อัลกอริทึมการแก้ปัญหาการชนแบบต้นไม้สำหรับอาร์เอฟไอดีโดยใช้การประมาณจำนวนแทกแบบเบยส์

นางสาวสานิกา กฤษณมาลี วิจายาเสกาวารา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560

A Tree-Based Collision Resolution Algorithm for RFID using Bayesian Tag Estimation

Miss Sanika Krishnamali Wijayasekara

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2017

Thesis Title	A Tree-Based Collision Resolution Algorithm for RFID using Bayesian Tag Estimation
By	Miss Sanika Krishnamali Wijayasekara
Field of Study	Electrical Engineering
Thesis Advisor	Associate Professor Lunchakorn Wuttisittikulkij
Thesis Co-Advisor	Assistant Professor Warakorn Srichavengsup

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

....Dean of the Faculty of Engineering
(Associate Professor Dr. Supot Teachavorasinskun)

THESIS COMMITTEE

....Chairman
(Professor Watit Benjapolakul)

....Thesis Advisor
(Associate Professor Lunchakorn Wuttisittikulkij)

....Thesis Co-Advisor
(Assistant Professor Warakorn Srichavengsup)

....Examiner
(Assistant Professor Widhyakorn Asdornwised)

....Examiner
(Assistant Professor Chaiyachet Saivichit)

....External Examiner
(Dr. Pisit Vanichchanunt)

สานิกา กฤษณมาลี วิจายาเสกาววารา : อัลกอริทึมการแก้ปัญหาการชนแบบต้นไม้สำหรับอาร์เอฟไอดีโดยใช้การประมาณจำนวนแทกแบบเบยส์ (A Tree-Based Collision Resolution Algorithm for RFID using Bayesian Tag Estimation) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ลัญฉกร วุฒิสิทธิกุลกิจ, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม: วรากร ศรีเชวงทรัพย์, หน้า.

การระบุด้วยความถี่วิทยุ (อาร์เอฟไอดี) คือ เทคโนโลยีติดตาม/ระบุวัตถุอัตโนมัติแบบไร้สายโดยใช้คลื่นความถี่วิทยุในการส่งผ่านข้อมูลระหว่างตัวอ่านอาร์เอฟไอดีและแท็ก ระบบอาร์เอฟไอดีมีการประยุกต์ใช้งานอย่างแพร่หลาย อาทิ อุตสาหกรรมการผลิต, ระบบดูแลสุขภาพ, การขนส่งสินค้า และการเกษตร แม้ว่าจะมีการใช้งานเทคโนโลยีอาร์เอฟไอดีอย่างแพร่หลาย แต่ปัญหาการชนกันของแท็กก็ยังคงมีปรากฏอยู่ในระบบอาร์เอฟไอดี ซึ่งเป็นเรื่องน่ากังวลใจและจัดว่าเป็นปัญหาที่มีความท้าทาย ปัญหาการชนกันของแท็กเกิดขึ้นในกรณีที่แท็กหลายตัวพยายามส่งข้อมูลไปยังตัวอ่านเดียวกันในเวลาใกล้เคียงกันโดยไม่ได้มีการประสานกันล่วงหน้า มาตรฐานอุตสาหกรรมของเลขรหัสสินค้าอิเล็กทรอนิกส์ในปัจจุบันเลือกใช้อัลกอริทึมคิวในการแก้ปัญหาการชนกันของแท็ก เนื่องจากอัลกอริทึมคิวทำงานโดยใช้หลักการของโพรโทคอลอโลฮา ประสิทธิภาพสูงสุดของระบบที่ทำได้มีค่าเพียงประมาณ 34%

ในงานวิจัยนี้ เรานำเสนอวิธีป้องกันการชนแบบใหม่ 2 วิธี คือ ต้นไม้พลวัตดัดแปลงด้วยการประมาณแบบเบส์ (บีอี-เอ็มดีที) และต้นไม้พลวัตดัดแปลงด้วยการตัดแบ่งแบบไบนารี (บีเอส-เอ็มดีที) ซึ่งมีสามารถทำงานได้ดีกว่าโพรโทคอลป้องกันการชนที่มีอยู่เดิมทั้งหมด โพรโทคอลป้องกันการชนทั้งสองวิธีแบ่งการทำงานออกเป็น 2 ช่วง คือ ช่วงการประมาณจำนวนแท็ก และช่วงการระบุแท็ก โดยในช่วงแรกของบีอี-เอ็มดีที เราเสนอวิธีการประมาณจำนวนแท็กอิงสล็อตอโลฮาแบบเบส์ ซึ่งสามารถสะสมและรวบรวมข้อมูลที่ได้ในแต่ละสล็อตสำหรับใช้ประมาณจำนวนของแท็ก และใช้ในการกำหนดขนาดของเฟรมเริ่มต้นที่ต้องใช้ในช่วงที่สอง ในช่วงที่ 2 ของบีเอส-เอ็มดีที เรานำเสนออัลกอริทึมต้นไม้พลวัตดัดแปลง (เอ็มดีที) ซึ่งใช้ค่าประมาณขนาดของเฟรมจากช่วงแรกเป็นค่าเริ่มต้นของช่วงที่สอง และตามด้วยการใช้อัลกอริทึมต้นไม้ไบนารีแบบข้ามสล็อตที่มีการชน สำหรับอัลกอริทึมบีเอส-เอ็มดีที การทำงานช่วงแรกใช้วิธีการประมาณจำนวนแท็กด้วยการตัดแบ่งแบบไบนารี และใช้อัลกอริทึมต้นไม้พลวัตดัดแปลงเอ็มดีที ในช่วงที่ 2 พร้อมกับเทคนิคในการประมาณขนาดของเฟรมเริ่มต้นเพื่อทำให้ประสิทธิภาพของระบบมีค่าสูงสุดสำหรับทุกค่าของจำนวนแท็ก

เรายังได้นำเสนอแบบจำลองการคำนวณทางคณิตศาสตร์สำหรับแต่ละอัลกอริธึมสำหรับคำนวณค่าประสิทธิภาพของระบบและประสิทธิภาพของระบบในเชิงเวลา แบบจำลองการคำนวณทางคณิตศาสตร์นี้ได้รับการตรวจสอบความถูกต้องโดยใช้เทียบผลกับการจำลองด้วยคอมพิวเตอร์ ผลลัพธ์เชิงตัวเลขยืนยันว่าบีอี-เอ็มดีที มีประสิทธิภาพการของระบบ 45% และประสิทธิภาพของระบบในเชิงเวลา 78% ในขณะที่บีเอส-เอ็มดีที มีประสิทธิภาพของระบบ 46% และประสิทธิภาพของระบบในเชิงเวลา 80%

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต ....
สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่อ อ.ที่ปรึกษาหลัก ....
ปีการศึกษา 2560 ลายมือชื่อ อ.ที่ปรึกษาร่วม ....

5871435121 : MAJOR ELECTRICAL ENGINEERING
KEYWORDS: ANTI-COLLISION, FRAME, TREE, RFID, TAG, READER

SANIKA KRISHNAMALI WIJAYASEKARA: A Tree-Based Collision Resolution Algorithm for RFID using Bayesian Tag Estimation. ADVISOR: ASSOC. PROF. LUNCHAKORN WUTTISITTIKULKIJ, CO-ADVISOR: ASST. PROF. WARAKORN SRICHAVENGSUP, pp.

Radio Frequency IDentification (RFID) is a promising wireless object identifying technology which uses radio frequency waves to transmit data between an RFID reader and tags. The RFID systems have been effectively applied in different areas, like manufacturing, healthcare, supply chain, transportation and agriculture. Despite the vast deployment of the RFID technology in practice, the inherent RFID tag collision problem still persists as a serious concern and remains a challenge. The tag collision problem happens when some tags in reader's vicinity try to transmit data to a reader simultaneously without priori coordination. The existing RFID Electronic Product Code (EPC) Class 1 Generation 2 (Gen 2) industrial standard family uses the Q algorithm as its anti-collision protocol to resolve the tag collision problem. As the Q algorithm relies on the concept of ALOHA protocols, the achievable maximum system efficiency is only around 34%.

In this thesis, we propose two novel anti-collision protocols, namely Bayesian Estimation based Modified Dynamic Tree (BE-MDT) and Binary Splitting Modified Dynamic Tree (BS-MDT), which outperform all existing anti-collision protocols. Both protocols use two phases of operations, i.e., estimate the amount of tags in the system and identify all of them. In the first phase of BE-MDT, we propose a slotted ALOHA based Bayesian tags estimation method which can accumulate the prior knowledge in each slot to estimate the amount of tags in the system and decide the initial frame size to use in the second phase. In the second phase of BE-MDT, we introduce Modified Dynamic Tree (MDT) algorithm which takes the estimated frame size in the first phase as the initial frame and follow by a definite collision skip binary tree algorithm to identify the tags. In our second algorithm, which is BS-MDT, we follow a binary splitting-based tag estimation method in the first phase and use the MDT algorithm in the second phase with a technique to estimate the initial frame size to maximize the system efficiency for any range of tags.

We also present the mathematical models for each algorithm to determine the system efficiency and time system efficiency. The mathematical models are validated through computer simulations. Numerical results confirm that the BE-MDT achieve the system efficiency of 45% and the time system efficiency is 78%, whereas the BS-MDT achieves the system efficiency of 46% and the time system efficiency of 80%.

Department:	Electrical Engineering	Student's Signature	
Field of Study:	Electrical Engineering	Advisor's Signature	
Academic Year:	2017	Co-Advisor's Signature	

ACKNOWLEDGEMENTS

This work has been carried out at Telecommunications System Research Laboratory (TSRL), Department of Electrical Engineering, Chulalongkorn University. The financial support by the 100th Anniversary Chulalongkorn University for Doctoral Scholarship and the top-up scholarship from Electrical Engineering are gratefully acknowledged. I am very thankful to Kingdom of Thailand and Chulalongkorn University for giving me an opportunity to do my PhD and to achieve one of my life goals.

I would like to express my deep appreciation for the invaluable guidance, continuous support and adequate patience I have received from my supervisor, Associate Professor Lunchakorn Wuttisittikulkij. The research and thesis could never be completed without his precious advice and encouragement.

Specially, I want to give my gratitude to Aj. Suvit Nakpeerayuth for his valuable time and tremendous help during my study and also for generous advice to my research. My sincere thanks go to Dr. Warakorn Srichavengsup for his excellent support to my research work. Next, I greatly appreciate all the Professors in my PhD thesis examination panel for their valuable and supportive comments and the time that they shared with me.

I truly thankful to all the Professors in Electrical Engineering department, Chulalongkorn University for sharing the useful background knowledge for me to understand the difficult theories and concepts very easily. I also admire the non-academic staff of Electrical Engineering department, Chulalongkorn University for their kindness support and patient towards me.

Many thanks to all the friends I met in TSRL lab and a special thanks goes to Dr.Muhammad Saadi, Mr. Quynh Le, Miss. Phyo May Thet, Miss Ei Ei Tun, Mr. Manus Pengnoo and Mr. Pruk Sasithong for their encouragement and support all the way through my study. Further, I thank to Dr. Robithoh Annur and all the colleagues I met in LWK group.

Most importantly, I will not complete or even start my study without the support from my father and mother. It is impossible for me to reach this step without your love, caring, sacrifice and understanding. I am grateful to my parents and my two sisters for their unconditional support. I dedicate this thesis to my beloved grandparents.

CONTENTS

Page

List of Figures

List of Tables

List of Abbreviations

RFID	Radio Frequency Identification
EPC	Electronic Product Code
Gen2	Generation 2
RF	Radio Frequency
MAC	Medium Access Control
NFC	Near Field Communication
SDMA	Space Division Multiple Access
FDMA	Frequency Division Multiple Access
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
ACK	Acknowledgement
NACK	Negative Acknowledgement
FSA	Frame Slotted ALOHA
DFSA	Dynamic Frame Slotted ALOHA
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
QT	Query Tree
MDT	Modified Dynamic Tree
BTA	Binary Tree Algorithm
MTA	Modified Tree Algorithm
ODT	Optimum Dynamic Tree
BE-MDT	Bayesian Estimation based Modified Tree Algorithm
BS-MDT	Binary Splitting Modified Tree Algorithm

Chapter 1 Introduction

The radio frequency identification (RFID) is one of the popular wireless technology used in automated object tracking industry and tag collision is one of the main problem which can negatively affect the system throughput by increasing tag identification delay. This PhD thesis examines the techniques to enhance the tag identification efficiency in RFID systems by introducing efficient anti-collision algorithms with highest tag identification efficiency than the available algorithms in literature to date. In this chapter, an introduction to the study is given by stating the background, motivation, objectives, problem concern and contributions. It also provides an overview of the subsequent chapters of this thesis.

1.1 Background

1.1.1 RFID Technology

The RFID is a mechanism of identifying an object automatically with the aid of appropriate communicational devices and protocols. RFID enabled applications are used to track people, assets, documents, health care, library systems and wherever tracking is required, RFID tag identification can play its role. As shown in Figure 1.1, an RFID system consists of readers and tags. A reader or the transceiver is an electronic device which uses radio waves to communicate with tags and transfer the data between the software application and tags to track the asset in its interrogation zone. There are three categories of RFID tags; passive, active and semi-active. Active tags are self-powered using its internal batteries and use the battery power to broadcast the radio waves to the reader. The semi-active tags are powered by its own battery power and depend upon the power supply from the reader to broadcast the radio waves. The passive tags fully rely on the reader and powered up by reader’s radio frequency (RF) signal. The passive tags are less expensive as compared to the active and semi-active tags due to the less complexity in the hardware structure [1, 2].

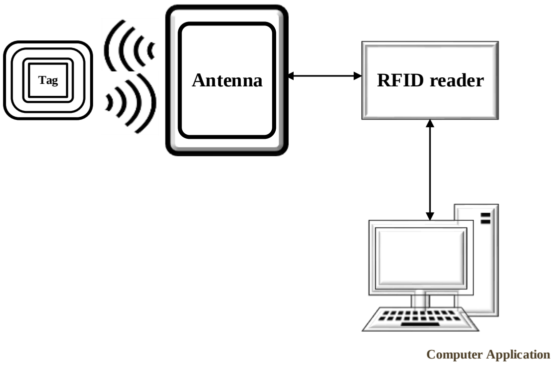


Figure 1.1 Components in a Basic RFID system

In earlier days, Barcode reading was a prominent technique used for inventory management. However, it has its limitations such as, barcode readers work for short ranges and only one object can be scanned at a time the identification. These limitations make the process slow. Currently, Near Field-Communication (NFC) is becoming as a prominent technology in the asset tracking system and it does not require any dedicated reader. NFC enabled devices such as smart phones, or tablets can operate as reader and communicate with other devices which are comprising NFC tags. These NFC enable devices can share the information simply by tapping the two devices in proximity. [3, 4].

RFID technology is playing a key role in current technological industry by revolutionizing the object tracking industry around the globe. Object tracking is one of the main prerequisite to internet of things (IoT) which is presently highly demanded. IoT is a process of monitoring the status of physical objects by capturing desired data from the objects and communicate those facts over the IP network to software application. The entire process is automated without depend upon the manual user interaction [5].

RFID tag collision is one of the main problem occurred during the reader and tags communication period. When several tags are activated by the reader and those activated tags reply the reader at same time, collision happens. Therefore, due to the bulk of acknowledgements from the tags, the reader fails to identify the tags. This failure increases the delay of tag identification in RFID systems and it wastes the system resources and energy. In order to mitigate such a type of problem, lots of Medium Access Control (MAC) layer protocols have been introduced in literature to serve as anti-collision protocols in RFID systems [2].

1.1.2 Anti-Collision Protocols

As outlined earlier, the simultaneous replies from the several tags to reader generate the collisions and this leads to delay in tag identification process. In order to resolve such a kind of collisions, some anti-collision protocols are invented. These protocols can be broadly being divided into, Space Division Multiple Access (SDMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA) and Time Division Multiple Access (TDMA). Among these four categories, the larger number of anti-collision protocols are initiated based on TDMA protocol [6-8]. These protocols can be divided in to reader driven and tag driven whereas, tag driven are further classified into pure ALOHA and reader driven is classified into slotted ALOHA, frame slotted ALOHA and tree. The classification structure of anti-collision protocols is given in Figure 1.2.

In 1970, Norman Abramson [9] proposed a novel medium access control protocol known as ALOHA or pure ALOHA. With respect to the communication processes between the tags and the reader in RFID, a tag randomly transmits its tag ID to the RFID reader and waits for an acknowledgement. An acknowledgement (ACK) is transmitted to a tag when the reader receives only one tag ID successfully and a negative acknowledgement (NACK) is sent when a collision has happened. Based on pure ALOHA concept, tags must wait for a random backing off time with a negative acknowledgement before sending their IDs. In 1975, Roberts [10] did a simple modification to pure ALOHA, where a synchronous data transmission happens in a specific time period called slot and the retransmission occur after a random number of slots. This is known as slotted ALOHA.

In pure ALOHA and slotted ALOHA, the RFID tag which is having a higher response rate will collide with other tags response frequently when accessing the shared channel or a slot. Therefore, the frame based slotted ALOHA concept is introduced to have only one response from each tag in a reader's range in a given frame, where the frame is a collection of slots. In Frame Slotted ALOHA (FSA) the frame size is set to fix number of slots, and when the frame size is not fixed and changes dynamically it is known as Dynamic Frame Slotted ALOHA (DFSA). DFSA operates in multiple rounds and in each round the frame size is dynamically changed based on previously used frame feedback, tag number, etc,. Therefore, DFSA requires some sort of tag estimation techniques to decide the next frame size. In Figure 1.3, an execution examples of FSA and DFSA algorithms are given to illustrate the process of FSA and DFSA algorithms. In 1983, F. C. Schoute presented the DFSA algorithm [11] by estimating previous frame tag count as $S+2.39C$, where S gives the number of success and C gives the amount of collision slots happened in the frame. Success slot means a single tag contains in a slot, while a collision slot indicates several competing tags are involved in a slot. In [12], Vogt introduces an estimation mechanism based on the minimum mean square error or the minimum distance between the mean of success, idle and collision in a frame and the actual read results. Idle means that the slot doesn't contain any tags. In [13], a Bayesian estimation based tag estimation method is given to identify the next frame size in DFSA. W.T.Chen [14], considered the three possible outcomes of one read cycle in a frame of empty, success and collision are independent events and derived a tag estimation method based on a posterior probability equation and multinomial distribution in order to get the next frame size use in DFSA. In [15], a better method of a posterior probability is reflected by taking in to account that the tags in a frame are multinomial distributed and are mutually dependent for different slot types. To enhance the precision of tag estimation and to reduce the computation complexity, [16] presented three Bayesian estimation models, with three different risk functions A linearized combinatorial model is presented in [17] to decide the optimum frame size in DFSA with less complexity and high accuracy.

Q protocol, is the currently used protocol in RFID standard such as ISO/IEC 18000-6 Type C and Electronic Product Code (EPC) global Class 1 Generation 2 [18]. In Q protocol, the reader initially broadcast the slot counter Q, which indicates the frame size of 2^Q and the reader allows to increase and decrease the frame size with a constant value C based on the ternary feedback of idle, success and collision from a slot, where $0.1 \le c \le 0.5$. Advanced properties for Q algorithm are proposed in [19, 20] to enhance the performance of convectional Q algorithm.

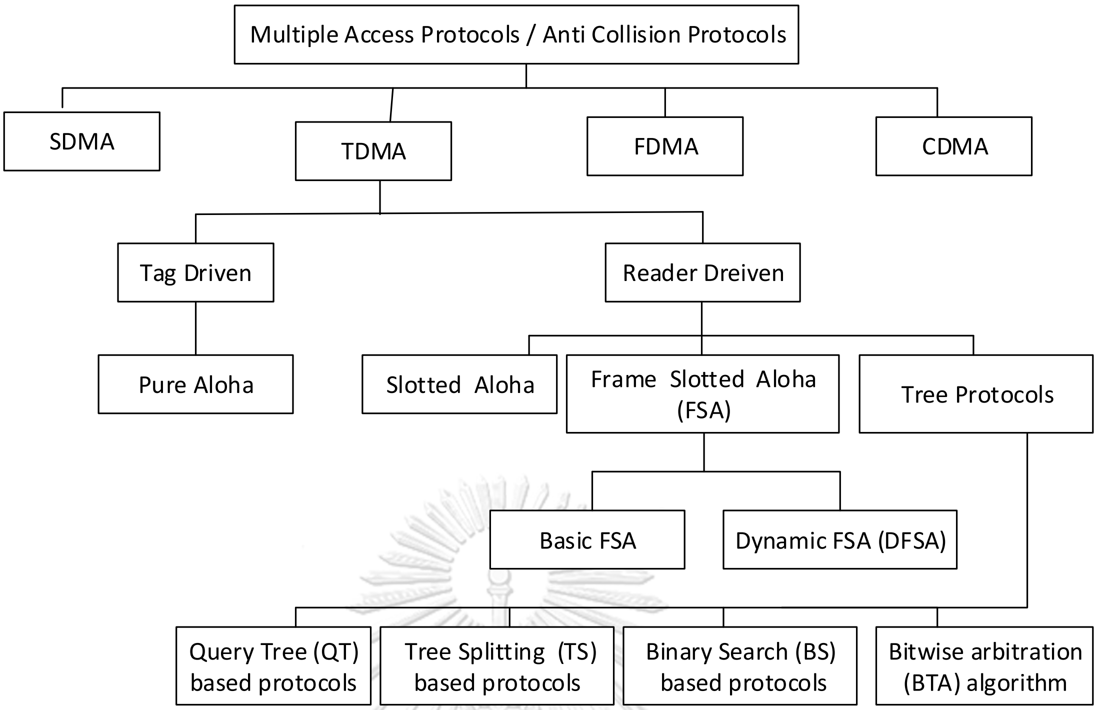


Figure 1.2 Anti-collision protocols [6]

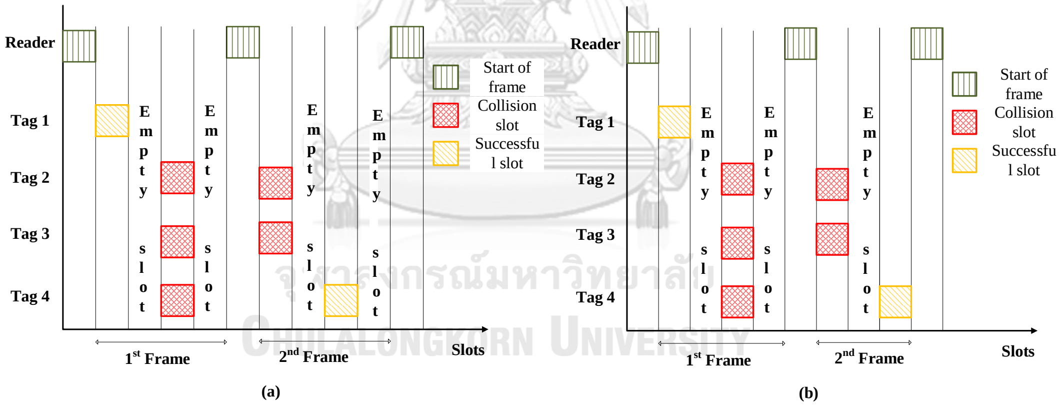


Figure 1.3 An execution example of FSA and DFSA protocol with respect to 4 tags.

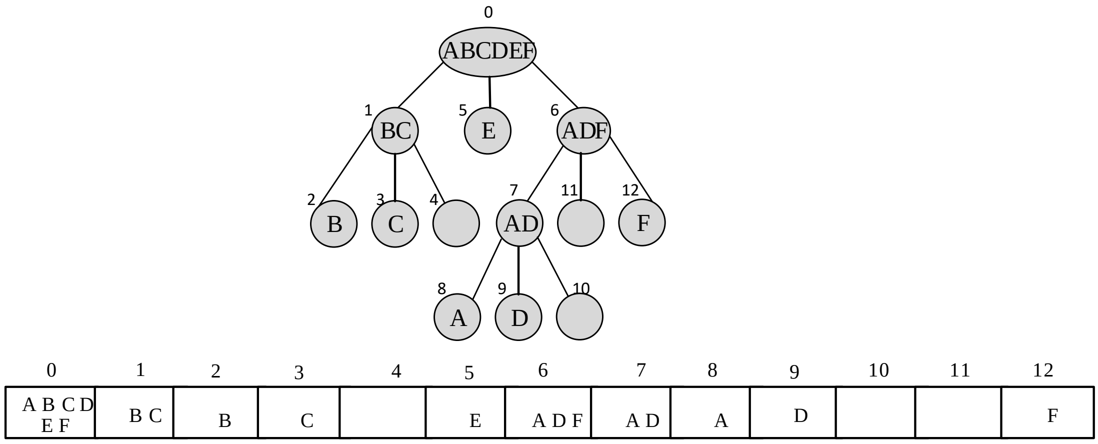


Figure 1.4 An execution of ternary tree algorithm with six tags

In 1979, J.I. Captanakis [21, 22] introduced the tree algorithm as a multiple access protocol in wireless communication systems. The introduction to binary tree has been discussed in [22], in which the collided tags are grouped into two subgroups along the tree structure, until the leaf nodes in the tree structure contains only one tag or none. In Figure 1.4, it shows an example for the execution of ternary tree algorithm with five initial competing tags. In addition, [22] initiated the dynamic binary tree algorithm, where the tree structure follows binary tree concept except the top level where the initially required slot count is decided in relation to the amount of tags in the system. [23], presented a Q-ary tree algorithm with the consideration of binary and ternary feedbacks in binary and ternary tree concepts. [24], introduced an advancement to the basic binary splitting tree known as adaptive binary splitting tree, which can improve the tag identification efficiency. The adaptive tree concept is given in [25], which decides the splitting factor of the current collided slot with respect to the collided number of tags in the current slot. This adaptive tree structure has been adopted in [26] namely, Tree Slotted ALOHA (TSA) along with the Vogt's estimation to guess the number of collided tags and decide the next subgroup size of the splitting tree.

In Query Tree (QT) protocol, the tags are divided into two subgroups based on their tag IDs as given in [27]. At the beginning of the tag identification process, the reader queries a prefix (which is a basically a bit string) to collect the information of the tags in its interrogation area. The tags who are having the matching prefixes in their IDs reply to the reader. When the reader experiences more than one reply, reader queries for one more bit longer prefix. However, according to [28], it is difficult to implement the query tree mechanism in the EPC Global standard. In [29], an improved query tree protocol was introduced, known as bit collision detection based QT (BQT) to detect the collision in each bit. [30], proposed a QT based protocol called as Adaptive Query Tree (AQT), where the reader additionally maintain a candidate queue (CQ) other than the main queue (Q) in conventional QT protocol.

In Binary Search (BS) algorithm, initially, a reader transmits a serial number to the tags. Then the tags, those who have the tag IDs equivalent or lesser than the receive serial number send a reply to the reader.

Next, the reader applies the bit by bit Manchester coding for the reply and when a collision is experienced, the reader splits the tags into subgroups based on the collided bit [31].

In Bitwise Arbitration Algorithm (BTA), from tag to reader a synchronous bit by bit transmission happens from most significant bit (MSB) to least significant bit (LSB) in a tag ID, with the bit position specifies by the reader. A collision is encounter when two tags respond with different bit values.

There are many hybrid algorithms introduced in the literature where the advantages of tree, and ALOHA algorithm are combined to create an efficient protocol in ant-collision paradigm. In [32], binary tree splitting tree slotted ALOHA (BSTSA) algorithm is presented which follows the binary splitting tag estimation method along with TSA [26]. [33], introduced three different tag anti-collision protocols using dynamic binary tree, binary splitting tag estimation and Q algorithm base optimum frame size selection mechanism. An optimal binary tracking tree (OBTT) is introduced in [34], where the bit estimation is used to estimate the number of tags in the system. Further, it introduced the optimal partitioning with frame slotted ALOHA concept and the collided tags are further identified using binary tree techniques.

Recently, [35] introduces an early frame breaking policy in DFSA to identify a suitable frame size to currently presenting tags in the system. After identifying the best frame size, the collided tags are sub-grouped and identified using DFSA. In [36], a binary splitting based an idle slot skipping mechanism is introduced by initiating a binary value of Q. In [37], Dynamic Sub-frame-based Maximum A Posterior probability method (DS-MAP) is introduced to estimate the backlog in a sub frame to decide the next frame sizes to use in DFSA. A Collision-tolErant Dynamic Framed Slotted ALOHA (CE-DFSA) algorithm is presented in [38], which tries to detect several tags in a slot to reduce the overall tag identification delay during the RFID tag identification process.

1.2 Motivations for the research

This section discusses the main motivation factors that persuaded the author to undertake this research in PhD study. Firstly, the RFID technology is playing a key role by revolutionizing the object tracking industry around the globe. With the evolution of IoT, tag identification importance has been increased many folds. Secondly, in order to enhance the tag identification process in RFID system, many research works were carried since ages in the field of anti - collision MAC protocols. There is an increased interest of researchers in the last five years in the said area. These researchers were able to introduce several novel anti- collision protocols with the system efficiency range from 40% to 42% [32, 33, 35].

1.3 Problem Concern

Initially, in an RFD system, the RFID reader doesn’t know anything about the number of tags in the system. During the tag identification process, firstly, the reader broadcast the frame size. Frame size the number of slots that the competing tags can choose randomly in order to communicate with the reader. Afterwards, the reader reads each slot in the frame and identifies the tags based on the feedback from each slot. There are mainly three types of feedback which can originate from a slot i.e. idle, success and collision, where these indicate no tag, one tag and many tags select a slot respectively. Usually, when the frame size is larger than the number of tags, the reader can experience more number of idle slots which results the wastage of slots. On the other hand, when the frame size is smaller than the tags more number of collisions happen, and this result in using more slots to detect the tags. Therefore, to enhance the performance of tag identification, it is necessary to identify the optimum frame size for the existing number of tags in RFID system. Furthermore, if the reader initially has some idea about the existing number of tags in the system, the usage of slots can be controlled, and this makes the tag identification process more efficient. Thus, an accurate tag estimation process can make a significant improvement to the tag identification process. Based on the tag identification, the tree based anti-collision algorithms can achieve higher tag identification efficiency than other ALOHA based algorithms and 42% tag identification efficiency has been achieved in literature. To the best of our knowledge, this is the highest achievable efficiency to date.

1.4. Objectives

The main concern of this thesis is to design an efficient tag identification protocol in RFID system with minimum tags collision resolution time. It is interesting to analyze some accurate tag estimation methods and collision resolution protocols to implement an efficient tag identification algorithm, which can achieve more than 42% identification efficiency.

1.5 Contributions

We have done three major contributions in this thesis. As a first contribution, we propose an algorithm which comprises two phases of operations. In the first phase, the algorithm aims to obtain the number of competing tags in the system. To acquire the estimated number as fast as possible, an efficient slotted based Bayesian tag estimation is applied. In the second phase, we use an efficient technique to dynamically set the frame size of first level of the tree, based on the estimated tag count from phase one. Then, we use definite collision skip binary tree to resolve the conflict among the tags and identify them. The proposed algorithm in the second phase is called as modified dynamic tree.

In our second contribution, we propose another efficient algorithm which also contains two phases for tag estimation and identification. In the first phase, the binary splitting tag estimation is adopted where the left most branch of the tree grows until an idle or a success experience in the left most leaf node in the tree structure. Then, the tags in the right most nodes in each level of the tree structure are resolved based on propose modified dynamic tree algorithm. The effectiveness of the proposed methods is validated using analytical models and computer simulations. The two proposed algorithms achieve around 45%-46% system efficiency in RFID tag identification process which is gives 4% higher system efficiency than the best performance algorithms to date.

Our third main contribution is to apply the proposed algorithms in EPCglobal Gen2 RFID standard. The system efficiency is the most common measure of performance for anti-collision protocols. However, the recent studies for RFID systems aim at maximizing the time system efficiency as it considers the fact that timing values of idle slots are shorter than timing values of successful and collision. It is necessary to do an investigation on the time system efficiency which is reflect the actual operation of RFID tag identification according to EPCglobal Gen2 standard with the defined various recommended time parameters.

1.6. Outline of the thesis

Chapter 2 reviews the literature on anti-collision algorithms and tag estimation methods. One purpose of this chapter is to review the literature of tag identification techniques which are already standardized and in use. This chapter includes a discussion of the nature of work undertaken at various levels to improve the tag estimation and identification practices in very recent years. Finally, the RFID standards for tag identification are reviewed with some simulation results.

Chapter 3 introduces five tag identification algorithms which are the variation of the binary tree algorithm. In this chapter, the newly derived mathematical models for each algorithm are given and the average amount of slots needed in tag identification for each algorithm is discussed in detail. Finally, the derived mathematical model and the simulation results are discussed.

Chapter 4 mainly focuses on the proposed BE-MDT tag identification method. This chapter reviews the literature on the importance of slotted based Bayesian tag estimation and on the creation of look up tables. Furthermore, it discussed the importance of selecting the proper frame size in tag the identification using binary tree. In addition, it justifies the reason why the proposed tag identification method is powerful using some simulation results.

Chapter 5 presents the next proposed algorithm of BS-MDT along with the mathematical models. The originated simulation results based on system efficiency, average delay, average number of collision slots and average number of idle slots are given to validate the mathematical model of the proposed method.

Chapter 6 presents the RFID EPC GlobalGen2 standard timing parameters to validate the propose algorithms in terms of time system efficiency and rate of tag identification. The test results of the proposed method are compared to seven other well-known tree-based algorithms.

Chapter 7 concludes the contributions of our work and indicate the areas where we can further improve in future works.

Chapter 2 Literature Review

This chapter reviews the research literature of anti-collision algorithms and tag estimation methods. Firstly, we discuss the algorithm uses in the EPC Gen2 RFID standard namely Q algorithm. Then, the basic binary tree with fair and bias splitting probability is reviewed. Next, the execution process of several variation of binary tree algorithms are introduced with examples while reweaving the algorithm which gives the highest efficiency in the literature up to date. Latter of the chapter, several well-known tag estimation techniques which are already introduced in RFID systems are discussed and the performance of those tag estimation methods with respect to the estimation error are compared.

2.1 Q Algorithm

Q algorithm [18] is an ALOHA based protocol which is used in ISO 18000-6 Type C and EPC Gen 2 standards. It uses an adaptive frame adjustment by analyzing the feedback of each slot. Usually, the RFID reader broadcast a value of *Q*, which useful to indicate the frame size of 2^Q to the tags in the RFID system. In Q algorithm, the *Q* value is adjusted dynamically by maintaining a constant called *C*, where it ranges from 0.1 to 0.5 with ternary feedback of idle, success and collision. For an idle feedback the *Q* decreases and for a collision the *Q* value increases by the constant of *C*. For a success, the *Q* doesn't change. When the frame size is larger than the number of tags, the probability of experiencing more number of idle slots increases, while the frame size is smaller than the number of tags, probability of occurrence of collision increases. Therefore, using this algorithm, a frame size closer to the remaining number of tags can be achieved without using any estimation method. This algorithm can achieve 0.34 efficiency in tag identification and Figure 2.1 illustrates the flow of Q algorithm.

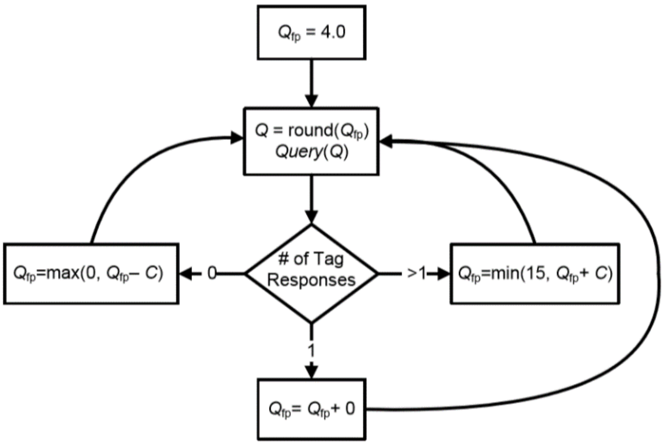
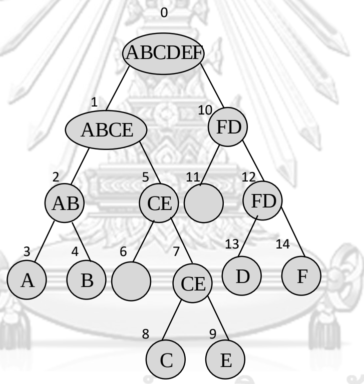


Figure 2.1 Execution flow of Q algorithm

2.2 Basic Binary Tree Algorithm

Binary Tree Algorithm (BTA) also known as fair tree is used in ISO/IEC 18000-6 Type B, EPCGlobal Class 0 and EPCGlobal Class 1 as the anti-collision protocols in RFID tag identification. This is originally developed by Capetanakis [22] in 1979 with the system efficiency of 0.346. In BTA, each tag maintains a counter to track the subgroup and the state of identification. Firstly, the counter value is 0. The tags receive the query command from the reader, the tags with the counter value equal to 0 sent their IDs back to the reader. Based on the acknowledgment from tags , the collision happens when the reader receives more than one tag ID. Then the tags in this colliding group create a 0 or 1 binary number and add it to their counter value, and all the other unidentified tags increase their counter value by one. For no tag response which indicates an idle, all the unidentified tags increase the counter value by one. For single tag responses all the unidentified tags decrease the counter value by one.

Figure 2.2 displays an example of tag identification using BTA where six users (A, B, C, D, E, F) are initially collided in the initial slot and required fourteen slots to resolve this collision and identify the tags.



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A B C D E F	A B C E	A B	A	B	C E		C E	C	E	D F		D F	D	F

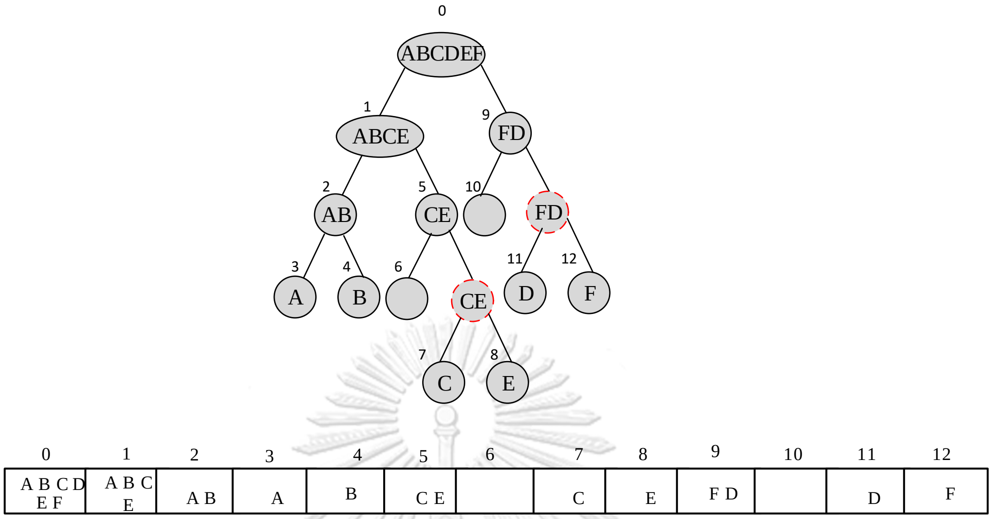
Figure 2.2 An execution example of BTA

2.3 Modified Tree Algorithm

Modified tree algorithm (MTA) [39, 40] is an algorithm which gives a 0.375 system efficiency by skipping the definite collision slots in basic binary tree algorithm with fair splitting probability. As aforementioned, in BT, the collided slots are further split into two subgroups. If the first subgroup is an idle, it is certain that the second subgroup is a collision. Therefore, a slot wastage can be reduced by just splitting the second subgroup into two subgroups by pretending that the collision has occurred.

Based on the example given in Figure 2.2, the slot 7 and 12 are followed by idle slots in slot 6 and 11. Therefore, all the collided users in slot 7 and 12 can be split into two subgroups without reading slot 7 and 12. The modified tree algorithm execution procedure using Figure 2.2 is shown in Figure 2.3.

Furthermore, in [39, 40], it is given that in MTA a 0.381 system efficiency is achieved by splitting the right subgroup of the binary tree structure with bias probability of 0.582.



0	1	2	3	4	5	6	7	8	9	10	11	12
A B C D E F	A B C E	A B	A	B	C E		C	E	F D		D	F

Figure 2.3 An execution example of MTA

2.4 Optimum Dynamic Tree

The original porotype of optimum dynamic tree (ODT) algorithm is proposed by Captanakis [22] in order to enhance the average efficiency in basic tree algorithm. As shown in Figure 2.4, the first frame size of ODT is based on the number of tags in the systems. The collided slots in this frame are resolved using binary tree concept. In [33] it is proofed that the optimum average efficiency of ODT algorithm is around 0.429 under infinite tag population. Therefore, the ODT algorithm gives a higher efficiency compared to the basic tree algorithms such as binary tree and ternary tree algorithms. To derive the optimal efficiency in ODT, the relationship between the number of tags and the first frame size is mathematically derived and given in Section 3.5.

Figure 2.4 shows the execution processes of ODT algorithm for six competing tags (A, B, C, D, E, F) with initial frame size of six.

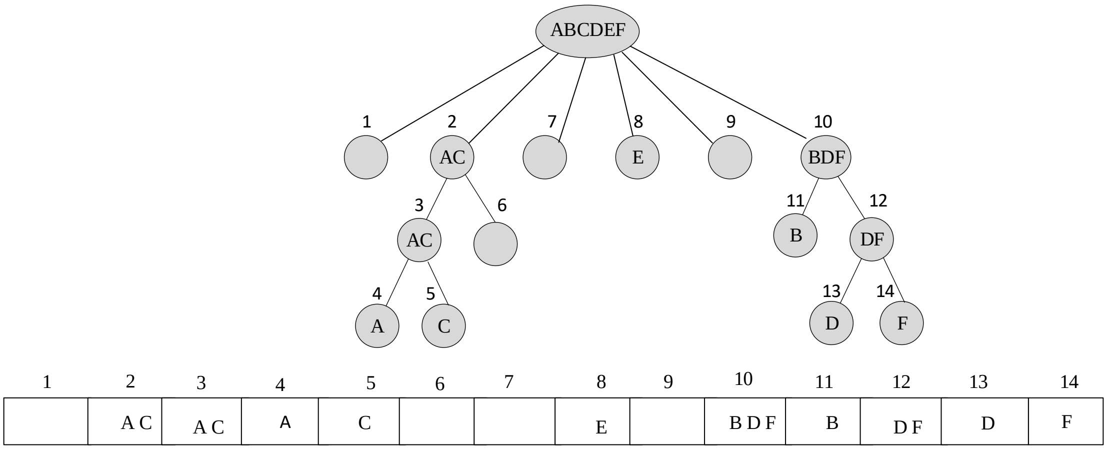


Figure 2.4 An execution example of ODT algorithm

2.4.1 Dynamic Binary Tree Slotted ALOHA

Dynamic Binary Tree Slotted ALOHA (dynamic BTSA) was introduced in [33] with the system efficiency around 0.40 where it adjusts the frame length similar to Q algorithm mentioned in Section 2.1. In this algorithm, firstly, it adjusts the frame size dynamically and inquires the feedback of the first slot in the frame. If the first slot is an idle, the Q value is decremented by one and for collision it is incremented by one. Therefore, it corresponds to read slot which follows the slotted ALOHA mechanism. The frame adjustment phase is terminated when a success happens and then it follows the ODT algorithm mentioned in Section 2.4 to identify the remaining tags. The process of dynamic BTSA is shown in Figure 2.5.

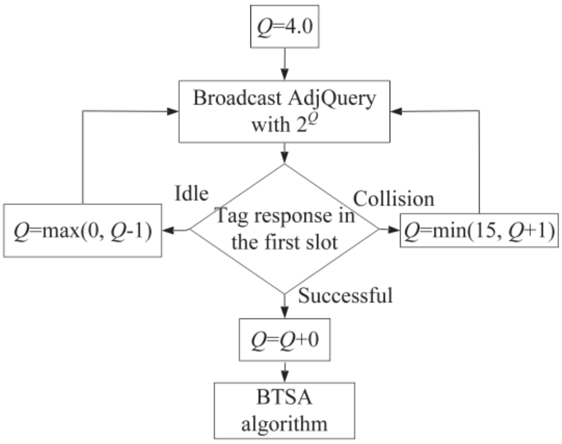


Figure 2.5 An execution example of dynamic BTSA algorithm [33]

2.5. Adaptive Tree

The adaptive tree (AT) algorithm, decide the next sub group size of current collided slot upon the number of collided tags of the current collided slot. Therefore, the adaptive tree performs well with some accurate tag estimation techniques and can achieve the optimum efficiency around 0.434 with an accurate tag estimation [25]. In [26], Tree Slotted ALOHA (TSA) algorithm is introduced by adapting the AT concept along with the tag estimation method introduced by Vogt in [12]. In TSA, the initial frame size is decided upon on the amount of tags in the systems and the collided slots in this initial frame size is resolve using AT algorithm and achieved the stable of system efficiency of around 0.38. The execution procedure of TSA for initially collided six tags (A, B, C, D, E, F). is given in Figure 2.6

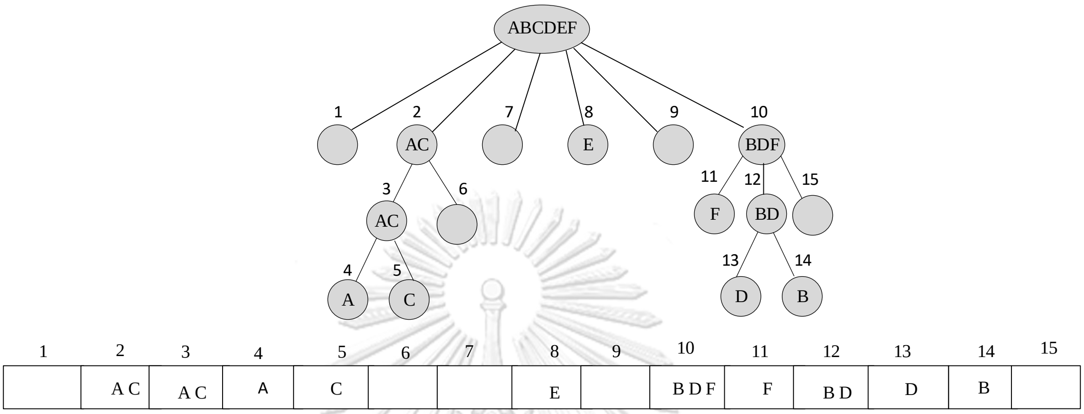


Figure 2.6 An execution example of dynamic TSA algorithm

2.6. Binary Splitting Tree Slotted ALOHA

In [32], binary splitting tree slotted ALOHA (BSTSA) is introduced which uses both BT and TSA concepts to achieve 0.415 system efficiency. Initially, the protocol splits tags into two subgroups by following BT until the leftmost leaf node of the tree contains no tags or a single tag. Then, all the right nodes are executed using TSA. This method enhances the TSA performance due to the initial elegant binary tree splitting tag estimation method. The process of BSTSA is given in Figure 2.7

2.6.1. Splitting Binary Tree Slotted ALOHA

BSTSA performance was significantly enhanced by using the splitting binary tree slotted ALOHA (splitting BTSA) as described in [33]. As shown in Figure 2.8, this follows the same binary tree splitting tag estimation method used in BSTSA with different tag identification approach. In this protocol, the authors have noticed that each node at a given level of the tree contains approximately half of the tags at its parent node. Therefore, when the splitting step is finished, the right nodes of the tree structure execute the ODT algorithm with initial frame size equal to the number of collided tags in the left-hand side. Using this approach 0.425 possible system efficiency is achieved for any number of tags.

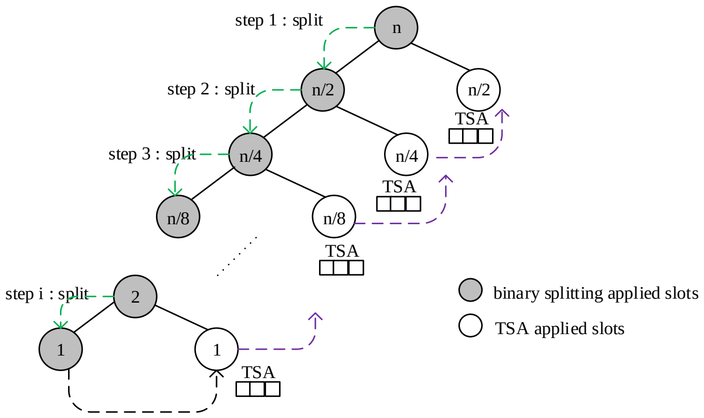


Figure 2.7 An execution example of dynamic BSTSA [32]

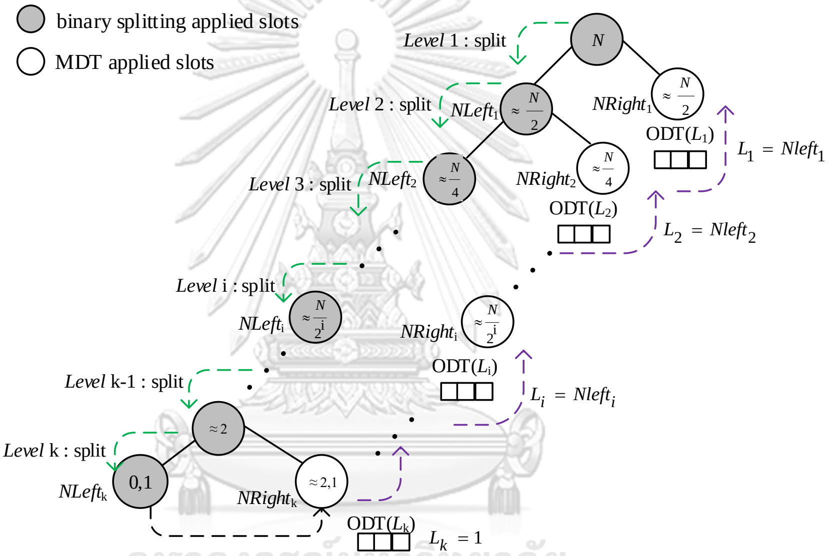


Figure 2.8 An execution example of dynamic splitting BTSA [33]

2.7. Tag Estimation Techniques

In general, an RFID reader broadcasts the next frame size information and each RFID tags in its interrogation zone select one slot among the given frame size. At the end of each slot, the state of the slot information is sent back to the reader with a ternary feedback of idle, success, and collision. A collision slot (C) indicates that many tags have chosen the slot, while success (S) and idle (I) slots specify one tag or none has selected the slot respectively. Most tag estimation methods consider the aforementioned ternary feedback to estimate the number of tags at the beginning of the former frame. In next sections several well know estimation methods are discussed in detail.

2.7.1. Schoute's tag estimation method

A simple, fast and accurate tag estimation method is introduced in [11] to decide the next frame size to use in DFSA algorithm based on the previous frame S and C with the backlog estimator of $2.39\times C$. The estimated number of tags $\hat{n}$ in the previous frame is given as,

$$\hat{n} = S + 2.39C \tag{2.1}$$

Based on Schoute's method, an accurate tag estimation can be observed when the number of tags in each slot follows a Poisson distribution with an integer mean.

2.7.2. Vogt's tag estimation method

A tag estimation model, based on minimum mean square error is given in [12] by considering the binomial distribution in slot occupancy. The binomial probability that i out of N tags transmit their ID for a given slot in a frame with L slots is given by,

$$B_{N,1/L}(i) = \binom{N}{i}\left(\frac{1}{L}\right)^{i}\left(1-\frac{1}{L}\right)^{N\text{-}i}, \tag{2.2}$$

where ${}^{1}\!/_{L}$ is the access probability for each tag to access the frame with L slot,

The expected number of idle a_e, successful a_s, and collision a_c slots in a frame with L slot can be derived using (2.2) as given in (2.3), (2.4) and (2.5) respectively.

$$a_e = LB_{N,1/L}(0) \tag{2.3}$$

$$a_s = LB_{N,1/L}(1) \tag{2.4}$$

$$a_c = L\left(1-a_e-a_s\right) \tag{2.5}$$

In this estimation method, using (2.6) the minimum mean square error or minimum distance between the vector of the expected values and the vector of the read results is calculated recursively until it reaches the minimum value. The initial estimation of $S+2C$ is considered by considering at least the two tags must collided in a slot to experience a collision.

$$\bar{n} = \min_{N}\left|\begin{pmatrix} a_e \\ a_s \\ a_c \end{pmatrix} - \begin{pmatrix} E \\ S \\ C \end{pmatrix}\right| \tag{2.6}$$

2.7.3. Chen’s tag estimation method

In [15], the authors derived a posterior probabilistic model by considering binomial distribution of idle (p_e), success (p_s) and collision (p_c) slots which occur in the previous frame as follows:

$$p_e = \left(1-\frac{1}{L}\right)^N \tag{2.7}$$

$$p_s = \left(\frac{N}{L}\right)\left(1-\frac{1}{L}\right)^{N-1} \tag{2.8}$$

$$p_c = 1 - \left(1-\frac{1}{L}\right)^N - \left(\frac{N}{L}\right)\left(1-\frac{1}{L}\right)^{N-1} \tag{2.9}$$

The authors assumed that the three outcomes of one read cycle: empty, successful and collision are independent and derived a posterior probability equation with multinomial distribution as shown in (2.10).

$$P(E,S,C) = \frac{L!}{E!S!C!} p_e^E p_s^S p_c^C \tag{2.10}$$

By maximizing the posterior probability expressed in (2.10), the estimated number of tags are approximated as follows:

$$\bar{n} = \arg\max_N P(E,S,C \mid N) \tag{2.11}$$

However, the three outcomes of success, collision and idle are not independent to each other. In Section 2.1.4, a better a posterior probabilistic model is discussed.

2.7.4. Vahedi’s tag estimation method

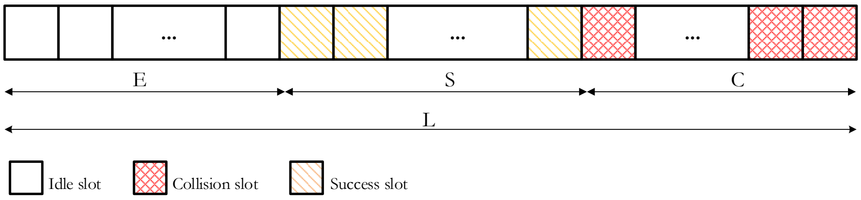


Figure 2.9 The Frame structure considered in the analytical model of Vahedi [15].

In [15], the authors reflect better approach of a posterior probability by assuming that tags in the frame are multinomial distributed and are mutually dependent for different slot types. As illustrated in Figure 2.9, a frame structure of E empty slot in the first part of the structure by following S success slots and C collision slots in the last section of the structure is being considered to derive the correct a posterior probability close form formula shown in Eq. (2.12).

$$P(E,S,C)=\left(\frac{L!}{E!S!C!}\right)P_1(E)P_2(S\mid E)P_3(C\mid E,S) \qquad (2.12)$$

The probability of observing empty slots in the first part of the frame can be expressed as

$$P_1(E)=\left(1-\frac{E}{L}\right)^{N} \qquad (2.13)$$

Next, the probability of observing successful slots in the second part of the frame when empty slots in the previous step is expressed as in (2.14), where the detail of its derivation is given in [15].

$$\begin{aligned}P_2(S\mid E)&=\binom{N}{S}\left(\frac{S}{L-E}\right)^{S}\left(1-\frac{S}{L-E}\right)^{(N\text{-}S)}\frac{S!}{S^{S}}\\&=\binom{N}{S}\left(\frac{(L-E-S)^{(N\text{-}S)}}{(L-E)^{N}}\right)S!\end{aligned} \qquad (2.14)$$

Finally, the probability of observing collision slots in the last part of the frame when empty slots and successful in the previous step is expressed as in (2.15), where the details of its derivation is given in [15].

$$\begin{aligned}P_3(C\mid E,S)&=\sum_{k=0}^{C}\sum_{v=0}^{C-k}(-1)^{(k+v)}\binom{C}{k}\binom{C-k}{v}\\&\times\frac{(N-S)!}{(N-S-k)!}\frac{(C-k-v)^{(N-S-k)}}{C^{(N-S)}}\end{aligned} \qquad (2.15)$$

By maximizing a posterior probability expressed in (2.12) the estimated tag count can be achieved.

2.7.5. Šoli´c’s tag estimation method

A system model called an Improved Linearized Combinatorial Model (ILCM) is introduced in [17] which uses interpolation method following the calculation of the combinatorial model. The expected number of tags $\bar{N}$ is derived by maximizing (2.16)

$$\bar{N}=\arg\max_{N}P(E,S,C\mid N) \qquad (2.16)$$

The introduced interpolation concept in ILCM reduces the complexity of the algorithm with the following estimated method:

$$P(E,S,C\mid N)=\frac{L!}{E!S!C!}\frac{N_S(n,S)N_C(n,S,C)}{L^{n}} \qquad (2.17)$$

where $N_S(n,S)$ stands for the number of ways to distribute remaining $n-S$ tags in C collision slots. To provide $N_C(n,S,C)$, the exponential generating function $G(x)=(e^{x}\text{ - }(1+x))^{C}$ and its Maclaurin expansion $\mathrm{G(x)} = (\mathrm{x}^2/2! + \mathrm{x}^3/3! + \mathrm{x}^4/4! + \mathrm{x}^5/5! + \cdots)^{\mathrm{C}}$ is used. Then, $N_C(n,S,C)$ is given by the coefficient of the $x^{(n\text{-}S)}/(n-s)!$ in the expansion of $G(x)$.

2.7.6. Annur's Bayesian estimation-based tag estimation method

In [41], authors proposed a novel efficient tag estimation concept to figure out the posterior probability distribution of number of tags in the system based on Bayesian estimation by following slotted ALOHA. This technique accumulates the information in each slot and produces a very consistent tag estimation for any number of tags. In this work, they assumed that the reader gets ternary feedback of idle (I), success (S) and collision (C) after reading a slot and the posterior probability distribution can be given as

$$\mathrm{P}(N \mid feedback) = \frac{\mathrm{P}(feedback \mid N)\mathrm{P}(N)}{\mathrm{P}(feedback)}, \tag{2.18}$$

where, $P(N)$ is the prior distribution of number of tags. The authors considered the initial prior distribution of tags as a uniform distribution with the initial slot access probability of $1/16$, where the number of tags distribute from 1 to $N_{\max}$, $N_{\max}$ is the length of the distribution. $\mathrm{P}(feedback)$ is the normalization constant.

The likelihood distribution $\mathrm{P}(feedback \mid N)$ for feedback of success S, collision C and idle E can be written as.

$$P(S \mid N) = B([1 \cdots N_{\max}], p, 1) \tag{2.19}$$

$$P(E \mid N) = B([1 \cdots N_{\max}], p, 0) \tag{2.20}$$

$$\begin{aligned} P(C \mid N) &= 1 - \left(P(S \mid N) + P(I \mid N)\right) \\ &= 1 - \left(B([1 \cdots N_{\max}], p, 1) + B([1 \cdots N_{\max}], p, 0)\right) \end{aligned} \tag{2.21}$$

where, $B(n, p, i) = \binom{n}{i} p^i (1-p)^{n\text{-}i}$ with access probability of p.

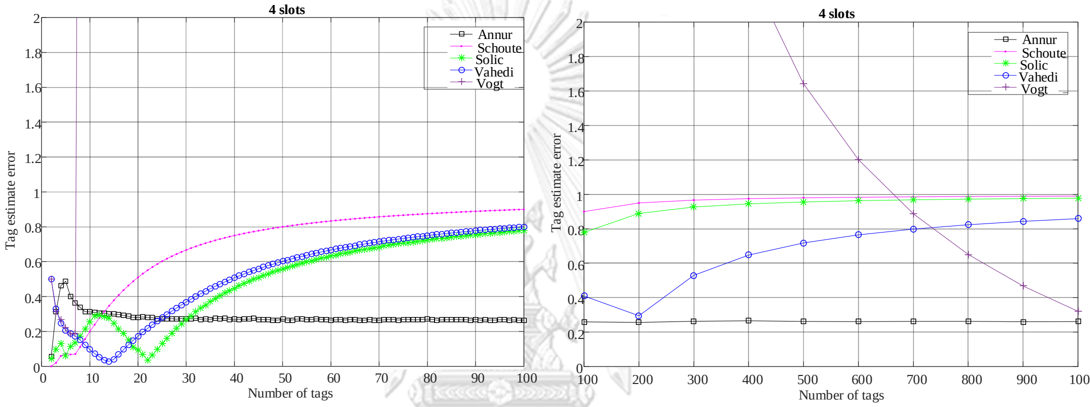
In success, the distribution is shifted to left hand side to indicate an identification of a tag.

The estimated number of tags $\bar{N}$ is derived from the mean of posterior probability distribution of number of tags $\mathrm{P}(N \mid feedback)$. Then the next access probability becomes $p = 1/\bar{N}$.

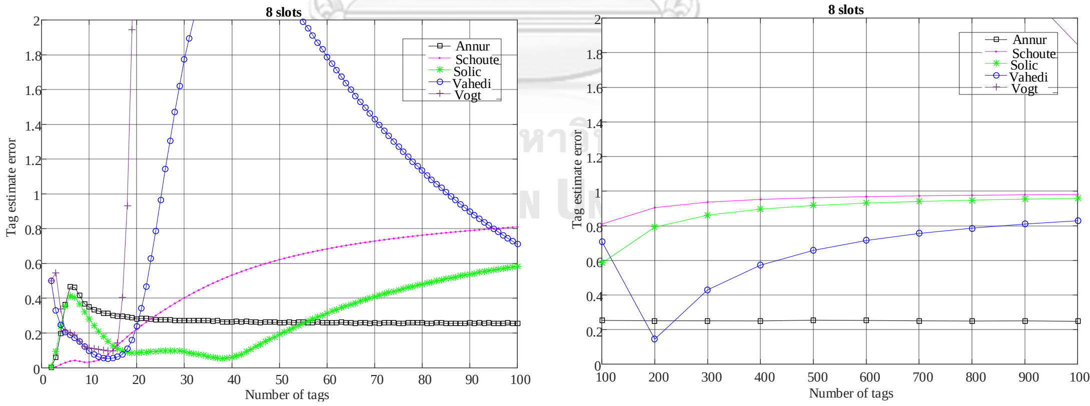
2.8 Simulation parameters and results with respect to tag estimation error in an RFID system

In this section, the performances of the stated tag estimation methods in Section 2.7 are compared with respect to the tag estimate error which is given in (2.22). Extensive simulations have been carried out with the frame lengths of 4, 8, 16, 32, 64, 128 based on the Monte Carlo technique for 10 to1000 range of tags.

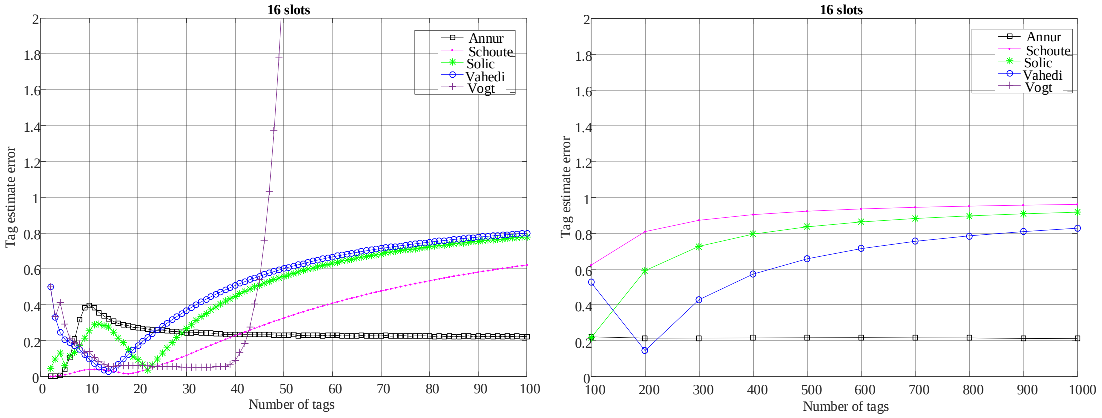
$$\text{tag estimate error} = \frac{\left|N - \bar{n}\right|}{N} \qquad \textbf{(2.22)}$$



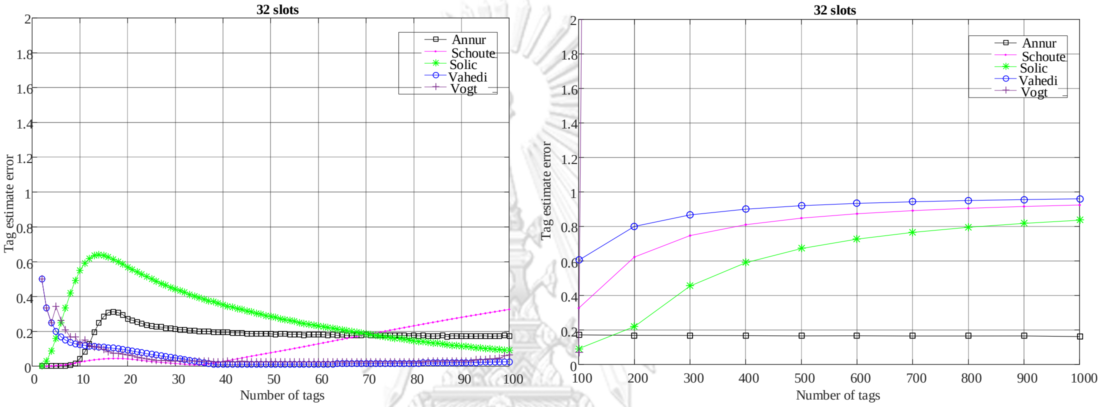
(a)



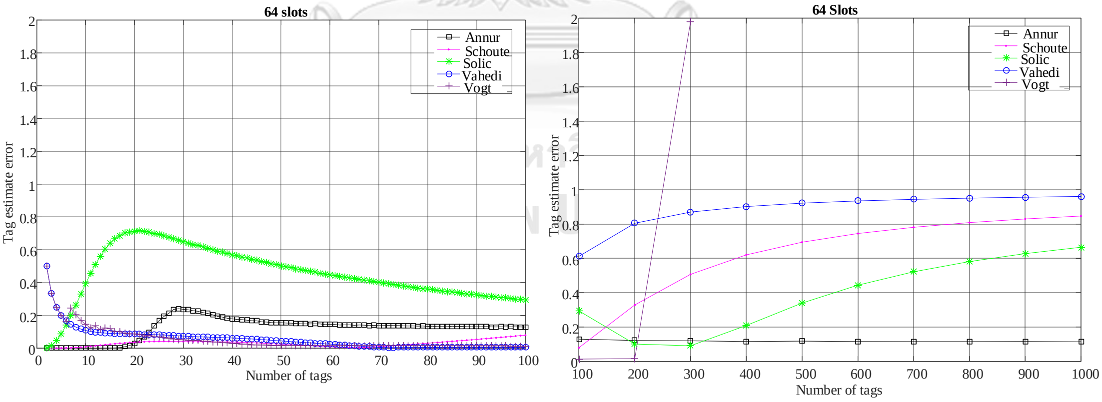
(b)


16 slots
Tag estimate error
Number of tags
Annur
Schoute
Solic
Vahedi
Vogt
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
10
20
30
40
50
60
70
80
90
100
16 slots
Tag estimate error
Number of tags
Annur
Schoute
Solic
Vahedi
Vogt
100
200
300
400
500
600
700
800
900
1000

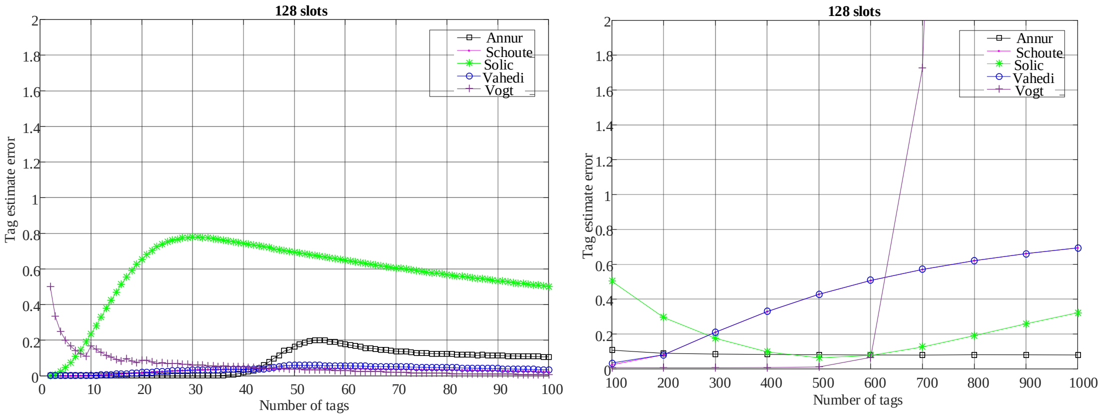
(c)


32 slots
Tag estimate error
Number of tags
Annur
Schoute
Solic
Vahedi
Vogt
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
10
20
30
40
50
60
70
80
90
100
32 slots
Tag estimate error
Number of tags
Annur
Schoute
Solic
Vahedi
Vogt
100
200
300
400
500
600
700
800
900
1000

(d)


64 slots
Tag estimate error
Number of tags
Annur
Schoute
Solic
Vahedi
Vogt
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
10
20
30
40
50
60
70
80
90
100
64 Slots
Tag estimate error
Number of tags
Annur
Schoute
Solic
Vahedi
Vogt
100
200
300
400
500
600
700
800
900
1000

(e)



(f)

Figure 2.10 Tag estimate error for distinct estimation methods

Discussion:

The results verify that Vogt's, Schoute's and Vahedi's methods give a lower tag estimate error when the frame size is equal or less than the given number of tags in the system. The Vog's minimum distance estimation poorly act upon the tag estimation process than the other stated methods, specially, when the first frame size is not bigger enough. In Schoute's backlog estimation and Vahedi's a posterior probability-based estimation method, the error in tag estimation increases when more number of tags exist in the system than the frame size. As illustrated in Figure 2.10 (f), the Šoli´c's tags estimation is suitable for larger number of tags such as more than twenty tags with larger frame sizes. The Annur's slot wise Bayesian estimation method gives a low and consistent error in tag estimation for any range of tags and for any frame sizes. The specialty behind Annur's estimation is that it accumulates the information in each slot. Therefore, by using the prior accumulate knowledge in each slot there is a higher probability to get a good estimation value. In Vogt's, Schoute's, Vahedi's and Šoli´c's methods, they utilized one frame information to estimate the available tags.

Chapter 3 Average number of slots required in collision resolution for various types of binary tree algorithms

In this chapter, the average number of slots needed in tag identification process is analyzed with respect to conventional Binary Tree, Binary Tree with bias probability, Optimum Dynamic tree (ODT), Modified Tree Algorithm (MTA) and proposed Modified Dynamic Tree (MDT) are mathematically analyzed.

3.1 System Model

In this analysis, a binary tree structure is considered and a node in the tree structure represents a slot uses in tags identification process. As depicted in Figure 4.1, at the k^{th} level of the tree structure, there are 2^k number of slots and the splitting probability of right and left branches are p and $1-p$ respectively. We consider the binary tree algorithm which allows ternary feedback of idle, success and collision returns at the end of each slots to the users. Based on this system model and the ternary feedbacks, the rest of this chapter will calculate the mean number of slots required in tag identification process in different types of binary tree-based algorithms.

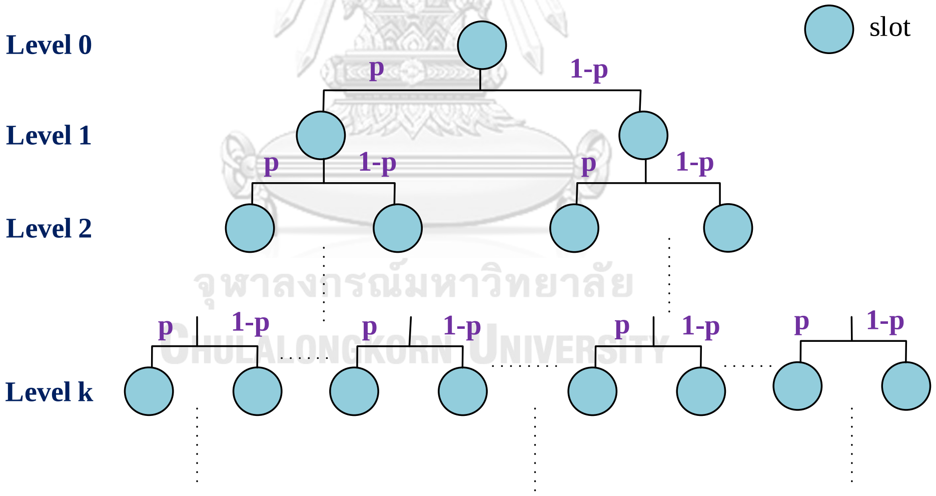


Figure 3.1 System model: binary tree structure

3.2 Binary Tree

Lemma 1**:** Let $C_1(N,k)$ denotes the average number of collision slot at the k^{th} level of binary tree structure with fair splitting probability of $p = 1/2^k$. $C_1(N,k)$ can be given as:

$$C_1(N,k) = \sum_{n=2}^{N} B(N,1/2^k,n) \qquad (3.1)$$

where, $B(n, p, i) = \binom{n}{i} p^{i}(1-p)^{n\text{-}i}$.

Proof: The number of collision slots happen when *n* out of *N* tags are colliding at the k^{th} level of binary tree structure is $\sum_{n=2}^{N} B(N, 1/2^{k}, n)$.

Also note that

$$\sum_{n=2}^{N} B(n, 1/2^{k}, n) = \sum_{n=2}^{N} 1 - B(N, 1/2^{k}, 0) - B(N, 1/2^{k}, 1) \qquad \textbf{(3.2)}$$

Theorem 1: Let $T_{BTA}(N)$ be the average number of timeslots required by the binary tree algorithm with a fair splitting to recognize *N* tags. Then, $T_{BTA}(N)$ can be expressed as:

$$T_{BTA}(N) = 2\left[1 + \sum_{k=1}^{\infty} C_{1}(N,k)\right] \qquad \textbf{(3.3)}$$

Proof: Since the tree structure will continue to grow with collisions i.e., every collided node will always produce two new nodes, by taking the summation of average number of collided slots in each level from Lemma 1 and multiply with the splitting factor of two, the Theorem 1 can be yielded.

Furthermore, using an approximated asymptotic expression in [23, 42], (3.3) can be simplified as:

$$T_{BTA} = \frac{2N}{\ln(2)} - 1 \qquad \textbf{(3.4)}$$

where $N > 2$

Hence, using (3.2), (3.3) can be given as:

$$T_{BTA}(N) = 2\left[1 + \sum_{k=1}^{\infty}\left[1 - \left(1 - \frac{1}{2^{k}}\right)^{N} - \frac{N}{2^{k}}\left(1 - \frac{1}{2^{k}}\right)^{N-1}\right]\right]. \qquad \textbf{(3.5)}$$

This is exactly the same expression as (3.40) in [23], although it is derived from different approach.

3.3 Binary tree with bias splitting probability

Lemma 2: Let $C_{2}(N,k)$ be the average number of collision slots in the k^{th} level of the binary tree structure starting with *N* tags, $p_{k,j}$ be the probability that a tag arrives at the *k*th level of the tree structure through the j^{th} path from all possible 2^{k} paths and p_{0} and $p_{1} = 1 - p_{0}$ are the probability that a tag selects the left and right branches respectively. Then, $C_{2}(N,k)$ is:

$$C_{2}(N,k) = \sum_{j=0}^{2^{k}-1} \sum_{n=2}^{N} B(N, p_{k,j}, \mathrm{n}) \qquad \textbf{(3.6)}$$

and $p_{k,j} = \prod_{i=0}^{k-1} p_{\delta_{i}}$ where $j = dec(\delta_{1}\delta_{2}\ldots\delta_{k})$ and δ_{i} is a binary value, which indicates the subgroup selected at the i^{th} level.

Proof: A collision is said to take place when two or more tags choose the same slot. Therefore, the average number of collision slots in the k^{th} level of the binary tree structure is the summation of the collision probability of all slots in the k^{th} level and can be given as in (3.6).

***Theorem 2*:** Let $T_{BTAbias}(N)$ be the average number of timeslots required by the binary tree protocol with a biased splitting to recognize *N* tags. Then, $T_{BTAbias}(N)$ can be expressed as:

$$T_{BTAbias}(N) = 2\left[1 + \sum_{k=1}^{\infty} C_2(N,k)\right] \qquad \textbf{(3.7)}$$

Proof: Since the tree structure will continue to grow with collisions i.e., every collided node will always produce two new nodes, by taking the summation of average number of collided slots in each level from Lemma 2 and multiply with the splitting factor of two, the Theorem 2 can be yielded. Also note that this is exactly the same expression as (3.31) in [23] , although it is derived from different approach. The (3.31) in [23] can be further simplified for binary tree algorithm with splitting probability of p and $1-p$ in left and right branches respectively as follows:

$$T_{BTAbias}(N) = 1 + \frac{\sum_{k=2}^{N} \binom{N}{k} (-1)^k \left[2(k-1) - k(1-p) + p^k - 1\right]}{1 - p^k - (1-p)^k}, \qquad \textbf{(3.8)}$$

where $L_0 = L_1 = 1$.

3.4 Modified Tree Algorithm

Lemma 3: The average number of definite colliding slots that can be predicted at level *k* is:

$$C_3(N,k) = \begin{cases} p_1^N & k = 1 \\ \sum_{\substack{i=1 \\ i:odd}}^{2^k-1} \sum_{n=2}^{N} \left[B(N, p_{k-1,(i+1)/2}, n) \times p_1^n \right] & k > 1 \end{cases} \qquad \textbf{(3.9)}$$

Proof: For the binary tree structure, a collision will definitely occur on the right branch, if no tag selects the left branch. The probability of such a definite collision on a slot i in the k^{th} level of the binary tree structure is the probability that there are more than two tags arriving at the $k-1^{th}$ level through the $(i+1)/2^{th}$ path and all of them select the right branch with probability of p_1. Note for the first level that all *N* tags are collided at the root slot. Therefore, $C_2(N,k)$ gives the average number of definite colliding slots that can be predicted at level *k.*

Theorem 3*:* Let $T_{MTA}(N)$ be the average number of timeslots consumed by MTA to identify all *N* tags. Then, $T_{MTA}(N)$ can be given by:

$$T_{MTA}(N) = T_{BTAbias}(N) - \sum_{k=1}^{\infty} C_3(N,k) \qquad \textbf{(3.10)}$$

Proof: In MTA, all inevitable collision slots that are be predicted are skipped and tags involve in the collision proceed to the next tree level by splitting them into two subgroups. Thus, by deducting (3.7) by average number of definite collision slots, (3.10) can be obtained.

Further, $T_{MTA}(N)$ can be derived by extending (3.13) in [23] with p as the splitting probability of the left branch in a binary tree structure as:

$$T_{MTA}(N) = 1-(1-p)^N + \sum_{i=0}^{N} \binom{N}{i} p^i (1-p)^i \left[p^{N-2i} + (1-p)^{N-2i} \right] T_{MTA}(i)\,, \qquad \textbf{(3.11)}$$

and the results are apparently similar to the one in (3.10).

3.5 Optimum Dynamic Tree

Lemma 4: Let $T_{ODT}\left(N,L\right)$ represents the average number of slots needed by optimum dynamic tree (ODT) to resolve the collided slots. Initially, when N tags distributed in a frame with L slots, then, $T_{ODT}\left(N,L\right)$ can be given by:

$$T_{ODT}\left(N,L\right) = L \times \left[\begin{array}{r} B(N,1/L,0) + B(N,1/L,1) + 5 \times B(N,1/L,2) \\ + \sum_{n=3}^{\infty} B(N,1/L,\mathrm{n}) \times T_{BTA}(n) \end{array} \right] \qquad \textbf{(3.12)}$$

Proof: In ODT tag identification, the tags are initially separated into L sets and the collided slots among these sets are resolved based on binary tree protocol. $B(N,1/L,n)$ representing the binomial probability of n collided tags out of N tags distribute among L initial slots. The $B(N,L^{-1},0)$, $B(N,L^{-1},1)$ and $B(N,L^{-1},2)$ represent the binomial probability of no tag, one tag and two tags arbitrary select L slots respectively. As given in [24], the average number of slots needed to resolve two tag collision in binary tree is 5.

Lemma 5: The relationship between the initial frame size L and the number of tags N in ODT can be obtained as $L = 0.87N$.

Proof:

By substituting (3.4) in (3.12) the $T_{ODT}\left(N,L\right)$ can be simplified as given in (3.13).

$$T_{ODT}\left(N,L\right) = \frac{2N}{\ln(2)} - L + 2\left(1-\frac{1}{L}\right)^{N-2} \left[\frac{N(N-1)}{2L}\left(3-\frac{2}{\ln(2)}\right) + L\left(1-\frac{1}{L}\right)^2 + N\left(1-\frac{1}{\ln(2)}\right)\left(1-\frac{1}{L}\right) \right] \qquad \textbf{(3.13)}$$

To discover the frame size in ODT with minimum slot consumption for any N, the function $T_{ODT}\left(N,L\right)$ given in (3.13) differentiates with respect to L. Let the resulting expression derived after the differentiation equal to zero and further analyze it with $L=\alpha N$, where α is a constant which gives the ratio of L and N. Then, for large number of tags, (3.13) can be simplified to

$$1=e^{-1/\alpha}\left[\left(3-\frac{2}{\ln(2)}\right)\left(\frac{1}{\alpha}-1\right)\frac{1}{\alpha^2}+2\left(\frac{1}{\alpha}+1\right)+\left(1-\frac{1}{\ln(2)}\right)\frac{2}{\alpha^2}\right] \qquad (3.14)$$

which can be solved with $\alpha \approx 0.871$.

Figure 3.2 illustrates the frame sizes which offer the minimum collision resolution interval length for different number of tags. CRI length is the average number of slots require in tag identification. By taking the ratio between the frame size *L*, tags *N* and constant value α is calculated and is given in Table 3.1 for different number of tags. Therefore, the relationship between the frame size *L* and number of tags *N* can be given as, $L=0.87N$.

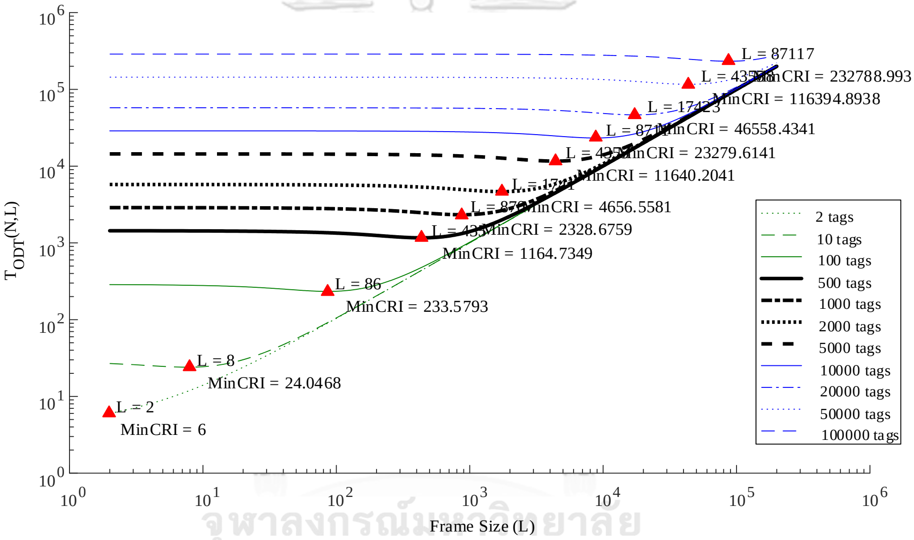


Figure 3.2 Frame sizes to obtain minimum CRI length for any N.

Table 3.1 α values for different N

No.Of Tags (N)	10	50	100	500	1000	5000	10000	50000	100000
α	0.800	0.820	0.860	0.870	0.870	0.871	0.871	0.871	0.871

Theorem 4: The optimum average efficiency of ODT is around 0.429 under infinite tag population.

Proof: Substituting $L=0.87N$ in Eq. (3.12),

$$T_{ODT}\left(N,L\right)= 2.3278N+0.04172$$

Definition 1: Let $T\left(N\right)$ represents the mean number of timeslots needed in anti-collision algorithm to identify *N* tags. Then, the system efficiency η can be given by

$$\eta = \frac{N}{T\left(N\right)} \tag{3.15}$$

Using (3.15), η_{ODT} which is the efficiency of ODT for large number of tags is

$$\eta_{ODT} \approx \frac{N}{2.3278N} \approx 0.42959 \tag{3.16}$$

Therefore, Theorem 4 can be yielded.

Note that the efficiency of ODT has derived in [33] with the assumption of frame size is equal to the number of tags. In this thesis, we derive optimum average efficiency of 0.429 based on the relationship of the frame size and the number of tags in ODT given in Lemma 5.

3.6 Modified Dynamic Tree

The first level of the tree is set according to the number of tags in the system. Then, the MTA use to resolve the conflict among the tags and identify them. This procedure is referred to as MDT.

***Theorem 5**:* Let $T_{MDT}\left(N,L\right)$ denotes the average number of timeslots used in MDT to resolve *N* tags with the initial frame size of L slots. Then, $T_{MDT}\left(N,L\right)$ can be given by:

$$T_{MDT}\left(N,L\right) = L \times \left[B(N,1/L,0) + B(N,1/L,1) + \sum_{n=2}^{N} B(N,1/L,\mathrm{n}) \times T_{MTA}(n) \right] \tag{3.17}$$

Proof: Tags are initially separated into L slots at random. Thus, the probability that each slot contains 0,1 and more than one tag are $B(N,L^{-1},0)$, $B(N,L^{-1},1)$ and $B(N,L^{-1},n)$ respectively. As only colliding slots are resolved by MTA, given in (3.11). The minimum $T_{MDT}\left(N,L\right)$ can be achieved when $L = 0.79N$ the frame size *L* and the number of tags

3.7 Simulation parameters and results

3.7.1 Performance analysis of optimum system efficiency for variations of binary tree algorithm

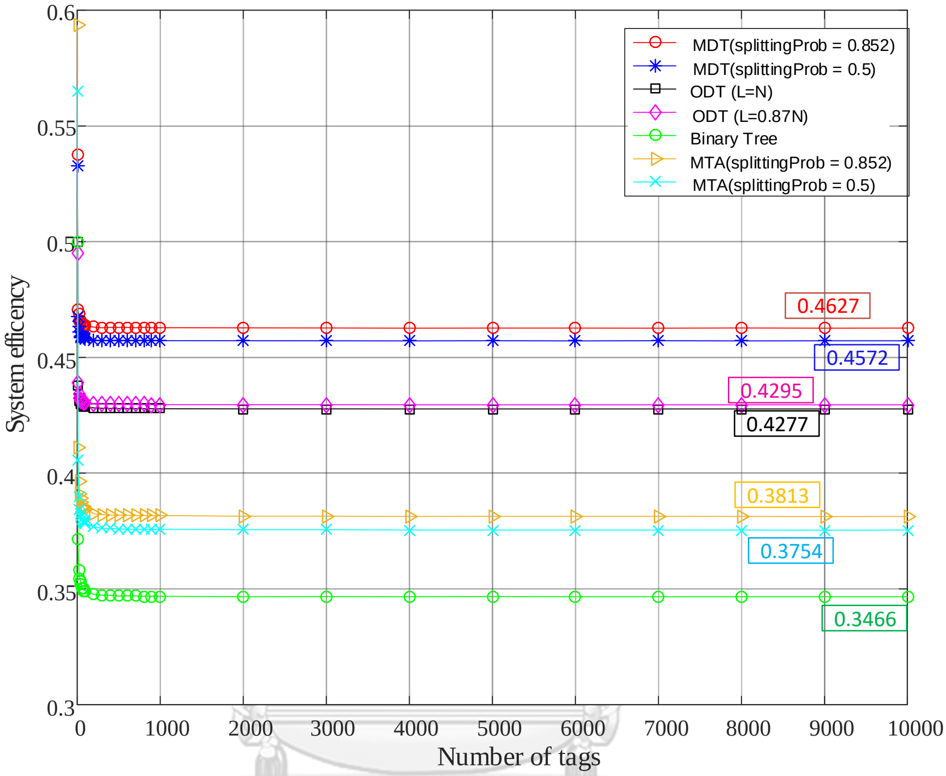


Figure 3.3 System efficiency of variations of the binary tree algorithm for large number of tags

In Figure 3.3, the system efficiency of ODT, MDT, MTA and conventional binary tree algorithm is plotted. For the comparison purpose, we have used the optimal splitting probability of 0.582 and fair splitting factor of 0.5 in MDT and MTA algorithms. Compared to MDT with fair splitting, MDT with optimal splitting factor of 0.582 offers a significant improvement with a 0.4627 consistent efficiency when consider the initial frame size of ODT process equal to the tags.

Compared to MTA and conventional binary tree, ODT shows a higher system efficiency. As given in Figure 3.4, based on the ODT process discussed in Section 3.5, when the tags are initially distributed among *L* slots, we can verify that the most frequently, two tags are collided in a slot. Therefore, when binary tree algorithm is used to resolve these collided slots, the efficiency gets increased. Further, ODT gives a higher efficiency

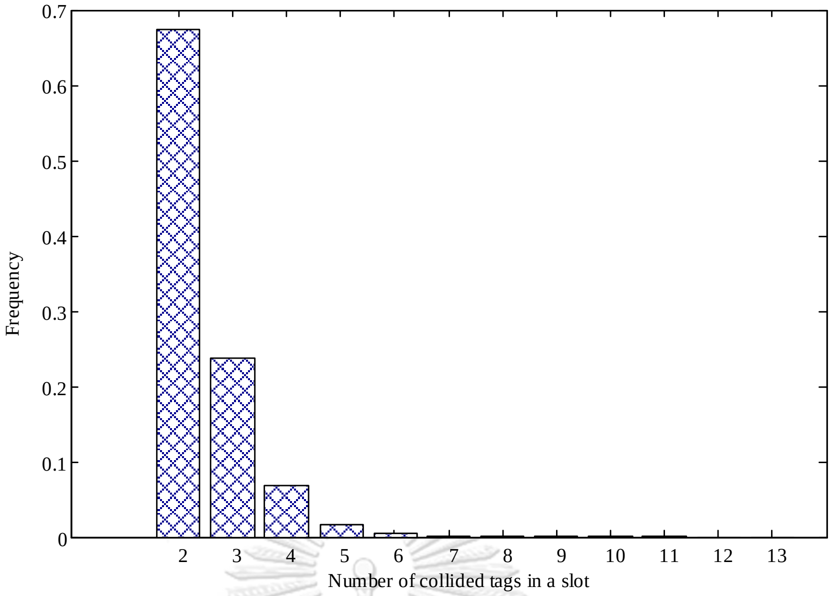


Figure 3.4 Frequency of tags colliding at the initial frame of ODT

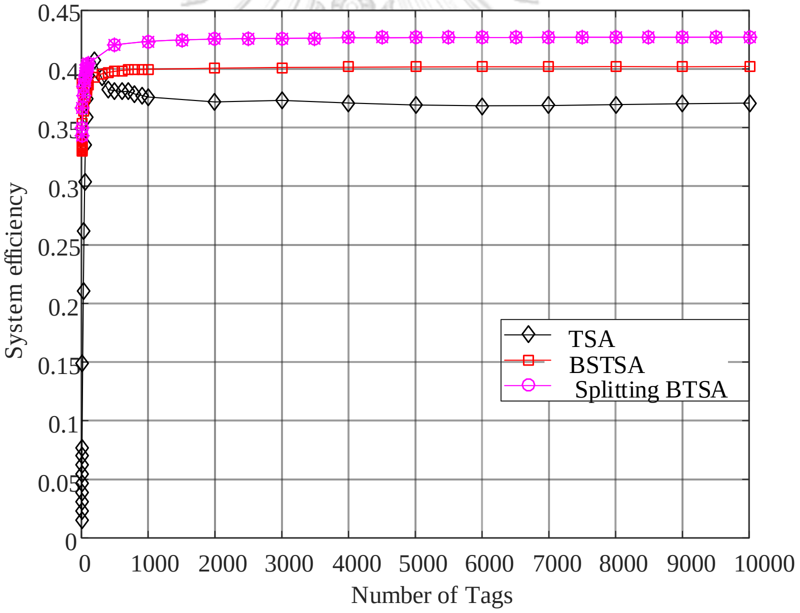


Figure 3.5 System efficiency of best performed tree-based algorithm up to date

when considered an initial frame size is equal to the 0.87 time of number of tags as given in Theorem 4. Therefore, it is proofed that using MTA and ODT algorithms, we can enhance the efficiency of conventional binary tree. In Figure 3.5, the system efficiency of well performed tree-based algorithms are given and it is verified that 42% is the best efficiency that have achieved up to date.

The analytical model results and simulation results of BTA and ODT algorithms are given in the following tables.

Table 3.2 Comparison of analysis and simulation results for BTA and ODT algorithms

BTA			ODT		
Tags	**Simulation**	**Analysis**	**Tags**	**Simulation**	**Analysis**
2	3.99	4	**2**	4	4
3	6.671	6.666	**3**	6.4074	6.4062
4	9.5656	9.528	**4**	8.7738	8.7721
5	12.3926	12.419	**5**	11.1273	11.1252
6	15.3028	15.313	**6**	13.4750	13.4726
7	18.2296	18.2	**7**	15.8197	15.8169
8	21.1076	21.085	**8**	18.1626	18.1594
9	23.9476	23.968	**9**	20.5043	20.5008
10	26.8164	26.853	**10**	22.8452	22.8413
11	29.7224	29.738	**11**	25.1856	25.1813
12	32.6082	32.623	**12**	27.5255	27.5209
13	35.526	35.509	**13**	29.8651	29.8601
14	38.3705	38.395	**14**	32.2045	32.1991
15	41.277	41.281	**15**	34.5436	34.5379
16	44.1452	44.166	**16**	36.8826	36.8766
17	47.052	47.051	**17**	39.2215	39.2151
18	49.9347	49.937	**18**	41.5603	41.5535
19	52.8598	52.822	**19**	43.8989	43.8919
20	55.7334	55.707	**20**	46.2375	46.2301

Chapter 4 Bayesian Estimation based Modified Dynamic Tree Algorithm

In this chapter, we discuss the proposed algorithm called Bayesian estimation based modified dynamic tree (BE-MDT). Throughout the chapter, a detailed clarification of whole the algorithm is presented with flow chart and other necessity diagrams.

4.1 Introduction to BE-MDT

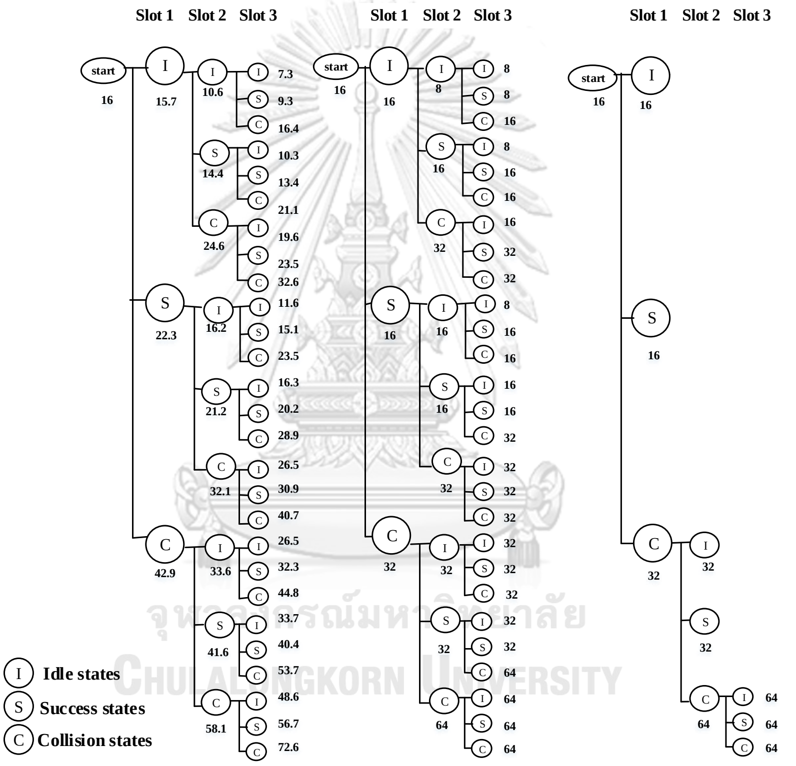
The proposed BE-MDT algorithm comprises two phases of operations. In the first phase, the algorithm aims to obtain the number of competing tags in the system. To acquire the estimated number as fast as possible, an efficient slotted based Bayesian tag estimation is applied. In the second phase, we use an efficient technique to dynamically set the frame size of first level of the tree, based on the estimated tag count from phase one. Then, we use MTA to resolve the conflict among the tags and identify them. This procedure in the second phase is referred as modified dynamic tree (MDT).

4.2 Process of BE-MDT

4.2.1 Slotted ALOHA based Bayesian tag estimation: Phase I

For tag estimation, most well-known methods such as [11-15, 17] rely on the feedback information in the previous frame, i.e. the number of idle, success and collision slots to estimate the collided tags. As the feedback information is from one whole frame, with efficient estimation techniques [15, 17], involving complicated calculations, the estimate can be quite accurate. However, when the current frame size is rather different from the actual number of tags, it is very important to change the frame size quickly for both over estimation or under estimation. If the decision to change of frame size takes several frames, a serious loss of efficiency may occur. That is why, several frame-based algorithms such as [6] adopt the early frame braking policy in which a frame can be terminated pre-mutually, if the discrepancy between the number of tags and frame size are detected so that a new and more appropriate frame size can be initiated promptly. However, frame-based estimation can exhibit slow response, even the early frame breaking technique in which a frame can be terminated before the end of the frame is applied. We propose to perform the estimation at every slot to achieve fast tag estimation. In order to obtain good estimates, it is important to use a technique which can extract feedback information from each slot and accumulate them over a series of slots. Bayesian estimation method is one of powerful mathematical tools that minimizes posterior expected value of a loss function while accumulating the prior information.

In this thesis, we follow the Bayesian estimation principles which were introduced in [41] and [43], for the purpose of tag estimation and not for collision resolution. The achievable system efficiency is reported to be 36%, which approaches the maximum limit of slotted ALOHA protocols. At each slot, the algorithm performs tag estimation, which involves very expensive computation, while resolving collision and identifying tags at the same time. Based on their method, a decision tree with dynamically updated frame sizes is shown in Figure 4.1(a). This is displayed only for the first three slots. The decision tree grows exponentially with the number of slots, making it practically impossible to store all the pre-assigned frame sizes in a lookup table. Therefore, the reader must compute a new frame size at the end of every slot in real time. This is computationally too demanding for the reader. Figure 4.1(b) represents the modified decision tree from Figure 4.1(a) with an additional constraint that the frame sizes are restricted to 2^k where $\{k \in 1, 2, 3, \ldots\}$. Such a constraint not only helps minimizing the signaling from the reader to tags, but also substantially simplify the tag estimation process.



(a) Conventional Bayesian (b) Bayesian with discrete frame sizes (c) Proposed approach

Figure 4.1 Comparison of decision tree diagrams

In our work, we aim to acquire an accurate tag estimate using the smallest number of slots, thus the decision tree should be terminated as soon as the tag estimate is deemed accurate enough. An appropriate termination criterion is based on the minimum ratio of the standard deviation over mean of the prior distribution of the number of tags. As this ratio gets smaller, the estimation becomes more accurate. Based on our extensive investigation over a broad range of tags from 2 to 65,535, we found that every colliding slot always exhibits a large standard deviation over mean which signifies that the estimation procedure should continue. In contrary, the idle and success slots give relatively small standard deviation over mean in prior distribution, which indicates that the estimation process can be terminated. Based on these investigations, the decision tree of the proposed method is shown in Figure .4.1(c). As we can see, the tree grows linearly with the number of slots, *i.e.* the number of leaves are two times the number of slots plus one. Consequently, it is possible using mathematical methods and simulations to store the precomputed frame sizes in a lookup table as given in Table 4.1. Thus, in our method, we get the benefit from the Bayesian estimation while the Bayesian computational complexity is no longer a problem.

Table 4.1 Lookup table for precomputed Bayesian estimation.

Slot	1	2	3-6	7-13	14-28	29-59	60-120	121-244	245-491	>491
Q	4	5	6	7	8	9	10	11	12	13

As given in [41], the appropriate access probability for the next slot is derived by maximizing the average probability of success as follows:

$$p_{opt} = \frac{\sum N(1-p_{opt})^N \, p(N)}{\sum N^2(1-p_{opt})^N \, p(N)}, \qquad (4.1)$$

where $p(N)$ is the prior probability distribution. Alternatively, a good estimate for (4.1) is:

$$p_{opt} = \frac{1}{\sum Np(N)} \qquad (4.2)$$

In our work, we utilize the Bayesian estimation concept solely for tag estimation purpose (only in Phase I), not at all for tag identification. Our main concern is to find out when the Bayesian estimation process in Phase I should be terminated. It is clear that the complexity is the main issue here as the number of states increases exponentially (see Figure 4.1(a)). To ensure that the proposed algorithm can be implemented, we considered an intelligent criterion that when the standard deviation over mean of the priori distribution is narrow enough, the estimation process can be terminated. As it appears that after an idle or a success slot, the standard deviation over mean is rather narrow, while a collision slot always results in large standard deviation over mean. Based on this finding, we construct Table 4.1 accordingly. Note that the Q value in Table 4.1 is defined as $\lceil \log_2(1/p_{opt}) \rceil$.

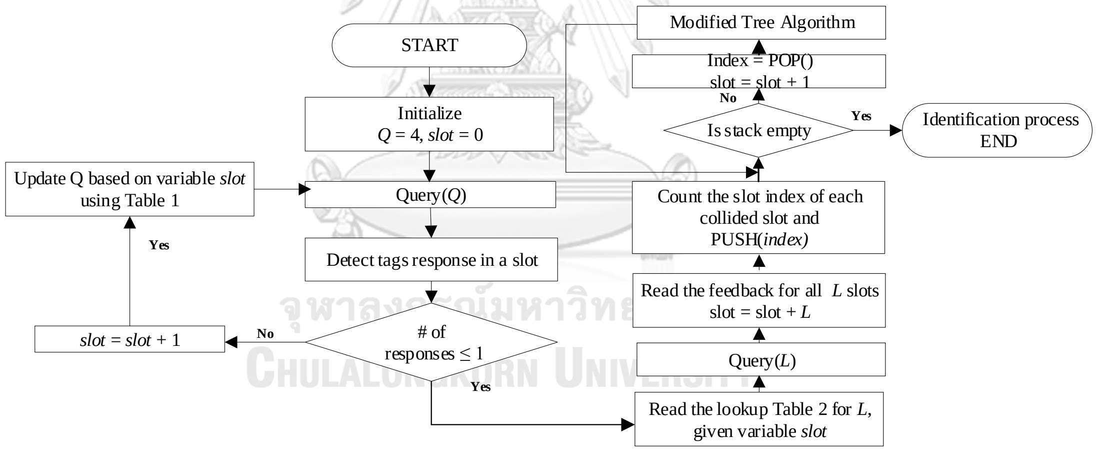
4.2.2 BE-MDT: Phase II

In Phase II, the remaining tags from the Phase I are resolved based on MDT algorithm. MDT is the combination of ODT and MTA algorithms and a detailed description of MTA and ODT algorithms is given in Section 2.4 and 2.5 respectively. The mathematical analysis of MDT algorithm is given in Section 3.6.

In Phase II, an initial frame size L is chosen based on the variable $slot$ using the lookup table given in Table 4.2. The reader sends the query command and waits for L slots. Only those collision slots are further resolved through MTA. Note that Table 4.2 is constructed in such a way that the system efficiency can be maximized across a broad range of tags using only frame sizes that have the powers of two. An efficient mean to identify the optimum frame sizes for every terminated slots is to assign the mode of a posterior distribution of the number of tags at each terminating slot as an initial frame size. We can then optimize them to obtain suitable frame sizes, as given in Table 4.2. The main purpose of Table 4.2 to assign an initial frame size for each terminating slot so that the system efficiency is maximized across a broad range of tags using only frame sizes that are the power of two. The posterior distribution of the number of tags at the terminating slot is the key to assign the frame size. We use the mode of the posterior distribution at the terminating slot to assign the frame size, as it is the value that appears most often. Therefore, $L = 2^{round(\log_2(\hat{N}))}$ where $\hat{N}$ is the mode of the posterior distribution. Based on these initial frame sizes, we can then optimize them to obtain suitable frame sizes, as given in Table 4.2.

Table 4.2 Initial frame sizes for MDT.

Slot	1-3	4-5	6-12	13-20	21-40	41-70	71-132	133-249	250-490	491-550	>550
Frame Size(L)	64	128	256	512	1024	2048	4096	8192	16384	32768	65536



a) Phase 1 b) Phase 2

Figure 4.2 Flow chat of proposed BE-MD

4.2.3 Flow chart of BE- MDT

Figure 4.2 shows the complete flow chart of the proposed algorithm. In the first phase, each unidentified tag accesses a slot with the probability of $1/2^Q$, where the initial broadcast frame size of $Q = 4$. When the reader experiences an idle or successful feedback from the current reading slot, it dismisses tag estimation phase and moves to the second phase. Otherwise, the reader continues in the first phase with an updated value of Q using Table 4.1. In the second phase, an initial frame size (L) is chosen based on the variable *slot* using the lookup table given in Table 4.2. The reader sends the query command and wait for L slots. Only those collided slots are further resolved through MTA.

4.3 Average number of slots required in collision resolution in BE-MDT

Let $T_{BE-MDT}(N, \max slot)$ be the average number of slots needed by BE-MDT to identify N tags, $p_i = 1/2^{Q_i}$ be the access probability that each tag uses in the i^{th} slot in the first phase where the Q_i value is given in Table 4.1 and L_i be the initial frame size used in the second phase for MDT. Then $T_{BE-MDT}(N, maxslot)$ can be calculated as:

$$\begin{aligned} T_{BE-MDT}(N, maxslot) = {} & (1-p_1)^N T_{MDT}(N, L_1) + Np_1(1-p_1)^{N-1} T_{MDT}(N-1, L_1) \\ & + \sum_{i=2}^{maxslot} [(1-p_i)^N T_{MDT}(N, L_i) + Np_i(1-p_i)^{N-1} T_{MDT}(N-1, L_i) \prod_{m=1}^{i-1} x_m] , \\ & + T_{MDT}(N, L_{maxslot}) \prod_{q=1}^{maxslot} x_q \end{aligned} \tag{4.3}$$

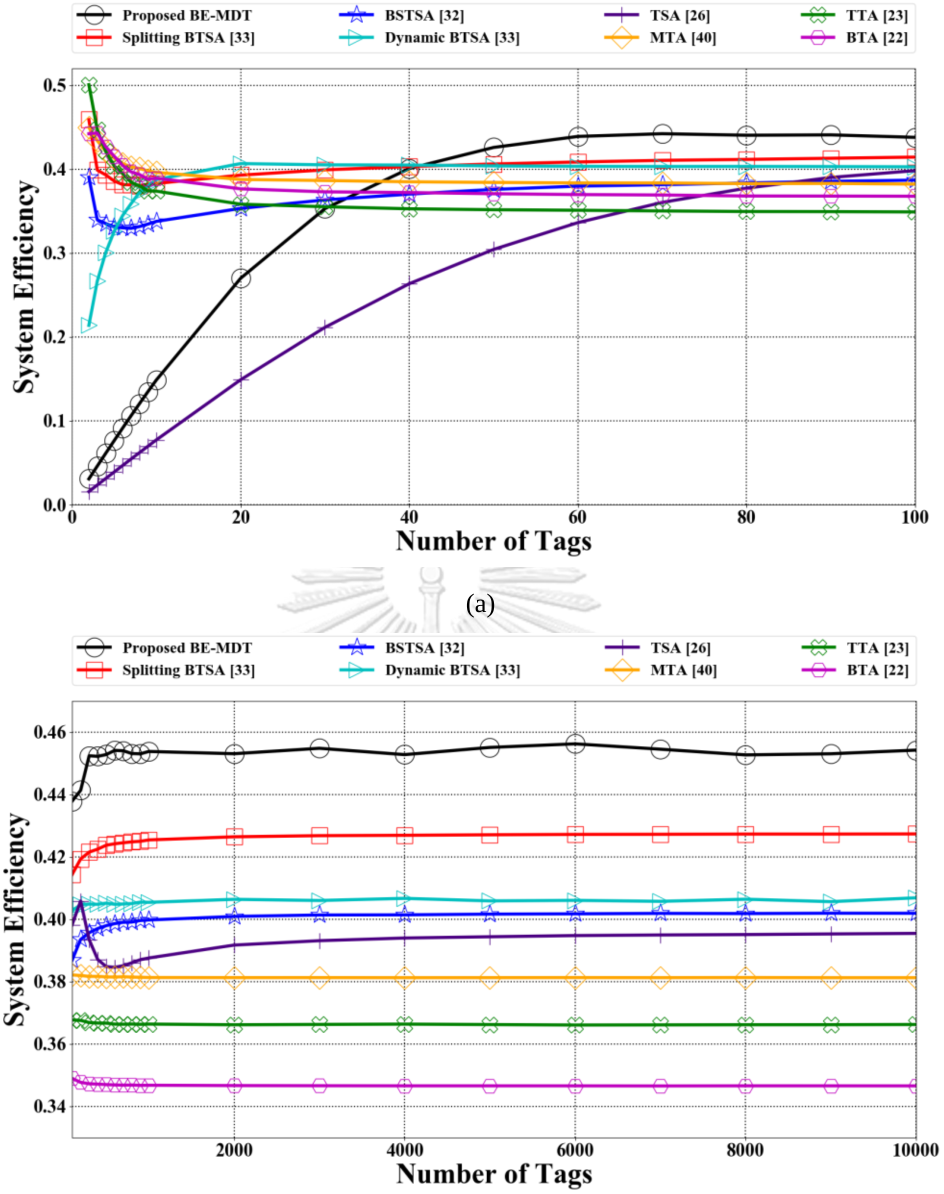
where $x_m = 1-(1-p_m)^N - Np_m(1-p_m)^{N-1}$.

For BE-MDT, tags stay in the first phase as long as they experience collision slots. Hence, the probabilities that all tags arrive at the $(i-1)^{th}$ slot while still in the first phase is $\prod_{m=1}^{i-1} 1-(1-p_m)^N - Np_m(1-p_m)^{N-1}$. If they transit into the second phase due to idleness or success, with the probability of $(1-p_i)^N$ and $Np_i(1-p_i)^{N-1}$ respectively, the number of tags entering the second phase are N and $N-1$ respectively and they will be resolved through MDT. $T_{MDT}(N, L_i)$ which can be derived using (3.17) in Section 3.6. By taking into account all possible events as shown in Figure 4.1(c) (4.3) can be yielded.

4.4 Simulation Results and Discussion

In this section, we compared the performance of the proposed BE-MDT with that of known tree-based anti-collision algorithms in terms of system efficiency, average delay, average number of collisions and idle slots for 2 to 10000 number of tags range. The system efficiency η_{SE} is the ratio of the number of tags N and the average number of time slots $T(N)$ required in an anti-collision algorithm to identify N tags as specified bellow:

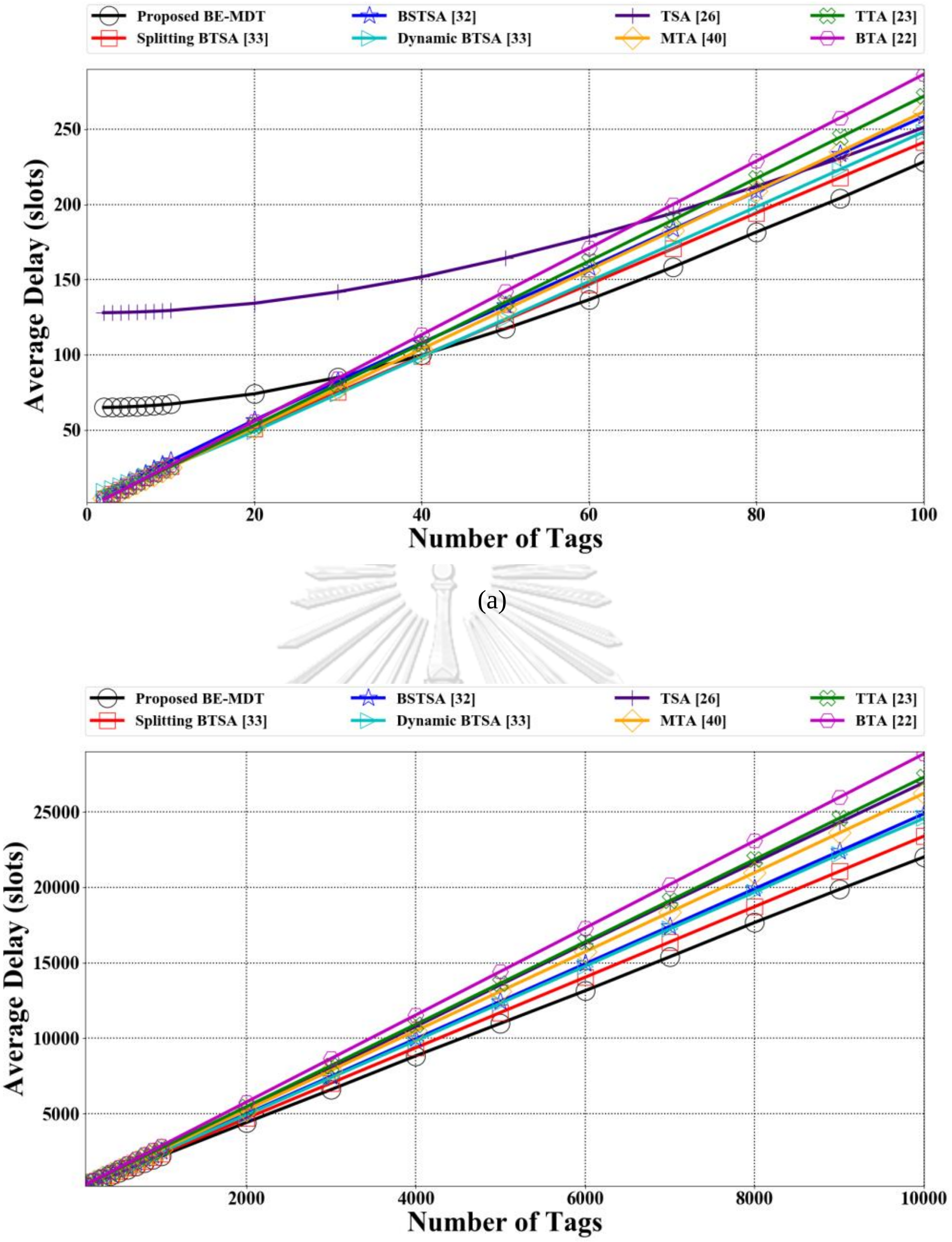
$$\eta_{SE} = \frac{N}{T(N)}. \tag{4.4}$$



(a)

(b)

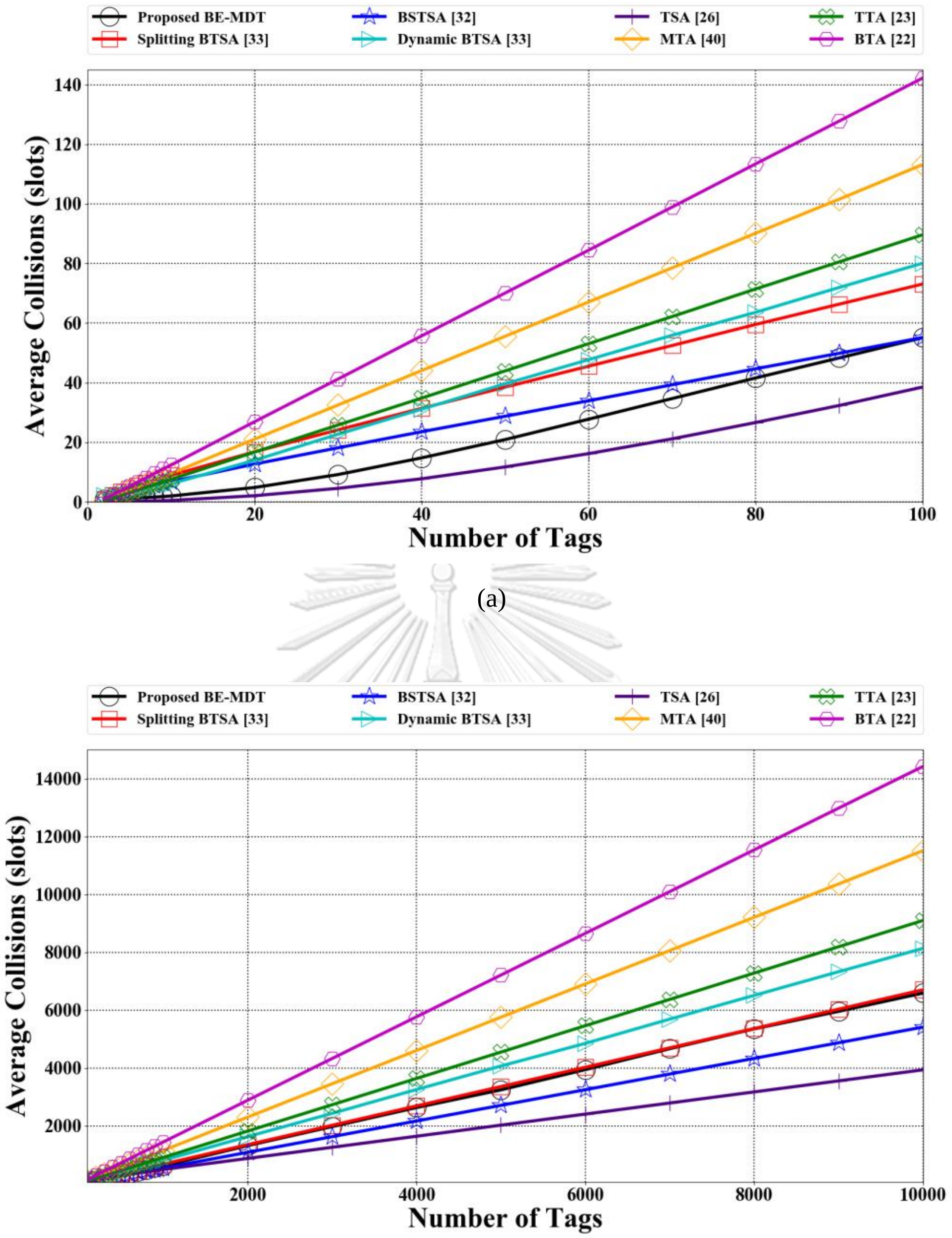
Figure 4.3 Comparison of system efficiency for various tree-based algorithms.



(a)

(b)

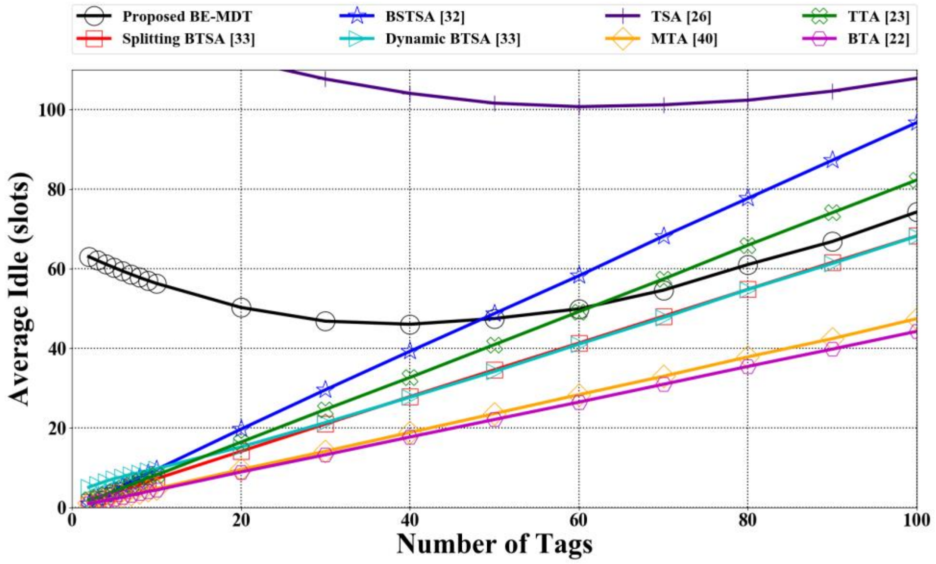
Figure 4.4 Average number of slots for identifying all tags for various tree-based algorithms.



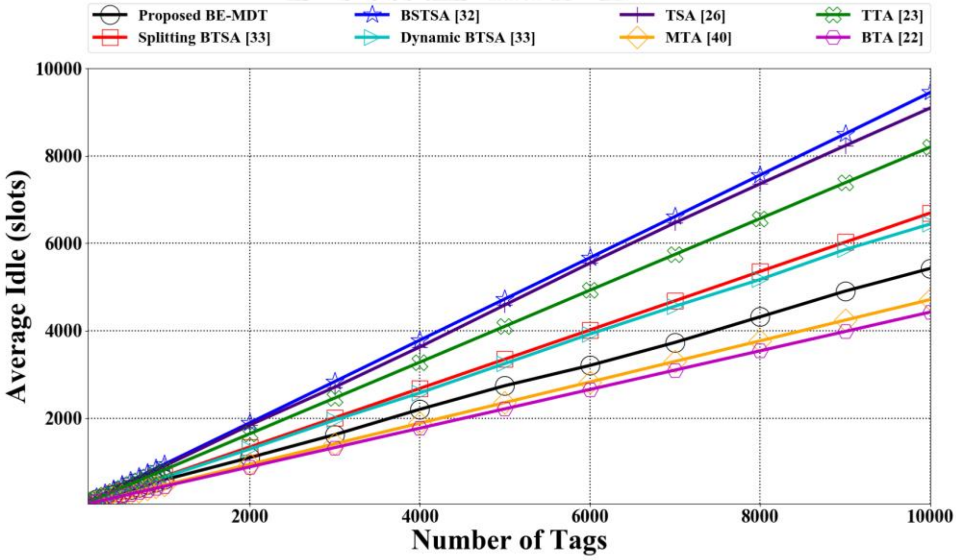
(a)

(b)

Figure 4.5 Average number of collsions for various tree-based algorithms.



(a)



(b)

Figure 4.6 Average number of idles for various tree-based algorithms.

Discussion:

Figure 4.3 presents the system efficiency comparison between the proposed BE-MDT algorithm and BTA [22], TTA [23], MTA [40] TSA [26], BSTSA [32], Dynamic BTSA [33] and Splitting BTSA [33]. Figure 4.3(a) and (b) show the system efficiency for tags $2 \le N \le 100$ and $100 \le N \le 1000$ respectively. It is clear that the BE-MDT consistently offers higher system efficiency of more than 45% than all other well-known tree-based algorithms for tags between 100-10,000 and beyond. Note that, for small number of tags, these numerical results for BTA, MTA, MDT and BE-MDT are obtained from our derived mathematical formula (3.7), (3.10) and (3.17) respectively and confirmed through computer simulations. For larger number of tags, due to floating point precision limitations, numerical results for all algorithms are obtained solely from computer simulations. To ensure high accuracy of the simulation results, 10,000 runs of tag identification are executed for each case. As given in Figure 4.3, the proposed BE-MDT can achieve better in system efficiency when $N > 40$ than all other algorithms.

Figure 4.4 (a) and (b) illustrate the average delay which is the number of slots required to identify all tags for $2 \le N \le 100$ and $100 \le N \le 1000$ respectively. As shown in the Figure 4.4, BE-MDT gives lowest delay when $N > 40$ than all other algorithms. This is in line with the highest system efficiency for $N > 40$, which we have discussed earlier. Further, Figure 4.5 and Figure 4.6 give the average number of collision and idle slots for identifying all tags, respectively. In these figures, our proposed protocols may not have the least number of collision or idle slots.

Chapter 5 Binary Splitting Modified Dynamic Tree Anti-Collision Algorithm for RFID Systems

In this chapter, another proposed algorithm called binary splitting modified dynamic tree anti-collision (BS-MDT) algorithm for RFID systems is explained. Throughout the chapter, a detailed clarification of whole algorithm is presented with a flow chart and other necessity diagrams.

5.1 Introduction to BS-MDT

The proposed BS-MDT algorithm first performs tag estimation by using the binary splitting through a tree structure as proposed by [33] and subsequently adopted by [40], until the left branch contains zero or one tag. Then, tags in the right branches of the binary tree structure are resolved from the lowest level to the top by using our proposed modified dynamic tree (MDT) algorithm. Hence, this algorithm is referred as binary splitting modified dynamic tree (BS-MDT) algorithm.

5.2 Execution of BS-MDT

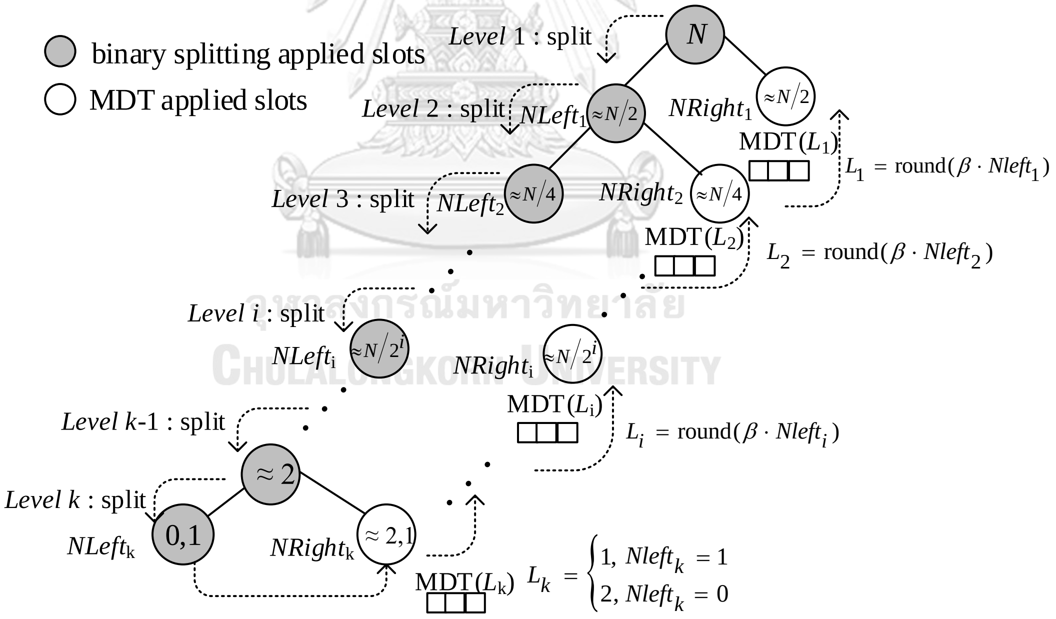
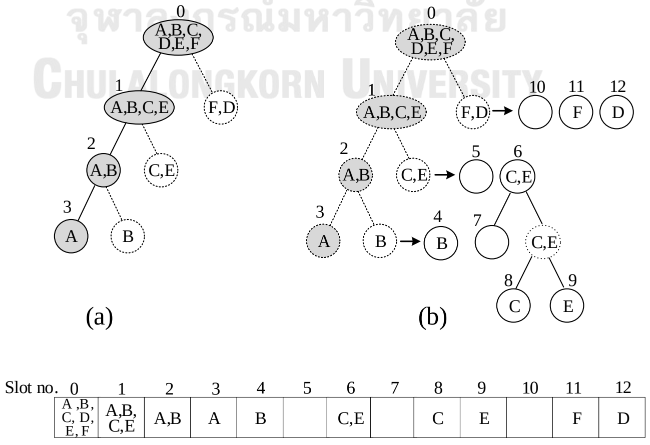


Figure 5.1 BS-MDT algorithm tree.

Figure 5.1 presents the binary splitting procedure, followed by our proposed initial frame size assignment method for the MDT. As we can see, tags are split into two subgroups, i.e. left and right, repeatedly in the depth first search manner, until reaching the slot or the k^{th} level which contains either zero or one tag. For an initial size of N tags, the value of k can be estimated as $log_2 N$, indicating that this process can finish very rapidly. If at the k^{th} level, the slot contains no tag, i.e. $NLeft_k = 0$, it is certain that the number of tags on the right branch of the k^{th} level must contain at least 2 tags, i.e., $NRight_k \geq 2$. Therefore, the initial frame size (L_k) should be set to at least two slots and in the proposed method, L_k is set to 2 for maximizing the system efficiency. On the other hand, if $NLeft_k = 1$, L_k is set to 1 for the same reason. Once all tags in the right branch of the k^{th} level are resolved by the MDT method described below, the actual value of $NRight_k$ can be known, thus, actual number of colliding tags at the $(k-1)^{th}$ level on the left branch can be readily obtained as $NLeft_{k-1} = NRight_k + NLeft_k$. The next step is to resolve tag collision on the right branch of each level starting from the $(k-1)^{th}$ level upwards to the 1^{st} level using MDT. Unlike most algorithms that usually assign the initial frame size equal to the estimated number of tags, we show that the system efficiency can be increases by 1% by assigning $L = round(\beta N)$, where $\beta < 1$. The MDT algorithm execution is based on two steps. In the first step, the competing tags randomly select a timeslot from a frame, which is made up of L slots. In the second step, the collided slots within the frame are resolved using the modified tree algorithm (MTA) [40, 41] until all the collided tags are identified.

5.3 Example of transmission process in the BS-MDT algorithm



Slot no.	0	1	2	3	4	5	6	7	8	9	10	11	12
	A ,B, C, D, E, F	A,B, C,E	A,B	A	B		C,E		C	E		F	D

(c)

Figure 5.2 Execution procedure of BS-MDT algorithm for six initial tags collision.

Figure 5.2 presents an example of the tag identification process of BS-MDT for six colliding tags (A-F) in Slot0. Figure 5.2(a) shows detail of the first step, where binary splitting is performed three times and terminates at Slot 3 with one tag in it, *i.e.*, tag A. Figure 5.2(b) shows details of the second step. Tag B is first resolved by MTA with an initial frame size of $L_3 = 1$ and the number of colliding tags in Slot 2 is known to be two. Therefore, the initial frame size for resolving tags on the right-hand side of the 2^{nd} level containing Tags C and E is $L_2 = 2$, assuming $\beta = 0.79$. Notice that Slot 7 is idle, implying that the next slot will be a sure collision, and hence it is skipped and immediately split into two slots, i.e., Slots 8 and 9. Since we can know that Slot 1 contains 4 colliding tags, the initial frame size for resolving the right-hand-side slot of the 1st level is $L_1 = 3$. A total of twelve slots are required to resolve the six tag collisions as shown in Figure 5.2(c).

5.4 Analysis of the BS-MDT Algorithm

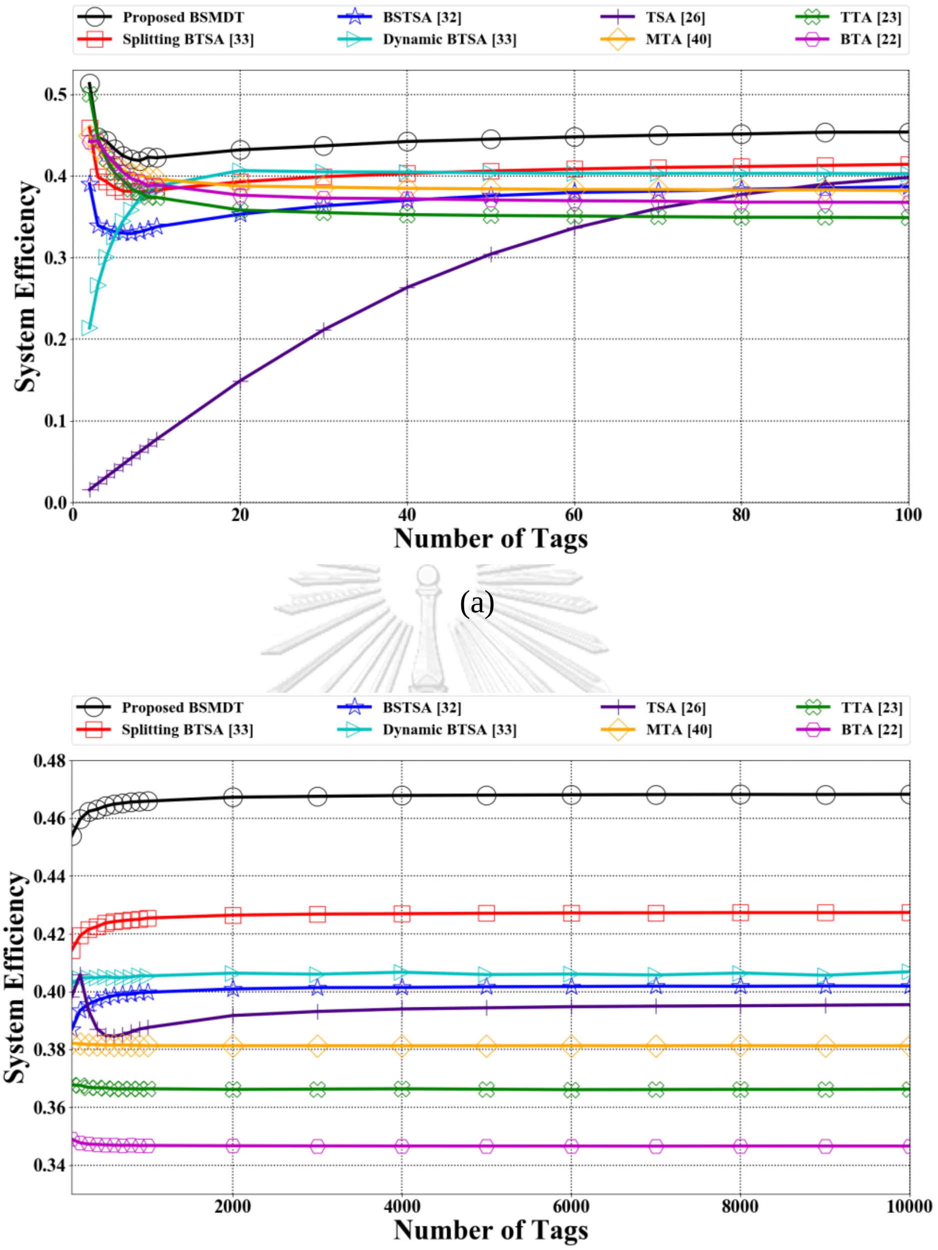
$T_{BSMDT}(N)$, which is the average timeslots used in the proposed BS-MDT can be derived using T_{MDT} in (3.17) and is given in (5.1).

$$T_{BSMDT}(N) = \sum_{j=0}^{N-1} \frac{1}{2^N} \binom{N}{j} \left[1 + T_{MDT}(j,q) + T_{BSMDT}(N-j)\right] + \frac{1}{2^N}\left[1 + T_{MDT}(N,2)\right] \quad \textbf{(5.1)}$$

where $q = round\left(\beta\left(N-j\right)\right)$ (*i.e.*, *rounds* $\left(\beta\left(N-j\right)\right)$ to the nearest integer) and $T_{BSMDT}(0) = T_{BSMDT}(1) = 0$.

5.5 Simulation parameters and results

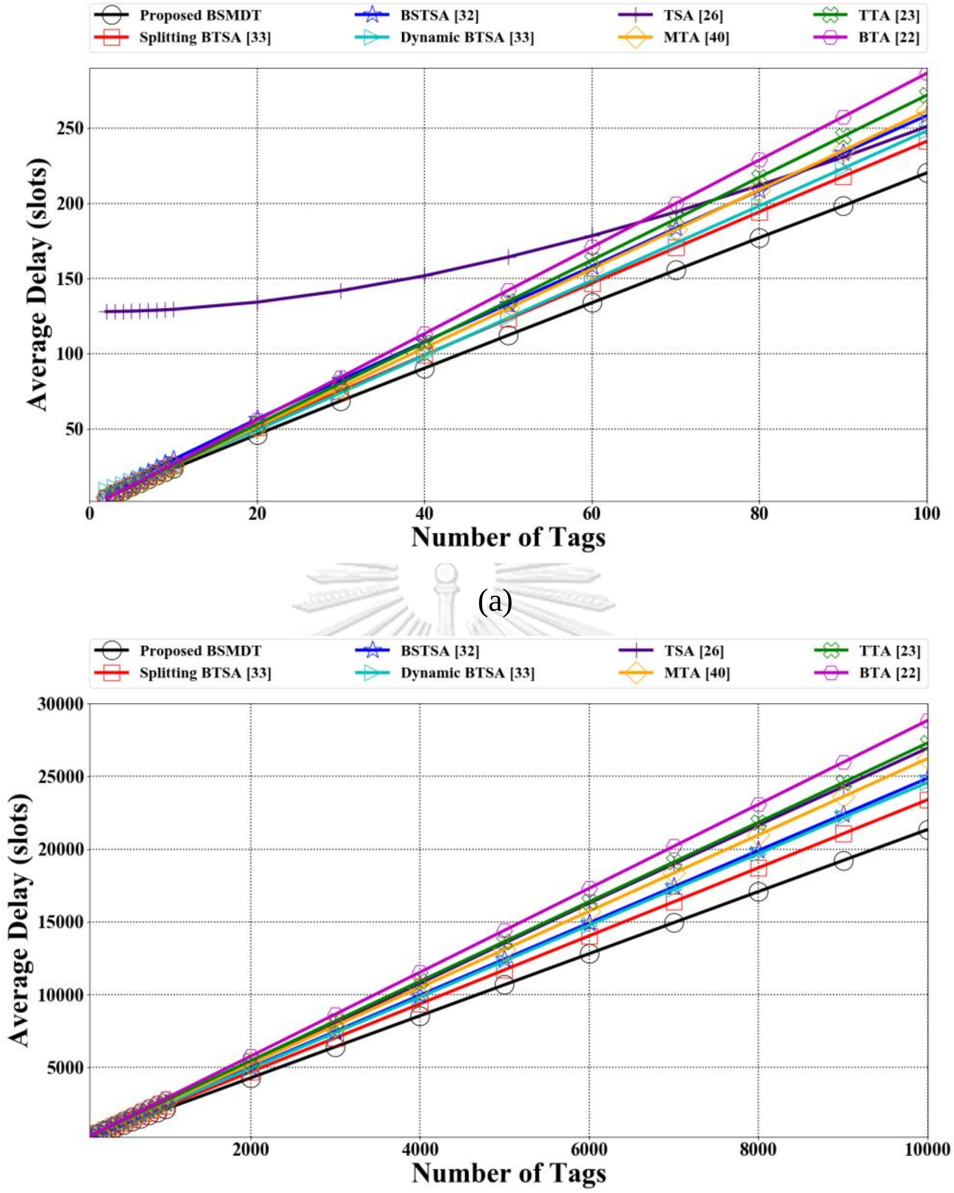
In this section, we analyze the average number of timeslots required by the BTA [22], TTA [23], MTA [40] TSA [26], BSTSA [32], Dynamic BTSA [33]and Splitting BTSA [33] algorithms to resolve all N tags, so that their corresponding system efficiency is determined using (4.4).



(a)

(b)

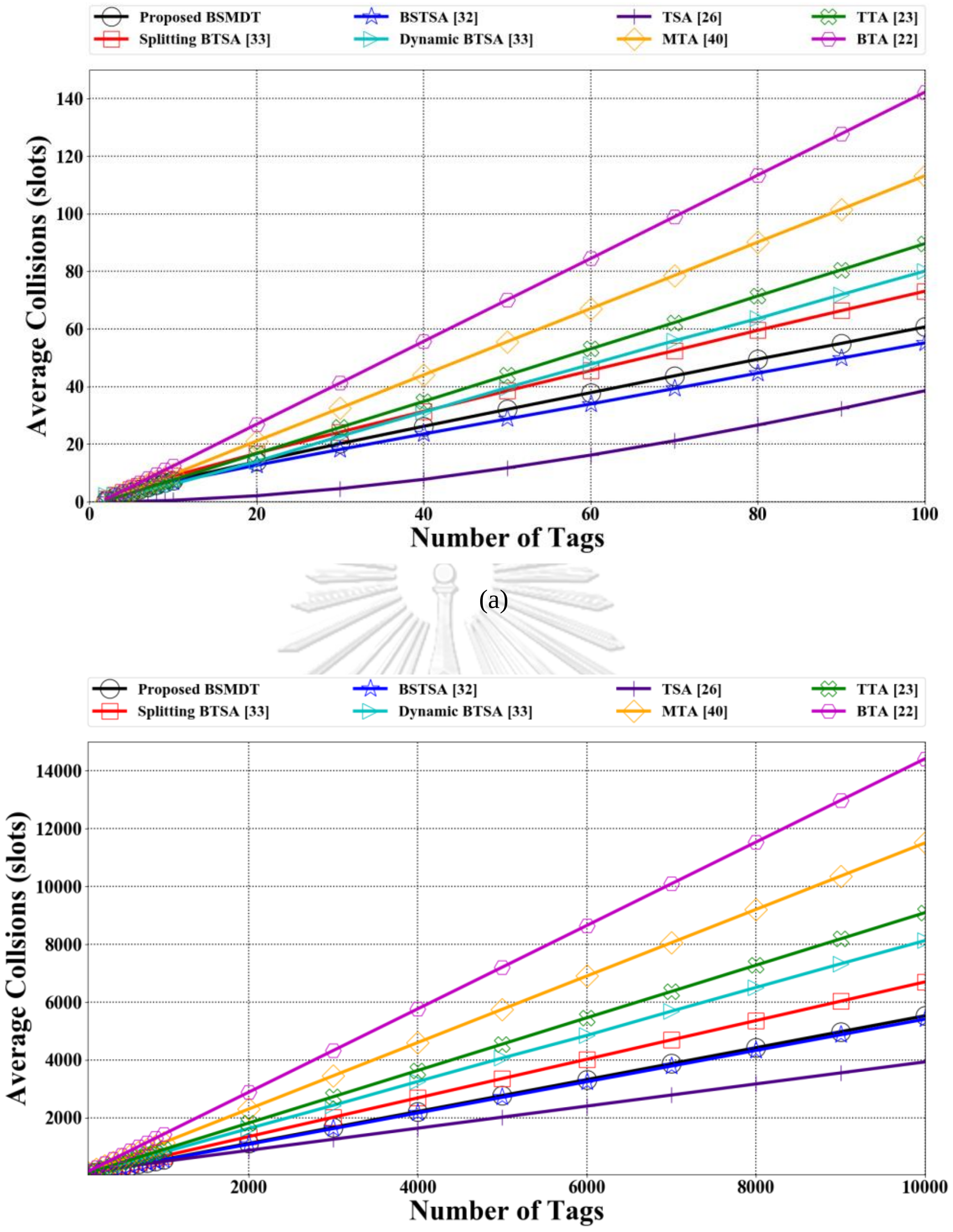
Figure 5.3 Comparison of system efficiency for various tree-based algorithms.



(a)

(b)

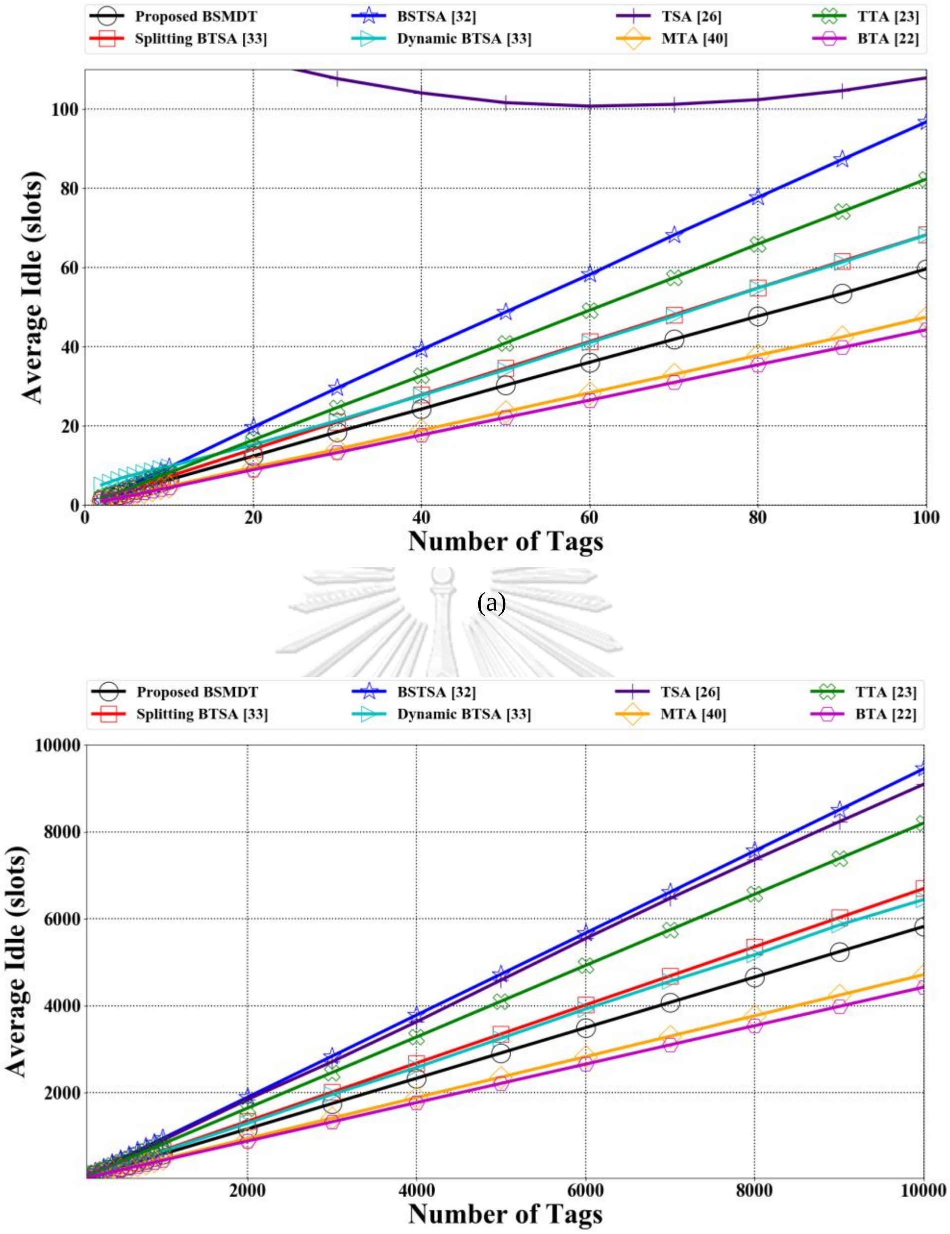
Figure 5.4 Average number of slots for identifying all tags for various tree-based algorithms.



(a)

(b)

Figure 5.5 Average number of collsions for various tree-based algorithms.



(a)

(b)

Figure 5.6 Average number of idle for various tree-based algorithms.

Discussion:

The performance of the proposed BS-MDT is evaluated and compared with BTA [22], TTA [23], MTA [40] TSA [26], BSTSA [32], Dynamic BTSA [33]and Splitting BTSA [33]. in terms of system efficiency and the results are shown in Figure 5.3. Numerical results are obtained through our derived analytical model and verified through computer simulation. The parameters β and p are optimized through extensive simulations as shown in Section 3 and the system efficiency is maximized when using $\beta = 0.79$ and $p = 0.418$. Figure 5.3(a) presents the comparison of system efficiency for small numbers of tags, namely 2 to 100 tags. The proposed BS-MDT algorithm achieves at least 42% efficiency, which is higher than all other algorithms: almost 4% higher than that of splitting BTSA and over 6% higher than that of BSTSA. For larger numbers of tags, i.e. 100-10,000, the system efficiency of BS-MDT algorithm is above 46% as reflected in Figure 5.3(b) and it is almost constant in the range for 10,000-50,000 tags (not shown here). This is a significant improvement from existing tree-based anti-collision algorithms. As depicted in Figure 5.4, the proposed method gives the lowest delay compare all other seven tree-based algorithms. However, it doesn’t give the lowest average number of idle or collision slots.

Chapter 6 BE-MDT and BS-MDT in RFID EPCglobal Gen2 standard

6.1 Standards for RFID tag identification

6.1.1 RFID standards bodies

International Standards Organization (ISO) and Electronics Product Code Global Incorporated (EPCglobal) are the two main international standards for RFID. In 1996, ISO joined with IEC to standardize the RFID technology and this standard specifies in the literature in the form of ISO/IEC. In 1999, with aim of doing research and standardize RFID technology, set of companies joined with MIT and form a group known as the Auto-ID consortium. In 2003, this consortium moved to a new unit call as EPCglobal.

The anti-collision protocols and standards for RFID tag identification are listed in Table 6.1.

Table 6.1 Standards and anti-collision protocols [28]

Anti-collision protocols	Standards for RFID tag identification
Q protocol	ISO/IEC 18000-3 Mode 1 EPCglobal Class 1 Generation 2
Pure ALOHA	ISO/IEC 18000-3 Mode 1 Extention
Slotted ALOHA	ISO/IEC 18000-3 Mode 2
Frame slotted ALOHA	ISO/IEC 18000-3 Mode 1 Extention ISO/IEC 18000-6 Type A EPCglobal Class 1 EPCglobal Class 1 Generation 2
Dynamic frame slotted ALOHA	ISO/IEC 18000-3 Mode 1 ISO/IEC 14443-3 Type B
Binary tree protocol	ISO/IEC 18000-6 Type B EPCGlobal Class 0 EPCGlobal Class 1
Query tree protocol	ISO/IEC 18000-3 Mode 1
Dynamic binary search algorithm	ISO/IEC 14443-3 Type A

The binary tree algorithm is used by the first-generation of EPC standards and in EPC 2nd generation (EPCGen2) [18] approach, it uses the Q algorithm which is a variant of originally known slotted ALOHA.

6.2 Slotted ALOHA in EPCGen2 Standard

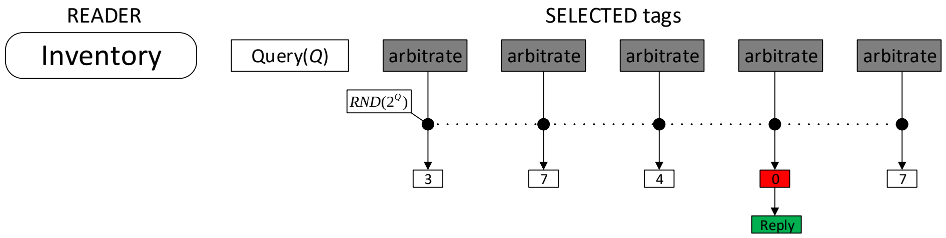


Figure 6.1 Function of slotted ALOHA in EPCGen2 standard [18].

In the inventory round, the reader broadcast Query command and inform the number of slots in the inventory. The tags receiving the Query command identify that they are the contestants in reader's interrogation zone. The Query command contains a numerical value Q which helps each tag to randomly select a number between 0 and ($2^{Q}-1$). This number indicates in which slot the tag will respond. If the random number is zero, then the tag responds in the current slot, else it stores this value in the slot counter and wait for it turn. During these inquiries, if the reader can decode receive, random number will send an acknowledgement to the tag. Next, the tag can send the electronic product code (EPC) which is the tag ID.

6.3 Time System Efficiency (TSE) using EPCGen2 Standard

Although the system efficiency has been the most common measure of performance for anti-collision protocols, recent studies for RFID systems aim at maximizing the time system efficiency as it takes into account of the fact that idle slots are shorter than successful and collision and various recommended time parameters as defined in the EPCglobal Gen 2 standard play its part in the actual time required for tag identification.

The time system efficiency η_{TSE} is defined as [32]

$$\eta_{TSE} = \frac{NT_S}{I(N)T_I + S(N)T_S + C(N)T_C}. \qquad (6.1)$$

where $I(N), S(N)$ and $C(N)$ are the expected number of idle, successful and collision slots respectively, while T_I, T_s and T_c are their corresponding time duration, according to EPCglobal C1 Gen2 standard [18].

In EPCglobal Gen 2 standard [18], link timing between reader→tags and tags→reader is shown in Fig. 6.2. When the reader issues a Query or QueryRep command, three possible outcomes may arise. First, single tag replies as depicted in Figure. 6.2(a). In this case, the reader responds by returning an acknowledgment and the tag sends its identity to complete a successful tag identification. Second, multiple tags reply as shown in Figure 6.2(b). A collision occurs, and no further action is taken. Third, no tag replies as illustrated in Figure 6.2(c). In this case, the reader is able to detect and recognize the absence of replied signals and thus decides to terminate earlier. The time durations for a single tag reply (T_S), collided reply (T_C) and no reply (T_I) for Phase II can be expressed as [18]:

$$T_S = 2T_1 + 2T_2 + T_{QRep} + T_{RN16} + T_{Ack} + T_{EPC} + T_{Preamble} \tag{6.2}$$

$$T_C = T_1 + T_2 + T_{QRep} + T_{RN16} + T_{Preamble} \tag{6.3}$$

$$T_I = T_1 + T_3 + T_{QRep} + T_{Preamble} \tag{6.4}$$

These expressions indicate clearly that in the EPCglobal Gen 2 standard, successful, collision and idle slots have different time durations, with $T_S > T_C > T_I$. We follow the timing values given in Table 6.2 to calculate T_S, T_C and T_I parameters.

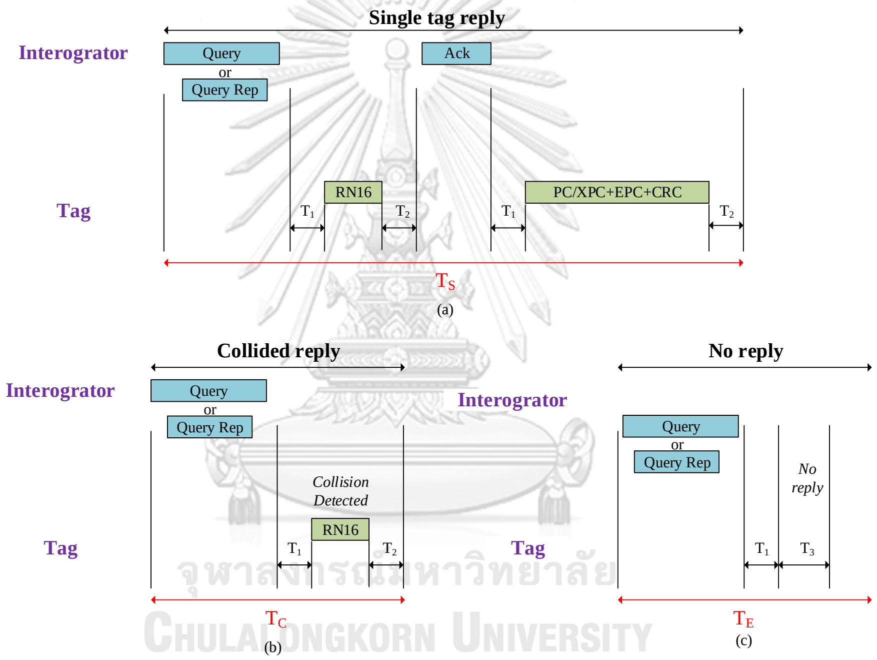


Figure 6.2 Link Timing according to EPCglobal Gen2 standard.

Table 6.2 Gen2 Reader Interrogation Parameters [37]

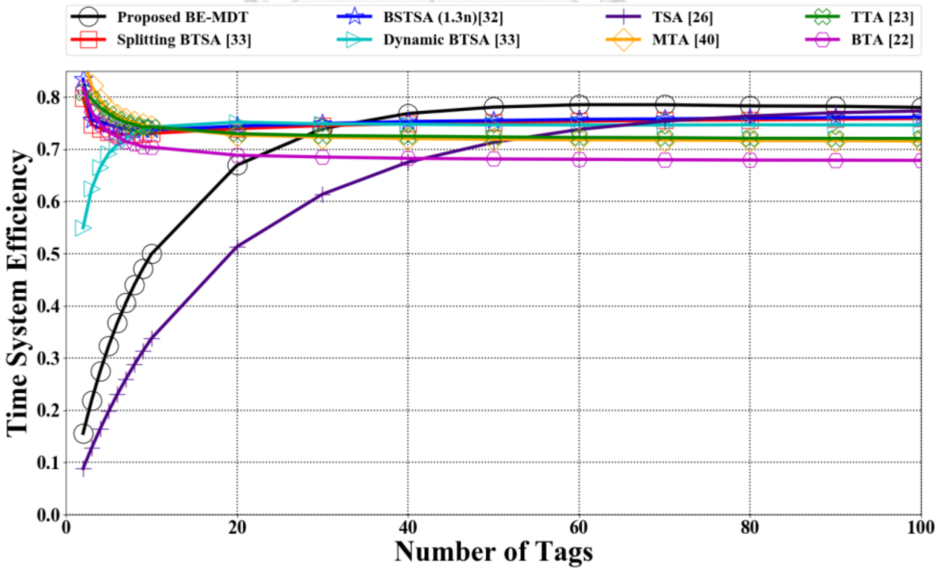
Parameter	Duration	Parameter	Duration
Reader- to -Tag Data0	1T$_{ari}$	RTcal	75μs
Tag - to -Reader Data1	2T$_{ari}$	TRcal	200μs
Reader-to-Tag Rate	40kbps	T1	250μs
Tag-to-Reader Rate	40kbps	T2	250μs
T$_{pri}$	25μs	T3	100μs
T$_{ari}$	25μs	RN16	16 bits
DR	8	EPC	96 bits
Query	22 bits	Ack	18 bits
Query Adju	9 bits	Query Rep	4 bits

The 22-bit long Query command was included to every slot in Phase I along with other timing parameters, such as $T_{RN16}, T_{Ack}, T_{EPC}, T_{Preamble}, T_1, T_2, T_3$. Therefore, the effect of extra-reader Query commands is considered together with all other necessary commands according to the EPCGen2 standard in the new and additional investigation of the time system efficiency. To be precise, in each slot of Phase I the timing for success (T_S), idle (T_I) and collision (T_C), slots are given as:

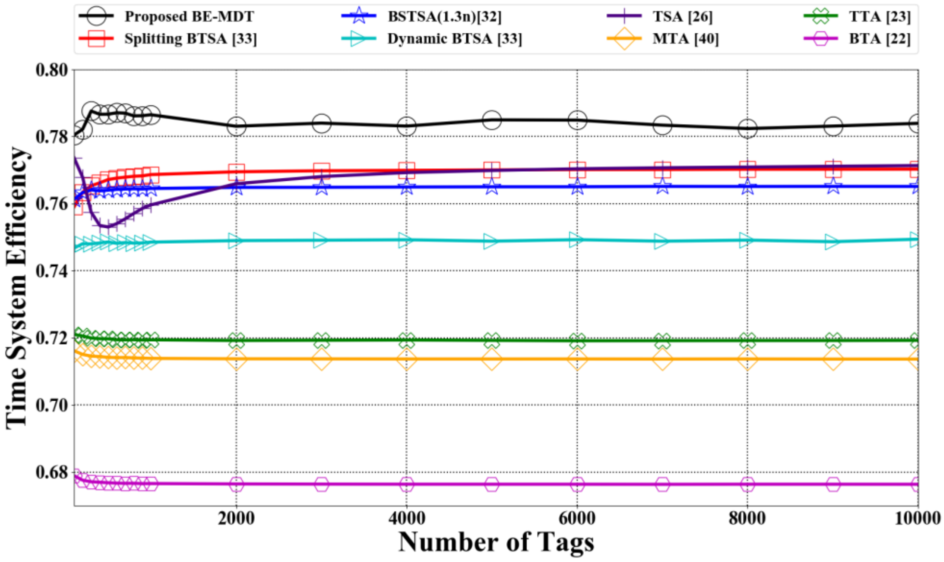
$$T_S = 2T_1 + 2T_2 + T_{Query} + T_{RN16} + T_{Ack} + T_{EPC} + T_{Preamble} \quad (6.5)$$

$$T_I = T_1 + T_3 + T_{Query} + T_{Preamble} \quad (6.6)$$

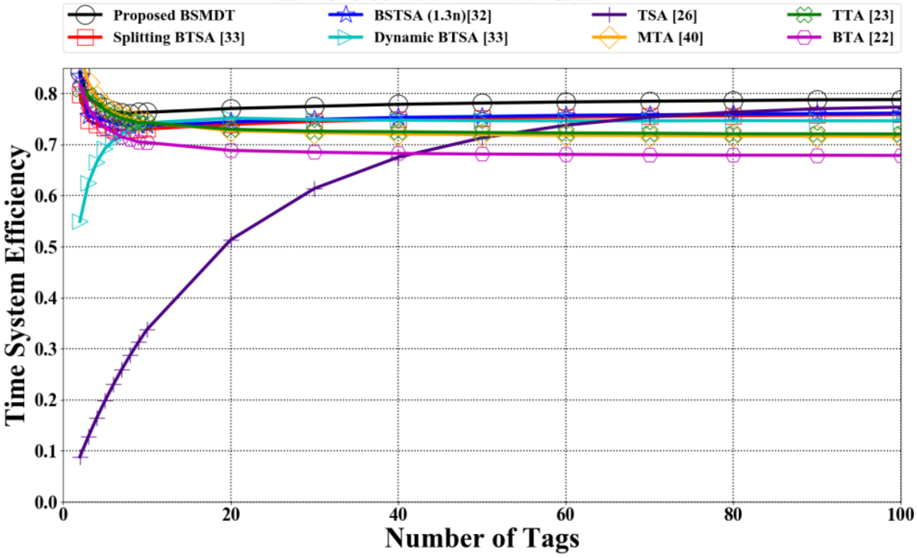
$$T_C = T_1 + T_2 + T_{Query} + T_{RN16} + T_{Preamble} \quad (6.7)$$



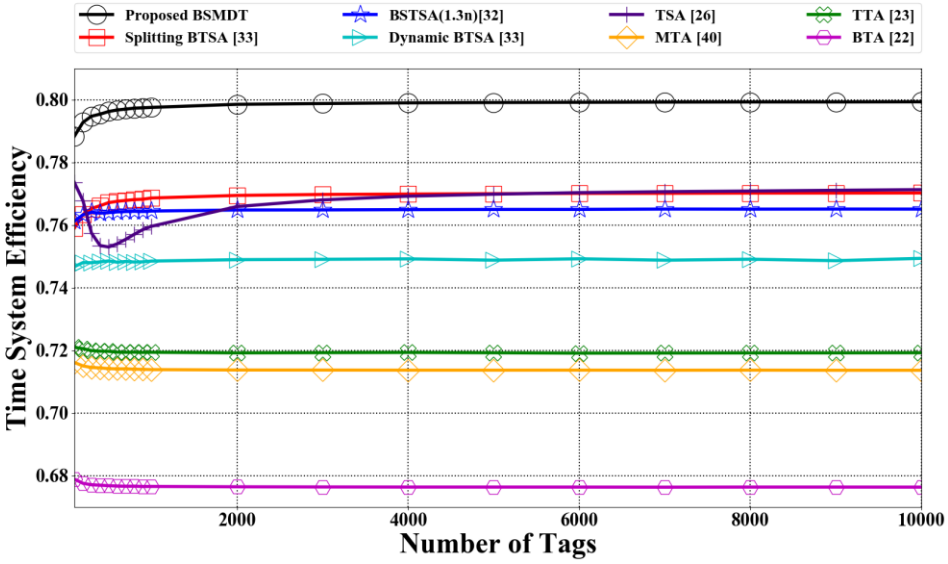
(a)


Proposed BE-MDT
Splitting BTSA [33]
BSTSA(1.3n)[32]
Dynamic BTSA [33]
TSA [26]
MTA [40]
TTA [23]
BTA [22]
0.80
0.78
0.76
0.74
0.72
0.70
0.68
Time System Efficiency
2000
4000
6000
8000
10000
Number of Tags

(b)

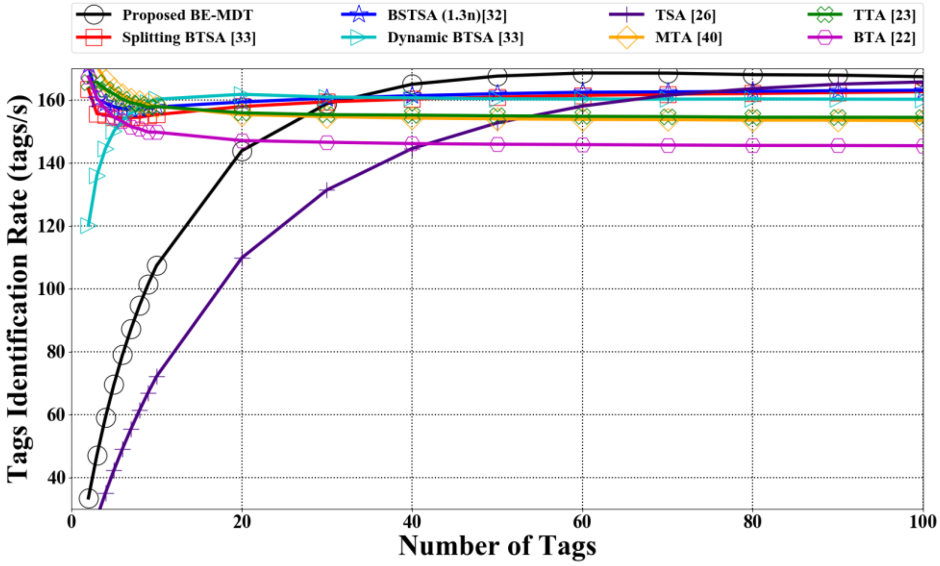

Proposed BSMDT
Splitting BTSA [33]
BSTSA (1.3n)[32]
Dynamic BTSA [33]
TSA [26]
MTA [40]
TTA [23]
BTA [22]
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Time System Efficiency
0
20
40
60
80
100
Number of Tags

(c)

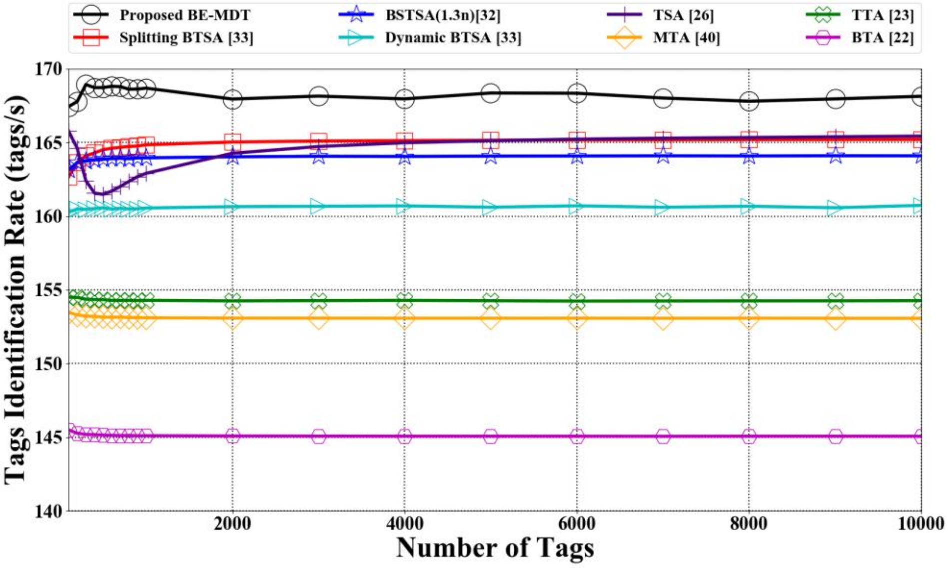


(d)

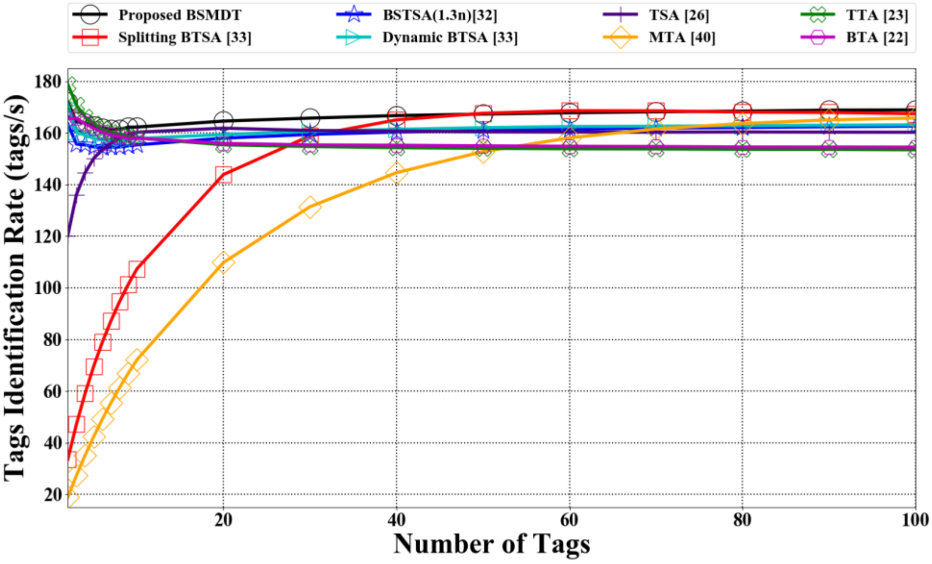
Figure 6.3 Time system efficiency for different variations of tree-based algorithms.



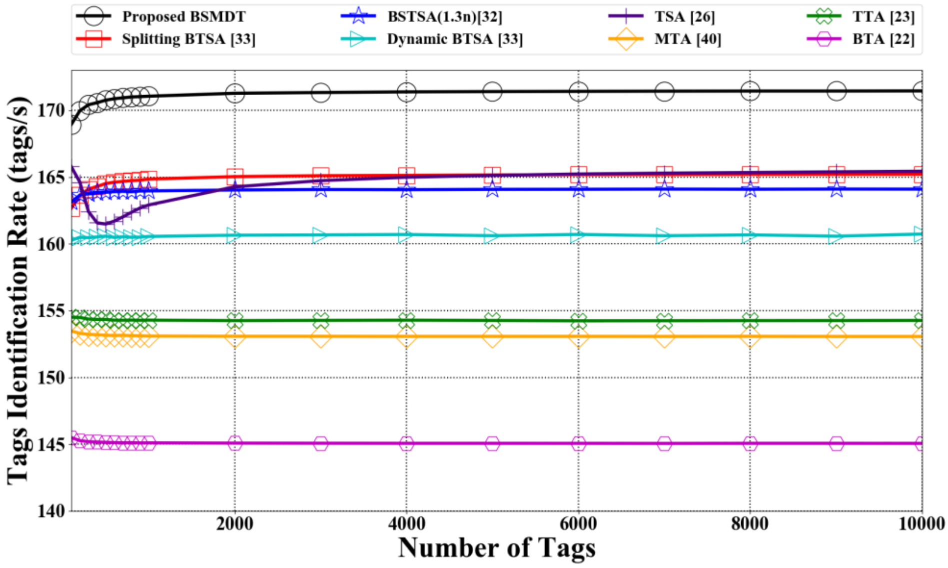
(a)


Proposed BE-MDT
Splitting BTSA [33]
BSTSA(1.3n)[32]
Dynamic BTSA [33]
TSA [26]
MTA [40]
TTA [23]
BTA [22]
Tags Identification Rate (tags/s)
170
165
160
155
150
145
140
2000
4000
6000
8000
10000
Number of Tags

(b)


Proposed BSMDT
Splitting BTSA [33]
BSTSA(1.3n)[32]
Dynamic BTSA [33]
TSA [26]
MTA [40]
TTA [23]
BTA [22]
Tags Identification Rate (tags/s)
180
160
140
120
100
80
60
40
20
20
40
60
80
100
Number of Tags

(c)



(d)

Figure 6.4 Tags identification rate for different variations of tree-based algorithms.

Discussion:

We applied the proposed algorithms to the RFID system according to the EPCglobal Gen 2 standard at data rate of 40 kbps with time parameters as given in Table 6.2. Figure 6.1 (a), (b) show the time system efficiency of BE-MDT algorithm. Figure 6.1 (c), (d) gives the time system efficiency of BS-MDT algorithm. Numerical results given in these figures show that the proposed BE-MDT and BS-MDT algorithms can achieve better time system efficiency than all other algorithms, even the BSTSA which is primarily designed and optimized for maximum time system efficiency. It is possible that some protocols which have good system efficiency may not be as effective with the time system efficiency, such as MTA. The reverse is also true for instance BSTSA. For our proposed algorithms, both system efficiency and time system efficiency are better than all known protocols. As can be observed from Figure 6.4 the proposed methods can achieve the best overall performance in terms in tag identification rate as well.

Chapter 07 Conclusions and Future Works

7.1 Conclusions

In this thesis, we proposed two highly efficient anti-collision algorithms with system efficiency of 45% for a large range of number of tags. Firstly, in BE-MDT, the slot-based Bayesian estimation technique is used along with the proposed modified dynamic tree algorithm for rapid tag identification. Secondly, an algorithm called BS-MDT is proposed in conjunction with binary splitting estimation method which can identify tags for RFID systems faster than other known tree-based algorithms across the entire range of tag population from 2 to at least 10000. We also presented a new and complete mathematical analysis to derive the average number of time slots required by several tree-based algorithms such as BTA, MTA, MDT and proposed BE-MDT, BS-MDT algorithms to resolve a group of tags. The mathematically analyzed results are verified by simulations. Numerical results confirm that the BE-MDT offers system efficiency greater than 40% for number of tags (N) greater than 40. For larger range of tags, i.e., $100 \le N \le 10000$ system efficiency is around slightly more than 45%. For smaller number of tag range i.e., $2 \le N < 40$ the BE-MDT algorithm does not perform well. The BS-MDT offers the average system efficiency of 42% for small numbers of tags i.e., $2 \le N < 100$ and slightly above 46% for larger numbers of tags. i.e., $100 \le N \le 10000$.

When applying the BE-MDT and BS-MDT algorithm to the RFID system using EPCglobal Gen2 standard, the achievable time system efficiency for large range numbers of tags is at least 78.5% and 80% respectively which is higher than that of all other existing algorithms. The BS-MDT algorithm gives the time system efficiency around 78% for small numbers of tags and BE-MDT algorithm gives 78%-time system efficiency for small numbers of tags greater than 40, i.e., $N > 40$. Furthermore, we use tags identification rate, average delay, average number of collision and idle slots required in the tag identification process as the performance evaluation parameters. The simulation results confirm that the proposed two methods are performed well than stated seven well-known methods with respect to all the performance evaluation criteria

7.2 Future Works

Though this thesis has revealed the potential of achieving the highly efficient binary tree based anti-collision algorithms several concepts for extending the scope of this thesis remain. The first concern is to implement the proposed algorithms in real world application (using hardware) and compare it with simulated results. It is possible to consider small to large range of tags in the practical test bed as defined in the thesis and measure the performance with respect to stated parameters in the thesis. Currently, ALOHA-based Q algorithm is the anti-collision algorithm used in RFID Gen2 standard which is a frame-based algorithm. It is interesting to analyze the optimum efficiency that can be achieve though frame-based algorithms and compare it with the splitting-based algorithms.

In order to further extend this work, we can implement an anti-collision protocol to enhance the tag identification rate when NFC devices appear in the identification range of RFID system. Since the NFC devices and RFID systems operate 13.56 MHz, the NFC devices cannot operate as usual when they coexist. in RFID identification range.

REFERENCES

[1] K. Finkenzeller, *RFID Handbook*, 3rd ed. 2010

[2] R. Want, "An Introduction to RFID Technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25-33, 2006.

[3] P. H. Cole, L. H. Turner, Z. Hu, and D. C. Ranasinghe, "The Next Generation of RFID Technology," in *Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks*, D. C. Ranasinghe, Q. Z. Sheng, and S. Zeadally, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3-23.

[4] V. Coskun, B. Ozdenizci, and K. Ok, "A Survey on Near Field Communication (NFC) Technology," *Wireless Personal Communications*, journal article vol. 71, no. 3, pp. 2259-2294, August 01 2013.

[5] X. Jia, Q. Feng, T. Fan, and Q. Lei, "RFID technology and its applications in Internet of Things (IoT)," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, 2012, pp. 1282-1285: IEEE.

[6] K. W. C. a. R. R. D. K. Klair, "A Survey and Tutorial of RFID Anti-Collision Protocols," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, pp. 400-421.

[7] Z. T. a. Y. He, "Research of Multi-access and Anti-collision Protocols in RFID Systems," in *International Workshop on Anti-Counterfeiting, Security and Identification (ASID),*, Xiamen, Fujian, 2007 pp. 377-380, 2007.

[8] D.-H. Shih, P.-L. Sun, D. C. Yen, and S.-M. Huang, "Taxonomy and survey of RFID anti-collision protocols," *Computer communications*, vol. 29, no. 11, pp. 2150-2166, 2006.

[9] N. Abramson, "THE ALOHA SYSTEM: another alternative for computer communications," presented at the Proceedings of the November 17-19, 1970, fall joint computer conference, Houston, Texas, 1970.

[10] L. G. Roberts, "ALOHA Packet System with and without Slots and Capture," *Computer Communications Review*, vol. 5, no. 2, pp. 28-42, April 1975.

[11] F. Schoute, "Dynamic frame length ALOHA," *IEEE Transactions on communications*, vol. 31, no. 4, pp. 565-568, 1983.

[12] H. Vogt, "Efficient Object Identification with Passive RFID Tags," presented at the Proceedings of the First International Conference on Pervasive Computing, 2002.

[13] C. Floerkemeier, "Bayesian transmission strategy for framed ALOHA based RFID protocols," in *RFID, 2007. IEEE International Conference on*, 2007, pp. 228-235: IEEE.

[14] W.-T. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA," *IEEE transactions on automation science and engineering*, vol. 6, no. 1, pp. 9-15, 2009.

[15] E. Vahedi, V. W. S. Wong, I. F. Blake, and R. K. Ward, "Probabilistic Analysis and Correction of Chen's Tag Estimate Method," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 3, pp. 659-663, 2011.

[16] H. Wu and Y. Zeng, "Bayesian tag estimate and optimal frame length for anti-collision aloha RFID system," *IEEE Transactions on Automation Science and Engineering,* vol. 7, no. 4, pp. 963-969, 2010.

[17] P. Šolić, J. Radić, and N. Rožić, "Energy efficient tag estimation method for ALOHA-based RFID systems," *IEEE Sensors Journal,* vol. 14, no. 10, pp. 3637-3647, 2014.

[18] EPCglobal, "EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID," vol. 2.0.1, Apr-2015.

[19] W.-T. C. a. W.-B. Kao, "A Novel Q-algorithm for EPCglobal Class-1 Generation-2 Anti-collision Protocol," *nternational Journal of Electronics and Communication Engineering,* vol. 5, no. 6, 2011.

[20] Y. M. a. R. Pappu, "An Optimal Q-Algorithm for the ISO 18000-6C RFID Protocol," *IEEE Transactions on Automation Science and Engineering,* vol. 6, no. 1, pp. 16-24, Jan 2009.

[21] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE transactions on information theory,* vol. 25, no. 5, pp. 505-515, 1979.

[22] J. Capetanakis, "Generalized TDMA: The multi-accessing tree protocol," *IEEE Transactions on Communications,* vol. 27, no. 10, pp. 1476-1484, 1979.

[23] P. Mathys and P. Flajolet, "Q-ary collision resolution algorithms in random-access systems with free or blocked channel access," *IEEE Transactions on Information Theory,* vol. 31, no. 2, pp. 217-243, 1985.

[24] J. Myung, W. Lee, and J. Srivastava, "Adaptive binary splitting for efficient RFID tag anti-collision," *IEEE communications letters,* vol. 10, no. 3, pp. 144-146, 2006.

[25] L. Devroye, "The Height and Size of Random Hash Trees and Random Pebbled Hash Trees," *SIAM Journal on Computing,* vol. 28, pp. 1215–1224, 1999.

[26] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Tree Slotted Aloha: a New Protocol for Tag Identification in RFID Networks," presented at the Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, 2006.

[27] C. a. L. Law, Kayi and Siu, Kai-Yeung, "Efficient Memoryless Protocol for Tag Identification (Extended Abstract)," presented at the Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Boston, Massachusetts, USA, 2000.

[28] Y. G. X.Jia, Q.Feng,L.Zhu and L.Yue, "Standards and Protocols for RFID tag identification " in *Electronics, Communications and Networks IV: Proceedings of the 4th International Conference on Electronics, Communications and Networks (CECNET IV),* Beijing, China, 2014, pp. 851-855.

[29] H.-c. J. a. Y. Y. Haosong Gou, "A Bit collision detection based Query Tree protocol for anti-collision in RFID system," presented at the 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, Niagara Falls, 2010.

[30] W. L. a. T. K. S. J. Myung, "An Adaptive Memoryless Protocol for RFID Tag Collision Arbitration," *IEEE Transactions on Multimedia,* vol. 8, no. 5, 2006.

[31] Z. Y. a. X. Liu, "Improvement of Dynamic Binary Search Algorithm used in RFID system," presented at the Proceedings of 2011 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, Harbin, 2011.

[32] T. F. La Porta, G. Maselli, and C. Petrioli, "Anticollision protocols for single-reader RFID systems: Temporal analysis and optimization," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 267-279, 2011.

[33] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary tree slotted ALOHA for passive RFID tag anticollision," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 19-31, 2013.

[34] L. Y. H. a. B. S. L. Y. C. Lai, "Optimal Slot Assignment for Binary Tracking Tree Protocol in RFID Tag Identification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 255-268, Feb 2015.

[35] J. Su, Z. Sheng, D. Hong, and G. Wen, "An Effective Frame Breaking Policy for Dynamic Framed Slotted Aloha in RFID," *IEEE Communications Letters,* vol. 20, no. 4, pp. 692-695, 2016.

[36] J. Su, Z. Sheng, L. Xie, and G. Wen, "Idle-slots elimination based binary splitting anti-collision algorithm for RFID," *IEEE Communications Letters,* vol. 20, no. 12, pp. 2394-2397, 2016.

[37] J. S. a. W. Y. Y. Chen, "An Efficient and Easy-to-Implement Tag Identification Algorithm for UHF RFID Systems," *IEEE Communications Letters,* vol. 21, no. 7, pp. 1509-1512, July 2017.

[38] Z. S. a. L. X. J. Su, "A Collision-Tolerant-Based Anti-Collision Algorithm for Large Scale RFID System," vol. 21, no. 7, pp. 1517-1520, July 2017.

[39] J. L. Massey, "Collision-resolution algorithms and random-access communications," in *Multi-user communication systems*: Springer, 1981, pp. 73-137.

[40] Y. Y. a. G. B. Giannakis, "SICTA: A 0.693 Contention Tree Algorithm Using Successive Interference Cancellation," *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies,* vol. 3, pp. 1908-1916, 2005.

[41] R. Annur, W. Srichavengsup, S. Nakpeerayuth, and L. Wuttitsittikulkij, "Bayesian Method of Slotted Aloha Based Anti-Collision Protocol for RFID Systems," in *2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems*, 2015, pp. 87-90.

[42] F. Borgonovo and M. Cesana, "ARPA: An Arbitration Protocol Based on Advanced Channel Feedback for Radio Frequency Identification," in *2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2008, pp. 178-183.

[43] R. Rivest, "Network control by Bayesian broadcast," *IEEE Transactions on Information Theory,* vol. 33, no. 3, pp. 323-328, 1987.

APPENDIX

Contributions

List of research papers by the author is as follows.

1. **Sanika K. Wijayasekara**, Pruk Sasithong,Annur Robithoh ,Pisit Vanichchanunt Suvit Nakpeerayuth, Lunchakorn Wuttisittikulkij,"A Reduced Complexity of Vahedi's Tag Estimation Method for DFSA", ENGINEERING JOURNAL, Chulalongkorn University, Volume 21, Issue 6, pp 111-125, October 2017.

2. **Sanika.K.Wijayasekara**, M. Saadi, W. Srichavengsup, R. Anuur, S. Nakpeerayuth and L. Wuttisittikulkij", Frame Size Analysis of Optimum Dynamic Tree in RFID", International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Bangkok, Thailand,2018 (accepted).

3. **Sanika K.Wijayasekara**, Robithoh Annur, Warakorn Srichavengsup, Suvit Nakpeerayuth, Lunchakorn Wuttisittikulkij, Evaluating the influence of tree protocol as a collision resolution protocol in RFID, Regional Conference on Electrical and Electronic Engineering (RCEEE) 2017, Bumi Surabaya Hotel Jl. Jendral Basuki Rahmat No.106-128. Surabaya, Indonesia, 28-29 August 2017

4. **Sanika.K.Wijayasekara,** Pruk Sasithong, Warakorn Srichavengsup,Chairat Phongphanphanee, Robithoh Annur, Suvit Nakpeerayuth and Lunchakorn Wuttisittikulkij,"Comparison of Frame Based Tag Estimation Methods with and without Priori Knowledge", International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Busan, Korea,2017.

5. **Sanika.K.Wijayasekara,** Pruk Sasithong, Warakorn Srichavengsup,Chairat Phongphanphanee, Robithoh Annur, Suvit Nakpeerayuth and Lunchakorn Wuttisittikulkij," A Performance Study of Enhancing the Anti-Collision Algorithms in RFID System", International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Busan, Korea,2017.

6. **Sanika.K.Wijayasekara,** Chairat Phongphanphanee,Robithoh Annur,Suvit Nakpeerayuth, Lunchakorn Wuttisittikulkij", An Improved Framed slotted ALOHA-Based Anti-collision Algorithm with Skipping Idle Slots for RFID System", "The 31st International Technical Conference On Circuits/Systems, Computers And Communications" (ITC-CSCC 2016), Okinawa Pref. Municipal Center, (Okinawa Jichikaikan), Japan, 10-13 July 2016.
7. **Sanika Krishnamali Wijayasekara**, Chairat Phongphanphanee, Robithoh Annur, Lunchakorn Wuttisittikulkij, Study of Idle Skipped Dynamic Frame Slotted Aloha For RFID Systems, "The 31st International Technical Conference On Circuits/Systems, Computers And Communications" (ITC-CSCC 2016), Okinawa Pref. Municipal Center, (Okinawa Jichikaikan), Japan, 10-13 July 2016.

8. **Sanika.K.Wijayasekara**, Kritsada Mamat, Robithoh Annur, Suvit Nakpeerayuth, Lunchakorn Wuttisittikulkij, A Comparison of Tag Estimation Methods for the Initial Phase of Collision Resolution Interval in RFID, The 9th AUN/SEED-Net Regional Conference on Electrical and Electronics Engineering (RCEEE 2016), Ta Quang Buu Library in Hanoi University of Science and Technology (HUST), Hanoi, 17-18 November 2016.

9. **Sanika K. Wijayasekara**, Pruk Sasithong,Annur Robithoh ,Pisit Vanichchanunt Suvit Nakpeerayuth, Lunchakorn Wuttisittikulkij,"A Reduced Complexity of Vahedi's Tag Estimation Method for DFSA", Electrical Engineering Conference (EECON-39), Hua Hin, Thailand, 2016

VITA

Sanika Krishnamali Wijayasekara was born in Colombo, Sri Lanka. She received her B.Sc in Computer System and Networking from Sri Lanka Institute of Information Technology (SLIIT), Sri Lanka in 2010 and M.Sc in Telecommunications from Asian Institute of Technology (AIT), Thailand in 2012. Since 2015 August, she pursues the Doctoral Degree in Department of Electrical Engineering, Chulalongkorn University, Thailand with the support of the 100th Anniversary Chulalongkorn University for Doctoral Scholarship. Her main research interests are multi-access communications, Anti – Collision protocols in RFID, MAC for industrial IoT and cross- layer approach in wireless network.