Other Ones

The second simulation has compared the presented FQT algorithm and QTALOHA algorithm with other anti-collision algorithms, 18000-6 Type A, Type B, Type C, and QT algorithm. This comparison sets the query-response number of times between a reader and tags as a comparison value while changing the number of tags from 32 to 1,024. On the other hand, in the algorithms based on FS-ALOHA algorithm, such as Type A, Type C and QT-ALOHA algorithm, the initial frame size is started arbitrarily at 32 regardless of the number of tags. The epoch size in FQT algorithm also begins at 32.

Fig.7 and Fig. 7.1 are the results of simulations. 18000-6 Type A and Type C use the same operating method, FS-ALOHA algorithm. But as the maximum frame size of Type A is 256, if executing the algorithm at the number of tags higher than it, a significant degradation in performance can be seen. Type C, however, if there occurs many collisions or no-responses even in the middle of a frame, stops the on-going frame and proceeds to the next frame, thereby improving the performance. Type B uses binary tree algorithm, one of the tree based algorithms.

The results show that the tree based algorithm typically is lower in the query response number of times than the ALOHA based algorithms. For tags identified in a frame of the ALOHA based algorithm send tags a command they have been identified so they cannot be included in the identification process of the next frame.

Fig.7.1 illustrates how many queries-responses are needed in order to identify one tag. Looking into this, it can be seen that FQT algorithm queries less for identifying one tag than any other algorithms. Its performance improvement is 10 to 50 percent of times than many other existing anti-collision algorithms.

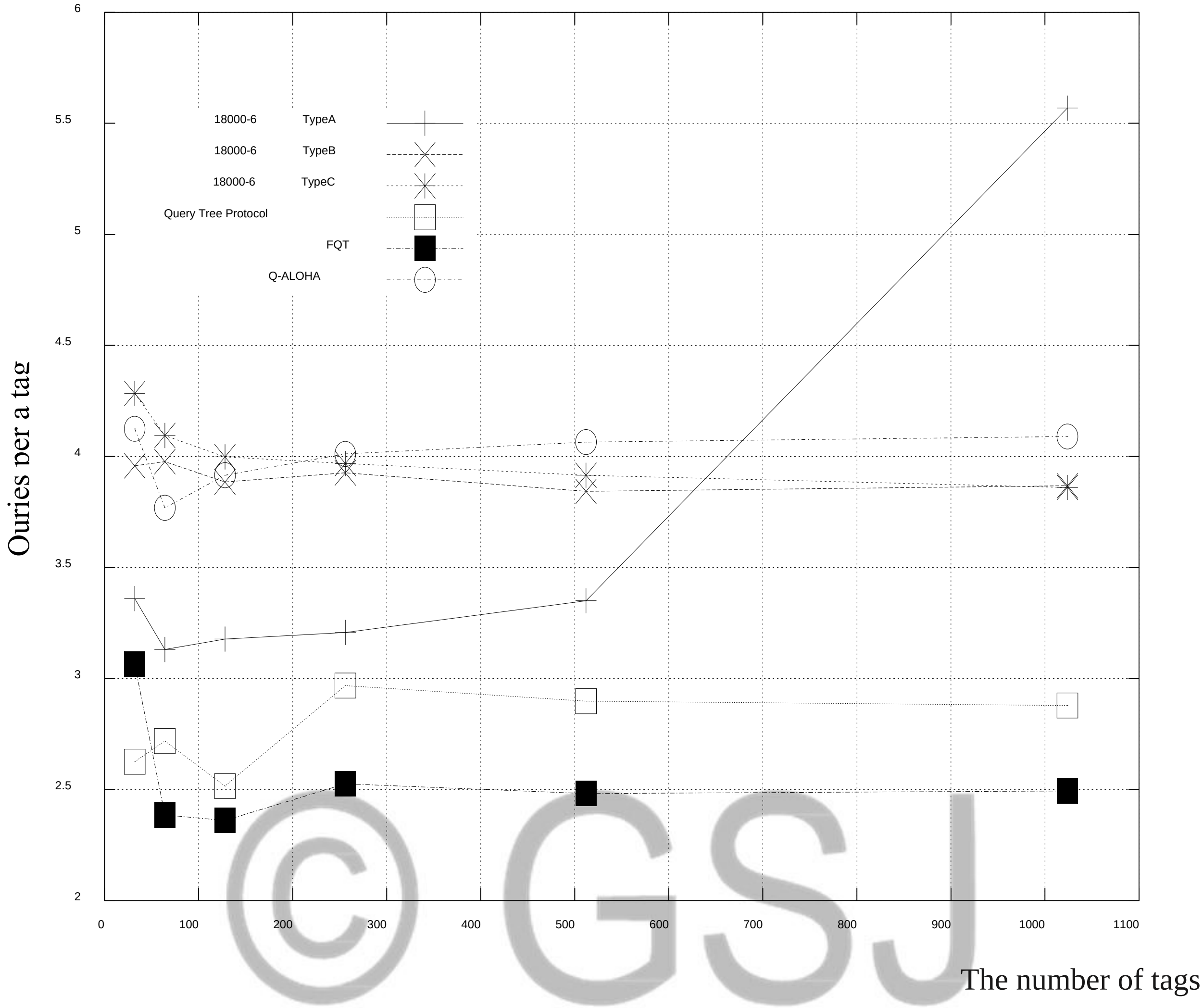


Fig.7.1. Comparison of query-response number needed for identification per tag.

8. Summary

This research work identifies the flaw in the existing system of collision in RFID System, and the need to for those flaws to be addressed. To eliminate these flaws, a new system was proposed, using the Hybrid Anti-collision Algorithms. To achieve the more efficient and reliable RFID system, other related works were reviewed. This gave more insight as to the requirements of the system and the expected outcome. For the development of the system, incremental development model was used. An algorithm showing the procedures of the operation of the system was also developed. This algorithm shows how the components of the system interact with each other to produce the required outcome

9. Conclusions

The emergence of wireless RFID technologies, identifying high density RFID tags is a crucial task in developing large scale RFID systems. Due to the nature of large scale RFID systems, many collisions may occur during the process of tag identification. Two new algorithms combining the characteristics of the two categories, ALOHA based and TREE based algorithm, have been presented in this research work. FQT algorithm of the two, in particular, has shown a big performance improvement on the tag - tag identification. This will also provide the ability for the gate to operate autonomously without collision and will set an example for firms to follow and keep up with the digital age.

10. Recommendations

The researchers recommends that more effective pre-detection scheme can be applied, that is, an extra layer of security can be added to the system by means of a Personal Identification Number (PIN), where after a tag is read, the tag holder types in his/her PIN to verify that it is actually him/her.

11. References

Ammiu J. Singh, E. Olsen, K. Vorst, and K. Tripp, (2017). "Rfid tag readability issues with palletized loads of consumer goods," Packaging Technology and Science, vol. 22, No. 8, pp. 431–441.

Ashane, K., Shah, H., and Kingston, P. (2017). RFID Applications: An Introductory and Exploratory Study. *International Journal of ComputerScience*.

Engels, D.W. and Sarma, S.E. (2017). 'The reader collusion problem in proceedings of IEEE International Conference 0n SMC 02.

Lee.S.R., Joosd .S.D. and L .C.W. (2020). 'An enhanced dynamic Framed slotted ALOHA algorithm for RFID tag identification proc. of MobiQuitous, pp.166-172.

Marrocco, G. (2019). *UHF Tags for Sensing Applications, Development and Implementation of RFID.* (C. Turcu, Ed.) InTech.

Manti W. Choi, N. Seong, J. M. Kim, C. Pyo, J. Chae, 2016. "A planar inverted-F antenna (PIFA) to be attached to metal containers for an active RFID tag". *IEEE, Antennas and propagation society international symposium* 1B, PP. 508-511.

Modin, N. (2016). *Contactless Electronic Door Key Using Radio Frequency Identification (RFID).*

Polytechnic of Hertfordshire, Department of Electronic, Communication and Electrical Engineering.

Myung, J and Lee,W. (2017). 'Adaptive binary splitting; A RFID Tag collusion Arbitration protocol for tag identification,; ACM.

Myung, J, Lee .W and Srivastava .J. (2019). 'Adaptive binary splitting for efficient RFID Tag, Anti- collusion IEEE, communication letters, vol 10, no. 3 pp. 144-146. protocol for tag identification,; ACM

Vogt, H. and Park.J. (2019). 'Efficient Object identification with passive RFID tags, proc of international conference on pervasive computing, Zurich, switzarland, pp 98-113.