UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Aloha-Based Radio-Frequency Identification (RFID) System With Early Frame Adjustment

Academic Session: Jan 2017

I, LEE KHAI YI declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ _________________________

(Author's signature) (Supervisor's signature)

Address:
29, Jalan Besar
11000 Balik Pulau,
Pulau Pinang.

____

Dr Robithoh Annur

Date: ____________________ **Date**:____________________

ALOHA-BASED RADIO-FREQUENCY IDENTIFICATION (RFID) SYSTEM WITH EARLY FRAME ADJUSTMENT

BY

LEE KHAI YI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfillment of the requirement

for the degree of

BACHELOR OF INFORMATION AND COMMUNICATION TECHNOLOGY (HONS)
COMMUNICATION AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2017

DECLARATION OF ORIGINALITY

I declare that this report entitled "**ALOHA-BASED RADIO-FREQUENCY IDENTIFICATION (RFID) SYSTEM WITH EARLY FRAME ADJUSTMENT**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ____________________________

Name : ____________________________

Date : ____________________________

ACKNOWLEDGEMENTS

Firstly, I would like to take this opportunity to present my gratitude to my supervisor, Dr. Robithoh Annur who has given me the chance to involve in this project and enables me to have the opportunity to study and explore the anti-collision algorithms which helps to enhance RFID tag identification process. A faithful thanks for all the assistance and guidance given throughout this project.

Next, I would like to thank my academic advisor, Miss Wong See Wan for giving me useful advices and suggestions whenever I face any difficulties in my study. Besides, I would also like to express my sincere appreciation to my family and loved one who has always been pillars of strength for me, giving me unconditional support and encouragement when any challenges or obstacles arise. A million thanks to them.

Last but not least, I would like to express my appreciation to my peers who willing to spend their time providing me ideas and solutions whenever I face any challenges or technical issues during project development phase.

ABSTRACT

Radio-Frequency Identification (RFID) is a technology which utilised electromagnetic fields for fast object tracking and identification. The beauty of this technology such as fast data reading without line of sight (LOS), large memory, long service life and strong penetrability has made RFID becomes a technology employed in different applications and commercial sectors to automate mundane tasks.

However, there are several drawbacks in RFID system has been discovered. One of the disadvantages of using passive RFID system is it will lead to tag collision when a reader is receiving signal sent from two or more tags at the same time. Consequently, these signals could not differentiate by the reader and the hence the tag information also could not receive correctly. Therefore, anti-collision algorithms that help to prevent tags collision are needed.

There are two main types of RFID anti-collision algorithm which are Binary Tree and ALOHA-based. In this project, we are mainly focus on ALOHA-based anti-collision algorithms and we are going to study Frame Slotted ALOHA (FSA) and Dynamic frame slotted ALOHA (DFSA) respectively. In most cases, DFSA is always adopted to resolve tag collision in RFID system as it could provide dynamic frame length that fits the collision situation during identification process. FSA is not preferable due to its static initial frame size that could lead to very high number of collision in the worst case.

However, the significant drawback of using DFSA is it has to predict the number of tags correctly in order to offer an optimal frame size. This is the most challenging task in DFSA. Thus, this project is going to propose a new timing concept which would enhance the tag identification process and mitigate RFID tag collision problem by utilising Manchester Coding, a bit-tracking technology in DFSA that allows RFID reader to recognise the location of collision bits within a time slot. Besides, Gen2 standard was applied in this project for the purpose of obtaining the slot duration during tag identification process and the tag identification rate of FSA, DFSA and proposed approach under different given scenarios.

TABLE OF CONTENTS

LISTS OF TABLES

LISTS OF FIGURES

LIST OF ABBREVIATIONS

DFSA	Dynamic frame slotted ALOHA
DS-MAP	Dynamic Sub-frame based Maximum a posteriori probability
EDFSA	Enhanced dynamic framed slotted ALOHA
EPC	Electronic Product Code
FDFSA	Fitted Dynamic Framed Slotted ALOHA
FSA	Framed slotted ALOHA
Gen2	Generation 2
HF	High Frequency
IC	Integrated Circuit
ILCM	Improved Linearized Combinatorial Model
LOS	Line Of Sight
LF	Low Frequency
RF	Radio Frequency
RFID	Radio-Frequency Identification
RN16	16-bit Temporary ID
SUBF-DFSA	Sub-frame DFSA
UHF	Ultra-High Frequency

CHAPTER 1: INTRODUCTION

1.1 Problem Statement and Motivation

Due to its contactless nature and faster read rate, RFID system has become popular and successfully attracted worldwide attention in supply market. However, RFID system is still consists of some limitations to be solved especially tag collision that occurred when more than one tag is transmitting data simultaneously to a reader. Consequently, the reader cannot rightly receive the tag information and lower the tag identification accuracy.

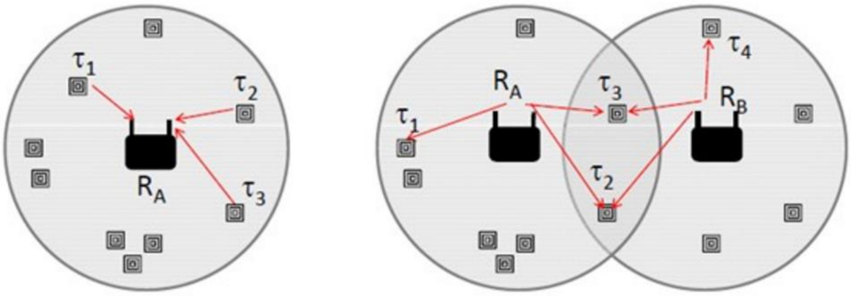
Among of different existing ALOHA-based anti-collision algorithms, DFSA is always adopted to resolve RFID tag collision problem. This is due to its advantage which is able to provide frame size that corresponding to the number of tags. But, DFSA is heavily relying on the result of tag estimation to perform frame size adjustment. Hence, accurate tag estimation is crucial in DFSA. This is because when a smaller frame size is offered, it would result in increasing the number of collision slots. Meanwhile, the number of empty slot would become higher when a bigger frame size is offered. Therefore, selecting an optimal frame size in DFSA is always not an easy task.

Besides, Generation 2(Gen2) protocol which involved timing during tag-reader communication is also adopted in this project. This is because we want to study the slot duration in Gen2 during its identification process and further shorten its timing during identification by introducing a new timing concept. In this project, we were going to utilise Manchester Coding which is a bit-tracking technology to mitigate the tag collision problem. Manchester Coding is very useful in collision detection during tag identification process as it could inform the reader the collision occurrence whenever it detects the position of collision bits from the information carried by RFID tags. Thus, it could help to improve system efficiency of RFID system as the reader could notice the collision and resolve it in a shorter period of time. As a result, it could help to reduce the time needed in tag identification process. In a nutshell, the motivation of this project is to mitigate the problem of tag, maximise the rate of tag identification and shorten the slot duration of RFID system based on DFSA.

1.2 Project Background

RFID is an identification system that uses electromagnetic fields to offer wireless communication between reader and tags. Due to its contactless nature, RFID has largely adopted in our daily routine and industries for fast object identification and tracking. Even with its fast identification and wireless nature, it also comes with several constraints. As discussed in previous section, RFID system always suffers from the problem of collision.

There are two different categories of RFID collision which are reader collision and tag collision. Reader collision occurred when one tag is read by multiple readers while tag collision happened when there is more than one tag sending signals to a reader at the same time. As a result, it prolongs the tag identification time as the reader could not recognise tags instantly. This is because the reader would need to solve the collision via anti-collision algorithm or retransmit the command. Figure 1.1 shows the two main types of collision in RFID system.



(a) Tag collision (b) Reader collision

Figure 1.1: Diagram of RFID system collision types (slideplayer.com, n.d.)

The main focus of this project is resolving RFID tag collision problem. There are two types of anti-collision algorithms had been proposed to encounter this problem which are ALOHA-based and Binary Tree. In this project, we are focusing on ALOHA-based anti-collision algorithms. The related existing works of this algorithm are Pure ALOHA, slotted ALOHA, FSA and DFSA algorithm.

In earlier time, FSA was usually adopted to resolve tag collision. This is because the previous Pure ALOHA and slotted ALOHA are not able to resolve the tag effectively and efficiently when there is huge number of tags involved in reader-tag communication. Therefore, FSA that enables the tags to send their data in random slots within a frame and the collided tags will be identified in future frames instead of competing with each other for the available time slots. This could reduce more collisions as compared to both of these algorithms. Figure 1.2 illustrates an example of FSA tag identification process.

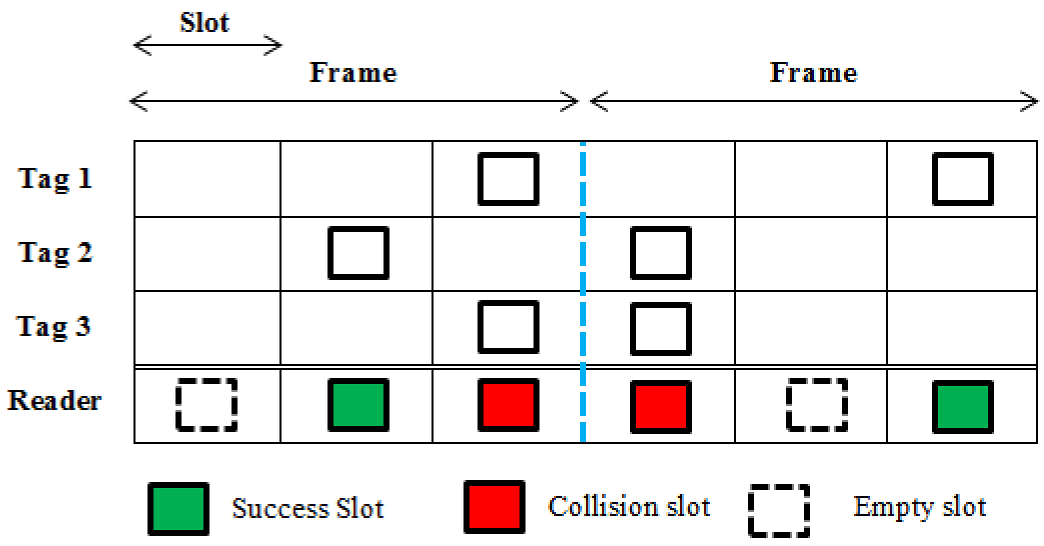


Figure 1.2: Example of FSA tag identification process

However, FSA is always using the same frame size throughout the whole identification process. This will lead to high collision rate and long identification time if an improper frame size is adopted. Hence, DFSA which able to provide dynamic frame size according to collision status of current frame is more preferable to use in resolving RFID tag collision. However, there is one issue arise while implementing DFSA which is selecting an optimal frame size. If an inappropriate frame size is selected, it will affect the rate of tag identification and system efficiency of RFID system. As a result, DFSA requires an accurate tag estimation algorithm in order to provide optimal frame size. Due to this reason, many tag estimation algorithms such as Lowbound, Schoute and Vogt's algorithm and other improved estimation algorithms had been introduced in order to tackle this problem.

1.3 Objectives

The overarching aim of this project is to mitigate tag collision problem when there is large volume of tags involved in the tag identification process. We are going to apply DFSA with Manchester Coding which is a bit-tracking technology in our proposed method in order to introduce a new timing concept which would provide the position of collision bits during RFID reader-tag communication process. By providing this kind of information, it would help to enhance the tag identification process of RFID system by accelerating the process of resolving tag collision. Unlike conventional ALOHA-based anti-collision algorithms, our proposed approach would adjust the frame size based on the Manchester Coding collision detection results in each time slot in a read cycle and it would resolve the tag collision slot by slot. This could help to improve the system efficiency of RFID system as it could reduce the number of collided tag involved in each future read cycle and thus the time slot used in the tag identification process would be lower.

The objectives of this project had been identified and listed as below:

i. Study and simulate FSA and DFSA anti-collision algorithms

ii. Study and apply ideal tag estimation, Schoute and ILCM tag estimation algorithm in DFSA to select optimal frame size

iii. Understand timing and slot duration of Gen2 standard in RFID system

iv. Propose a prototype scheme which:

    a. utilise Manchester Coding to resolve tag collisions during tag identification process

    b. shorten the timing in Gen2 standard

1.4 Proposed Approach

Figure 1.3 shows the system flow diagram of the proposed approach of this project.

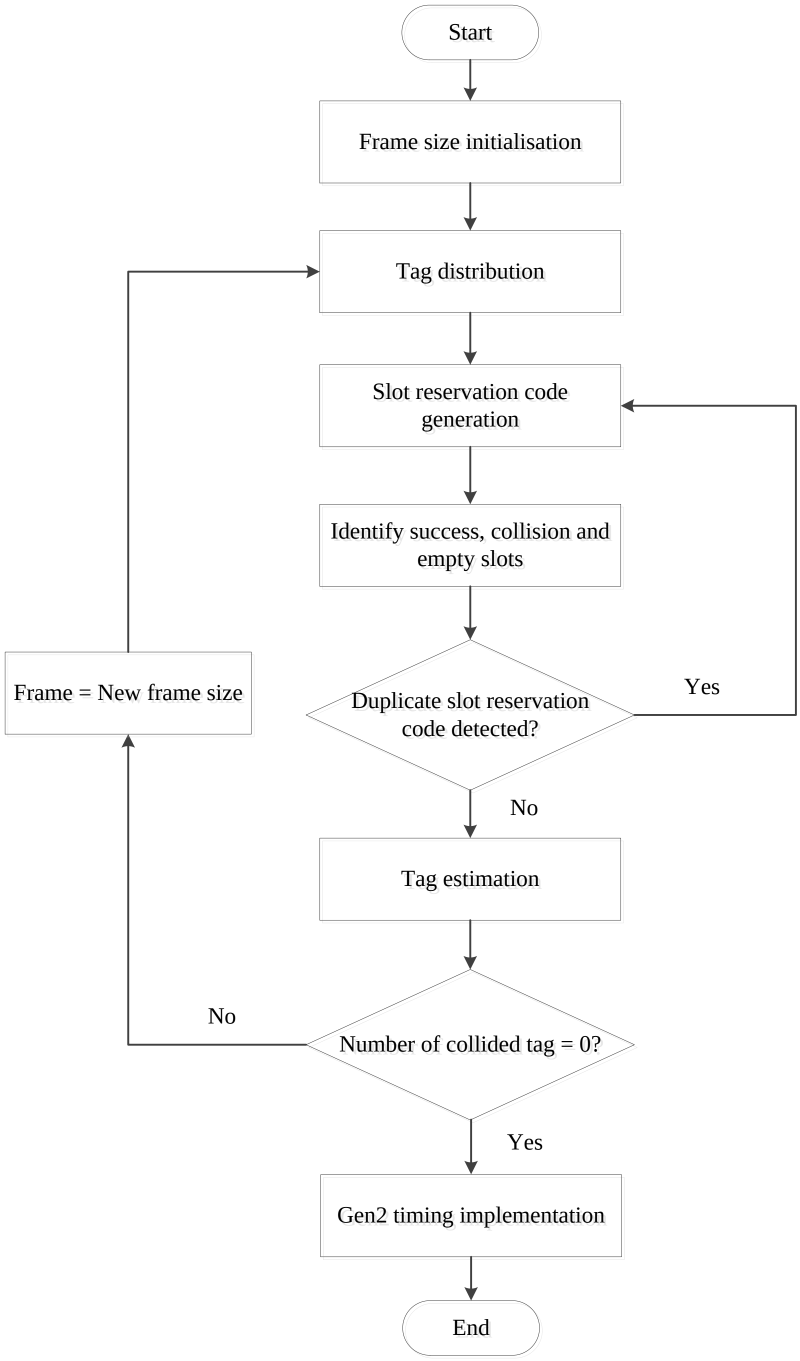


Figure 1.3: Proposed Approach Flow Diagram

For the first step of the proposed approach, it would be used to define the initial frame size that used for tag identification process. This step is the crucial part of the proposed approach as it would provide the number of time slot that available for the tags to reserve for identification process.

Step 2 is mainly used to perform tag distribution which allows each tag to reserve a time slot through the random number generated. In the proposed approach, each tag is required to have a slot reservation code in order to inform the reader regarding their slot reservation information.

Step 3 is designed to generate an 8-bit binary slot reservation code after all the tags have selected the time slot to be reserved. The actual time slot allocation and collision detection of reader in our proposed approach would heavily rely on this 8-bit binary slot reservation code.

Step 4 will be used to identify success, collision and empty time slots. In this step, we would apply Manchester Coding to identify success and collision slots and provide reader the actual time slot allocation decision for each tag. If any collision bits are found in a particular slot, the tags which reserved this slot would need to return to step 2 in order to reserve a new time slot. In contrast, if no collision bit is found, the reader would grant the tags its reserved time slot. Hence, this step is important for the reader to perform the actual time slot allocation and determine the system efficiency of the proposed approach. Besides, step 4 is also designed to prevent another type of collision which caused by the same reservation code used by the tags to reserve a time slot.

For step 5, it would be used to generate a new frame size in future read cycle for the tags which unable to obtain reserved time slot in previous step. All the collided tags would go through the slot reservation process again and thus a new frame size is created in order to provide time slots that are available for reservation.

The last step would be used to study and determine the tag identification rate and timing used by the proposed approach throughout the tag identification process by applying Gen2 standard timing parameters.

1.5 Report Organisation

This report consists of five chapters and is organised as following:

Chapter 1: Introduction

This chapter consists of problem statement and motivation, project background information, objectives and the basic idea of the proposed approach.

Chapter 2: Literature Review

This chapter consists of literature reviews of technology used in this project and discussion of previous related works.

Chapter 3: System Design

This chapter would discuss about the system design of the proposed approach and it also consists of flow diagram and detailed steps that describing the implementation of the proposed approach.

Chapter 4: Implementation and Analysis

This chapter would consist of the design specification of this project such as hardware and software used and experiment design of the proposed approach. Besides, it also includes the analysis and discussion for the simulation results and performance of the proposed approach with previous related works.

Chapter 5: Conclusion

This chapter would consist of the project review, discussion, the contributions of the proposed approach and future works to be done.

CHAPTER 2: LITERATURE REVIEW

2.1 Review of technologies

2.1.1 RFID system

RFID is an identification system that utilising electromagnetic fields to automatically identify tags attached to objects. According to McDowell (2009), there are four basic components in RFID systems which are tags, reader, antenna and computer system for data collection and processing. A RFID reader consists of a transceiver and an antenna. The transceiver will generate radio signal and transmit it through the antenna. This signal is used as a form of energy to activate the tags. Later, the RFID tag will receive the signal and the transponder will convert the radio frequency into usable power to send message back to reader. Then, the reader will receive the radio waves sent by transponder and interpret the radio frequencies as meaningful data. Lastly, the reader will send the information to the host computer for interpreting and processing. Figure 2.1 illustrates how a RIFD system works.

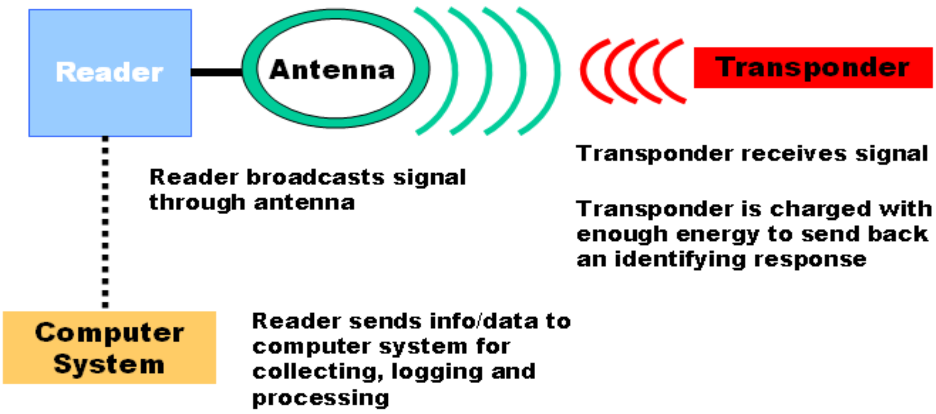


Figure 2.1: Diagram of how a RFID system works (lakshmi and profile, 2012)

RFID system consists of three different kinds of tags. The first one, passive tags which do no internal power source and usually relying on the radio signal that generated by reader to power up. Therefore, passive tags always have shorter read range when compared to active tags and required high signal strength for communication. Unlike passive tags, active tags have their own power source that enables them broadcast signal and have longer read ranges. They could be also act as “beacons” to initiate a communication with reader or other tags. In this project, we are only focusing on the passive tag collision in RFID system.

For semi passive tag, it is a kind of passive tag that contains the feature of active tag. Like active tag, it has internal power source that used to power itself up. Besides, the internal power source of semi passive tag will be activated and powers integrated circuit (IC) while a reader radio frequency (RF) signal is received. Figure 2.2 shows comparison between three types of RFID tag.

Tags and Features	Passive Tag	Active Tag	Semi Passive Tag
Internal Power Source	No	Yes	Yes
Signal by backscattering the carrier wave from the reader	Yes	No	Yes
Response	Weaker	Stronger	Stronger
Size	Small	Big	Medium
Cost	Less expensive	More expensive	Less
Potential Shell life	Longer	Shorter	Longer
Range	10 centimeters to few meters	Hundreds of meters	Hundreds of meters
Sensors	No	Yes	Yes

Figure 2.2: Comparison between Three Types of RFID tag (content, 2008)

2.1.2 Generation 2 (Gen2) technology

Gen2 is the first air-interface protocol that introduced by EPCglobal for ultra-high frequency (UHF) band in 2004. It is operating in UHF frequency range of 860MHz to 960MHz. Most of the recent RFID systems are adopting Gen2 technology as it allow the tags that do not have own power source to have the ability to reach the reader up to the distance of 10 meters. Similar to RFID passive tags, Gen2 tags are activated by utilising the reader generated radio waves that transmit through the antenna. As Gen2 tags do not powered by batteries, they required an energy-efficient technique to perform tag-reader communication. Hence, Medium Access Control (MAC) layer that allows energy-efficient multiple tags communication is applied in Gen2 technology.

Besides, Gen2 technology is always worked with DFSA based on Q-algorithm for the purpose of improving the system efficiency of RFID system. The frame size in Gen2 is usually initialised in the range of 0 to 2^Q-1 and Q is an integer which is ranging from 0 to 15 and it is broadcast by reader through Query command. The reader is able to change the Q parameter based on the collision condition in current frame by issuing an adjust command.

2.1.3 Manchester Coding

In this project, Manchester Coding would be utilised to resolve the problem of tag collision. Manchester Coding is a bit-tracking technology that allows RFID reader to recognise the location of collision bits within a collision slot. Moreover, it uses voltage level transition to represent the value of a bit. If there is a positive transition, the value of bit will be equal to "0" and "1" is for negative transition. Manchester Coding allows individual bit tracing when there are two or more tags are sending different bits and caused tag collision. Figure 2.3 illustrates an example of bit-tracking in Manchester Coding.

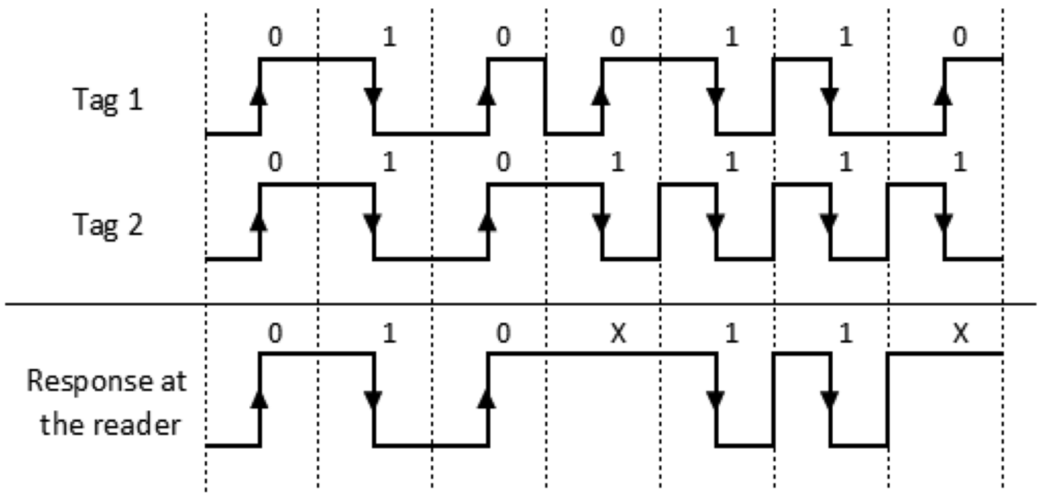


Figure 2.3: Example of bit-tracking technology in Manchester Coding (Landaluce, Perallos, and Angulo, 2014)

As showed in Figure 2.3, there are two tags transmitting their tag identifier which are 0100110 and 0101111 respectively. However, the reader does not manage to receive the correct tag information as the signals sent by these two tags are being interfered. As a result, the reader is receiving 010X11X and X is denoted as the collision bits. There is no voltage level transition when there is a collision occurred. In the given example, the collision bits are located the 4th and 7th time slots. By providing this information, Manchester Coding could help to accelerate the tag identification process and lower the rate of tag collision.

2.2 Review of -based anti-collision algorithms

Throughout the years, many RFID anti-collision algorithms have been proposed by researchers as it is crucial to enhance the performance of RFID system. We are going to discuss different ALOHA-based anti-collision algorithms in the section below.

2.2.1 Pure ALOHA

Pure ALOHA is the first algorithm that had been introduced to encounter the problem of tag collision. In this algorithm, a tag will simply transmit data to a reader whenever it has data to be sent. However, the retransmission time needed is very long when there is tag collision between two or more tags. The problem become worse if there is large volume of tags sending signal to a reader at the same point of time. Thus, the maximum throughput of Pure ALOHA is only 18.4%. (Raja and Perumal, no date)

2.2.2 Slotted ALOHA

In Slotted ALOHA, it consists of time slots that divided from the data transmission time. Each available tag is required to select one slot to their transmit data. (Cheng and Jin, 2007) Thus, the collision interval in Slotted ALOHA is halved as compared to Pure ALOHA. However, the efficiency of this approach is degraded if there is huge number of tags involved in data transmission. The maximum throughput of Slotted ALOHA is 36%. (Raja and Perumal, no date)

2.2.3 Frame Slotted ALOHA (FSA)

In order to encounter the limitation of slotted ALOHA, FSA that consists of frame that make up by a group of time slots has been introduced. In FSA, the tags are randomly distributed to the time slots. The time slot is considered to be a successful slot when only one tag occupied that slot. A collision slot would be created when there is two or more tags select one time slot at the same time. These tags would then transmit their data in the next read cycle. A time slot becomes an empty slot when there is no tag selects it. This process will keep on going till all the tags are successfully identified. However, system efficiency of FSA would influence badly by the frame size as the frame size in FSA is remaining unchanged throughout the identification process because it has no way to know the number of unread tags.

2.2.4 Dynamic Slotted ALOHA (DFSA)

The major limitation of FSA is its static frame size during the tag identification process. Hence, the frame size provided might be smaller or bigger than the number of tags. Thus, DFSA was introduced to encounter this drawback by enabling the adjustment of the frame size during identification process. Hence, it has lower number of used time slots as compared to FSA. The time duration used by the reader to identify the tags is also shorter. However, system efficiency of DFSA is always affected by the frame size offered. This makes tag estimation process become very important in DFSA as the fame size offered is depend on the results of tag estimation process. However, there is no standardised tag estimation algorithm available in current stage.

2.2.5 Summary of ALOHA-based anti-collision algorithms

The table below shows the summary of different ALOHA-based anti-collision algorithms.

ALOHA-based Anti-collision algorithm	**Strength**	**Weakness**
Pure ALOHA	Easy to implement	Collision become higher as the number of tags become bigger
Slotted ALOHA	• Reduce collision interval to half • Enhance throughput	Efficiency is degraded when the number of tags become bigger
FSA	Increase throughput and reduce the rate of collision	Number of tags cannot be recognised
DFSA	Number of tags could be known through estimation	Tag estimation algorithm is not standardised

Table 2.1: Summary of ALOHA-based anti-collision algorithms

2.3 Review of existing tag estimation algorithms

After reviewing the previous related works, the anti-collision algorithm that we are going to adopt in this project is DFSA. However, selecting an optimal frames size in DFSA is a very challenging task. In this section, we are going to discuss different existing tag estimation methods that assist DFSA in selecting optimal frame size.

2.3.1 Lowbound algorithm

In Lowbound algorithm, it will predict the number of unknown tags bay assuming that there are two or more collision tags. Therefore, it predicts the number of tags using the formula $S + 2C$ where C represents the number of collision slot and S represents successful slots within one frame. However, the tag estimation error increases when there are more than two collision tags.

2.3.2 Schoute algorithm

Schoute algorithm was using Poisson distribution to obtain the expected number of collision slots. P_{succ} and P_{coll} is the probability of success and collision occurred in a time slot respectively. The formula used by Schoute to predict the number of unknown tags is $S + 2.39C$. Therefore, tag estimation of Schoute is more efficient than Lowbound. However, Schoute algorithm is having the same drawback as Lowbound algorithm as they are doing estimation without considering the actual collision condition of current frame. Thus, it would have large estimation error when there is large number of collision tags.

2.3.3 Improved Linearized Combinatorial Model (ILCM)

ILCM is a tag estimation algorithm with low computational cost and was introduced by Solic et al.(2013). ILCM is a scheme that performs frame break when the next frame has higher expected number of successful slots than current frame. The tag estimation of ILCM is done through $p(E,S,C|n) = \frac{L!}{E!S!C!}\frac{N_S(n,S)N_C(n,S,C)}{L^n}$ where frame size, L is equal to E+S+C. Figure 2.4 shows the parameters that involved in ILCM tag estimation equation and their definition.

Parameter	Definition
$N_s(n,s)$	Number of ways to distribute n tags to S slot
$N_c(n,S,C)$	Number of ways in how to distribute the remaining n-S tags to C slots
L^n	Total number of ways to distribute tags in L slots
$\frac{L!}{E!S!C!}$	Number of ways to permute E,S and C and positions in the frame

Figure 2.4: ILCM tag estimation equation parameters and definition (Solic et al., 2013)

However, this relation is computationally heavy as it needs both hardware and software to perform the calculation. Thus, it was later simplified into $\widehat{N} = kS + l$ to reduce the computational cost. In this project, we are going to adopt ILCM as one of tag estimation algorithm used during tag estimation process.

2.3.5 Summary of existing tag estimation algorithms

Table 2.2 shows the summary of different existing tag estimation algorithms.

Tag estimation algorithm	Strength	Weakness
Lowbound	Easy to implement	Tag estimation error increases when collision tag is more than two
Schoute	More accurate tag estimation	Tag estimation error increases when collision tag is more than two
ILCM	Low computational cost	Complex calculation

Table 2.2: Summary of existing tag estimation algorithms

2.4 Review of existing improved ALOHA-based anti-collision algorithms

As existing ALOHA-based anti-collision algorithms could not resolve tag collision problem effectively, numerous related improved works has been proposed throughout the years. Out of the many improved related works, three relevant ones were selected to be reviewed in detail as follows.

2.4.1 Fitted Dynamic Framed Slotted ALOHA Anti-Collision Algorithm in RFID Systems

In Shakiba, Zavvari, and Sundararajan paper (2011), their proposed method, Fitted Dynamic Framed Slotted ALOHA (FDFSA) was to shorten the tag identification time by using minimum slots number. The proposed algorithm consists of four parts.

The first part which is the main part of the proposed algorithm is to define an initial frame size. This is to initiate a read cycle. Next, all tags would be assigned to different time slots based on the random number generated by distribution function. After that, the tags would send their IDs to reader. In next step, the slots would be read one by one after calling the read function. A tag is successfully identified when it is a successful slot and the tag will also assign a number of -1. This read function would count the number of successful slots (C_l) and collision slots (C_K).

Lastly, curve fitting estimation function is called to predict the number of tags according to C_l and C_K. If there is a collision occurred in current read cycle, these tags would be identified in future read cycle with a new frame size which is created based on tag estimation results. This process would be stopped when all the tags are successfully identified. Shakiba, Zavvari, and Sundararajan had compared and evaluated the performance of FDFSA with curve fitting estimation with other existing algorithms such as FSA, DFSA and enhanced dynamic framed slotted ALOHA (EDFSA) using the total number of slots used during tag identification with the initial frame size of 64. Figure 2.5 shows the simulation results of FSA, DFSA, EDFSA and FDFSA.

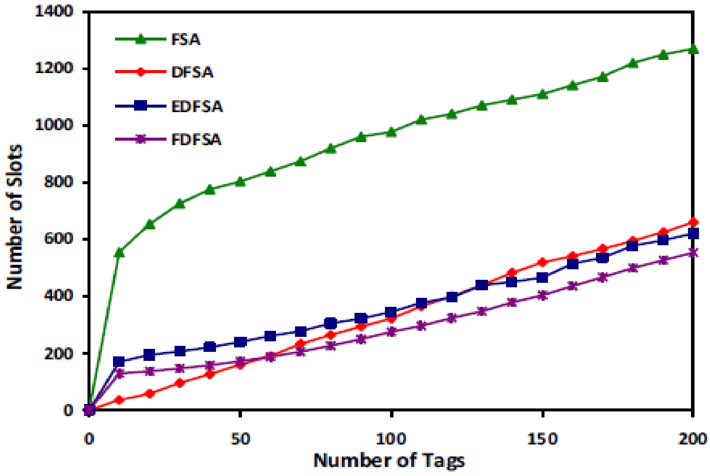


Figure 2.5: Simulation results of FSA, DFSA, EDFSA and FDFSA (Shakiba, Zavvari, and Sundararajan, 2011)

From their simulation results, they had concluded that FDFSA used the least number of required slots in tag identification process as compared to other algorithms. But, their proposed method did not consider the number of idle slots and thus efficiency of the proposed method would be reduced when there is high number of idle slots.

2.4.2 An Efficient and Easy-to-implement Tag Identification Algorithm for UHF RFID Systems

In this paper, a sub-frame based DFSA algorithm, Dynamic Sub-frame based Maximum a posteriori probability (DS-MAP) was proposed with the purpose of improving the tag identification efficiency of RFID systems. In this proposed method, it will not perform tag estimation calculation in the reader itself but utilise tables to pre-store the estimation results. By looking up the tables, it could reduce MAP computation overhead which caused by nested loop. However, it might require more memory to store the tables when the number of trials when n tags compete for slots. Thus, sub-frame structure is used in order to save memory and limit the table size.

In DS-MAP, it would divide a frame into sub-frames and assume the estimated tag numbers are equal in each sub-frame under the condition that all the tags ware evenly distributed. By referring to the number of successful and collision slots, it will predict the number of tags in first sub-frame by looking up the pre-stored

DS-MAP tables using the formula, $N_{est} = \hat{n} * K$ where $K = F/F_{sub}$ and F represents the frame length. If the estimated number of tags fits the current frame length, the algorithm will return to original DFSA and the frame length would be considered as optimal. As a result, the chances of new frame size fits to the number of backlog will be higher.

If the estimated number of tags did not fit the current frame length, it would adjust the frame length and calculate the estimated number of backlog using the formula $N_{back} = N_{est} - N_S$ where N_S denotes the number of successful slot in sub-frames. After that, the new frame length will be determined by the reader based on N_{back} and then it will issue a *QueryAdj* command to update the frame length. Chen, Su, and Yi (2017) had compared their proposed algorithms with SUBF-DFSA, MAP, FEIA, ILCM and Q-algorithm. Figure 2.6 shows the comparison of these algorithms.

Method	Avg. system throughput	Fluctuate ratio of system throughput	Computation complexity	Memory requirement (Bytes)	Overall ranking
Q-algorithm	0.295 (6)	18.54% (6)	$0.9 \times 10^3(1)$	≤ 100	5
MAP	0.327 (5)	12.35% (3)	$6.3 \times 10^5(6)$	≤ 100	6
ILCM	0.332 (4)	12.95% (4)	$1.44 \times 10^4(4)$	≤ 100	3
FEIA	0.336 (3)	16.12% (5)	$4.39 \times 10^4(5)$	≤ 100	4
SUBF-DFSA	0.349 (2)	3.47% (2)	$2.07 \times 10^3(3)$	≤ 100	2
DS-MAP	0.357 (1)	2.11% (1)	$1.28 \times 10^3(2)$	2728	1

Figure 2.6: Comparison of DS-MAP, SUBF-DFSA, MAP, FEIA, ILCM and Q-algorithm (Chen, Su, and Yi, 2017)

From figure 2.7, we could note that the proposed method is the second least computation complexity as it only needs to look up tables in order to count the number of success, idle and collision slots. DS-MAP also required lesser memory to store the tables. However, the significant drawback of this proposed method was it trade-off the computation complexity that would largely affect the energy consumption and RFID tag identification time with memory size.

2.4.3 A Dynamic Framed Slotted ALOHA Anti-collision Algorithm Based on Tag-Grouping for RFID Systems

In this paper, the authors had introduced a tag-grouping method that RFID reader identifies the tags group by group and it was divided into two main parts which were randomisation grouping process and collision slots identification process. Firstly, an initial frame length, $L = 2$ is set and the reader would send this parameter with a query command. Then, all the tags would randomly select a slot from 0 to L-1 after receiving the query command and then transmit reader a 16-bit random number in the corresponding slot. Next, the reader would identify success slot, idle slot and collision slot when reader received all the tag responses.

The next operation that would be performed by the reader is based on the response slots. The reading process would be terminated if all the response slots are idle slot. If the response slots are single slots then the reader will read the tags in this kind of slot and these tags would exclude from the reading process. If the responded slots are collision slots, firstly, the reader adjust the frame length become $L * 2$ and send this parameter with a query command. Then, all the unread tags will randomly select a slot in range of 0 to 3. The reader could read all the tags if there is no collision occurred and stop the reading process. If collision occurs, reader will transmit a new query command with a parameter which double the number of collision slots in order to read other tags. Figure 2.7 shows the efficiency of proposed algorithm and DFSA algorithms.

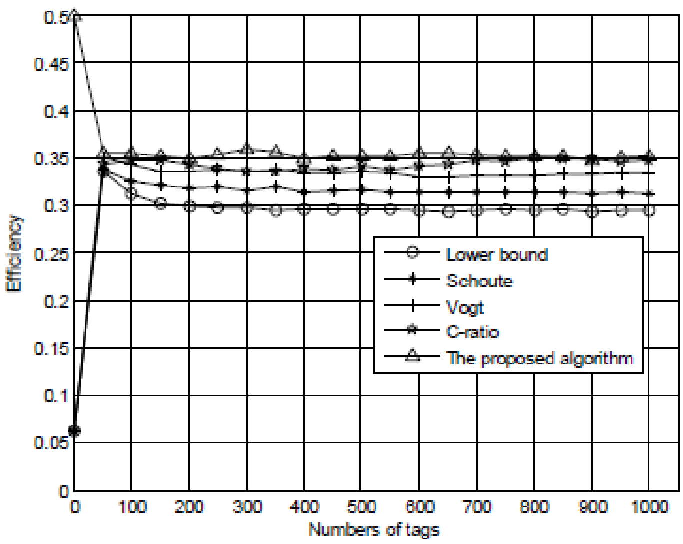


Figure 2.7: Efficiency of proposed algorithm and DFSA algorithms (Qing et al., 2012)

Qing et al. had designed 50 experiments by setting the initial frame length to 16 with 0 to 1000 tags in order to compare their proposed algorithm with other DFSA algorithms. From Figure 2.8, we could observe that the efficiency of their proposed algorithm was able to achieve up to 35% which is the highest among other DFSA algorithms. However, in order to reach highest efficiency of this proposed algorithm, a probability of 0.9 containing one tag in at least one slot in a frame needs to be achieved which is very biggest challenge faced by their proposed method.

2.4.4 Summary of existing improved ALOHA-based anti-collision algorithms

The following table shows the summary of different existing improved works that has been studied in the literature review.

Existing improved anti-collision algorithm	Strength	Weakness
Fitted Dynamic Framed Slotted ALOHA Anti-Collision Algorithm (FDFSA)	• Shorten tag identification time • Consume lesser time slots	Number of idle slots is not considered and thus the efficiency of the algorithm would be reduced when there is large number of idle slots
Dynamic Sub-frame based Maximum a posteriori probability (DS-MAP)	Able to adjust the frame length to fit the tag number in shorter time	• Complex computation • Higher energy consumption • Slower tag identification speed
DFSA based on Tag-grouping	• Able to disperse tags quickly and evenly • Enhance the efficiency of RFID systems	Probability of 0.9 containing one tag in at least one slot in a frame needs to be achieved in order to reach highest efficiency

Table 2.3: Summary of existing improved ALOHA-based anti-collision algorithms

CHAPTER 3: SYSTEM DESIGN

3.1 System flow

In this project, we were going to apply DFSA that could adjust the frame size dynamically based on the collision situation in our proposed approach together with a bit-tracking technology, Manchester Coding to mitigate the tag collision problem occurred in RFID system. Figure 3.1 shows the implementation flowchart of the proposed approach of this project.

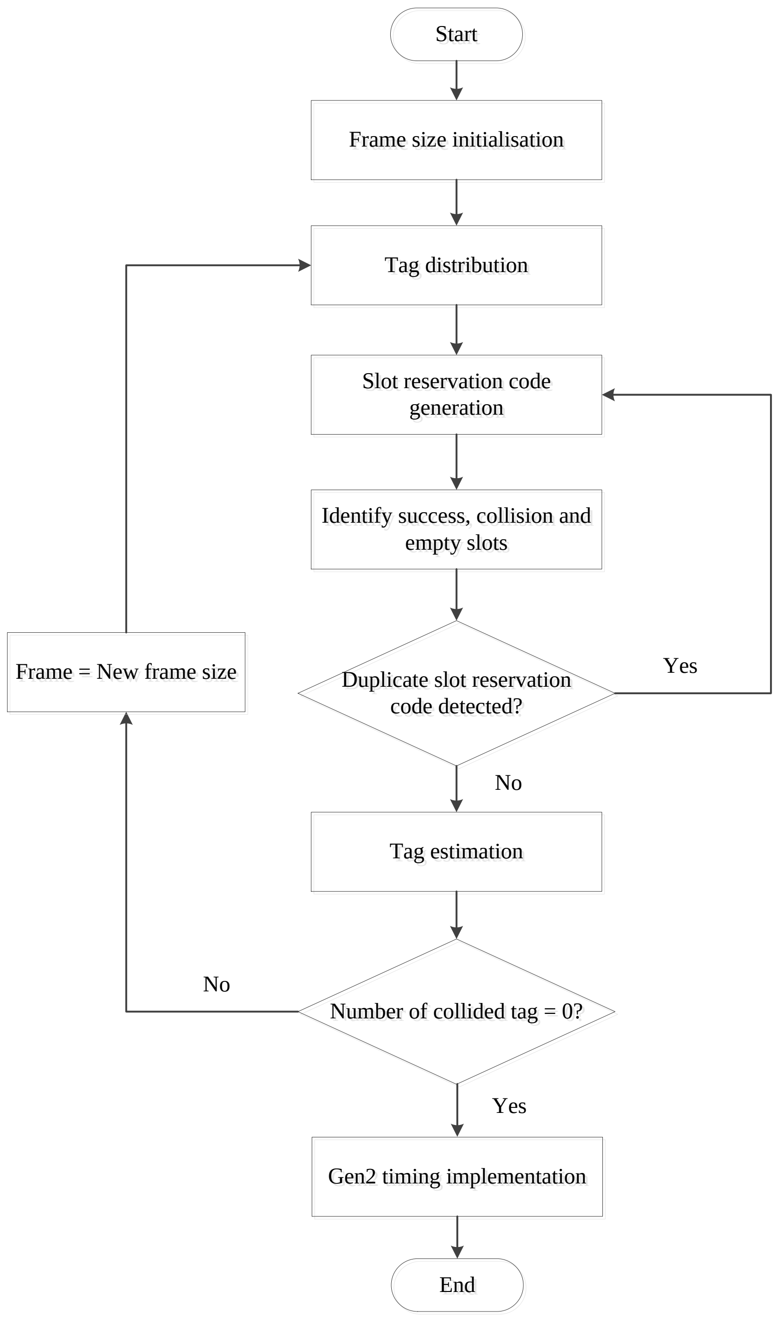


Figure 3.1: Flowchart of project implementation

Step 1: Frame size initialisation

The first step of RFID tag identification process is to initialise the frame size, *L* used to identify the tags. In the proposed approach implementation, we always assumed that the initial frame size for DFSA is always equal to the number of tags. As we want to provide frame size that is optimal for the tag identification process. For instance, if there are five tags to be identified by the reader then *L* would be set equal to 5.

Step 2: Tag distribution

In this step, each tag would generate random numbers within the range of frame size via randi() function. These generated random numbers are representing the time slot that would be reserved by the tags. For example, when $L = 5$ and then the possible random numbers that would be generated by the tags is ranging from 1 to 5. The tag distribution process is illustrated in Figure 3.2.

Tag	Time slot to be reserved
1	2
2	2
3	3
4	5
5	5

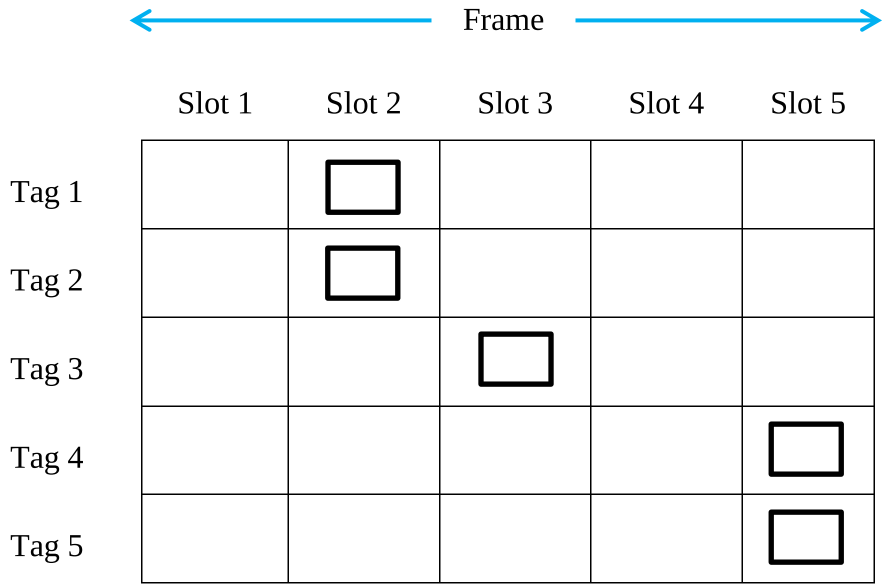


Figure 3.2: Tag distribution process

Step 3: Slot reservation code generation

After selecting the time slot to be reserved, each tag would generate a random number in the range of 1 to 255 using randi() function and convert to an 8-bit binary bit string using de2bi() function. Then, these 8-bit binary codes would be sent to the reader by the tags and this completes the slot reservation process. The reader would detect the collision bits from these slot reservation codes via Manchester Coding in the next step. This is to detect whether any collision is happened before the reader allocates the reserved time slot to the tags. The slot reservation code generation for five tags is showed in Figure 3.3.

Tag	Reserved Time Slot	Generated Random Number	8-bit slot reservation code							
1	2	5	0	0	0	0	0	1	0	1
2	2	10	0	0	0	0	1	0	1	0
3	3	6	0	0	0	0	0	1	1	0
4	5	1	0	0	0	0	0	0	0	1
5	5	1	0	0	0	0	0	0	0	1

Figure 3.3: Slot reservation code generation for 5 tags

Step 4: Identify success, collision and empty slots

After the tags reserving the time slots using its slot reservation code generated in previous step, the reader would categorise the time slots into success, collision and empty slot. In this step, we are going to apply Manchester Coding to identify success and collision slots.

A time slot would be treated as a success slot if there is one tag reserve that time slot and its slot reservation code could correctly receive by the reader. Then, the reader would allocate that particular time slots for these tags. The tags in success slot would be successfully identified by the reader and store in success list in the following step. Meanwhile, when there is no tag reserve a particular time slot then that time slot would become an empty slot. All the empty slots would not go through the collision bits detection process and the reader would not receive any slot reservation code from these slots.

On the other hand, when the reader could not receive the slot reservation codes correctly and that particular time slot would become a collision slot. This phenomenon occurs whenever there is a time slot reserved by more than one tag. As a consequence, these tags could not be granted the reserved time slot from the reader and stored in collision list and need to reserve new time slots in future read cycle with a new frame size.

Besides, there is another type of collision that occurred when there are more than one tag is using the same slot reservation code to reserve a particular time slot. In this case, the reader might mistreat that slot as a success slot as there is no collision bits found during Manchester Coding collision bits detection process. In order to resolve this kind collision, a new slot reservation code would be generated for a tag when there is more than one tag using the same slot reservation code to reserve a time slot. After regenerating new slot reservation codes, these tags would go through Manchester Coding collision bits detection process once again in order to mark it as a collision slot.

Step 4.1: Identify empty slots

In step 4, firstly, we would identify which time slot is not reserved by any tags to be identified by the reader. Then, we would mark this type of slot as empty slot and these slots are not going through Manchester Coding collision bits detection process. Thus, the reader would not receive any slot reservation code from these slots. Meanwhile, those time slots which were reserved by the tags in previous step would go through the collision bits detection process in order to find out which time slot is suffering collision. From Figure 3.2, we could notice that Slot 1 and 4 are empty slots and hence these two slots will skip all the following steps.

Step 4.2: Detect collision bits using Manchester Coding

After identifying empty slots, we would perform Manchester Coding collision bits detection for the time slots which were reserved by tags in previous step. In order to identify collision bits using Manchester coding, we would obtain the number of bit '1' of tags' slot reservation codes in each slot by using sum() function. Before that, we would identify the number of tags in each time slot. Thus, if the checking result is either less than the number of tags that reserved that particular time slot or not equal to 0, we could know that there is a collision bit detected. Figure 3.4 illustrates the reader collision bits detection process.

Reader Manchester Coding collision bits detection process:

Read Cycle 1

Frame 1

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5
Tag 1		**00000101**			
Tag 2		**00001010**			
Tag 3			**00000110**		
Tag 4					**00000001**
Tag 5					**00000001**
Check Result	-	**00001111**	**00000110**	-	**00000002**
Reader	-	**0000????**	**00000110**	-	**00000001**
	(Empty)	**(Collision)**	**(Success)**	**(Empty)**	**(Success)**

Figure 3.4: Collision bits detection process by reader using Manchester Coding

Step 4.3: Check duplicate slot reservation code

Instead of using Manchester Coding to detect collision, we would also acquire collision slots information through the number of tags in each time slot. This is done because we need this information to resolve the collision which caused by the same slot reservation code used by two or more tags to reserve a time slot. Besides, we would also check the tags' slot reservation code in each time slot. Thus, when there is more than one similar slot reservation code is used to reserve a time slot, then all the tags which reserved that particular time slot would be required to regenerate a random number using randperm() and convert to 8-bit slot reservation code using de2bi().

From Figure 3.4, we could notice that Slot 5 was mistreating as a success slot as there is no collision bits detected. This is because both tag 4 and 5 were reserving Slot 5 by using the same slot reservation code. However, by referring to Figure 3.3, we could know that Slot 5 is collision slot as there is more than one tag were reserving it. Hence, tag 4 and 5 would need to regenerate a new slot reservation code before proceeding to the next step. Besides, the reader would perform collision bits detection process once again in order to mark Slot 5 as collision slot. Figure 3.5 shows the collision bits detection process after slot reservation code regeneration.

Slot reservation code regeneration for tags in Slot 5:

Tag	Reserved Time Slot	New Generated Random Number	New 8-bit slot reservation code							
4	5	155	1	0	0	1	1	0	1	1
5	5	65	0	1	0	0	0	0	0	1

Reader Manchester Coding collision bits detection process after slot reservation code regeneration:

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5
Tag 1		**00000101**			
Tag 2		**00001010**			
Tag 3			**00000110**		
Tag 4					**10011011**
Tag 5					**01000001**
Check Result	-	**00001111**	**00000110**	-	**11011011**
Reader	-	**0000????**	**00000110**	-	**??0??0??**
	(Empty)	**(Collision)**	**(Success)**	**(Empty)**	**(Collision)**

Figure 3.5: Collision bits detection process after slot reservation code regeneration

Step 5: Tag estimation

After identifying success, collision and empty slots using Manchester Coding, the tags which not managed to obtain their reserved time slot would go to next read cycle with a new frame. In DFSA, the frame size would be adjusted before going to the next read cycle and hence tag estimation that provide the estimated number of unknown tags would be taken place. In our proposed approach, we are going to apply ideal tag estimation which the frame size is always equal to the number of tags.

In this step, we would also generate a success and collision list to store the tags which are either successfully or failed to recognise by the reader. If the number of tags stored in collision list is not equal to zero, then we would redistribute new time slots for these tags in future read cycle with a new frame size. If there is any collision is detected during tag redistribution process, these tags will repeat the whole slot reservation process until it is successfully read by the reader. In our proposed approach, the tag redistribution process would start from the first collision slot that stored in collision list and perform tag estimation to create new frame. In our proposed approach, the collision occurred in first collision slot must be resolved before proceeding to the next collision slot. Figure 3.6 shows the tag estimation and redistribution process of the proposed approach.

Success List:

Tag	Reserved Time Slot	8-bit slot reservation code
3	3	0 0 0 0 0 1 1 0

Collision List:

Tag	Reserved Time Slot	8-bit slot reservation code
1	2	0 0 0 0 0 1 0 1
2	2	0 0 0 0 1 0 1 0
4	5	1 0 0 1 1 0 1 1
5	5	0 1 0 0 0 0 0 1

1st collision slot

Tag redistribution for 1st collision slot:

Number of tags in Slot 2 = 2, New frame size = 2

Tag	New Reserved Time Slot	8-bit slot reservation code
1	1	0 0 0 0 0 1 0 1
2	2	0 0 0 0 1 0 1 0

Read Cycle 1 | Read Cycle 2

Frame 1 | Frame 2

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 1	Slot 2
Tag 1		**00000101**				**00000101**	
Tag 2		**00001010**					**00001010**
Tag 3			**00000110**				
Tag 4					**10011011**		
Tag 5					**01000001**		
Check Result	-	**00001111**	**00000110**	-	**11011011**	**00000101**	**00001010**
Reader	-	**0000????**	**00000110**	-	**??0??0??**	**00000101**	**00001010**
	(Empty)	**(Collision)**	**(Success)**	**(Empty)**	**(Collision)**	**(Success)**	**(Success)**

Updated Success List:

Tag	Reserved Time Slot	8-bit slot reservation code
3	3	0 0 0 0 0 1 1 0
1	1	0 0 0 0 0 1 0 1
2	2	0 0 0 0 1 0 1 0

Updated Collision List:

Tag	Reserved Time Slot	8-bit slot reservation code
4	5	1 0 0 1 1 0 1 1
5	5	0 1 0 0 0 0 0 1

Figure 3.6: Tag redistribution process

From Figure 3.6, we could observe that the first collision slot is Slot 2 which was reserved by tag 1 and 2. Thus, the collision occurred in Slot 2 would be resolved first before proceeding to Slot 5. From the collision list, we could notice that there are two tags reserved Slot 2 and thus by using ideal tag estimation we would set the value of new frame size as 2. Both tag 1 and 2 would reserve a new time slot by generating a random number in range of this new frame size which from 1 to 2.

After that, the reader would detect collision bits via Manchester Coding once again. From Figure 3.6, we could notice that the new time slot which reserved by tag 1 and 2 are Slot 1 and 2 respectively and thus there is no collision bits detected and the reader could receive the slot reservation codes correctly. Hence, the reader would assign these two tags to their reserved time slots and insert into success list and then remove from the collision list. However, the whole tag redistribution process would be repeated again if there is any collision detected. This process would repeat for all the following collision slots until the collision list becomes empty.

Step 6: Gen2 timing implementation

The last step is to implement Gen2 timing in the proposed approach in order to study the slots duration used during tag identification process. In Gen2 protocol, the reader will send the information about a frame using 22-bit Query and send 4-bit Query Rep to tags at every beginning of the slot. Later, random number would be generated by the tag and the number of QueryRep is counted. When generated random number is same to the counted number, the tag would respond to the reader query. As it allows the tag to send 16-bit temporary ID (RN16) during its slot time and this can help to decrease the collision and empty slots time indirectly.

In Gen2 standard, a successful slot will be created when the reader could successfully decode tags RN16 and acknowledge it by replying an ACK command that comes with tag identifier, Electronic Product Code (EPC). If the tags are not successfully identified, the reader will reply a negative acknowledge (NAK) command and these tags would be read in following read cycles. The tag identification process of Gen2 would not be terminated whenever there are collision slots. Figure 3.7 illustrates the timing details for successful, empty and collision slots in Gen2.

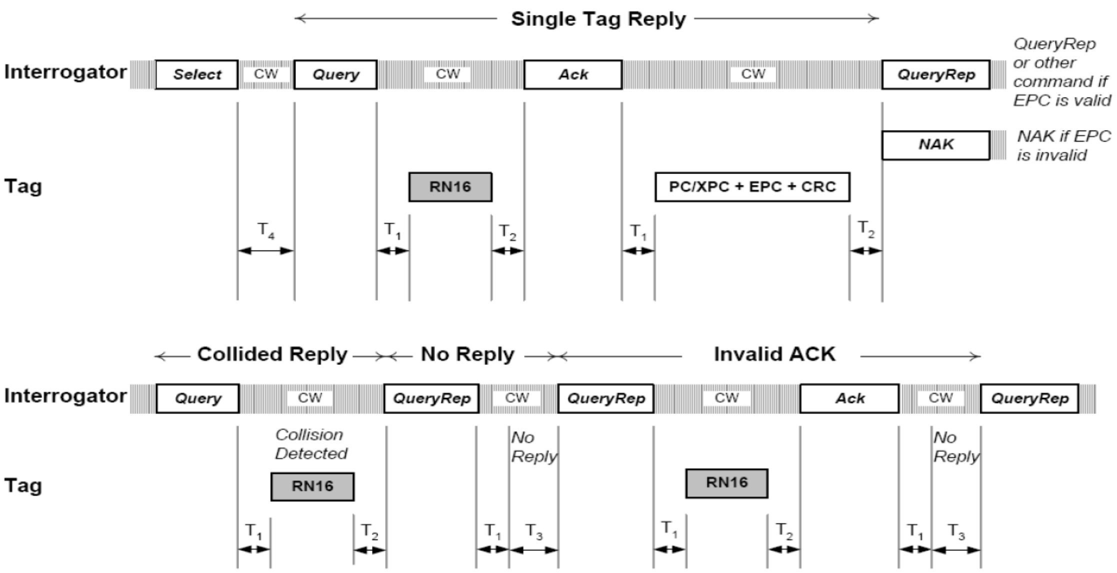


Figure 3.7: Timing details for successful, empty and collision slots in Gen2 (Solic et al., 2013)

In Gen2 standard there is a list of parameters involved in the tag identification process. Figure 3.8 shows the Gen2 standard parameters and its description.

Parameter	Description
Tari	Reference time interval for a data-0 in Interrogator-to-Tag signalling
PRT	Reader tag preamble
BLF	Backscatter-link frequency that represents tag-reader response frequency
Tpri	Backscatter-link pulse-repetition interval
DR	Division Ratio that used to define tag-reader symbol rate
TRCal	Tag-reader calibration symbol
RTCal	Tag-reader calibration symbol
Rbl	Reader bit length that measure the duration of each reader bit
TFS	Time Frame Sync
T_{Qrep}	4-bit query that send to tags at every beginning of the slot
T_{Query}	Duration of Query command duration
T_{ACK}	Duration Acknowledgement command
T1	Time from Interrogator transmission to Tag response for an immediate Tag reply
T2	Time from Tag response to Interrogator transmission
T3	Time an Interrogator waits, after T1, before it issues another command
M	Number of Miller subcarrier cycles in tag response, which could be set to 1 (FM0-code), 2, 4 and 8
TRext	Value specified in the Query that initiated the inventory round, which could be set to 0 ($Trext_0 = 4$) or 1 ($Trext_0 = 16$)
T_{RN16}	Duration of RN16 command
T_{EPC}	Duration of EPC command
T_E	Duration of empty slot
T_C	Duration of collision slot
T_S	Duration of successful slot

Figure 3.8: Gen2 standard parameters and description (Nov-2013, version 2.0 EPC™ radio-frequency identity protocols generation-2 UHF RFID specification for RFID air interface protocol for communications at 860 MHz – 960 MHz version 2.0.0 ratified, 2013)

The Gen2 Timing implementation steps are described as following:

Step 6.1 Select Tari, DR and BLF

The first step to implement Gen2 timing, we have to select Tari, DR and BLF value to be used in the experiment from the given value range. The following table shows the given value ranges of Tari, DR and BLF provided in Gen2 standard.

Parameter	Value Range
Tari	$6.25\mu s\ to\ 25\mu s$
DR	$\frac{64}{3}\ or\ 8$
BLF	$40kHz \leq BLF \leq 640kHz$

Table 3.1: Tari, DR and BLF and their value ranges

Step 6.2 Calculate Rbl, PRT, Tpri, TRCal and RTCal

After selecting the value of Tari, DR and BLF values, we are going to calculate the value of Rbl, PRT, Tpri, TRCal and RTCal which is based on the value of parameters in previous step. The following table shows Rbl, PRT, Tpri, TRCal and RTCal with their equation and given value range for calculation.

Parameter	Equation/Value Range
Rbl	$(2Tari + 0.5Tari)/2 \leq Rbl \leq 3Tari/2$
Tpri	$1/BLF$
TRCal	$DR * Tpri$
RTCal	$1.5Tari \leq RTCal \leq 2Tari$
PRT	$12.5 * 10^{-6} + Tari + 2.5Tari + 1.1\,TRCal$

Table 3.2: Equations for Rbl, PRT, Tpri, TRCal and RTCal and Value Range

Step 6.3: Calculate T_{Query}, T_{ACK}, T_{Qrep}, T1, T2 and T3

Next, we will calculate the duration of T_{Query}, T_{ACK}, T_{Qrep}, T1, T2 and T3. The equations for calculation and their given value range is showed in Table 3.3.

Parameter	Equation/Value Range
T_{Query}	$PRT + 22Rbl$
T_{ACK}	$TFS + 18Rbl$
T_{Qrep}	$TFS + 4Rbl$
TFS	$12.5 * 10^{-6} + Tari + 2.5Tari \leq TFS \leq 12.5 * 10^{-6} + Tari + 3Tari$
T1	$\max(RTCal,10Tpri) * (0.9) - 2*10^{-6} \leq T1 \leq \max(RTCal,10Tpri) * (1.1) + 2*10^{-6}$
T2	$3Tpri \leq T2 \leq 20Tpri$
T3	Minimum of 0.1Tpri

Table 3.3: Equations for T_{Query}, T_{ACK}, T_{Qrep}, T1, T2 and T3 and Value Range

Step 6.4 Calculate T_{RN16} and T_{EPC}

In this step, we are going to calculate the value of T_{RN16} and T_{EPC}. Before that, the value of M and $TRext$ has to be selected. In this project, we are using $M = 1$ and $Trext_0 = 4$. Table 3.4 shows the equations to calculate the value of T_{RN16} and T_{EPC}.

Parameter	Equation
T_{RN16}	$((Trext_0 * M)/BLF) + ((6M)/BLF) + ((17M)/BLF)$
T_{EPC}	$((Trext_0 * M)/BLF) + ((6M)/BLF) + ((M * (16 + 96 + 17))/BLF)$

Table 3.4: Equations for T_{RN16} and T_{EPC}

Step 6.5 Calculate T_S, T_C and T_E

The last step of Gen2 timing implementation is to calculate the duration of empty, collision and successful slot. Table 3.5 below shows the equations to calculate the value of T_S, T_C and T_E.

Parameter	Equation
T_S	$T_{Qrep} + T1 + T_{RN16} + T2 + T_{ACK} + T1 + T_{EPC} + T2$
T_C	$T_{Qrep} + T1 + T_{RN16} + T3$
T_E	$T_{Qrep} + T1 + T3$

Table 3.5: Equations for T_S, T_C and T_E

CHAPTER 4: IMPLEMENTATION AND ANALYSIS

4.1 Design Specifications

In this project, experiments with different number of tags which changing from 5 to 1000 had been carried out to analyse and evaluate the performance of FSA, DFSA and our proposed approach in Gen2 standard. This project is using MATLAB to perform simulation among the different algorithms and produce simulation results in graphical form for analysis purpose. The following are the minimum system requirements for this project sorted into hardware and software categories.

4.1.1. Hardware

A) Personal Computer (PC)

- Pre-installed with MATLAB.

4.1.2. Software

The software that would be equipped in this project is MATrix LABoratory (MATLAB). The following figure shows the image logo of MATLAB.

Figure 4.1: Image logo of MATLAB

B) MATrix LABoratory (MATLAB)

- MATLAB is the easiest and most productive software that provides high-level language for numerical computation, data analysis, and application development. (The MathWorks, Inc., n.d.)
- This project involves simulation of different ALOHA-based anti-collision algorithm and the simulation involved many calculations and parameters during the tag estimation and identification process. Therefore, MATLAB which includes an extensive set of built-in math functions and 2D and 3D plotting functions is required in the project implementation in order to provide fast mathematical calculations and a visualized data and communication results.

4.2 Implementation of FSA and DFSA

In this project, we were also carried out the simulation for existing ALOHA anti-collision algorithms which are FSA and DFSA. This is because we were going to evaluate and compare the performance among FSA, DFSA and the proposed approach in term of system efficiency and the tag identification rate. For DFSA, three tag estimation methods which are ideal tag estimation, Schoute algorithm and ILCM were applied. This is because we want to study the efficiency of DFSA with these three algorithms. The following section would describe the steps involved in both FSA and DFSA simulations. Figure 4.2 shows the flowchart the implementation of FSA and DFSA simulations.

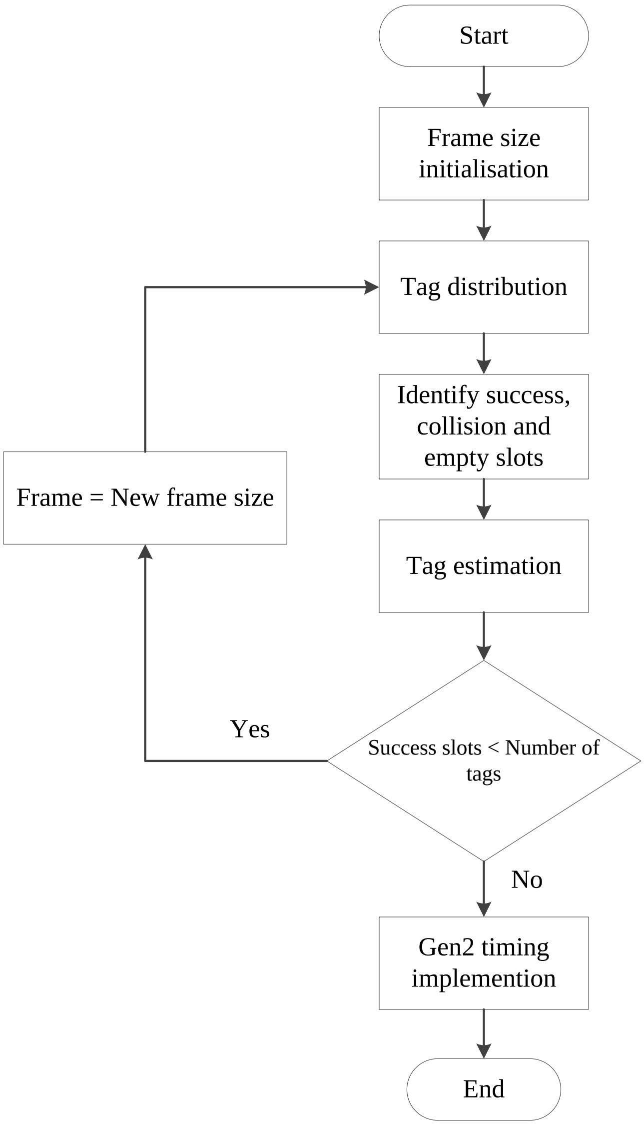


Figure 4.2: Flowchart of FSA and DFSA simulations

Step 1: Frame size initialisation

Similar as the proposed approach, we need to initialise the frame size, L to be used to identify RFID tags. In our simulation, the initial frame size for DFSA is always equivalent to the number of tags that are available whereas the initial frame sizes used in FSA are 100, 150 and 200 respectively. Hence, there would be a total of three simulations would be done in FSA.

Step 2: Tag distribution

After defining the frame size, the tags would randomly select the time slot by generating random numbers within the range of frame size via randi() function. The generated random numbers are representing the time slot which would be chosen by the tags. For instance, if there are 3 tags to be identified by reader and then the possible random numbers that would be generated by the tags is ranging from 1 to 3.

Step 3: Identify number of empty, success and collision slots

After distributing the tags to their respective time slots, the next process is to identify the tags either are collided or successfully identified by the reader. The slot would be considered as successful slot when there is one and only one tag is assigned to it. In contrast, if there is more than 1 tag are choosing the same time slots and that time slot would become a collision slot. When there is none of the tags are assigning to a particular time slot, the time slot would become an empty slot. For the tags which are suffering from tag collision would not be identified and they will go to the next reading cycle and then a new frame would be used.

Step 4: Tag estimation

Whenever there is a tag collision detected in previous step, the unread tags would go to next read cycle. Thus, tag estimation would be taken place in order to adjust the frame size. FSA would not go through this step as the frame size used will remain unchanged throughout the identification process. For DFSA, we are going to apply ideal tag estimation, Schoute algorithm and ILCM for tag estimation process. In ideal tag estimation, we assume the frame size is always equal to the number of tags.

Step 4.1 Schoute tag estimation algorithm implementation

As mentioned in section 2.4.2, Schoute algorithm would estimate the number of unknown tag, $\widehat{N}$ by using the formula $\widehat{N} = S + 2.39 * C$. Hence, the new frame size is determined by using the formula $2.39 * C$. For example, if there are two collision slots in current frame. Then, the new frame size would be $2.39 * 2$. The result 4.78 will later round to the nearest integer which is 5.

Step 4.2 ILCM tag estimation algorithm implementation

In ILCM, the frame size, L is set to $S + C + E$. The estimation for unknown tags of ILCM is done through the simplified relation, $\widehat{N} = kS + l$ where

$$k = \frac{C}{(4.334L - 16.28) + \left(\frac{L}{-2.282 - 0.273L}\right)C + 0.2407\ln(L + 42.56)}$$

$l = (1.2592 + 1.513L)tan(1.236L^{-0.9907}C)$. (Solic et al., 2013)

Besides, there are two scenarios that ILCM estimation would be bounded in. The first scenario is when value of k is less than 0 is given to smaller L and then the estimation should return $k=0$. Another scenario is when there involved an estimation error where $C=0$. In such scenario, $\widehat{N}$ would set to S. Figure 4.3 show the implementation of ILCM tag estimation.

```
L = E + S + C
k = C/((4.334L-16.28)+(L/(-2.282-0.273L))C+0.2407ln(L+42.56))
l=(1.2592+1.513L)tan(1.236L^(-0.9907)C)
if k < 0 then
   k = 0
end if
N̂ = kS + l
if C = 0 then
   N̂ = S
end if
```

Figure 4.3: Implementation of ILCM tag estimation (Solic et al., 2013)

Step 5: Gen2 timing implementation

The last step is to implement Gen2 timing in both FSA and DFSA in order to study the slots duration during tag identification process.

4.3 Results and discussion

The simulation results discussion would be organised into two parts which are comparison between FSA, DFSA and our proposed approach, DFSA with Manchester Coding with and without Gen2 standard respectively.

4.3.1 Comparison between FSA, DFSA and Proposed Approach without Gen2 standard

In this section, we are going to compare the simulation results of FSA, DFSA and proposed approach which is DFSA with Manchester Coding without Gen2 standard. This is to discuss and evaluate the performance of proposed approach and the existing ALOHA anti-collision algorithms which are FSA and DFSA by comparing the average time slot used during tag identification and system efficiency.

A. Average time slot used

The following figure presents the average time slot used in FSA (with frame size of 200), DFSA and proposed approach with 5 to 1000 tags.

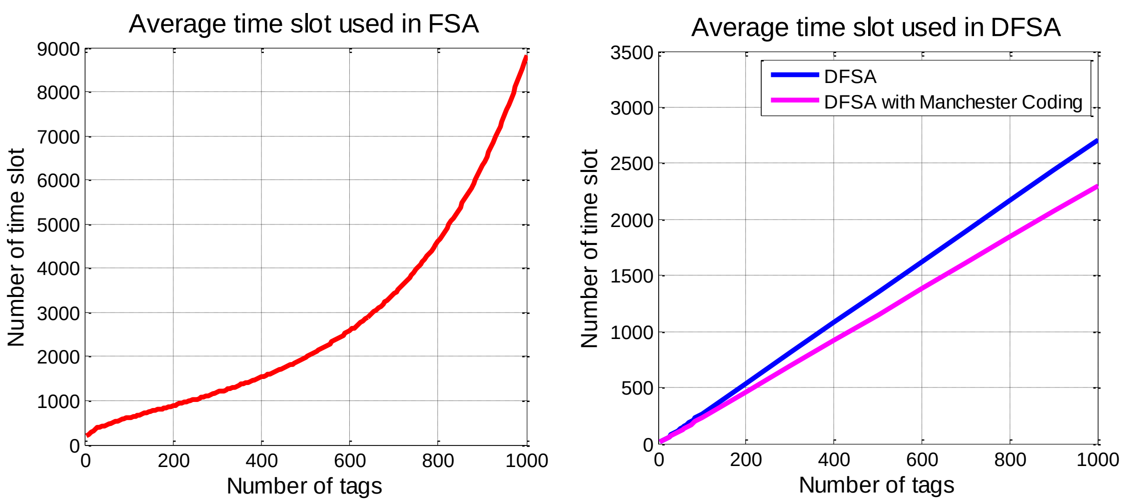


Figure 4.4: Average time slot used in FSA, DFSA and Proposed Approach

From Figure 4.4, it shows that the average of time slot used in FSA is increased exponentially as the number of tags becomes bigger. When there are 1000 tags involved in the identification process, the time slot used is approximately 8800. This happened is due to the initial frame size provided in the experiment which is 200 could not cater the needs of this huge number of tags. Furthermore, this frame size is remained static throughout the identification process. If smaller frame size is initialised, the rate of tag collision will become higher. As a result, the tags will go through many rounds of read cycle and the number of time slot used will become higher.

Meanwhile, we could learn that DFSA has reduced the number of average used time slot to only around 2700 which is more than half as compared to FSA. This is because DFSA could dynamically adjust the frame size as the collision situation reported in the current frame. However, DFSA would resolve the tag collision detected from different time slots in one read cycle and this process will require higher number of time slot if there is large number of collided tags found in current read cycle. For instance, if there are 100 collided tags detected in one read cycle, DFSA would need to prepare 100 time slots in the consequent read cycle. This would also create higher number of empty slots especially in the worst case which all the tags select the same time slot.

Unlike the conventional DFSA, our proposed approach will resolve the tag collision slot by slot. For example, when there are 3 collision slots and 10 collided tags detected in one read cycle, our proposed approach would sort these collided tags according to their reserved time slot. After that, it would resolve the collision accordingly based on the slot number. For instance, if there are 3 collision slots which are Slot 1, 2 and 3 found in current read cycle. Our proposed approach would first resolve the collision in first collision slot which is Slot 1. If there are 3 collided tags found in Slot 1, our proposed method would provide time slot available for reservation in the next read cycle that fit to this number of collided tags. By doing this, it could reduce the number of empty slot and slot used per read cycle. From Figure 4.4, we could notice that the number of time slot used in our proposed approach when there are 1000 tags is reduced to approximately 2300. Therefore, it could provide higher system efficiency as compared to traditional DFSA.

B. System Efficiency

System efficiency is another main concern of selecting anti-collision algorithm. In ideal case, maximum efficiency that could be theoretically achieved by FSA is 36.8%. However, the system efficiency of FSA would decrease gradually when an improper frame size is used in the identification process. Figure 4.5 shows the system efficiency of FSA with frame size of 100, 150 and 200.

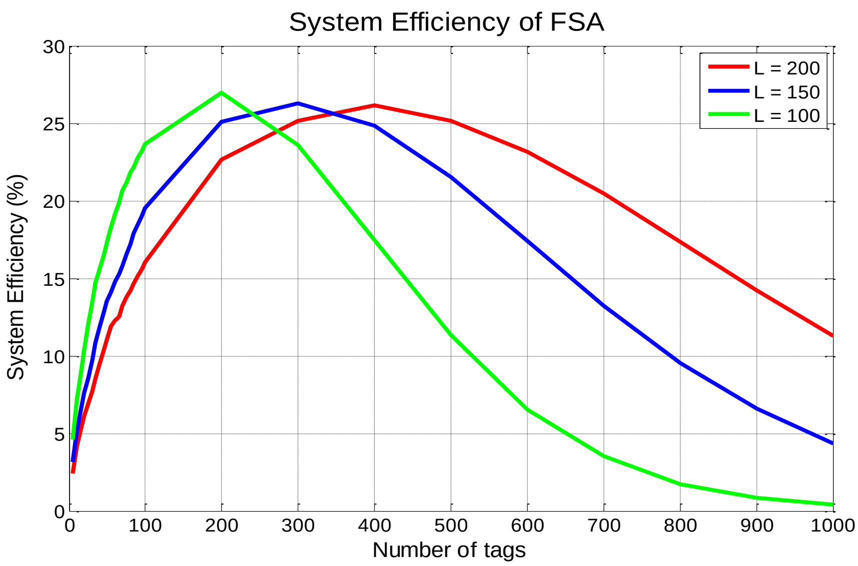


Figure 4.5: System efficiency of FSA with different frame sizes

From Figure 4.5, the maximum figure that FSA could achieve in the simulations is ~27%. Especially when frame size, $L = 100$, the system efficiency of RFID system is 0.3842% which almost 0% when there are 1000 tags. This is because of the frame size used was smaller than the number of tags. Hence, the tags need to compete with each other in order to obtain a time slot and this would cause high rate of tag collision. In FSA, an optimal frame size is very important as it could influence the efficiency of RFID system badly if the frame size is not properly defined. Therefore, it is proven that FSA is no longer feasible to resolve tag collision as there is no way to decide the initial frame size to be used and hence DFSA is introduced.

However, DFSA is also having the difficulty in selecting an optimal frame size. It needs to have prior notice about the number of tags before selecting a frame size. Hence, tag estimation is essential to resolve DFSA dilemma in choosing frame size. In this project, the selected tag estimation algorithms to be simulated are ideal tag estimation, Schoute algorithm and ILCM respectively. In our proposed method, we adopted Manchester Coding collision bits detection results together with ideal tag estimation to provide optimal frame size for tag identification process. Figure 4.6 shows the system efficiency of DFSA with ideal tag estimation, ILCM, Schoute algorithm and Manchester Coding.

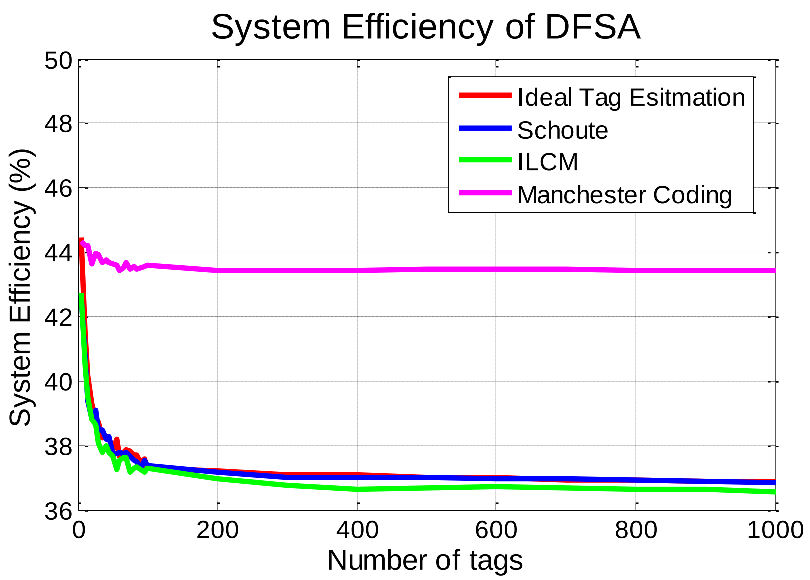


Figure 4.6: System efficiency of DFSA with different tag estimation algorithms and Manchester Coding

As showed in Figure 4.6, DFSA is giving more stable system efficiency as compared to FSA. The system efficiency of DFSA with ideal tag estimation, Schoute algorithm and ILCM would able to provide a more stable system efficiency which is ~36% as the number of tags becomes larger. Besides, we could observe that ILCM has the lowest system efficiency among other algorithms. This is because the performance of ILCM is affected by the initial frame size. Thus, ILCM could not adjust the frame size which fits the collision status in current frame when a small initial frame size is offered and there is large number of tags and hence it will reduce the efficiency of ILCM. Meanwhile, Schoute algorithm which uses static estimation has higher system efficiency than ILCM. But, it would also lead to large estimation error if there is high number of collision tags.

In our proposed approach, Manchester Coding and ideal tag estimation are used to select optimal frame size. From Figure 4.6, we could notice that the proposed approach could provide the highest system efficiency and it could remain ~43% even the number of tags becomes bigger. This is because the proposed approach would tune the frame size according to the number of collided tag obtained from each collision slot using Manchester Coding collision bits detection results and thus it could provide frame size that closely reflected to the collision situation. As a result, it would able to reduce the number of empty slot and thus the objective to enhance the system efficiency of RFID system is achieved.

4.3.2 Comparison between FSA, DFSA and Proposed Approach with Gen2 standard

In this section, we were going to FSA, DFSA and our proposed approach with Gen2 standard in order to study the slot duration during tag identification process. There were three scenarios with different BLF values which are 40 kHz, 340 kHz and 640 kHz was designed for the simulations. Different BLF are used is because RFID system may operate in different frequency bands. For example, RFID system that operates at low frequency (LF) such as access control, high frequency (HF) is including payment and tolling system and UHF such as antenna design. Table 4.1 shows the Gen2 parameters used in this project.

Parameter	**Scenario 1**	**Scenario 2**	**Scenario 3**
BLF	40 kHz	340 kHz	640 kHz
Tari	16µs	16µs	16µs
Rbl	22µs	22µs	22µs
PRT	276µs	81.88µs	69.75µs
RTCal	Tari + 0.75Tari = 28µs	Tari + 0.75Tari = 28µs	Tari + 0.75Tari = 28µs
TRCal	200µs	23.5294µs	12.5µs
TFS	60µs	60µs	60µs
T1	275µs	31.3529µs	30.8µs
T2	75µs	8.8235µs	4.6875µs
T3	2.5µs	0.2941µs	0.1563µs
M	1	1	1
TRext	4	4	4
T_{Query}	760µs	565.8824µs	553.75µs
T_{ACK}	456µs	456µs	456µs
T_{Qrep}	148µs	148µs	148µs
T_{RN16}	675µs	79.4118µs	42.1875µs
T_{EPC}	3475µs	408.8235µs	217.1875µs
T_S	5450µs	1174.6µs	934.35µs
T_C	1173µs	268.5882µs	225.675µs
T_E	425.5µs	180.6471µs	178.9563µs

Table 4.1: Gen2 parameters used in Scenario1, 2 and 3

The rate of tag identification with Gen2 is calculated through the formula as follow:

$$Timing\ of\ empty\ slot = T_E * E$$

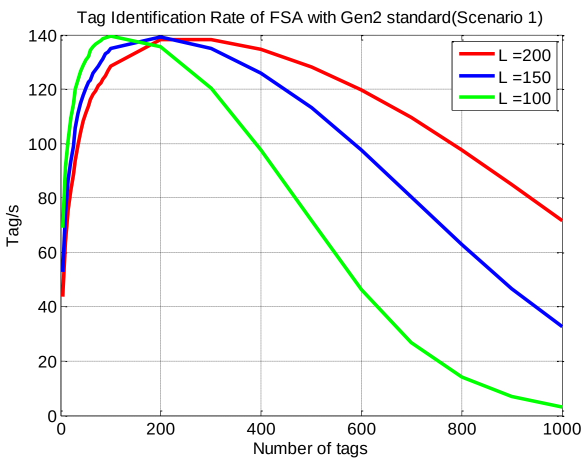
$$Timing\ of\ collision\ slot = T_C * C$$

$$Timing\ of\ success\ slot = T_S * S$$

E = Number of empty slot
C = Number of collision slot
S = Number of successful slot

Tag identification rate = $\frac{Number\ of\ tags}{((T_E*E)+(T_E*C)+(T_S*S)+T_{Query})}$

The parameters showed in Table 4.1 are later applied in FSA with frame size of 100, 150 and 200, DFSA simulation with ideal tag estimation, Schoute algorithm, ILCM and our proposed approach, DFSA with Manchester Coding. We started the experiments from Scenario 1 which has the lowest BLF to Scenario 3 with highest BLF. Figure 4.7, 4.8 and 4.9 presents the simulation results of FSA, DFSA and proposed approach in three different scenarios.



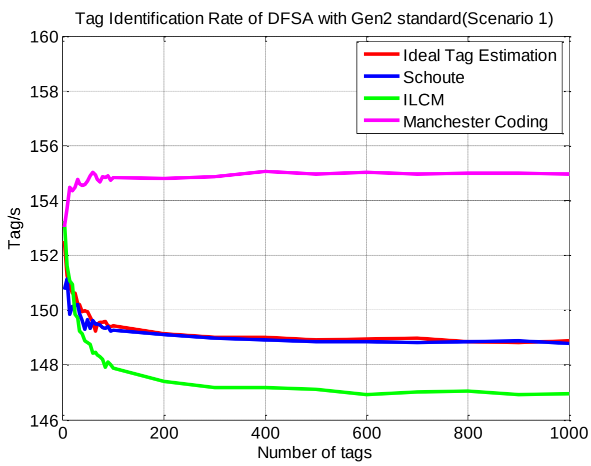


Figure 4.7: Scenario 1 tag identification rate of FSA, DFSA and Proposed Approach

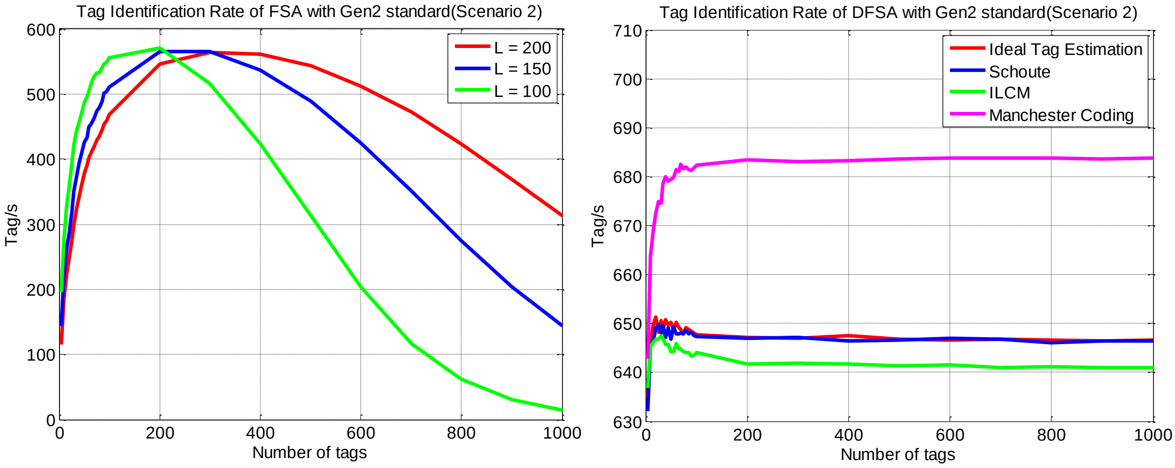


Figure 4.8: Scenario 2 tag identification rate of FSA, DFSA and Proposed Approach

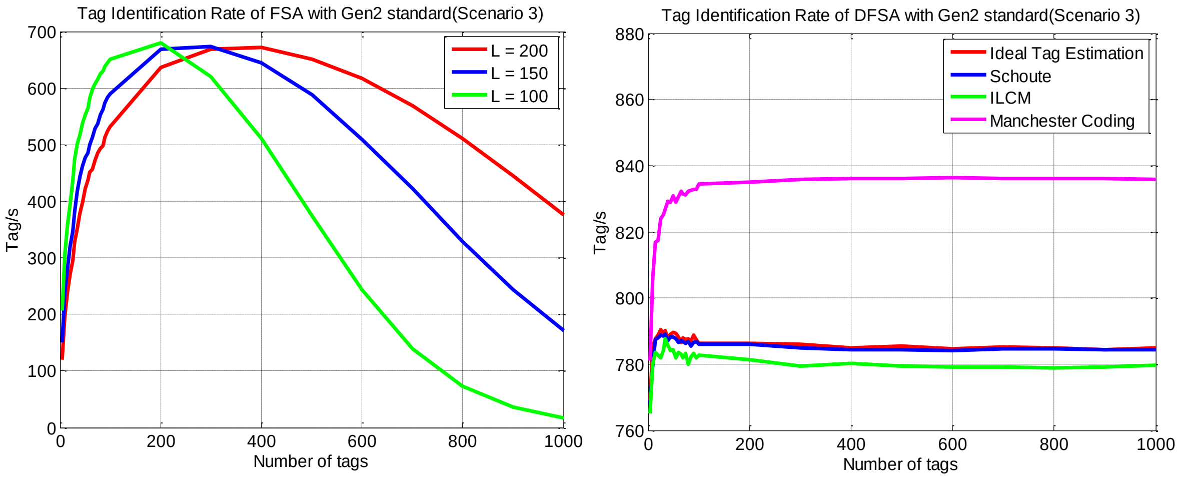


Figure 4.9: Scenario 3 tag identification rate of FSA, DFSA and Proposed Approach

As showed in Figure 4.9, we could observe that FSA, DFSA and proposed approach have the highest tag identification rate in Scenario 3 as compared to Scenario 1 and 2. This is because when larger BLF is used, the read rate of RFID system is higher and hence more tags could read within a second.

From Figure 4.7, 4.8 and 4.9, we could also notice that the number of tags/s that could be recognised by reader in FSA is getting lower when the frame size became smaller and the number of tags became larger. Especially when $L = 100$ and there are 1000 tags, the number of tags could be identified by FSA in three scenarios are only 3, 14 and 17 per second respectively. This is because the frame size provided did not fit the number of tags to be identified. Consequently, the tags are having limited number of time slot that is available to be selected and they might select the same time slot with each other and cause tag collision. As a result, a smaller number of tags would be successfully recognised by the reader and the reader might need to spend a longer period of time in order to recognise all the tags. This proved that the frame size is playing an important role in throughput of FSA as it would affect the tag identification rate severely if it is selected inappropriately.

For DFSA, the simulation results of ideal tag estimation and Schoute algorithm tends to provide a stable tag identification rate as compared to FSA which are 148 tags/s in first scenario, 645 tags/s in second scenario and 784 tag/s in third scenario. ILCM had the lowest tag identification rate as compared to ideal tag estimation and Schoute algorithm due to its limitation which could not adjust the frame size optimally when a small initial frame size is offered. In the traditional DFSA, it would resolve the tag collision from different collision slots in future read cycle. Thus, it might have the possibility to cause high rate of tag collision in future read cycle and more time slots would be used if there is large number of collided tags found in current read cycle. As a result, the time needed by the collided tags to be identified by the reader would be longer. Hence, the number of tags that could be successfully identified by the reader per second in DFSA would be lower as compared to the proposed approach.

In our proposed approach, if any collision bits are detected from the slot reservation codes during Manchester Coding collision bits detection process, the collided tags from each collision slot would store in a collision list. After that, the reader would resolve the tag collision slot by slot instead of resolving all the tag collision in one read cycle. Thus, the time slot used in the proposed approach in each tag redistribution process would be lesser than the traditional DFSA. This would reduce the number of empty slot and also shorten the duration for the reader to resolve the collision indirectly. As a result, more tags could be identified successfully by the reader per second in the proposed approach as compared to the traditional DFSA. From Figure 4.7, 4.8 and 4.9, we could observe that our proposed approach is able to provide the highest and stable tag identification rate which are 155 tags/s in first scenario, 684 tags/s in second scenario and 835 tag/s in third scenario as compared to FSA and DFSA. Therefore, this achieve the objectives of this project which is to utilise Manchester Coding to resolve tag collisions during tag identification process and shorten the timing in Gen2 standard.

In conclusion, the time needed for the reader to successfully read all the tags in FSA heavily rely on the initial frame size used during tag identification process. If an inappropriate frame size is selected, it might require a longer time in resolving tag collision and reduce the system efficiency and tag identification rate of RFID system. Thus, FSA which use static frame size throughout the tag identification process is not recommended to be used in resolving tag collision of RFID system anymore. DFSA which could dynamically adjust the frame size has become a more preferable solution to resolve the tag collision problem in RFID system as compared to FSA. However, traditional DFSA would resolve all the tag collisions detected from different time slots in one read cycle. The problem arises when there is large number of collided tags found in current read cycle. According to our study, DFSA tends to provide optimal frame size that fits the current collision situation and this also means that the number of time slot required for each tag redistribution process would be higher when then number of collided tag is large. Therefore, the slot duration of DFSA during tag identification process would be longer when there is large volume of collided tags found in current read cycle. In our proposed approach, we tried to shorten the slot duration by resolving the collision slot by slot. This is to reduce the number of collided tag in each tag redistribution process and further shorten the time needed to resolve collision. This is because the time slots needed to resolve the collision would be lower when there is smaller number of collided tags. At the same time, this is also able to reduce the number of empty slots created in each future read cycle. As a result, it enables the reader to identify the tags in a shorter duration and also improve system efficiency of RFID system. In short, the proposed approach is more efficient and provides shorter slot duration during tag identification process as compared to FSA and DFSA with ideal tag estimation, Schoute algorithm and ILCM.

CHAPTER 5: CONCLUSION

5.1 Project Review

Some related previous works have been reviewed before developing this project as we need to have basic the idea of how the main problem which is tag collision occurred in RFID system and the solutions used to encounter this problem. In this project we had reviewed the technology involved which is RFID system and ALOHA-based anti-collision algorithms that help to resolve tag collision. In this project, we were mainly focus on two algorithms which are FSA and DFSA respectively.

In this project, we had also provided the critical remark which includes the strengths and weaknesses for all the reviewed related previous works. This is to understand the problems encountered in previous works and we could improve or enhance it in our proposed approach. The main objectives to be achieved in this project are to mitigate the tag collision problem and shorten the slot duration in RFID tag identification process.

The proposed approach in our project would adopt DFSA together with Manchester Coding to reduce the rate of collision in RFID system. Unlike the conventional DFSA, our proposed approach would allow the tags to reserve their selected time slot via an 8-bit binary slot reservation code instead of allocating a time slot directly to a tag. Thus, if the reader found any collision bits in a particular time slot during Manchester Coding, the time slot would mark as a collision slot and will go through tag redistribution process to reserve a new time slot. Besides, the proposed approach would resolve the tag collision slot by slot. Therefore, it could lower the number of time slots used and shorten the time needed by the reader to resolve the collision as the number of collided tags involved in each future read cycle is reduced. As a result, it enables the reader to recognise more tags during tag identification process and enhance the system efficiency of RFID system.

5.2 Discussion

The aim of this project is to pinpoint the importance of anti-collision algorithm in RFID system. As we know, tag collision is always the biggest obstacle for RFID system to achieve high identification accuracy and tag identification rate in RFID system. Thus, it is important to study and improve the existing related works in order to further mitigate the tag collision problem in RFID system.

In this project, we have proposed a new timing concept which applying a bit-tracking technology, Manchester Coding in DFSA. Before that, we had done the study for both FSA and DFSA which are ALOHA-based anti-collision algorithms. This is because these two algorithms are widely used in resolving this problem. In this paper, we had found that FSA is very time consuming while resolving tag collision due to its fixed frame size. Thus, we had adopted DFSA which could dynamically adjust the frame size based on the slot information in our proposed approach.

Furthermore, we also recognised a significant limitation of DFSA which is resolving the collision of all the collided tags in one read cycle. This would increase the number of time slot and time needed for the reader to resolve the collision when there is a large number of a collided tag found in current read cycle. Consequently, this would increase the time needed during the tag identification process. Thus, we tried to resolve this issue by letting the tag to reserve their time slots using an 8-bit reservation code and resolve the collision in slot basis after detecting collision bits using Manchester Coding from each time slot. According to our simulation results, we found that the number of slot used in our proposed approach is lower than FSA and DFSA as the number of collided tags involved in future read cycle is reduced. Besides, it is also able to provide a stable and higher tag identification rate and system efficiency which is ~43% during tag identification process. This is because the time needed for reader to resolve the tag collision would be reduced if the number of collided tag becomes smaller and therefore more tags could be recognised each read cycle. Thus, this achieved the main objectives of this project which is to utilise Manchester Coding to resolve tag collisions during tag identification process and shorten the timing in Gen2 standard.

5.3 Contributions

During the implementation and development of our proposed approach, we had recognised the limitations of existing ALOHA-based anti-collision algorithms. For instance, FSA which remains its initial frame size throughout the tag identification process would cause high rate of tag collision when there is large volume of tag is involved. For DFSA, the frame size is always heavily relies on the tag estimation result and thus the adjusted frame size might not be optimal when there is any estimation error. Besides, DFSA would resolve all the collided tags determined in current read cycle with a new frame size in a new read cycle. Consequently, the reader might spend longer time and use higher number of time slot to resolve the collision if there is large volume of collided tag found in current read cycle.

In our proposed approach, we would detect collision through Manchester Coding by tracing the slot reservation codes in each time slot. If there is any collision bits detected, the collided tags in a time slot would need to restart the whole slot reservation process in future read cycle with a new frame size. Unlike the traditional DFSA, our proposed approach would solve the collision slot by slot. This could cut down the number of collided tags and time slot used in each future read cycle. Therefore, the success rate for a tag to be recognised by the reader would be higher as the reader might spend a shorter period of time to resolve the collision. As a result, our proposed approach could help to reduce the slot timing used in tag identification process and improve the system efficiency of RFID system.

In a nutshell, the proposed approach of this project could mitigate the tag collision problem and enhance identification time in RFID system. Therefore, it enables RFID system to achieve its main advantage which is fast data reading and helps to improve the productivity of the industries or application systems which employed RFID system in their business operations.

5.4 Future works

During the implementation of the proposed approach of this project, we found that the time needed to complete the simulation for 1000 experiments would last for hours or days while the number of tags involved is becoming larger. This happened is due to there might be collision occurred during the tag redistribution process. Thus, these tags might go through many rounds of tag redistribution process if there is any collision detected.

Besides, the slot reservation code used in the slot reservation code regeneration process in proposed approach would only cater up to maximum 255 tags which mean that for each collision slot the maximum number of tags that could receive a new reservation code is only up to 255. This is because the maximum number of tags we used in our experiments is only up to 1000 tags and thus we used randperm() function to regenerate unique slot reservation code for the tags which used the same code to reserve a time slot in slot reservation code regeneration process. As we know, the maximum number for 8-bit binary string is 255. Thus, the proposed approach might not able to cater the case of a collision slot which contained 255 tags and with duplicate slot reservation code.

Thus, the future enhancement for our proposed method should have reduced the chances of collision during tag redistribution process. This is to further reduce the time needed for the reader to resolve collision and the number of time slot used in each future read cycle. Besides, the future work should provide a way to cater for the cases when there are more than 255 tags with duplicate slot reservation code as there might more than 1000 tags to be recognised in real life application.

BIBLIOGRAPHY

Bang, O., Kim, S. and Lee, H. (2009) ‘Identification of RFID Tags in Dynamic Framed Slotted ALOHA’, *Advanced Communication Technology,* 01, pp. 354–357.

Chen, Y., Su, J. and Yi, W. (2017) ‘An efficient and easy-to-implement tag identification algorithm for UHF RFID systems’, *IEEE Communications Letters*, , pp. 1–1. doi: 10.1109/lcomm.2017.2649490.

Cheng, T. and Jin, L. (2007) ‘Analysis and simulation of RFID anti-collision Algorithms’, *The 9th International Conference on Advanced Communication Technology,*. doi: 10.1109/icact.2007.358450.

content (2008) Using RFID & IPv6. Available at: http://ipv6.com/articles/applications/Using-RFID-and-IPv6.htm (Accessed: 18 August 2016).

Use, I.T. of, Policy, P. and Trademarks (2002) *The different types of RFID systems*. Available at: http://www.impinj.com/resources/about-rfid/the-different-types-of-rfid-systems/ (Accessed: 15 August 2016).

Jin, X., Wei, D., Xu, Y., Jin, L. and Huang, X. (2015) ‘A novel RFID tag estimation algorithm based on DFSA’, *2015 IEEE 5th International Conference on Electronics Information and Emergency Communication*, . doi: 10.1109/iceiec.2015.7284479.

Kaewsirisin, S., Supanakoon, P., Promwong, S., Sukutamtanti, N. and Ketprom, U. (2008) ‘Performance study of dynamic framed slotted ALOHA for RFID systems’, *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, . doi: 10.1109/ecticon.2008.4600459.

Landaluce, H., Perallos, A. and Angulo, I. (2014) ‘Managing the number of tags bits transmitted in a bit-tracking RFID collision resolution protocol’, Sensors, 14(1), pp. 1010–1027. doi: 10.3390/s140101010.

McDowell, G. (2009) How does RFID technology work? [Technology Explained]. Available at: http://www.makeuseof.com/tag/technology-explained-how-do-rfid-tags-work/ (Accessed: 22 June 2016).

Qing, Y., Jian-cheng, L., Hong-yi, W. and Rong-jun, S. (2012) ‘A dynamic framed slotted ALOHA anti-collision algorithm based on tag-grouping for RFID systems’, *2012 IEEE 11th International Conference on Solid-State and Integrated Circuit Technology*, . doi: 10.1109/icsict.2012.6467907.

Nov-2013, version 2.0 EPC™ radio-frequency identity protocols generation-2 UHF RFID specification for RFID air interface protocol for communications at 860 MHz – 960 MHz version 2.0.0 ratified (2013) Available at: http://www.gs1.org/sites/default/files/docs/epc/uhfc1g2_2_0_0_standard_20131101.pdf (Accessed: 20 August 2016).

Raja, T. and Perumal, V. (no date) T RAJA AND V PERUMAL: ADVANCED DYNAMIC FRAMED SLOTTED ALOHA ADVANCED DYNAMIC FRAMED SLOTTED ALOHA. Available at: http://ictactjournals.in/paper/IJCT_Vol3_Iss2_P5_548_550.pdf (Accessed: 12 August 2016).

Shakiba, M., Zavvari, A. and Sundararajan, E. (2011) ‘Fitted dynamic framed slotted ALOHA anti-collision algorithm in RFID systems’, *ICIMU 2011 : Proceedings of the 5th international Conference on Information Technology & Multimedia*, . doi: 10.1109/icimu.2011.6122776.

Solic, P., Radic, J., Dujmic, H., Saric, M., Russo, M., Stella, M., Begusic, D. and Rozic, N. (2013) ‘Improved Linearized Combinatorial model (ILCM) for optimal frame size selection in ALOHA-based RFID systems’, *2013 IEEE International*

Conference on Communications Workshops (ICC), . doi: 10.1109/iccw.2013.6649396.

Soli´c, P. and Stella, M. (2013) ‘Tags/s -RFID reader-tag communication throughput using gen2 Q-algorithm frame adaptation scheme’, *Recent Advances in Information Science*, pp. 100-105

Su, J., Sheng, Z., Hong, D. and Wen, G. (2016) ‘An effective frame breaking policy for dynamic framed slotted Aloha in RFID’, *IEEE Communications Letters*, 20(4), pp. 692–695. doi: 10.1109/lcomm.2016.2521839.

Tong, Q., Zou, X. and Tong, H. (2009) ‘Dynamic framed slotted ALOHA algorithm based on Bayesian estimation in RFID system’, *2009 WRI World Congress on Computer Science and Information Engineering*, . doi: 10.1109/csie.2009.248.

APPENDIX A - Weekly Report

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 4 Semester 1	**Study week no.:** Week 2
Student Name & ID: Lee Khai Yi, 14ACB00454	
Supervisor: Dr Robithoh Annur	
Project Title: ALOHA-Based Radio-Frequency Identification (RFID) System With Early Frame Adjustment	

1. WORK DONE Find work related articles and figure out the ideas on how to apply Manchester Coding in DFSA.
2. WORK TO BE DONE • Design the logic flow and flow diagram of proposed approach • The possible limitations of the proposed method
3. PROBLEMS ENCOUNTER Since Manchester Coding technology is always applied on Binary Tree algorithm and thus applying it in DFSA is a new idea.
4. SELF EVALUATION OF THE PROGRESS • Understand the basic idea of how Manchester Coding work in Binary Tree algorithm • Find out the rough idea of how to apply Manchester Coding in DFSA

___________________________ ___________________________

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 4 Semester 1	**Study week no.:** Week 4
Student Name & ID: Lee Khai Yi, 14ACB00454	
Supervisor: Dr Robithoh Annur	
Project Title: ALOHA-Based Radio-Frequency Identification (RFID) System With Early Frame Adjustment	

1. WORK DONE

- Literature review of existing improved RFID anti-collision algorithms
- Understand how Manchester Coding works

2. WORK TO BE DONE

- Produce flow diagram for proposed idea
- Program the proposed idea

3. PROBLEMS ENCOUNTER

Some proposed idea might not be implemented and thus other alternative solutions need to be figured out

4. SELF EVALUATION OF THE PROGRESS

Understand the program flow of proposed idea and bit tracing technology of Manchester Coding

_________________________ _________________________

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 4 Semester 1	**Study week no.:** Week 6
Student Name & ID: Lee Khai Yi, 14ACB00454	
Supervisor: Dr Robithoh Annur	
Project Title: ALOHA-Based Radio-Frequency Identification (RFID) System With Early Frame Adjustment	

1. WORK DONE • Bit-tracking of Manchester Coding • Calculate the tag identification rate and system efficiency of proposed approach
2. WORK TO BE DONE • Apply Gen2 timing in proposed method • Check logic flow of proposed method
3. PROBLEMS ENCOUNTER The time taken for proposed method to identify the tags is extremely long and the binary code generated could only cater for not more than 10000 tags.
4. SELF EVALUATION OF THE PROGRESS Done Manchester code generation for collision checking purpose and slot selection method used in the proposed method.

____________________________ ____________________________

Supervisor’s signature Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 4 Semester 1	**Study week no.:** Week 8
Student Name & ID: Lee Khai Yi, 14ACB00454	
Supervisor: Dr Robithoh Annur	
Project Title: ALOHA-Based Radio-Frequency Identification (RFID) System With Early Frame Adjustment	

1. WORK DONE

- Program code for proposed approach
- Simulations for proposed approach

2. WORK TO BE DONE

- Resolve the slot reservation code collision occurred when there are more than one tags using the same slot reservation code to reserve a time slot
- Redesign the checking process for collision slots

3. PROBLEMS ENCOUNTER

- The tags that reserved the same time slot were not inserted into remaining tag list.
- The unread tags which has the same bit pattern as the tags stored in the success list are removed from the list.

4. SELF EVALUATION OF THE PROGRESS

- Change the slot reservation code generation using randi() instead of randperm().
- Change the checking process for collision time slot and redesign the slot reservation process when there is a collision detected by using Manchester Coding.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 4 Semester 1	**Study week no.:** Week 10
Student Name & ID: Lee Khai Yi, 14ACB00454	
Supervisor: Dr Robithoh Annur	
Project Title: ALOHA-Based Radio-Frequency Identification (RFID) System With Early Frame Adjustment	

1. WORK DONE

- Report writing for Chapter 1, 2 and 3

2. WORK TO BE DONE

- Obtain the simulation results of proposed approach
- Graph plotting for proposed approach and existing ALOHA-based anti-collision algorithms
- FYP poster design

3. PROBLEMS ENCOUNTER

N/A.

4. SELF EVALUATION OF THE PROGRESS

End of program enhancement

____________________________ ____________________________

Supervisor’s signature Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 4 Semester 1	**Study week no.:** Week 12
Student Name & ID: Lee Khai Yi, 14ACB00454	
Supervisor: Dr Robithoh Annur	
Project Title: ALOHA-Based Radio-Frequency Identification (RFID) System With Early Frame Adjustment	

1. WORK DONE Report and poster design enhancement
2. WORK TO BE DONE • Enhance Chapter 3 discussion • Perform Turnitin check
3. PROBLEMS ENCOUNTER N/A.
4. SELF EVALUATION OF THE PROGRESS Reorganise some contents and finalise the report

__________________________ __________________________

Supervisor’s signature Student’s signature

APPENDIX B - Poster

ALOHA-BASED RADIO-FREQUENCY IDENTIFICATION (RFID) SYSTEM WITH EARLY FRAME ADJUSTMENT

LEE KHAI YI
UNIVERSITI TUNKU ABDUL RAHMAN

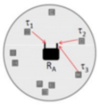
Introduction

Problem Statement

- RFID system is always suffer from tag collision that could lead to low throughput and long identification time
- DFSA that used to solve tag collision has difficulty in selecting an optimal frame size

What is tag collision?

A phenomena when there are more than one RFID tags respond to a reader simultaneously at the same time.



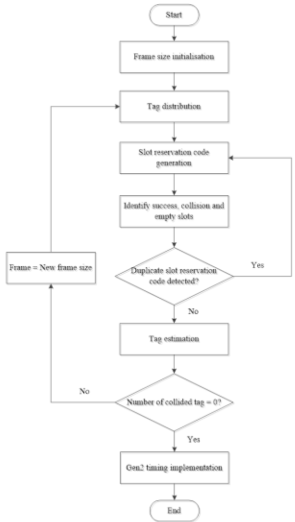
Objective

- Study and simulate Frame slotted ALOHA (FSA) and Dynamic Frame Slotted ALOHA (DFSA) anti-collision algorithms
- Study and apply ideal tag estimation, Schoute and ILCM tag estimation algorithm in DFSA to select optimal frame size
- Understand timing and slot duration of Gen2 standard in RFID system
- Propose a prototype scheme which:
  - utilise Manchester Coding to resolve tag collisions during tag identification process
  - shorten the timing in Gen2 standard

Methodology

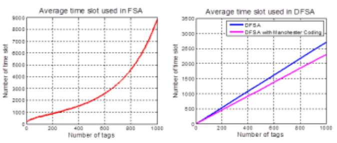
- The simulation of all the algorithms are conducted by using MATLAB.
- 1000 experiments have been conducted for each simulation by changing the number of tags from 5 to 1000.

System Design

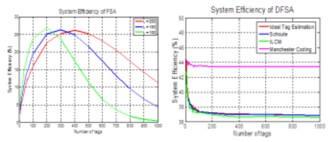


Results

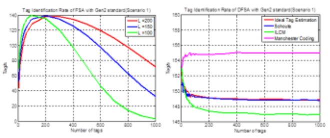
- **Average Time Slot Used in FSA, DFSA and Proposed Approach:**



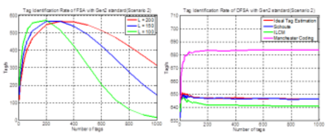
- **System Efficiency of FSA, DFSA and Proposed Approach:**



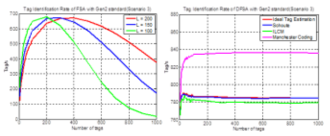
- **Tag Identification Rate of FSA, DFSA and Proposed Approach (Scenario 1-40kHz):**



- **Tag Identification Rate of FSA, DFSA and Proposed Approach (Scenario 2-340kHz):**



- **Tag Identification Rate of FSA, DFSA and Proposed Approach (Scenario 3-640kHz):**



Discussion

- **Average time slot used**
  - Both FSA and DFSA resolve the collision of all collided tags in one future read cycle with a new frame size that fits the number of collided tag.
  - If the number of collided tag found in each read cycle is large, the larger the number of time slot used by the reader to resolve the collision and it would also increase the rate of creating empty slots.
  - In proposed approach, the tag collision is resolved in slot basis and this helps to reduce the number of collided tag in each read cycle and thus the number of time slot used by the reader to resolve collision would be lower than FSA and DFSA.
- **System efficiency**
  - In the proposed approach, when there is a collision bits detected in a particular time slot using Manchester Coding, we would provide a new frame size using ideal tag estimation that fits the number of collided tag of that time slot.
  - This reduces the number of collided tag in each new read cycle and hence the number of time slot used and empty slot produced would be reduced.
  - As a result, proposed approach is able to provide ~43% of system efficiency even the number of tag becomes larger.
- **Tag identification rate and Timing**

  As the number of collided tag involved in each new read cycle is reduced, the time needed by the reader to resolve collision would become shorter and thus the proposed approach could recognise more tag per second as compared to FSA and DFSA.

Conclusion

- FSA is not preferably used to resolve tag collision due to its static initial frame size.
- DFSA which resolves all collided tags in one read cycle would increase the number of time slot and time needed by the reader to resolve tag collision.
- The proposed approach which utilised Manchester Coding resolves tag collision slot by slot could reduce the number of collided tag and time to resolve collision in each read cycle.
- As a result, proposed approach could provide higher system efficiency and tag identification rate than FSA and DFSA as it reduces the number of time slot and time used by the reader to resolve tag collision in each read cycle.

APPENDIX C - Turnitin Similarity Report

Aloha-Based Radio-Frequency Identification (RFID) System With Early Frame Adjustment

ORIGINALITY REPORT

%3	%0	%3	%0
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Qing, Yang, Li Jian-cheng, Wang Hong-yi, and Shen Rong-jun. "A dynamic framed slotted ALOHA anti-collision algorithm based on tag-grouping for RFID systems", 2012 IEEE 11th International Conference on Solid-State and Integrated Circuit Technology, 2012. Publication	%1
2	Solic, Petar, Josko Radic, and Nikola Rozic. "Energy Efficient Tag Estimation Method for ALOHA-Based RFID Systems", IEEE Sensors Journal, 2014. Publication	<%1
3	Sarangan, V.. "A framework for fast RFID tag reading in static and mobile environments", Computer Networks, 20080410 Publication	<%1
4	Dhakal, Sunil, and Seokjoo Shin. "Parametric heuristic schemes for performance improvement in Frame Slotted Aloha based RFID systems", International Conference on ICT for Smart Society, 2013.	<%1
5	Masoud Shakiba. "Fitted dynamic framed slotted ALOHA anti-collision algorithm in RFID systems", ICIMU 2011 Proceedings of	<%1

the 5th international Conference on Information Technology & Multimedia, 11/2011
Publication

6	Ramez Elmasri. "Robust and Dynamic Bin Slotted Anti-Collision Algorithms in RFID System", 2008 IEEE International Conference on RFID, 04/2008 Publication	<%1
7	Shakiba, Masoud, Mandeep Jit Singh, Elankovan Sundararajan, Azam Zavvari, and Mohammad Tariqul Islam. "Extending Birthday Paradox Theory to Estimate the Number of Tags in RFID Systems", PLoS ONE, 2014. Publication	<%1
8	Seungnam Kang, Zornitza Prodanoff. "Chapter 14 RFID Model for Simulating Framed Slotted ALOHA Based Anti-Collision Protocol for Muti-Tag Identification", InTech, 2011 Publication	<%1
9	CAO, Jichang, Lin SHU, and Zhengding LU. "Synchronous Dynamic Adjusting: An Anti-Collision Algorithm for an RF-UCard System", International Journal of Communications Network and System Sciences, 2009. Publication	<%1
10	Agrawal, Tanvi, P. K. Biswas, A. D. Raoot,	

	and Rohit Sharma. "Optimum frame size evaluation framework for efficient tag identification in passive RFID systems", 2013 IEEE Wireless Power Transfer (WPT), 2013. Publication	<%1
11	Submitted to Universiti Teknologi MARA Student Paper	<%1
12	Zurinah Suradi. "A framework of tag anti-collision algorithm for fast identification in RFID system", 2008 International Conference on Computer and Communication Engineering, 05/2008 Publication	<%1
13	Shengli Lai. "ALOHA-Based Anti-Collision Algorithms Used in RFID System", 2006 International Conference on Wireless Communications Networking and Mobile Computing, 09/2006 Publication	<%1
14	Yejun He, , and Xiaoye Wang. "An ALOHA-based improved anti-collision algorithm for RFID systems", IEEE Wireless Communications, 2013. Publication	<%1
15	works.bepress.com Internet Source	<%1
16	Jehn-Ruey Jiang. "Adaptive k-Way Splitting and Pre-Signaling for RFID Tag Anti-	<%1

Collision", IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society, 11/2007
Publication

17 Der-Jiunn Deng. "Optimal Dynamic Framed Slotted ALOHA Based Anti-collision Algorithm for RFID Systems", Wireless Personal Communications, 12/25/2010 <%1
Publication

18 Rushikesh S. Khasgiwale. "Extracting information from tag collisions", 2009 IEEE International Conference on RFID, 04/2009 <%1
Publication

19 Sree Kalyan Ravilla. "Anti-collision policy for RFID systems: fast predict tags in field algorithm", International Journal of Radio Frequency Identification Technology and Applications, 2011 <%1
Publication

20 Urachada Ketprom. "Performance study of dynamic framed slotted ALOHA for RFID systems", 2008 5th International Conference on Electrical Engineering/Electronics Computer Telecommunications and Information Technology, 05/2008 <%1
Publication

21 Kazem Sohraby. "A New Slot-Count Selection Algorithm for RFID Protocol", 2007 Second International Conference on Communications <%1

and Networking in China, 08/2007
Publication

22	Lecture Notes in Computer Science, 2006. Publication	<%1

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	< 5 WORDS
EXCLUDE BIBLIOGRAPHY	ON		

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lee Khai Yi
ID Number(s)	14ACB00454
Programme / Course	Bachelor of Information Technology (Hons) Communications and Networking
Title of Final Year Project	ALOHA-Based Radio-Frequency Identification (RFID) System With Early Frame Adjustment

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: ________ % **Similarity by source** Internet Sources: _______________% Publications: _______________ % Student Papers: _______________ %	
Number of individual sources listed of more than 3% similarity: _________	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor

Name: __________________________

Date: ___________________________

____
Signature of Co-Supervisor

Name: __________________________

Date: ___________________________