Zhang, Lijuan (2017) ***High efficient tag identification protocols for large-scale RFID systems.*** **PhD thesis, James Cook University.**

https://doi.org/10.4225/28/5acd7406bf34f

College of Science and Engineering

High Efficient Tag Identification Protocols for Large-scale RFID Systems

A thesis submitted by

Lijuan Zhang, B Eng

for the award of

Doctor of Philosophy

Dec. 2017

Abstract

As one of the most important auto-identification technologies, radio frequency identification (RFID) is the most basic technology used to connect physical objects in support of intelligent decision-making in Internet-of-Things (IoT) networks. Building up the connections between physical objects and virtual networks, RFID systems have been in widespread use in numerous large-scale applications, such as logistics, management and inventory tracking. Since the reader and tags share the same wireless channel in such systems, tag collision occurs when multiple tags reply to the reader simultaneously, which not only increases the identification delay but also wastes bandwidth. In this thesis, three efficient tag identification protocols are proposed to reduce the collisions in three typical application scenarios, i.e., missing tag identification, moving tag recognition and energy-saving of passive systems with a portable reader.

Firstly, with RFID systems being used more and more widely in warehouses and logistics applications, effectively and efficiently identifying missing tags becomes one of the most fundamental tasks, especially for asset management and anti-theft purposes. To identify missing tags, we propose a time-efficient pair-wise collision-resolving missing-tag identification (PCMTI) protocol through designing novel pair-reply and two-collision slot resolving strategies. Compared with previous work, PCMTI can verify two tags in one short response slot simultaneously and identify the tags in all the two-tag collision slots, resulting in less identification time than previous works.

Secondly, in mobile systems with moving tags, many tags move in and out of the system continuously, resulting in limited time for the reader to identify the tags within its reading range. To improve the identification time and reduce the tag-lost ratio of mobile systems, an efficient bit-detecting (EBD) protocol is

proposed. With a new bit monitoring strategy and an M-ary bit-detecting tree recognition method proposed, EBD can effectively verify the known tags using a few number of slots and rapidly identify unknown tags without generating any idle slots. EBD shows better time performance and lower tag-lost ratio than existing protocols reported in the literature.

Thirdly, in passive systems with a portable reader, the energy cost of the reader is caused not only by its own communication operations but also by powering all the tags around it. To prolong the reader's battery life, an M-ary collision tree (MCT) protocol is proposed for time- and energy-saving of the tag identification process. Using the positions of colliding bits, MCT can identify all the tags with fewer collision slots and transmitted message bits, which greatly reduces the time and energy costs.

Through solving the tag identification problem in various applications, this thesis is of great significance and practical use for wider implementation of large-scale RFID systems.

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

________________________ / /

Lijuan Zhang, Candidate

________________________ / /

Prof. Wei Xiang, Primary advisor

________________________ / /

Prof. Ian Atkinson, Secondary advisor

Acknowledgments

The completion of my thesis is attributed to many people's help and encouragement. I would like to express my sincere gratitude to all those people.

I would like to extend my heartfelt gratitude to my principal supervisor Prof Wei Xiang for his instructive guidance and valuable suggestions in my academic studies. He gave me much help and advice during my PhD study and the preparation of this thesis. I am deeply grateful for his help. I gratefully appreciate my secondary advisor Prof Ian Atkinson for his constructive suggestions. I also acknowledge Prof Xiaohu Tang from the Southwest Jiaotong University in China and Prof John Billingsley from the University of Southern Queensland in Australia, for their instructive comments and discussions about my research work.

Special thanks to my beloved family for their continuous support and encouragement. Without their loving consideration and great confidence in me, I would not have been able to reach this far in my study. I am also thankful to my friends and my fellow group members for their discussions and comments about my thesis. I would also thankful to Dr. John Cokley for proofreading my thesis.

LIJUAN ZHANG

James Cook University, Australia

Dec. 2017

List of publications

The following publications were produced during the period of candidature:

Journal Papers

[1] **L. Zhang**, W. Xiang and X. Tang, “An efficient bit-detecting protocol for continuous tag recognition in mobile RFID systems”, accepted by *IEEE Trans. Mobile Computing*, 2017. (ERA ranking 2010: **A***, JCR Impact Factor 2016: **3.822**)

[2] **L. Zhang**, W. Xiang, I. Atkinson, and X. Tang, “A time-efficient pair-wise collision-resolving protocol for missing tag identification”, *IEEE Trans. Communications*, vol. 65, no. 12, Dec. 2017. (ERA ranking 2010: **A***, JCR Impact Factor 2016: **4.058**)

[3] **L. Zhang**, W. Xiang, X. Tang, Q. Li, and Q. Yan, “An efficient Collision Tree Protocol for Energy- and Time-aware Tag Identification in Large-scale RFID Systems”, accepted by *IEEE Trans. Industrial Informatics*, 2017. (ERA ranking 2010: **A**, JCR Impact Factor 2016: **6.764**)

Conference Papers

[1] **L. Zhang** and W. Xiang, “An energy- and time-efficient M-ary detecting tree RFID MAC protocol,” in *Proc. IEEE international conference on communications (ICC)*, London, Jun. 2015, pp. 2882-2887. (ERA ranking 2010: **B**)

Contents

Notation

n	Number of tags to be identified
f	Frame length
t_s	The time of transmitting a 1-bit message
RID	Reader's ID
SC	Session code of a reading cycle
S_x	The x-th reserved slot
R	Hash seed
h	Number of hash operations for identifying n tags in PCMTI
a	Number of hash operations for each 2-collision slot
t_Q	Time taken to transmit the $Query$ command
t_R	Time taken to transmit the $Qrep$ command
$T(n)$	The time for identifying n tags using MCT
$E(n)$	The energy cost for identifying n tags using MCT
PID	Pseudo-identifier
BP	Bit position
$Query$	Frame start command
$Qrep$	Slot start command
$DSG()$	Detecting string generating function
$CNT()$	Bit position calculating function
$\mathcal{C}_{sg}$	Number of singleton slots in each frame
$\mathcal{C}_{cs}$	Number of 2-collision slots in each frame
$\mathcal{T}_{ave}$	Average identification time for verifying one tag
$\mathcal{T}_{ave}^{rd}$	Average time for reader message transmission
$\mathcal{T}_{ave}^{tg}$	Average time for tag message transmission

$H()$	Hash function, note that $H()_k$ refers to the value of the k-th b-it of the hash result
α	Impact factor, which is defined as the ratio of the frame length to the number of tags
t_p	The time of transmitting a one-symbol in tag response
P_d	Detection probability, i.e., the probability that tag's signal can be detected by the reader
P_c	Capture probability, i.e., the probability that capture effect oc-curs
P_e	Transmission bit error probability
Tc	Recognized tag counter maintained by the reader and tags to r-ecord the number of tags identified
Ac	Allocated slot counter maintained by the tags to decide in whi-ch slot the tag will reply to the reader
t_1	Time taken from the reader transmission to a tag response
t_2	Reader response time required if a tag is to demodulate the int-errogator signal
t_3	The time a reader waits after t_1 if there is no tag response
$MT(n)$	Time needed to monitor the presence of n known tags in KTBM
$RT(\beta)$	Recognition time needed to recognize β unknown tags in MBTR

List of Figures

List of Tables

Acronyms

RFID	Radio frequency identification
IoT	Internet-of-Things
CW	Carrier waves
MAC	Multiple access control
SDMA	Space division multiple access
FDMA	Frequency division multiple access
CDMA	Code division multiple access
TDMA	Time division multiple access
FLOPS	Floating-point operations per second
CRC	Cyclic redundancy code
SNR	Signal-to-noise ratio
FP	False positive errors
FN	False negative errors
N	No errors
QT	Query tree
SA	Slotted Aloha
FSA	Frame slotted Aloha
DFSA	Dynamic frame slotted Aloha
TSA	Tree Slotted Aloha
MFML	Multi-frame maximum likelihood
STT	Smart trend tree traversal
ImATSA	Improved assigned tree slotted Aloha
AdATSA	Adaptive assigned tree slotted Aloha
PCMTI	Pairwise collision-resolving missing tag identification
EBD	Efficient bit detecting
MCT	M-ary collision tree

KTBM	Known tag bit monitoring
MBTR	M-ary bit-detecting tree recognition
BSTSA	Binary splitting tree slotted Aloha
SpBTSA	Splitting binary tree slotted Aloha
OQTT	Optimal query tracking tree
OBTT	Optimal binary tracking tree
BEA	Bit estimation algorithm
SRB	Single resolution blocking
PRB	Pair resolution blocking
RBA	Re-blocking algorithm
MUIP	Multi-pairing unknown tag identification protocol
MS	Multi-slotted scheme
MSS	Multi-slotted scheme with selective sleep scheme
MAS	Multi-slotted scheme with assigned slots scheme
ISS-TCA	Identified slot scan-based tag collection algorithm
DQ	Distributed queuing protocol
CwT	Collision window tree protocol
DPPS	Dual prefix probe scheme
TRP	Trust reader protocol
MUIP	Multi-pairing unknown tag identification
PMTI	Physical-layer missing tag identification
THP	Two-hash protocol
MMTI	Multi-hashing based missing-tag identification
SFMTI	Slot filter-based missing tag identification
ProTaR	Probabilistic tag retardation

Chapter 1

Introduction

1.1 Background

With recent developments in automatic identification technology, radio frequency identification (RFID) systems have been in widespread use in numerous large-scale applications to help expedite the progress of automatically identifying and tracking tags attached to objects, such as inventory management, logistics tracking and precision agriculture, etc. Through building up connections between physical objects and virtual networks, RFID is one of the most fundamental techniques used to connect physical objects in support of intelligent decision-making in Internet-of-Things (IoT) networks.

As one of the most important auto-identification technologies which support IoT networks, RFID can efficiently recognise multiple tags without line-of-sight signal propagation. Meanwhile, it also has the advantages of low manufacture costs, easy implementation, long service lifetime, robustness in harsh environments, and the ability of anti-duplication [1, 2]. During the past decade, most retail giants and large farms have implemented their RFID solutions for more efficient management. Moreover, RFID technology, which provides efficient wireless object identification, is envisioned to bridge the physical and virtual worlds. Many large companies have set foot in this area, providing hardware and software solutions as well as contributing to global standardisation efforts [3]. In the past decade, RFID systems have championed and greatly affected our lives in many practical applications. For example, tracking the movements and health of animals via RFID tags has been successfully implemented in large farms. Moreover, implementing RFID tags on products greatly benefits in many aspects, such as real-time asset management, warehouse management and supply chain visibility, etc.

Generally speaking, a typical RFID system consists of a reader, multiple tags and application systems as shown in Fig. 1.1. The tag (or transponder) representing the actual data-carrying device of RFID system is usually affixed to the object to be identified. A reader (or interrogator) is used to collect the data carried on the tags remotely and forward them to the back-end system for further utilization. A reader typically contains a radio frequency module (i.e., the transmitter and receiver), a control unit and a coupling element to the transponder. In passive systems, where the tags do not possess their own voltage supply (battery), the power required to activate the tag is supplied through the coupling unit (contactless), as are the timing pulse and data [1].

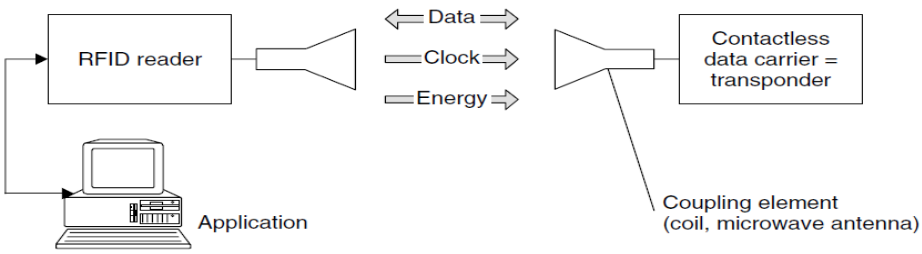


Figure 1.1: Components of an RFID system [1].

In recent research, RFID technology has attracted considerable attention, and a large number of challenging problems have been deeply investigated in the literature, such as tag identification, security and privacy protection, tag positioning and tracing problems etc. Firstly, the tag identification technique focuses on solving the tag collision problem caused by multiple simultaneously transmitted signals. When multiple tags transmit at the same time, the reader cannot correctly decode the received signal. Without correct tag information, the subsequent operations of the RFID system are no longer functional properly. Therefore, tag identification, as one of the most basic and important RFID techniques, is of great importance, especially in large-scale RFID systems, and it is also the research focus of this thesis.

Secondly, in security-concerning systems, such as military and financial applications, information security and users' privacy protection are of great importance. Restricted by the limited computational and storage capabilities of RFID tags, designing simple but efficient security protocols are very challenging. Last but not least, affected by the application environment, traditional satellite, infrared or ultrasonic positioning technologies cannot provide accurate location information of indoor objects. Making use of the signal strength between the reader and tag, RFID technology can help solve the indoor positioning problem. Localisation and tracking is also one of the most investigated research topics of RFID systems.

1.2 Motivations

The first and most important issue of any RFID-based IoT network is to effectively and efficiently identify all the tags within a certain range. In many applications, there are hundreds even thousands of tags within the reader's range and the tags usually do not have the ability to detect channel conditions. Tag collision occurs when multiple tags are trying to send responses to a reader concurrently [4,5]. Tag collision has a significant impact on the performance of RFID systems. For instance, it substantially wastes bandwidth and increases identification delays, especially in large-scale RFID systems. More importantly, if a tag is left unread because of the the weak backscattering signal or the capture effect, it will affect subsequent operations. So efficient RFID tag identification protocols play a critical role in improving the performance of RFID systems.

In general, RFID tags can be divided into passive and active ones. Active tags, which are powered by battery, are more prevalent in applications covering large areas or requiring security protection. However, such tags usually incur high manufacturing costs and have very limited lifetime. Powered through backscattering the reader's signal, passive tags are usually of lower costs and have much longer lifetime. In view of these features, passive tags are preferred in large-scale inventory and logistics applications. In a passive RFID system, tags usually have constrained storage, computation and communication capabilities. Therefore, developing simple and efficient tag identification protocols is very challenging in such systems.

In the past decade, many RFID tag identification protocols have been proposed in the literature. However, the efficiency of existing protocols is low, so that high-efficiency protocols are highly desirable. Moreover, a few existing protocols take into account the influence of fast-moving tags and practical channel conditions, such as the capture effect and channel errors. To implement RFID tag identification protocols in various applications, all kinds of channel conditions and practical issues should be considered. Finally, with the wide-ranging use of portable readers in RFID systems, energy-efficient RFID tag identification protocols are of great importance to prolong the battery life of these readers. However, existing protocols seldom consider the energy consumption at the reader side in passive systems.

Motivated by the importance of the tag identification processes and the deficiency of existing tag identification protocols, this thesis try to improve the time and energy performances of large-scale RFID systems.

1.3 Objectives

The specific objectives of this thesis are as follows.

- For missing tag identification, the objective is to improve the time performance by developing new recognition strategies. Since the average time taken by existing missing tag identification protocols is more than the time needed for transmitting one bit message, this average time can be further reduced.

- For mobile RFID systems with moving tags, the aim is to design a novel identification protocol with the objective of reducing the recognition time for low-mobility mobile systems, and to improve the throughput and robustness of the identification process for high-mobility systems.

- For energy-saving of the portable reader, the objective is to reduce the number of message bits transmitted by the reader and tags, and to develop a novel tree traversal protocol for practical applications.

1.4 Research Problems

This thesis project concentrates on an important but insufficiently investigated problem, i.e., the tag identification problem in large-scale RFID systems. The research problems include: known tag monitoring, tag identification in mobile systems, and energy-saving for passive RFID systems with portable readers.

These research problems are significant due to the following reasons. First, real-time monitoring known tags which have been recognised by the reader, and timely identifying missing tags is a significant issue for management and anti-theft purposes. Any delay in reporting a missing tag could lead to a potential loss of goods. Secondly, in mobile RFID systems, the reader not only needs to effectively monitoring all the known tags but also quickly identifying the unknown ones. Moreover, in high mobility applications, a tag may move out of the reader's reading range without having been recognised. Finally, in passive systems, energy saving for portable readers is a major concern. Consequently, the aforementioned tag identification problems are of great significance in large-scale RFID systems. In what follows, we will elaborate on these problems.

1.4.1 Missing Tag Identification

The missing tag identification problem exists in most warehouse and large farm management applications. Imagine a large warehouse that houses tens of thousands of items of merchandise, such as shoes, apparel, appliances, electronics, etc. One of the most important tasks is to find out whether anything is missing due to theft or management errors [6]. Manually checking the items one by one is unrealistic in most cases. Firstly, it is difficult to discover the presence of some items that are blocked by other items. Secondly, the manual checking process is usually laborious and incurs a long time [7, 8]. If each item is attached with a tag, a fully automated counting procedure assisted by an RFID reader that can be frequently performed will be very helpful.

In RFID systems, a simple method of monitoring missing-tags is to scan all the tags and compare them with the records in the database. However, collecting every tag's ID using this straightforward method is overly time-consuming as the number of tags becomes large. Although a great deal research efforts have been devoted to improve hardware and protocol design to improve our ability to collect the data, as the number of tags increases rapidly, simply collecting all the data is no longer feasible. For instance, Walmart is estimated to generate 7 terabytes of RFID data per day when RFID tags are attached to individual items [9]. So fast approaches for monitoring large numbers of tags are imperative.

Although the missing-tag monitoring problem has attracted extensive attention in both academia and industry, this problem is relatively new and under-investigated. Existing work can hardly satisfy the stringent real-time requirements. To tackle this problem, we proposes a new pair-wise collision-resolving missing-tag identification (PCMTI) protocol for rapid identification of missing tags.

1.4.2 Tag Identification in Mobile RFID Systems

Apart from stationary RFID systems, tags may move along a fixed path in the reader's recognition range in many practical applications, such as baggage processing, retail distribution, correspondence/parcels auto-sorting, and food management [10]. This is also known as mobile RFID systems with moving tags, and the mobility of the tags poses new challenges to the recognition process of current RFID systems.

In low-mobility systems, the reader needs to identify both the known and unknown tags. For known tags, an efficient missing tag identification protocol is very important as discussed before. For unknown tags, since the reader does not have any prior knowledge of the number of tags as well as the tags' ID information prior to the identification process in most applications [11], it can only broadcast the request command to all the tags to collect their ID information. After receiving the reader's request command, all the tags within its reading range will reply. If more than one tag sends messages to the reader simultaneously, a collision occurs as the reader cannot decode any tags message.

The unknown tag identification (also known as tag collision) problem has a significant impact on the performance of RFID systems. For instance, it wastes bandwidth and delays identification. Unlike traditional collision problems in wireless communications, RFID tags are resource constrained. Consequently, traditional anti-collision protocols, such as space division multiple access (SDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA) are not applicable to such environments [3]. Thus, most unknown tag identification protocols are time division multiple access (TDMA)-based methods. However, the performance of existing work is unsatisfactory, especially when the number of tags is large.

In high-mobility RFID systems, the products which are labelled with an RFID tag are usually placed on the conveyor belt, and pass through the reader's coverage area. The faster the speed of the conveyor belt, the higher the probability of the field identifying the products. Since all the tags within the range share the same wireless channel, and the identification of these tags is in a random manner, the tags entering the coverage area earlier may be identified later. Given a high moving speed, the tags have very few time to remain in the reader's recognition range. The limited tag staying time may not be sufficient for the tags to be recognized by the reader. Therefore, an elegant trade-off between rapidity and reliability of the identification process is very important.

In practical applications, it is highly desirable that a mobile RFID system is able to support a high moving speed of tags, while maintaining a high identification rate [10]. However, existing tag identification protocols cannot well support these two features simultaneously. The second contribution of this thesis is that we developed an efficient bit detecting (EBD) protocol to offer a better time performance and to achieve a lower tag-lost ratio than other comparative protocols.

1.4.3 Time- and Energy-saving in Passive RFID Systems

For commercial RFID systems located at industrial premises, readers are typically mounted on static locations and connected to wall-socket power supplies. The throughput of tag reading is important when scanning large-scale tagged items. It is quite common that a user carries a portable reader that is battery operated or rechargeable and scans tags, such as in supermarkets, garages, refrigerators and so on. Using a portable reader to scan tags results in faster depletion of battery energy supply, which requires more frequent recharging or replacement of the reader's battery [12]. In such applications, the tag identification protocol should strike a balanced trade-off between the time of the identification process and the energy consumption of the reader. That is, the reading time should be controlled within acceptable limits while minimizing the energy consumption of the reader.

Besides, as the time required for tag identification increases, the battery consumption on the reader also increase, which is one of the major concerns in passive RFID systems with portable readers. Consequently, in terms of the overall performance of such systems, the efficiency with witch the reader identifies a massive number of tags and the energy-saving on the reader are very significant.

As more and more applications require portable readers, energy has become an important resource to conserve. Energy saving in these systems enables longer operational lifespan of the reader and promotes the adoption of portable reader systems. However, energy consumption has not received serious attention in the literature to date. Existing work is either too time-consuming or energy-consuming. To prolong the portable reader's lifetime and to reduce the battery recharge frequency, a new M-ary collision tree (MCT) protocol in an attempt to reduce both the time and energy costs of the tag identification process is proposed in thesis.

To summarize, in this thesis three protocols are proposed to tackle three different application scenarios, i.e., known tag monitoring, moving tag identification and energy-saving of portable readers. These protocols are capable of substantially improving the performance of the tag identification process and well suited for implementation in practical RFID-based IoT networks. The moving tag identification protocol is an extension of the known tag monitoring work by considering the presence of the unknown tags and the conveyor belt applications. Both PCMTI and EBD focus on the time efficiency of the identification process, without consideration of the energy cost of the portable reader. Finally, the proposed MCT protocol generally considers both the time and energy costs of the identification process, which is more suitable for wide-ranging applications. In this thesis, we will introduce these protocols one-by-one in Chapters 3, 4 and 5.

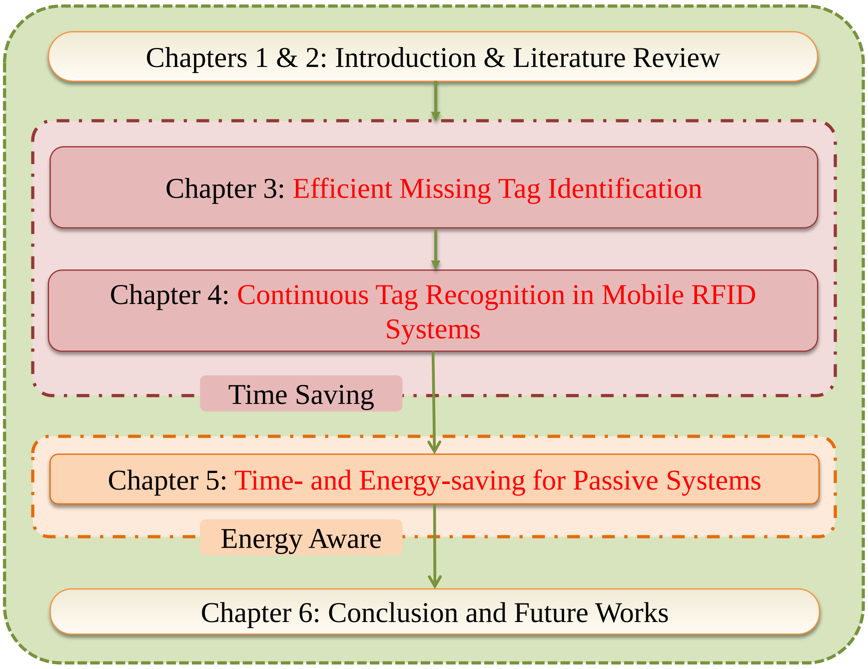


Figure 1.2: Framework of the thesis.

1.5 Organization

In this section, the framework of this thesis is illustrated in Fig. 1.2, followed by an outline of the remaining chapters.

As is shown in Fig. 1.2, the remainder of this thesis begins with a review of existing RFID tag identification protocols in the literature in Chapter 2. Next, Chapter 3 discusses the missing-tag monitoring problem and proposes an effective missing-tag identification protocol. Then, Chapter 4 presents a recognition model for mobile RFID systems and proposes an efficient bit detection protocol in an attempt to solve the moving tag problem. Then, the time and energy models of passive RFID systems alongside a time- and energy-aware tag identification protocol are presented in Chapter 5. Finally, Chapter 6 summarizes this thesis with some comments for future research directions.

Chapter 2

Literature Review

This chapter overviews the state-of-the-art works of RFID tag identification protocols in four aspects, i.e., the missing-tag identification protocols, the unknown tag identification protocols, protocols for moving tag identification, and energy-saving protocols.

2.1 Missing-tag Identification Protocols

In many practical applications, identifying missing tags in real-time is one of the most fundamental tasks for management and anti-theft purposes. In recent research, the missing-tag problem has attracted much attention but been under-investigated in the research community. Existing missing-tag protocols can be generally classified into two categories: probabilistic detecting and precise missing-tag monitoring protocols [6, 8]. Firstly, probabilistic detection protocols concentrate on detecting a missing-tag event or identify missing tags with a certain predefined probability. Such protocols can be scheduled to execute frequently to catch any loss event such as theft. Secondly, the precise missing-tag monitoring, also known as exact detection, protocol focuses on identifying exactly which tags are missing. An exact detection protocol gives much stronger results but its overhead is far greater than a probabilistic detection protocol. Hence, they both have their values and should be used together.

Probabilistic detecting protocols [6–9, 13, 14], including missing-tag event detecting and probabilistic missing-tag identification protocols, aim to detect whether any tags are missing with a certain predefined detecting probability, or identifying a portion of the missing tags, respectively. To efficiently address the missing-tag identification problem, Tan et al. introduced the basic detecting method in [6]. In their follow-up work [8], a Trust Reader Protocol (TRP) is proposed to detect a missing-tag event with a given probability when the number of missing tags exceeds a threshold. In TRP, a missing-tag event is detected if no tag transmits during a slot when there is supposed to be one or more tags transmitting. However, the reader cannot guarantee the detection of a missing-tag event in the situation when a tag is missing and its slot is kept busy by transmission from another tag. To solve this problem, Luo et al. investigated the birthday paradox to detect the missing-tag event [7]. Recently, Zheng *et al.* developed a physical-layer missing-tag identification (P-MTI) protocol to further reduce the detecting time in small-scale applications [15]. However, when the number of missing tags exceeds an estimated maximum value, this method may not identify all the missing tags. Generally speaking, these protocols can quickly detect a missing-tag event or identify a portion of the missing tags. However, they are unable to identify all the missing tags.

Recently, many tag monitoring protocols have been proposed to reduce the transmission message and increase the monitoring efficiency for precise missing-tag identification [15–23]. Benefiting from the ID information of known tags, these protocols usually use hash functions to assign known tags into some predicted slots. Thanks to the IDs stored in the database, the reader knows which tags should reply in each slot. For a predicted singleton slot, where only one tag is expected to reply, the reader confirms the presence of a staying tag if the reader successfully receives the tag's response. Otherwise, this tag is declared as missing.

In [18], Li *et al.* proposed three missing-tag identification protocols. Among them, the most time-efficient protocol is the two-hash protocol (THP), in which two hash processes are implemented to solve part of the collision slots and to make use of the empty slots. In [17], a multi-hashing based missing-tag identification (MMTI) protocol was proposed to improve the utilisation of the time frame used for identification. Using multiple hash process, MMTI turns most collision slots into singleton slots and relocates the extra tags in the collision slots to empty ones. Later, the slot filter-based missing-tag identification (SFMTI) protocol [19] was proposed to turn a k-collision ($k = 2$ or 3) slot (a collision slot with k tags assigned) into a singleton slot, and to filter out the expected empty and other collision slots. SFMTI takes $1.18t_s$ average time for one tag identification, which is the best protocol reported in the literature to date. Recently, Shao *et al.* presented a probabilistic tag retardation (ProTaR) protocol [20], which takes almost the same time as SFMTI. However, generating a pseudo-identifier (PID) in ProTar is a time-consuming process. As such, the authors adopted an off-line method. This still results in a large time gap between two consecutive reading rounds.

In summary, there is still room to improve the performance of existing missing-tag identification protocols for rapid identification of missing tags in large-scale RFID systems.

2.2 Unknown Tag Identification Protocols

Unknown tag identification protocols are also called tag ID collecting protocols. In the past decade, a variety of such protocols have been proposed in the literature, including tree-based [14, 24–34], Aloha-based [35–49] and hybrid protocols [50–56].

In tree-based protocols, colliding tags are recursively split into disjoint subgroups until there is at most one tag in each group. These protocols have the advantage of successfully recognising all the tags even when the number of tags is vast. Their system efficiency, however, is low because tags may experience many collisions at the beginning of identification. The most well-known tree-based protocol is the query tree (QT) protocol. In the QT protocol, the reader firstly broadcasts a query command with a query prefix. When receiving this command, tags that have the same prefix in their ID respond immediately. The QT protocol is very easy to implement; however, its system efficiency is only about 35%. Till now, various advances in the QT protocol have been proposed [24, 28, 29] with improved system efficiency, better than that of the original QT but below 50%. To further improve the identification efficiency, a smart trend tree traversal

(STT) protocol [16] was proposed to reduce the number of collision slots through a well-designed traversal route. Next, the improved assigned tree slotted Aloha (ImATSA) [52] was proposed to quickly identify tags with a new n-tree structure, which shows better time performance than previous works.

Aloha-based protocols can be divided into slotted Aloha (SA), frame slotted Aloha (FSA) and dynamic frame slotted Aloha (DFSA) protocols. In these protocols, a tag responds to the reader at a randomly selected back-off time. For example, in DFSA, the reader uses a dynamic frame structure to identify tags, where time is divided into frames, with each frame consisting of several slots. At the beginning of a frame, the reader informs tags with the frame length F. A tag selects a random number $i \in [0, F-1]$, and replies in the i-th slot. At the end of the frame, the reader estimates the number of colliding tags, then adjusts F accordingly. Aloha-based protocols are also easy to implement but have the well-known “tags starvation” problem, in the sense that some tags may not be identified even after a long period. Among various Aloha-based protocols, the tree slotted Aloha (TSA) protocol [57] has the highest system efficiency of about 37%, peaking at 43%, but has complicated implementation and high identification delay. Recently, a multi-frame maximum likelihood dynamic Aloha-based protocol, (MFML for short) [44], has been proposed. This protocol first uses a maximum likelihood estimator to estimate the number of contenders accurately, then implements an appropriate Q-selection method to maximise the throughput. It shows very good performance in terms of (lower) average identification delay and computational costs in contrast with the previous DFSA protocols.

In [25, 33, 50, 51, 53], some hybrid protocols have been proposed by combining the advantages of tree-based and Aloha-based protocols. Most of them first implement a tree-based procedure or an estimation procedure to obtain an approximate number of tags, then combine a variation of Aloha or tree protocol to reduce the identification delay. In [51], the binary splitting tree slotted Aloha (BSTSA) protocol is proposed to estimate the number of tags with a traditional tree splitting method, and to identify the tags with a TSA algorithm. The estimated tag number greatly helps the identification process of TSA, resulting in high time performance of BSTSA. Later, Wu et al. also proposed a splitting binary tree slotted Aloha (SpBTSA) protocol in [53] with a similar method to accelerate the identification process. However, using the random splitting method, colliding tags may generate the same random number such that all the colliding tags collide again in the next slot in BSTSA and SpBTSA. Recently, Lai *et al.* proposed an optimal query tracking tree (OQTT) protocol which employs a bit estimation algorithm (BEA) to partition tags into small groups, and then uses a query tracking tree to quickly identify tags [25].

To sum up, although the existing tag ID collecting protocols offer high system efficiency, the average identification time needs to be further reduced to support large-scale RFID systems.

2.3 Moving Tag Identification Protocols

In low-mobility RFID systems, such as warehouse management systems, the tags move so slow that the tag set is unchanged during a single reading cycle. To solve the tag identification problem in such systems, some two blocking protocols are proposed to identify the previously identified tags (i.e., known tags) and newly arrived tags (i.e., unknown tags) separately.

Making use of the information of the previously recognised tags stored on the readers memory or database, Myung et al. first proposed two schemes [58,59], the adaptive query splitting algorithm and the adaptive binary splitting algorithm. These algorithms use the ID information obtained from the last reading cycle to reduce the number of collision slots among known tags. However, they fail to prevent collisions between the unknown tags and the known ones. Later, Lai *et al.* propose two types of two-blocking algorithms, i.e., the single resolution blocking (SRB) algorithm and the pair resolution blocking (PRB) algorithm in [60]. In SRB, the reader identifies the known tags without generating any collision slots through recording the recognition indices of all the recognised tags in the preceding reading cycle. To further reduce the number of slots used to recognise the known tags, PRB uses the pair resolution method to confirm two known tags in one slot simultaneously. Compared with SRB, it takes PRB half the number of slots to recognise the known tags. Similarly, Li *et al.* also proposed a new re-blocking algorithm (RBA) [61] to further improve the performance of SRB and PRB in practical scenarios. These algorithms can save some time in the known tag recognition phase. However, they suffer from two drawbacks. On the one hand, if many known tags move out of the reader's reading range, these algorithms will yield many idle slots. On the other hand, the unknown tag recognition process of the algorithms generates many collision slots, which greatly lowers the efficiency.

Recently, Liu *et al.* [62] propose three unknown tag identification protocols. In their work, the reader identifies known tags and labels unknown tags in the first phase. Next, the dynamically framed Aloha protocol is implemented to recognise unknown tags. Among the three proposed protocols, the most efficient is the multi-pairing unknown tag identification (MUIP) protocol. By executing the proposed multiple hash pairing method, and transmitting short messages in each slot, the MUIP protocol takes little time to identify the known tags. However, MUIP suffers from the same drawbacks as the missing-tag identification protocols. Moreover, if there are missing tags in the current reading cycle, Liu's work needs an extra tag monitoring process prior to their proposed protocols to detect and delete missing tags from the database.

In high-mobility RFID systems, such as the conveyor belt systems, the tags quickly pass through the reader's reading range. Some tags may not be identified before they move out of the reader's reading area, resulting in a tag-lost problem. In recent research, very little work considers the impact of fast-moving tags. In [63], Xie *et al.* optimise the frame size of Aloha-based protocols in mobile environments. In [64], Alcaraz *et al.* present a dynamic system model for optimising the configurations of mobile RFID systems. Later, they provide a reader-scheduling method for moving tags in applications with multiple readers [65]. Recently, Zhu *et al.* proposed a grouping method to reduce the identification time [10]. However, these works are based on the Aloha protocols. Their performances are not high enough for fast-moving tag identification. Therefore, a more efficient tag identification protocol is needed for large-scale RFID systems with fast moving tags.

2.4 Energy-saving Protocols

In previous research, there is a little work concentrating on reducing the energy cost of RFID systems [12, 66–75], and most of it focuses on active systems, i.e., they consider the energy cost of the reader and tags separately.

The work on energy-aware RFID tag identification protocols for active systems was pioneered by Namboodiri et al. in [12]. In their work, several energy-aware protocols, including multi-slotted scheme (MS), multi-slotted scheme with selective sleep (MSS) and multi-slotted scheme with assigned slots (MAS), were proposed to minimise the energy consumption of a portable reader by using large frame lengths for each frame. However, their protocols generated many idle slots which increased recognition time. Besides, in each slot, tags needed to transmit the full ID information which increased the number of transmitted message bytes at the tags. Following their work, some efforts have been made for energy saving in active RFID systems. In [66], Klair et al. examined 12 variants of Aloha-based RFID MAC protocols to identify the most energy-efficient one. The analytical model is of great significance but the analysed protocols are of low efficiency. Li et al. presented an energy-efficient estimation protocol in [68]. They only considered the energy consumption of the estimation process, which is part of the recognition process of Aloha-based protocols. The overall energy consumption of the recognition process was not analysed. Zhu et al. also developed an ePath approach to minimise the power consumption of the reader without consideration of the energy consumption on the tags [72].

For active systems, a little work has been done to save tags energy while reducing the time cost of the tag identification process. In [69], an identified slot scan-based tag collection algorithm (ISS-TCA) protocol was proposed to reduce the number of messages transmitted between the reader and tags. However, ISS-TCA generates many collision slots, resulting in an increase in tags response message. Vazquez-Gallego et al. also proposed a distributed queuing (DQ) protocol [70] to reduce energy consumption. Similar to MAS, DQ wasted many idle slots and resulted in long recognition time. Generally speaking, these protocols only consider active systems by calculating the energy consumption on the reader and tag sides separately. In passive systems, the reader not only needs to provide power for its own operations but also to energise all the tags within its reading range. Consequently, these protocols are not suitable for passive RFID systems.

Recently, Landaluce *et al.* proposed a collision window tree (CwT) protocol [74] to save the reader's energy cost in passive RFID systems. Making use of a binary collision tree method and the heuristic bit window strategy, the number of message bits transmitted by the tags was effectively reduced. Compared with previous works, the CwT protocol can reduce both time and energy costs. However, the authors failed to consider the cost of the head message in each query command. According to [27, 76], head information contained in each query command is very important for reader-to-tag communications, and should not be ignored. Needing many slots, CwT requires the use of many query commands, which degrades the time and energy performance. In [27], the dual prefix probe scheme (DPPS), which makes use of consecutive collision bits and a dual response strategy, is proposed. With fewer numbers of slots and reader requests, DPPS is more time and energy-efficient than previous work and transmits fewer number of message bits and takes a shorter time for passive RFID tag identification.

Chapter 3

Time-efficient Pair-wise Collision-resolving Protocol for Missing Tag Identification

3.1 Introduction

Recent developments in radio frequency identification (RFID) technology have induced revolutions in a wide range of application domains, such as warehouse management, inventory tracking, supply chain control and so on. In most RFID applications, effective and efficient identification of missing tags is one of the most fundamental objectives, especially for management and anti-theft purposes. For example, in a large warehouse with thousands of items of goods, the set of tags attached to the items may vary because of management faults or theft [17]. To effectively sense such variations, rapid updates of missing tags to the reader are of great significance. Any delay in reporting a missing tag could lead to a potential loss of goods.

In recent research, many missing tag identification protocols have been proposed in the literature [15–23]. These protocols usually use a hash method to assign tags to different slots and verify the presenting tags with only 1-bit messages transmitted in each slot. In [18], Li *et al.* propose a two-hash protocol (THP) to identify a tag with $1.9t_s$ time, where t_s refers to the time of transmitting a 1-bit message. Later, a multi-hashing based missing tag identification

(MMTI) protocol [17], which takes $1.46t_s$ time to verify a tag, is proposed. Recently, the proposed slot filter-based missing tag identification (SFMTI) [19] and probabilistic tag retardation (RproTaR) [20] protocols achieved $1.18t_s$ time for one tag identification, much better than other competing protocols reported in the literature. However, none of the reported protocols in the literature so far can achieve identification time less than t_s. This work will try to achieve this goal.

In this work, a novel two-tag collision-slot resolving protocol is proposed. This is a new cross-layer design, which controls tag responses in the MAC (multiple access control) layer and resolves two-tag collision slots in the physical layer. In MAC layer, a new hash method is developed to distinguish the tags in each two-collision slot, and a pair-reply strategy is proposed to allow two tags to reply simultaneously. In the physical layer, Manchester coding is used to distinguish signals from three cases, i.e., two tags reply, one tag replies and no tag replies. With the decoded message, the reader identifies two tags simultaneously in each slot. More specifically, the major contributions of this paper are fourfold:

1) A novel pair-reply strategy is proposed to verify two tags in one response slot simultaneously. With the new pair-reply strategy, the time cost for tag message transmission is greatly reduced.

2) A new two-collision slot resolving strategy is proposed to increase the number of tags that can be verified in each frame. The new strategy can resolve all two-collision slots, further reducing the identification time.

3) The effect of the impact factor, which is defined as the ratio of the frame length to the number of tags, on the average identification time of the proposed PCMTI protocol is analysed. The existence of only a single optimal impact factor that minimises the average identification time of PCMTI is demonstrated. The optimal impact factor is derived.

4) A theoretical analysis is presented to demonstrate that the proposed PCMTI protocol takes only $0.825t_s$ on average to verify one tag. Compared with the best protocol reported in the literature, i.e., the slot filter-based missing tag

identification (SFMTI) protocol [19], the reduction in identification time is close to 30%.

The remainder of this chapter is organized as follows. Section 3.2 demonstrates the system model and the link-timing of this paper. Section 3.3 describes the proposed PCMTI protocol in detail. Next, a theoretical analysis is presented to evaluate the performance of the proposed protocol and to optimize the parameter settings in Section 3.4. Comparative simulation results are given in Section 3.5, followed by concluding remarks in Section 3.6.

3.2 System Configurations

In this section, we first introduce the system model used in the paper. Then, the link-timing between the reader and tags and some assumptions are described.

3.2.1 System Model

Considering a practical inventory system, tags attached to the items are placed on different shelves. A group of tags located on one shelf (or several neighboring shelves) is monitored by a fixed reader periodically. When new objects move into the system, the tag IDs and their corresponding shelve numbers are collected and stored in a back-end database through implementing some traditional tag ID collecting protocols [14, 24, 52, 77]. Given the IDs and the shelve numbers of all the tags, the reader can periodically monitor the statuses of these tags, i.e., detecting whether these tags are still within the range.

The system model, which is composed of a single reader, a large number of passive tags, and a back-end database, is illustrated in Fig. 3.1. In the system, each tag has a unique 96-bit ID and powered via an RF signal from the reader. The reader has access to the back-end database which records the tag IDs and their corresponding shelf numbers. With this information, the reader knows all the known tags, i.e, the staying and missing tags. Note that there are also some unknown tags which are misplaced. To effectively verify the known tags, the known tags should be separated from their unknown counterparts.

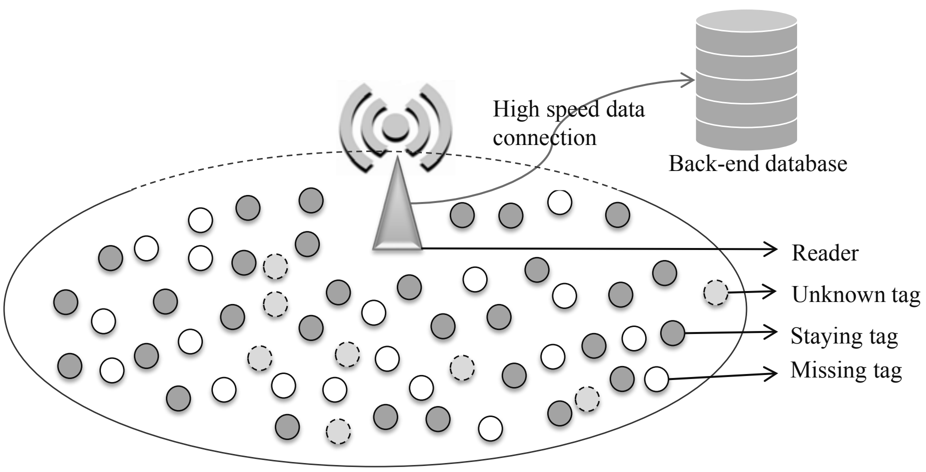


Figure 3.1: System model of a single reader RFID system, which consists of a reader, a large number of passive tags, and a back-end database.

A *reading round* is defined as the reading process in which the reader identifies all the tags within its reading range. A tag that stays within the reader's reading range for two consecutive reading rounds is said to be a *staying tag*. A tag that appeared in the preceding reading round but moves out of the reader's range in the current reading round is referred to as a *missing tag*. The IDs of both staying and missing tags can be retrieved from the database at the beginning of each reading round. To be more practical, we also consider the existence of unknown tags. Although the reader can collect the IDs and the shelf numbers of newly arrived tags, some misplaced tags from other shelves may appear around the reader [14]. The *objective* of this work is to design a highly time-efficient protocol for missing tag identification.

3.2.2 Link-timing

Since RFID tags are of limited computation and communications resources, the reader and tags usually adopt the reader-talk-first mode [17, 77]. That is, the tags reply to the reader according to the reader's request message. The communications between the reader and tags are carried out with the frame-slotted time structure [19, 77], where time is divided into frames, and each frame is composed of multiple synchronized time slots. A frame starts with the reader's $Query()$ command, which informs the tags of the frame length, hash seeds and other information needed in the current frame. After receiving the $Query()$ command, each tag replies a short response message in its corresponding slot. Each slot starts with the *QueryRep* command, except for the first slot in the frame, because the first slot starts automatically after the *Query*() command.

3.3 Proposed Pair-wise Collision-resolving Missing Tag Identification Protocol

In this section, we describe the proposed pair-wise collision-resolving missing tag identification (PCMTI) protocol in detail. In the protocol, two novel strategies are proposed, i.e., the pair-reply and two-collision slot resolving strategies. The pair-reply strategy can verify two tags in one short response slot, which greatly reduces the time costs for tag message transmission. Moreover, the two-collision slot resolving strategy resolves all the two-collision slots, which further reduces the identification time. In this section, we first introduce a simple idea to separate the known tags from the unknown ones. Then we demonstrate the basic method for missing tag identification, followed by a detailed elaboration of the proposed two strategies. Finally, the entire identification process of the proposed PCMTI protocol is given.

3.3.1 Separating Known Tags from Unknown Tags

In most existing missing tag identification protocols [17–20], it is assumed that there are no unknown tags in the reader's reading range. The reader knows all the tags' IDs. This assumption is not impractical. Shahzad *et al.* [14] illustrated several practical situations that unexpected tags whose IDs are unknown often exist in many applications. Although an unexpected tag estimation method was proposed to reduce the impact of unknown tags, Shahzad's method is a probabilistic method which is unable to eliminate all the unknown tags.

To separate the known tags from the unknown ones, the reader's unique ID and the session code of each reading cycle is used. Denote by RID_p and SC_p the reader ID and session code of the preceding reading cycle. RID_c and SC_c are the reader ID and session code of the current reading cycle. In each reading cycle, RID_p, SC_p, and the ID of all the identified tags are recorded in the back-end database. The identified tags also keep RID_p and SC_p in their inner memories.

At the beginning of a new reading cycle, the reader obtains RID_p, SC_p and all the tag IDs from the database, and broadcasts $Ini(RID_c, RID_p, SC_p)$ to initiate all the tags within its reading range. After receiving this command, a tag having the same values with RID_p and SC_p marks itself as a known one, and updates the reader ID and session code with RID_c and $SC_p + 1$ in its memory. Other tags mark themselves as unknown tags and will keep silent in the current reading cycle. At the end of each reading cycle, the reader also updates the values of the reader ID and session code in the database. Since RID_p and SC_p are obtained from the database, these values are the same with those in the known tags' memories even when multiple moving readers are used.

3.3.2 Basic Solution for Missing Tag Identification

After all the unknown tags are separated, the reader can retrieve the information of all the known tags from the preceding reading round, including their number and IDs. Instead of transmitting a tag's ID repeatedly, the basic solution is to use a hash function to assign the known tags to slots as in [17–20]. For example, a tag with ID_i is assigned to a reserved slot S_x,

$$x = \big(H(ID_i||R) \ mod \ f\big) + 1, \tag{3.1}$$

where x is the slot index, $H()$ is the hash function, R is the selected hash seed, and f is the frame length.

According to the number of tags assigned in each slot, the following three types of slots are defined.

- *Reserved empty slots*: reserved slots with no tags assigned;
- *Reserved singleton slots*: reserved slots with only one tag assigned;
- *Reserved collision slots*: reserved slots with more than one tags assigned. Specifically, a reserved k-collision slot refers to a slot with k exact tags assigned.

At the beginning of each frame, the reader broadcasts a query message to inform tags with $H()$, R and f. With the common knowledge of $H()$, R, f and

ID_i at the reader and tags, the reader can verify the tags with only 1-bit short response message transmitted in the reserved singleton slots. This helps greatly save time for tag ID transmission. However, this method also yields many collision slots.

To improve reading efficiency, a variety of protocols [17–20] have been proposed to reduce the number of reserved collision slots or to transfer some reserved collision slots into singleton slots. However, the average time of such protocols for one tag identification is larger than t_s. Moreover, these protocols can only make use of the reserved singleton slots and part of the two-collision slots. To further improve performance, this paper proposes two novel strategies, i.e., the pair-reply and two-collision slot resolving strategies, to reduce the time costs for tag message transmission and to resolve all reserved two-collision slots, respectively.

3.3.3 Proposed Pair-reply Strategy

To accelerate the recognition process, a new pair-reply strategy based on Manchester coding is proposed. Attributable to its capability of detecting the positions of colliding bits, Manchester coding has been widely used in a great number of RFID tag identification protocols to accelerate the recognition process [25,27,74], and both ISO/IEC 18000-6 and ISO/IEC 14443 standards also specify this coding method in RFID systems [76]. In Manchester coding, a logic 0 or 1 symbol is encoded by a positive and negative transition, respectively [2]. If two (or more) transponders simultaneously transmit symbols of different values, the positive and negative transitions of the received symbols cancel each other out, resulting in an invalid symbol.

The new pair-reply strategy is designed based on the Manchester coding. In the pair-reply strategy, two tags, which are assigned to two consecutive singleton slots, are expected to reply with two different messages ms_0 and ms_1 simultaneously. In this work, symbols 0 and 1 are used in the tag response messages. Tags use ASK to modulate messages into physical-layer symbols as in [15, 78]. With Manchester coding, the reader is able to distinguish the scenarios, where only one tag replies, two tags reply simultaneously, and no tag replies according to the decoded message.

More specifically, in PCMTI two tags, e.g., tags A and B, are expected to reply with symbols 0 and 1 in the same slot, respectively. With Manchester coding, symbols 0 and 1 are encoded with bits "01" and "10", respectively. If only tag A (or B) replies, the received message can be correctly decoded as 0 (or 1). With the decoded symbol 0 (or 1), the reader determines that tag A (or B) is a staying tag, while tag B (or A) is a missing one. If the two tags reply simultaneously, the decoded message becomes an invalid symbol. From this message, the reader determines that both tags are staying tags. If the reader receives no message, the reader determines that both tags are missing. To clearly show the received signals under these four situations, Fig. 3.2 illustrates the received waveforms at the reader side with ASK modulation and Manchester coding. It should be noted that only two tags are allowed to reply in each slot. The reader determines whether there are missing tags according to the received message.

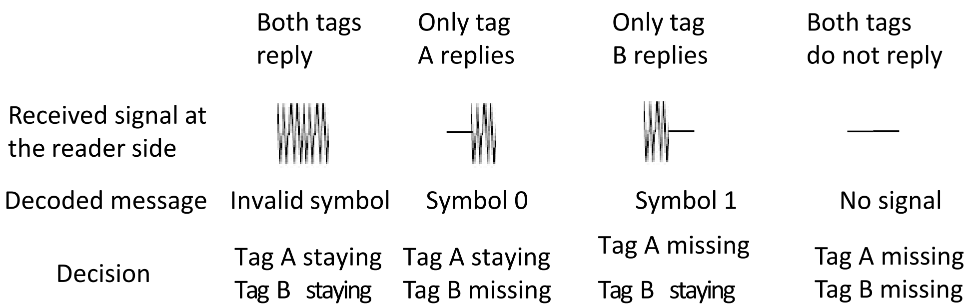


Figure 3.2: Received signals at the reader side with Manchester coding and ASK modulation.

3.3.4 Proposed 2-Collision Slots-resolving Strategy

Since the reader knows which tags are assigned to the reserved two-collision slots, the pair-reply strategy can also be used to verify the tags in the reserved two-collision slots. However, with the hash function of (3.1), the assigned tags only know in which slot they should reply. If the tags assigned to a reserved two-collision slot randomly select ms_0 or ms_1 to reply, the reader cannot determine from which tag the received message is. Then the reserved two-collision slots still cannot be resolved.

In this work, a new two-collision slot resolving strategy is proposed. For every reserved two-collision slot, a second hash seed is selected and transmitted to distinguish the two tags assigned. More specifically, the second hash seed R' is selected to enable $H(ID_A||R')_k = 0$ and $H(ID_B||R')_k = 1$, where ID_A and ID_B are the tag IDs assigned to the reserved two-collision slot, $H()_k$ is the value of the k-th bit of the hash result. Instead of using the *mod* function, using the value of one bit in the hash result can help simplify the tag's operation. With the hash seeds R and R', both the reader and tags know that tag A should reply ms_0 and tag B should reply ms_1, simultaneously.

Although some previous work also tried to resolve the two-collision slots, their methods need at least two hash functions [18] or need to store a long hash seeds ring in tags' memory [17, 19], resulting in additional hardware complexity and memory costs at the tags. Besides, these existing works cannot resolve all the two-collision slots. By contrast, the protocols proposed in this work can fully resolve all the two-collision slots without requiring extra hardware or memory at the tags. The only cost is to select and transmit the hash seeds. In what follows, the overhead of selecting the second hash seed for each two-collision slot is analysed. The time costs for transmitting the hash seeds will be given in Sections 3.4 and 3.5.

Overhead of the Second Hash Seeds Selecting Operation:

Since selecting a second hash seed for every reserved two-collision slot needs extra reader operations, Table 3.1 lists the number of hash selection operations needed to distinguish the tags in a reserved two-collision slot. In the test, the tag IDs in the reserved two-collision slot are randomly generated. The lengths of the IDs and the hash seeds are 128 bits and 8 bits, respectively. The results are averaged over 10,000 trials. The test hash functions uses the most popular hash functions, such as MD5 [79], SHA-1 [80], and SHA-256 [81].

From Table 3.1, the following conclusions can be drawn for all the three types of hash functions.

- The probability that the tags in a two-collision slot can be separated within two hash seed selection operations is about 60%. The probability that such tags can be separated with no more than four operations is more than 80%. This means that the tags in more than half of the two-collision slots can

Table 3.1: Number of hash seeds selecting operations needed to distinguish the tags in a two-collision slot

Hash functions		MD5	SHA-1	SHA-256
Percentage of t operations for obtaining an appropriate hash seed for a two-collision slot, $t=$	1	37.78%	37.62%	37.55%
	2	23.49%	23.46%	23.46%
	3	14.62%	14.68%	14.66%
	4	9.05%	9.29%	9.03%
Maximum number of hash operations		27	23	24
Average number of hash operations		2.654	2.655	2.668

be distinguished with two or less hash operations. Most of them can be distinguished with around four hash operations.

- The maximum number of operations is 27 for MD5, 24 for SHA-1, and 24 for SHA-256. The worst case to distinguish the tags in a two-collision slot is no more than 30 hash operations. And the average number of operations is about three.

Next, the computational complexity of hash operations is briefly analysed. In each frame, the reader has to execute a hash operation for each un-verified tag and several hash operations for each 2-collision slot. Suppose the number of hash operations for each 2-collision slot is a. The total number of hash operations for identifying n tags is obtained as follows.

$$\begin{aligned} h &\leq n\frac{n+a\mathcal{C}_{cs}}{\mathcal{C}_{sg}+2\mathcal{C}_{cs}} \\ &= \frac{n+\frac{an(n-1)}{2f}\left(1-\frac{1}{f}\right)^{n-2}}{\left(1-\frac{1}{f}\right)^{n-1}+\frac{(n-1)}{f}\left(1-\frac{1}{f}\right)^{n-2}}, \end{aligned} \tag{3.2}$$

where $\mathcal{C}_{sg}$ and $\mathcal{C}_{cs}$ are the numbers of singleton and 2-collision slots in each frame, respectively. Let $n=f$ and $n\to\infty$, we have $h\leq\frac{(2e+a)n}{4}$, where e is the Euler's number. Then, we know that the number of hash operations increases linearly with the number of tags. Since the reader is usually connected with a high performance back-end computer, the time cost for the hash operations is very small. Therefore, an off-line mechanism can be adopted to calculate the hash operations as in [15, 20]. In summary, the second hash seed selection for each two-collision slot is feasible. With a high performance back-end computer or an off-line mechanism, the overhead due to the reader's operations will not affect the performance of the proposed PCMTI protocol.

3.3.5 General Identification Process of the Proposed PCMTI Protocol

In PCMTI, the communications between the reader and tags follow a frame-slotted manner, where time is divided into frames and each frame consists of several slots. Generally speaking, each frame in PCMTI consists of two stages, i.e., the pre-assigning and tag confirming stages. The pre-assigning stage is executed at the reader side at the beginning of each frame. In this stage, the reader assigns the tags to reserved slots, and selects the second hash seed for each reserved two-collision slot. The interactive communications between the reader and tags start from the tag confirming stage. In this stage, the tags reply to the reader in a pair-wise manner. The detailed identification process is described in the following.

Stage 1: Pre-assigning stage

Firstly, the reader calculates the reserved slot index for each known tag using equation (3.1). The hash seed R is a random number, and the frame length f is set according to the number of unconfirmed known tags. Next the reader selects the second hash seed for each reserved two-collision slot.

For example in Fig. 3.3, there are 10 known tags $T_i,\ i \in [1-10]$. In the first stage, the tags are assigned to 8 reserved slots according to equation (3.1). Slots S'_1, S'_3, and S'_6 are reserved singleton slots, while slots S'_5 and S'_8 are reserved two-collision slots. To resolve the two-collision slots S'_5 and S'_8, two extra hash seeds RS_1 and RS_2 are selected, respectively. In the example, the value of the 3rd bit of the hash results are used to separate the tags in the two-collision slots.

To inform the tags of the status of each slot, a $2f$-bit mask M is constructed. In the mask, the values of every two bits represent the status of a reserved slot.

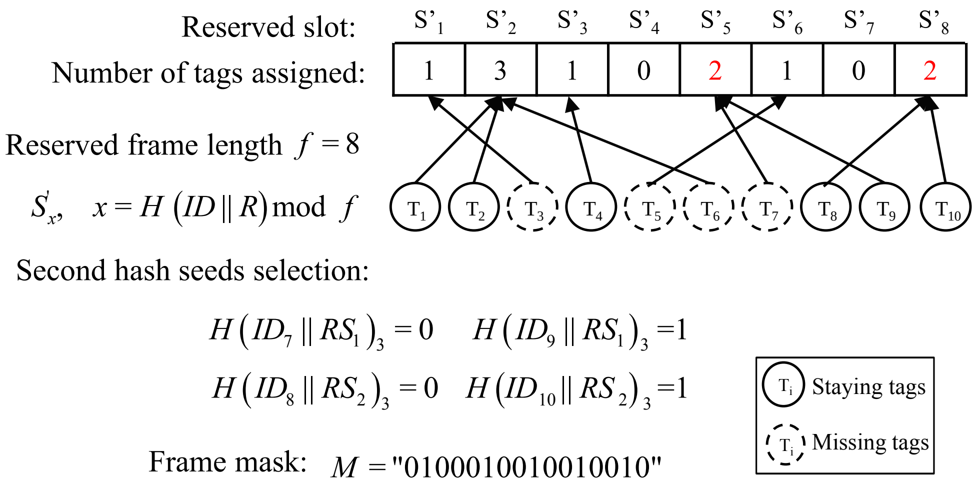


Figure 3.3: An example of the pre-assigning stage of PCMTI.

More specifically, if S'_x is a reserved singleton slot, the reader sets $M(2x-1,\ 2x) =$ "01". If it is a reserved two-collision slot, $M(2x-1,\ 2x) =$ "10"[1]. Otherwise, $M(2x-1,\ 2x) =$ "00". In Fig. 3.3, the mask $M =$ "01 00 01 00 10 01 00 10". In detail, Algorithm 1 gives the pseudo-codes of reader's operations in the pre-assigning phase.

Stage 2: Tag confirming stage

To implement the pair-reply strategy, the reader needs to rearrange the response slots in the second stage. Let the jth reserved singleton and two-collision slots in the first stage by S_j^{sg} and S_j^{2cs}, respectively. Denote by G and C the numbers of the reserved singleton and two-collision slots, respectively. Then the number of slots in the second stage is set to be $\lceil G/2 \rceil + C$.

- The tags assigned to S_i^{sg} and S_{i+1}^{sg}, $i \in \{2,\ 4, ...,\ \lceil G/2 \rceil\}$, are expected to reply in the $(i/2)$th slot with ms_0 and ms_1, respectively;
- The tags assigned to S_j^{2cs}, $j \in \{1,\ 2, ...,\ C\}$, are expected to reply in the $(\lceil G/2 \rceil + j)$th slot. The response message of the tags is decided by RS_j as mentioned in Section 3.3.4.

[1]Note that using "01" and "10" to represent the reserved singleton and two-collision slots can simplify the tags' operations. At the tag side, they can calculate the numbers of reserved singleton and two-collision slots by counting the number of "1"s in the odd and even bits, respectively.

Algorithm 1 PCMTI pseudocode at the reader side –Pre-assigning phase

Set $sg = 0$, $cs = 0$, $M = zeros(1, 2f)$, and choose a random hash seed R;
for $i = 1 \rightarrow n$ **do**
  if $tag(i).verified$ ==FALSE **then** Calculate $x = [H(tag(i).ID||R) \mod f] + 1$, and set $slot(x).t = slot(x).t + 1$; $slot(x).tg = QueueIn(slot(x).tg, i)$;
  end if
end for /* Assign slots for all the un-verified known tags */
for $x = 1 \rightarrow f$ **do**
  if $slot(x).t == 1$ **then**
    Set $sg = sg + 1$; $M(2x, 2x-1) =$‘01’; $A = QueueOut(slot(x).tg)$; $t_{sg} = QueueIn(t_{sg},\ A)$;
  else if $slot(x).t == 2$ **then**
    Set $cs = cs + 1$; $M(2x, 2x-1) =$‘10’; $A = QueueOut(slot(x).tg)$; $B = QueueOut(slot(x).tg)$;
    Choose a random hash seed RS', so that $H(tag(A).ID||RS')_3 \neq H(tag(B).ID||RS')_3$;
    Set $RS = QueueIn(RS, RS')$;
    if $H(tag(A).ID||RS')_3 == 0$ **then** $t_{cs} = QueueIn(t_{cs},\ A,\ B)$;
    else $t_{cs} = QueueIn(t_{cs},\ B,\ A)$.
    end if
  else Set $M(2x, 2x-1) =$‘00’;
  end if
end for /* Organize the frame parameter and record the tags in the singleton and 2-collision slots. */

For example, Fig. 3.4 illustrates an example of the slot re-arrangement process and the tag confirming stage. According to the frame mask M given in Fig. 3.3, both the reader and tags know that S'_1, S'_3, and S'_6 are the reserved singleton slots S_1^{sg}, S_2^{sg}, and S_3^{sg}, respectively. S'_5 and S'_8 are the reserved two-collision slots S_1^{2cs} and S_2^{2cs}, respectively.

With pair-reply, the tags assigned to S_1^{sg} and S_2^{sg}, i.e., tags T_3 and T_4, are expected to reply ms_0 and ms_1, respectively, in slot S_1. Tag T_5, which is assigned to the last reserved singleton slot, is expected to reply ms_0 in slot S_2. With only three reserved singleton slots, the confirmation of the tags assigned to the reserved

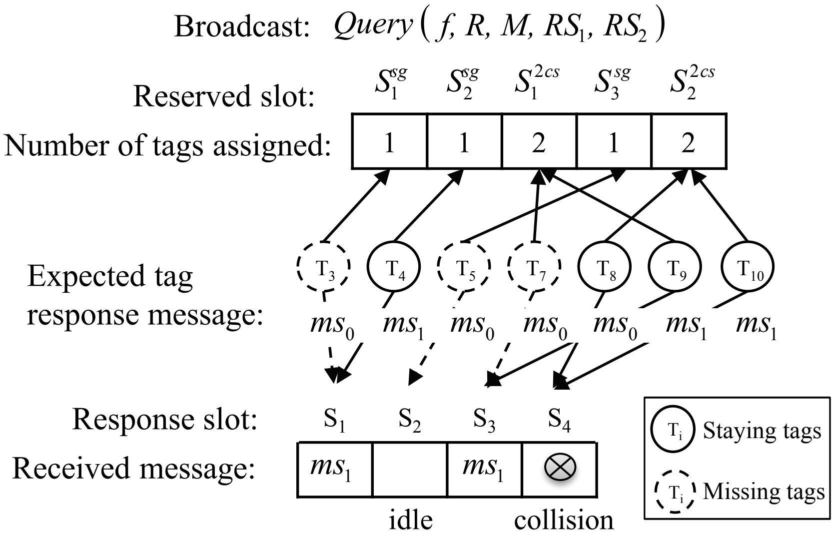


Figure 3.4: An example of the slot re-arrangement process and the tag confirming stage of PCMTI.

two-collision slots starts in S_3 and S_4. Tags T_7 and T_9, which is assigned to the reserved two-collision slots S_1^{2cs}, are expected to reply ms_0 and ms_1 respectively in slot S_3. Similarly, tags T_8 and T_{10} are expected to reply ms_0 and ms_1 in S_4, respectively.

After slot re-arrangement, the reader broadcasts $Query\big(f,\ R,\ M,\ RS_j\ (1 \leq j \leq TR)\big)$ to inform tags of the length of the reserved frame, the first hash seed, the frame mask, and the second hash seeds of all the reserved two-collision slots. Note that TR is the number of two-collision slots in the current frame. After receiving the $Query()$ message, tag T_i first computes the reserved slot index $x = \big(H(ID_i,\ R) \bmod f\big) + 1$. Then it checks the values of $M(2x-1,\ 2x)$:

- If $M(2x-1,\ 2x) =$ "00", the tag will remain silent until the next frame;

- If $M(2x-1,\ 2x) =$ "01", the tag counts the number of "1"s in the even bits of $M(j),\ j \in [1,\ 2x]$. Denote by y the calculated value. If $y \bmod 2 = 0$, the tag will reply ms_1 in slot $S_{\lceil y/2 \rceil}$. Otherwise, it will reply ms_0 in slot $S_{\lceil y/2 \rceil}$; and

- If $M(2x-1,\ 2x) =$ "10", the tag first calculates the number of reserved singleton slots G by counting the number of "1"s in the even bits in $M(j),\ j \in [1,\ 2f]$. Then the tag calculates the number of "1"s in the odd bits in

$M(j),\ j \in [1,\ 2x]$. Denote by y the calculated value. If $H(ID_j,\ RS_y)_k = 0$, the tag will reply ms_0 in slot $S_{\lceil G/2 \rceil + y}$. Otherwise, it will reply ms_1 in the same slot.

Algorithm 2 PCMTI pseudocode at the reader side –Tag confirming phase

Broadcast $Query(f, R, M, RS)$
for $x = 1 \to \lceil \frac{sg}{2} \rceil + cs$ **do**
  if $x \neq 1$ **then** Broadcast $Qrep$;
  end if
  if $x \leq \lceil \frac{sg}{2} \rceil$ **then** $A = QueueOut(t_{sg})$; $B = QueueOut(t_{sg})$;
  else $A = QueueOut(t_{cs})$; $B = QueueOut(t_{cs})$;
  end if
  Waiting for tags' responses;
  if Receiving an invalid symbol **then** Set $tag(A).state$ =‘staying’; $tag(B).state$ =‘staying’;
  else if Receiving symbol ‘0’ **then** Set $tag(A).state$ =‘staying’; $tag(B).state$ =‘missing’;
  else if Receiving symbol ‘1’ **then** Set $tag(A).state$ =‘missing’; $tag(B).state$ =‘staying’;
  else if Receiving no signal **then** Set $tag(A).state$ =‘missing’; $tag(B).state$ =‘missing’.
  end if
  Set $tag(A).verified$ =TRUE; $tag(B).verified$ =TRUE;
end for
Repeat the pre-assigning and tag confirming phases until all the tags are verified.

During the second stage, the reader first confirms the tags assigned to the reserved singleton slots in slot $S_{1,2,\ldots,\lceil G/2 \rceil}$. Then the reader verifies the tags assigned to the reserved two-collision slots in slot $S_{\lceil G/2 \rceil+1,\ldots,\lceil G/2 \rceil+C}$. For example in Fig. 3.4, the reader receives ms_1 in slot S_1. It confirms that T_4 is a staying tag and T_3 is a missing tag. Similarly, the reader verifies tags T_5, T_7, T_8, T_9, T_{10} in the remaining slots. At the end of each frame, all the confirmed tags remain silent until the next reading round. The reader continues the recognition process in the following frames until all the tags are confirmed. In detail, Algorithms 2 and 3 give the psudo-codes of the reader's operations in the tag confirming phase and the tags' operations, separately.

Algorithm 3 PCMTI pseudocode of tag's operations

```
Initialize: All the known tags set tag.ss = CSS; tag.RID = RRID; tag.state = NULL; tag.verified = FALSE
After receiving the reader's Query(f, R, M, RS) command, a tag first checks
if tag.verified = FALSE then  Calculate x̄ = [H(tag(i).ID||R) mod f] + 1;
    if M(2x̄, 2x̄ − 1) ==‘00’ then  Keep silent until the next frame;
    else if M(2x̄, 2x̄ − 1) ==‘01’ then  Calculate the number of 1s before the 2x̄-th bit position in the odd bits of M and record as x;
        if x̄ mod 2 == 0 then  Reply symbol ‘0 in the ⌈x/2⌉-th slot;
        else  Reply symbol ‘1’ in the x-th slot;
        end if
    else if M(2x̄, 2x̄ − 1) ==‘10’ then  Calculate the number of 1s before the 2x̄-th bit position in the even bits of M and record as x;
        Calculate the number of 1s in all the odd bits in M and record it as sg;
        Obtain the second hash seed RS′ from the x-th hash seeds in RS;
        if H(tag.ID||RS′)_3 == 0 then  Reply symbol ‘0’ in the (⌈sg/2⌉ + x)-th slot;
        else  Reply symbol ‘1’ in the (⌈sg/2⌉ + x)-th slot.
        end if
    end if
else  Keep silent until the next frame.
end if
```

Similar to most missing tag identification protocols [7, 8, 17–21], the reader needs to pre-assign tags to slots, and broadcasts a frame mask to inform the tags of the states of each slot. However, the main contributions of this work are the two novel strategies, i.e., the pair-reply and two-collision slots resolving strategies, designed in the identification process. These two new strategies can not only reduce the number of tag response slots, but also resolve all the two-collision slots. More importantly, with the proposed PCMTI protocol, the identification time is greatly reduced.

It should be noted that only the reserved two-collision slots are resolved in the proposed PCMTI protocol. This work can also be extended to resolve k-collision

($k > 2$) slots with more symbols transmitted in tag responses. Take the case of 3-collision slot as an example. Suppose tags A, B, and C are assigned in the same slot. Let tags A, B, and C simultaneously reply symbols 001, 010, and 100, respectively. At the reader side, the received signals of different situations can be illustrated in Fig. 3.5.

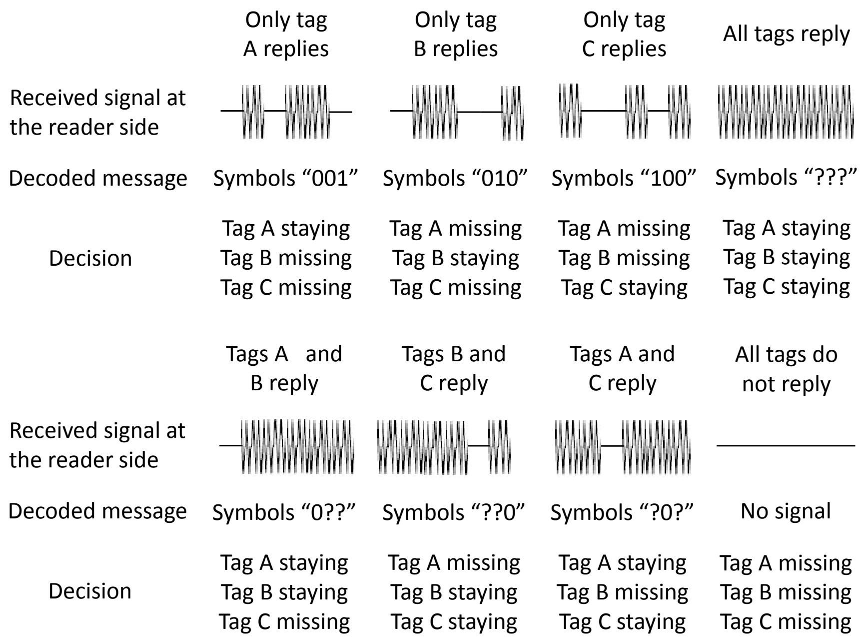


Figure 3.5: Received signals and reader decisions in a 3-collision slot.

As can be observed from Fig. 3.5, the eight cases of received signals can uniquely represent eight different tag response situations. From the received signal, the reader can effectively determine the states of three tags in one slot. However, the solving of k-collision slots ($k > 2$) requires more symbols transmitted in tag responses and more hash operations in the pre-assigning stage. Therefore, only the 2-collision slots are resolved in the proposed PCMTI protocol.

3.4 Performance Analysis

In this section, we first analyse the time performance of the proposed PCMTI protocol under the perfect channel condition. Then, we investigate the effect of channel unreliability on our proposed protocol.

3.4.1 Time Performance of the Proposed PCMTI Protocol

The average identification time $\mathcal{T}_{ave}$ for verifying one tag consists of two parts, i.e., the average time for reader message transmission $\mathcal{T}_{ave}^{rd}$ and the average time of tag response transmission $\mathcal{T}_{ave}^{tg}$. That is,

$$\mathcal{T}_{ave} = \mathcal{T}_{ave}^{rd} + \mathcal{T}_{ave}^{tg}. \tag{3.3}$$

In this section, we first analyse the average time costs at the reader and tags separately. Then the overall identification time is optimized by obtaining an appropriate impact factor, which is defined as the ratio of the frame length to the number of tags. In PCMTI, the frames are independent from each other and have the same reading process. Therefore, the performance of the proposed PCMTI protocol is analysed in a single frame $\mathcal{F}_i$ as per most Aloha-based protocols [2,76]. Denote the number of tags to be verified in the ith frame by $\mathcal{N}_i$, and the reserved frame length by f_i. According to its definition, the impact factor $\alpha = \frac{f_i}{\mathcal{N}_i}$.

In the frame, the reserved singleton slots are re-arranged into pair-wise slots, and the reserved two-collision slots are executed after the pair-wise slots. Other types of slots are eliminated in the interactive communications. Denote by t_p the time of transmitting a one-symbol in tag response. The time of tag message transmission $\mathcal{T}_i^{\text{tg}}$ is given as

$$\mathcal{T}_i^{\text{tg}} = \left(\left\lceil \frac{\mathcal{C}_i^{\text{sg}}}{2} \right\rceil + \mathcal{C}_i^{\text{2cs}} \right) t_p, \tag{3.4}$$

where $\mathcal{C}_i^{\text{sg}}$ and $\mathcal{C}_i^{\text{2cs}}$ are the numbers of reserved singleton and 2-collision slots in frame $\mathcal{F}_i$, respectively.

Since the number of tags verified in the frame equals $\mathcal{C}_i^{\text{sg}} + 2\mathcal{C}_i^{\text{2cs}}$, the average time for tag message transmission is given by

$$\mathcal{T}_{\text{ave}}^{\text{tg}} = \frac{\left(\left\lceil \frac{\mathcal{C}_i^{\text{sg}}}{2} \right\rceil + \mathcal{C}_i^{\text{2cs}} \right) t_p}{\mathcal{C}_i^{\text{sg}} + 2\mathcal{C}_i^{\text{2cs}}} \approx \frac{t_p}{2}. \tag{3.5}$$

This shows that the average tag message transmission time is constant, i.e., about half time of a short response slot.

At the beginning of each frame, the reader needs to transmit a $Query()$ command to initiate the current frame and inform the tags of the parameters. As given in Section 3.3, the reader's $Query()$ command consists of a frame length parameter, a mask of $2f_i$ bits length, and $(\mathcal{C}_i^{\mathrm{2cs}}+1)$ hash seeds. In this work, a 16-bit message is used to transmit the frame length parameter, while a 8-bit message is used to transmit each hash seed. Since the reader message is separated into 96-bit segments, the time cost $\mathcal{T}_i^{\mathrm{rd}}$ for reader message transmission is given as

$$\mathcal{T}_i^{rd} = \left\lceil \frac{2f_i + 16 + 8\big(\mathcal{C}_i^{\mathrm{2cs}}+1\big)}{96} \right\rceil t_{id}. \tag{3.6}$$

Lemma 1 *In frame $\mathcal{F}_i$, the numbers of singleton and 2-collision slots $\mathcal{C}_i^{sg}$ and $\mathcal{C}_i^{2cs}$ are given as*

$$\mathcal{C}_i^{sg} \approx \mathcal{N}_i e^{-\frac{1}{\alpha}}, \tag{3.7}$$
$$\mathcal{C}_i^{2cs} \approx \frac{\mathcal{N}_i}{2\alpha} e^{-\frac{1}{\alpha}}. \tag{3.8}$$

Proof 1 *In PCMTI, the tags are randomly assigned to the slots with an uniform hash function. Thus, the number of tags assigned to one slot follows approximately a binomial distribution with $\mathcal{N}_i$ Bernoulli experiments $B(\mathcal{N}_i,\ 1/f_i)$. The probability that k tags are assigned to the same slot is given by*

$$\mathcal{P}(k) = \binom{\mathcal{N}_i}{k}\left(\frac{1}{f_i}\right)^k\left(1-\frac{1}{f_i}\right)^{\mathcal{N}_i-k}. \tag{3.9}$$

Accordingly, the probabilities of the expected singleton and 2-collision slots are

$$\mathcal{P}^{sg} = \mathcal{P}(1) = \frac{\mathcal{N}_i}{f_i}\left(1-\frac{1}{f_i}\right)^{\mathcal{N}_i-1}, \tag{3.10}$$
$$\mathcal{P}^{2cs} = \mathcal{P}(2) = \frac{\mathcal{N}_i}{2f_i^2}\left(1-\frac{1}{f_i}\right)^{\mathcal{N}_i-2}. \tag{3.11}$$

Letting $f_i = \alpha\mathcal{N}_i$, where α is the impact factor, we have

$$\mathcal{P}^{sg} = \frac{1}{\alpha}\left(1-\frac{1}{\alpha\mathcal{N}_i}\right)^{\mathcal{N}_i-1}, \tag{3.12}$$
$$\mathcal{P}^{2cs} = \frac{\mathcal{N}_i+1}{2\alpha^2\mathcal{N}_i}\left(1-\frac{1}{\alpha\mathcal{N}_i}\right)^{\mathcal{N}_i-2}. \tag{3.13}$$

According to the well-known equality $e^x = \lim\limits_{x\to\infty}\left(1+\frac{x}{n}\right)^n$ [82], the following approximations are attainable when $n \to \infty$

$$\mathcal{P}^{sg} \approx \frac{1}{\alpha} e^{-\frac{1}{\alpha}}, \tag{3.14}$$
$$\mathcal{P}^{2cs} \approx \frac{1}{2\alpha^2} e^{-\frac{1}{\alpha}}. \tag{3.15}$$

The numbers of singleton and 2-collision slots are $\mathcal{C}_i^{sg} = f_i\mathcal{P}^{sg}$ *and* $\mathcal{C}_i^{2cs} = f_i\mathcal{P}^{2cs}$*, respectively. Therefore, we can prove that*

$$\begin{aligned}\mathcal{C}_i^{sg} &\approx \mathcal{N}_i e^{-\frac{1}{\alpha}},\\ \mathcal{C}_i^{2cs} &\approx \frac{\mathcal{N}_i}{2\alpha}e^{-\frac{1}{\alpha}}.\end{aligned}$$

In PCMTI, the number of tags that can be verified in frame $\mathcal{F}_i$ is $\mathcal{C}_i^{\rm sg} + 2\mathcal{C}_i^{\rm 2cs}$. Therefore, the average time for reader message transmission is given by

$$\mathcal{T}_{\rm ave}^{rd} = \frac{\mathcal{T}_i^{\rm rd}}{\mathcal{C}_i^{\rm sg} + 2\mathcal{C}_i^{\rm 2cs}}. \tag{3.16}$$

Substituting (3.4), (3.6), (3.7), and (3.8) into (3.16) yields

$$\begin{aligned}\mathcal{T}_{\rm ave}^{rd} &= \frac{\left\lceil \frac{2f_i+16+8\left(\mathcal{C}_i^{\rm 2cs}+1\right)}{96}\right\rceil t_{id}}{\mathcal{C}_i^{\rm sg} + 2\mathcal{C}_i^{\rm 2cs}}\\ &\approx \frac{\left\lceil \frac{2\alpha\mathcal{N}_i+24+\frac{4\mathcal{N}_i}{\alpha}e^{-\frac{1}{\alpha}}}{96}\right\rceil t_{id}}{\mathcal{N}_i e^{-\frac{1}{\alpha}} + \frac{\mathcal{N}_i}{\alpha}e^{-\frac{1}{\alpha}}}\\ &\approx \frac{\left(\alpha^2 + 2e^{-\frac{1}{\alpha}} + \frac{12\alpha}{\mathcal{N}_i}\right)\frac{t_{id}}{48}}{(1+\alpha)e^{-\frac{1}{\alpha}}}\\ &\approx \frac{\left(\alpha^2 e^{\frac{1}{\alpha}} + 2\right)\frac{t_{id}}{48}}{1+\alpha}.\end{aligned} \tag{3.17}$$

Therefore, the average identification time for verifying one tag is given by

$$\mathcal{T}_{ave} = \mathcal{T}_{ave}^{rd} + \mathcal{T}_{ave}^{tg} \approx \frac{\left(\alpha^2 e^{\frac{1}{\alpha}} + 2\right)\frac{t_{id}}{48}}{1+\alpha} + \frac{t_p}{2}. \tag{3.18}$$

Lemma 2 *There exists only one optimal impact factor* α_{opt}*, which minimizes the average identification time of the proposed PCMTI protocol.*

Proof 2 *To minimize* $\mathcal{T}_{ave}$*, we compute the first derivation of* $\mathcal{T}_{ave}$ *as follows*

$$\frac{d\mathcal{T}_{ave}}{d\alpha} = \frac{(\alpha^2+\alpha-1)e^{\frac{1}{\alpha}} - 2}{(1+\alpha)^2}\frac{t_{id}}{48}. \tag{3.19}$$

Since $\alpha > 0$*, we have* $(1+\alpha)^2 > 0$*. Letting* $\frac{d\mathcal{T}_{ave}}{d\alpha} = 0$*, we have*

$$g(\alpha) = (\alpha^2+\alpha-1)e^{\frac{1}{\alpha}} - 2 = 0. \tag{3.20}$$

It is straightforward to obtain the first derivation of $g(\alpha)$ *as follows,*

$$\begin{aligned}\frac{dg}{d\alpha} &= \frac{(2\alpha^3-\alpha+1)e^{-\frac{1}{\alpha}}}{\alpha^2}\\ &= \frac{(\alpha+1)\left(2(\alpha-\frac{1}{2})^2+\frac{1}{2}\right)e^{\frac{1}{\alpha}}}{\alpha^2}.\end{aligned} \tag{3.21}$$

When $\alpha \in [0, \infty)$, we have $g(0) = -\infty$, $g(\infty) = \infty$, and $\frac{dg}{d\alpha} > 0$, which proves that $g(\alpha)$ is a monotonically increasing function. Therefore, there exists only one solution α_{opt} satisfying $g(\alpha_{opt}) = 0$. Then, we can prove: 1) when $0 \leq \alpha < \alpha_{opt}$, $\mathcal{T}_{ave}$ decreases; and 2) when $\alpha \geq \alpha_{opt}$, $\mathcal{T}_{ave}$ increases. Thus, it is proven that α_{opt} is the only solution which minimizes the average identification time of PCMTI.

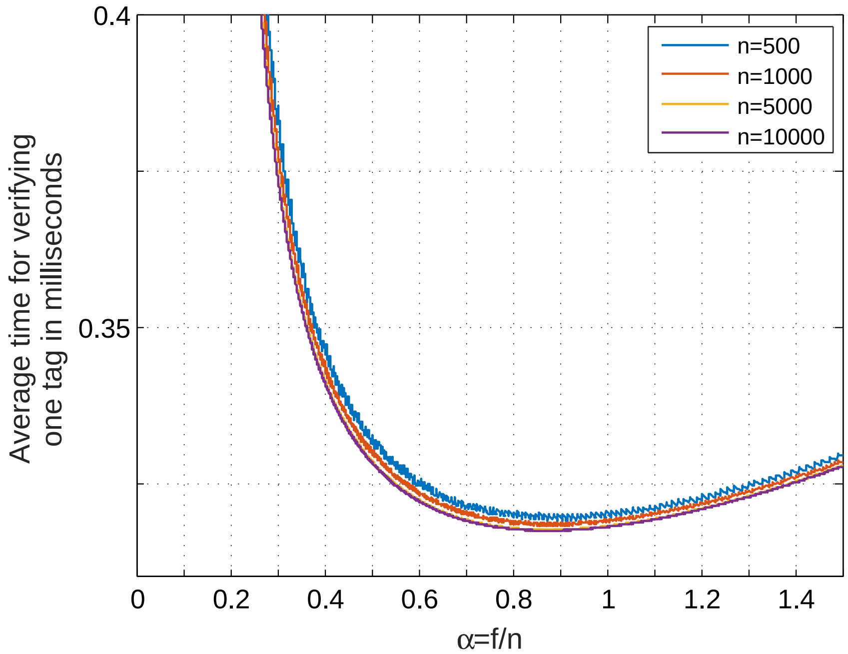


Figure 3.6: Numerical results of the average tag recognition time versus the impact factor α.

According to Philip's I-Code [83], the times for 1-bit short message and 96-bit ID transmission are $t_s = 0.4$ ms and $t_{id} = 2.4$ ms. The data rate is about $96/2.4 = 40$ kbps [17], and the tags' response messages are of 2-bit length. Therefore, the time for the tags' short response $t_p = 0.425$ ms in PCMTI. With these parameters, Fig. 3.6 plots the numerical values of $\mathcal{T}_{\text{ave}}$ when α ranges from 0 to 1.5, according to (3.18). As can be observed from Fig. 3.6, when $\alpha \approx 0.8735$, the minimum value of the average identification time for identifying one tag is

$$\mathcal{T}_{\text{ave}} \approx 0.3299 \text{ ms}.$$

Fig. 3.6 also shows that the proposed PCMTI protocol takes about 17.53% less time than the widely accepted lower bound given in [15, 17, 19–21].

Extra time cost for hash seeds transmission:

Since the proposed PCMTI protocol needs to transmit a second hash seed for every two-collision slot, the extra time cost for hash seeds transmission is given by

$$\mathcal{T}_{\text{ave}}^{\text{hs}} = \frac{\left\lceil \frac{8\mathcal{C}_j^{2cs}}{96} \right\rceil t_{id}}{\mathcal{C}_j^{sg} + 2\mathcal{C}_j^{2cs}} \approx \frac{t_{id}}{24(1+\alpha^2)} = 0.0567 \text{ ms.} \tag{3.22}$$

With the pair-reply and two-collision slot-resolving strategies, the average time for tag message transmission is greatly reduced to 0.2 ms as can be seen from (3.5), compared to 0.4 ms required by traditional methods [17–20]. The reduced time for tag message transmission (i.e., 0.2 ms) outweighs the time penalty attributable to extra hash seeds transmission (i.e., 0.0567 ms). Therefore, the proposed PCMTI protocol is faster than all the previous work.

3.4.2 Effect of Unreliable Channels

In the above analysis, the channels between the reader and tags are assumed to be error-free. This may not always be tree in practical RFID applications due to weak tag backscattering signals or transmission errors [84]. Since transmission errors can be effectively detected by appending a cyclic redundancy code (CRC) in each message, traditional re-transmission strategies can be used to ensure the correctness of the transmission of the reader's messages [17].

In our protocol, we do not use any CRC code in tag responses as previous works, such as MMTI [17], SFMTI [19], ProTar [20] and so on. The reasons are in three aspects. First, in missing tag identification protocols the tag only replies one symbol in each slot. The use of CRC code increases the time duration of every slot, resulting in increased identification time of each reading cycle. Secondly, since the missing tag identification protocol is usually repeatedly executed, if a missing tag is not detected because of transmission errors in the current reading cycle, it has a big chance to be detected in the following reading cycles. Thirdly, Griffin *et al.* demonstrated that when the distance between the reader and tags is smaller than 10m in indoor environment, the SNR in passive RFID systems is greater than 15dB, and the bit error probability is smaller than 10^{-6} [85]. Since the missing tag identification protocols are usually implemented in warehouse systems, the probability of transmission errors is too small to be negligible. Therefore, there is no need to use CRC in tag response messages.

In the following, we analyse the performance of our protocol in the real scenarios. Generally speaking, there are three main factors that affect the performance of RFID systems, i.e., detection probability, capture probability, and transmission bit error probability. Errors caused by these factors may result in two types decision faults in missing tag identification protocols, i.e., the false positive and false negative errors. The false positive error occurs when a staying tag is declared as missing mistakenly, and the false negative error occurs if a missing tag is mistakenly detected as a staying one. Denote by P_d, P_c and P_e the detection probability, capture probability, and transmission bit error probability, respectively. Let p_{fp} and p_{fn} be the probabilities of false positive and false negative errors.

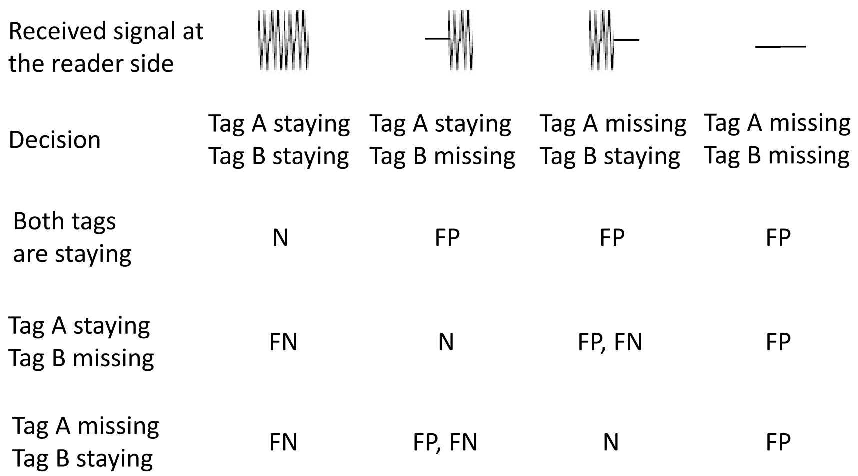


Figure 3.7: Received signals and reader decisions in the real scenarios, where “FP”, “FN” and “N” indicate the false positive error, false negative error and no error, respectively.

In our protocol, we allow two tags, e.g. tags A and B, reply in the same slot with symbols 0 and 1, respectively. In the physical layer, we use ASK modulation and Manchester coding methods to encode tag response message. In the ideal situation, if both tags are staying tags, the received signal should be an invalid symbol; if only tag A (or B) is staying, the received signal should be 0 (or 1); and if both tags are missing, the reader receives no signal. In the real scenarios, the received signal may be one of the four cases, i.e., invalid symbol, symbol 0, symbol 1 or no signal. More specifically, Fig. 3.7 illustrates the received signals and the corresponding reader decisions, where “FP”, “FN” and “N” indicate the occurrence of false positive error, false negative error and no error, respectively. Note that if a tag does not reply to the reader, the reader will not receive any signal from this tag. Therefore, when both tags are missing the reader always receives no signal.

Moreover, Tables 3.2, 3.3 and 3.4 list the decision cases when transmission errors, detecting errors and capture effect occur. In the table, C (or R) represents that the corresponding bit is correctly (or erroneously) transmitted.

Table 3.2: Decision errors caused by transmission errors when there is no detection errors and capture effect – both tags are staying tags

Transmission errors				aggregated signal	decision errors
Tag A		Tag B			
first bit	second bit	first bit	second bit		
C	C	C	C	“11”	N
C	C	C	R	“11”	N
C	C	R	C	“01”	FP
C	C	R	R	“01”	FP
C	R	C	C	“10”	FP
C	R	C	R	“11”	N
C	R	R	C	“00”	FP
C	R	R	R	“01”	FP
R	C	C	C	“10”	FP
R	C	C	R	“11”	N
R	C	R	C	“11”	N
R	C	R	R	“11”	N
R	R	C	C	“10”	FP
R	R	C	R	“11”	N
R	R	R	C	“10”	FP
R	R	R	R	“11”	N

Next, we calculate the probabilities of the false positive and false negative

Table 3.3: Decision errors caused by transmission errors, capture effect and detecting errors – Both tags are staying tags

Tag A is staying and tag B is missing: Tag A's signal is detected					
Transmission errors				aggregated signal	decision errors
Tag A		Tag B			
first bit	second bit	first bit	second bit		
C	C	–	–	"01"	FP
C	R	–	–	"00"	FP
R	C	–	–	"10"	FP
R	R	–	–	"11"	N

Tag A's signal is not detected, tag B's signal is detected (or tag B's signal is much stronger than tag A's signal).					
Transmission errors				aggregated signal	decision errors
Tag A		Tag B			
first bit	second bit	first bit	second bit		
–	–	C	C	"10"	FP
–	–	C	R	"11"	N
–	–	R	C	"00"	FP
–	–	R	R	"01"	FP

Both tags' signals are not detected					
Transmission errors				aggregated signal	decision errors
Tag A		Tag B			
first bit	second bit	first bit	second bit		
–	–	–	–	"00"	FP

Table 3.4: Decision errors caused by transmission errors when capture effect and detecting errors occur – One or more tags are missing

Tag A's signal is detected, tag B's signal is not detected (or tag A's signal is much stronger than tag B's signal)					
Transmission errors				aggregated signal	decision errors
Tag A		Tag B			
first bit	second bit	first bit	second bit		
C	C	–	–	"01"	N
C	R	–	–	"00"	FP
R	C	–	–	"10"	FP, FN
R	R	–	–	"11"	FN
Tag A is staying and tag B is missing: Tag A's signal is not detected					
Transmission errors				aggregated signal	decision errors
Tag A		Tag B			
first bit	second bit	first bit	second bit		
–	–	–	–	"00"	FP
Tag B is staying and tag A is missing: Tag B's signal is detected					
Transmission errors				aggregated signal	decision errors
Tag A		Tag B			
first bit	second bit	first bit	second bit		
–	–	C	C	"10"	N
–	–	C	R	"11"	FN
–	–	R	C	"00"	FP
–	–	R	R	"01"	FP, FN
Tag B is staying and tag A is missing: Tag B's signal is not detected					
Transmission errors				aggregated signal	decision errors
Tag A		Tag B			
first bit	second bit	first bit	second bit		
–	–	–	–	"00"	FP

errors. Firstly, when both tags are staying ones, from Tables 3.2 and 3.3 we can obtain the probability that a false positive error occurs in a slot as follows.

$$\begin{aligned} p_{fp}^b &= p_d^2\{p_c[(1-p_e)^2+3p_e(1-p_e)]+(1-p_c)[3(1-p_e)^3p_e \\ &\quad +4(1-p_2)^2p_e^2+2(1-p_e)p_e^3]\}+2p_d(1-p_d)+2(1-p_d)^2. \end{aligned} \tag{3.23}$$

Secondly, when one tag is missing and another one is staying, the probability that a false positive error occurs in a slot can be calculated in the following.

$$p_{fp}^e = p_d[p_e(1-p_e)+p_e^2]+(1-p_d) = p_dp_e+1-p_d. \tag{3.24}$$

The probability that a false negative error occurs in a slot with the case that one tag is staying and another one is missing is

$$p_{fn}^e = p_d[p_e(1-p_e)+p_e^2] = p_dp_e. \tag{3.25}$$

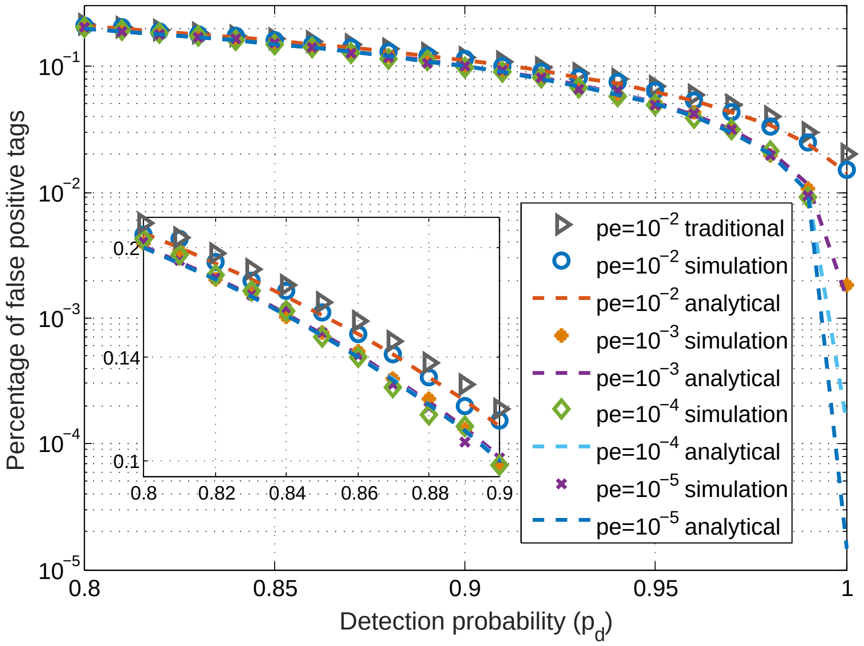
Suppose the number of total tags within the reader's reading range is n, the number of missing tags is m. The probability that a slot with both tags are staying ones is $\left(\frac{n-m}{n}\right)^2$, and the probability that a slot with one tag is staying and another one is missing is $\left(\frac{n-m}{n}\right)\left(\frac{m}{n}\right)$. With the pair-reply and two-tag collision-slot resolving strategies, each slot is used to identify two tags simultaneously. Thus, the total number of slots is $\lceil\frac{n}{2}\rceil$. Therefore, the percentage of false positive errors among all the staying tags is

$$\begin{aligned} p_{fp} &= \frac{(\frac{n-m}{n})^2p_{fp}^b\lceil\frac{n}{2}\rceil+\frac{m(n-m)}{n^2}p_{fp}^e\lceil\frac{n}{2}\rceil}{n-m} \\ &\approx \frac{n-m}{2n}\Big\{p_d^2\{p_c[(1-p_e)^2+3p_e(1-p_e)] \\ &\quad +(1-p_c)[3(1-p_e)^3p_e+4(1-p_2)^2p_e^2 \\ &\quad +2(1-p_e)p_e^3]\}+2p_d(1-p_d)+2(1-p_d)^2\Big\} \\ &\quad +\frac{m}{n}\left[p_dp_e+1-p_d\right]. \end{aligned} \tag{3.26}$$

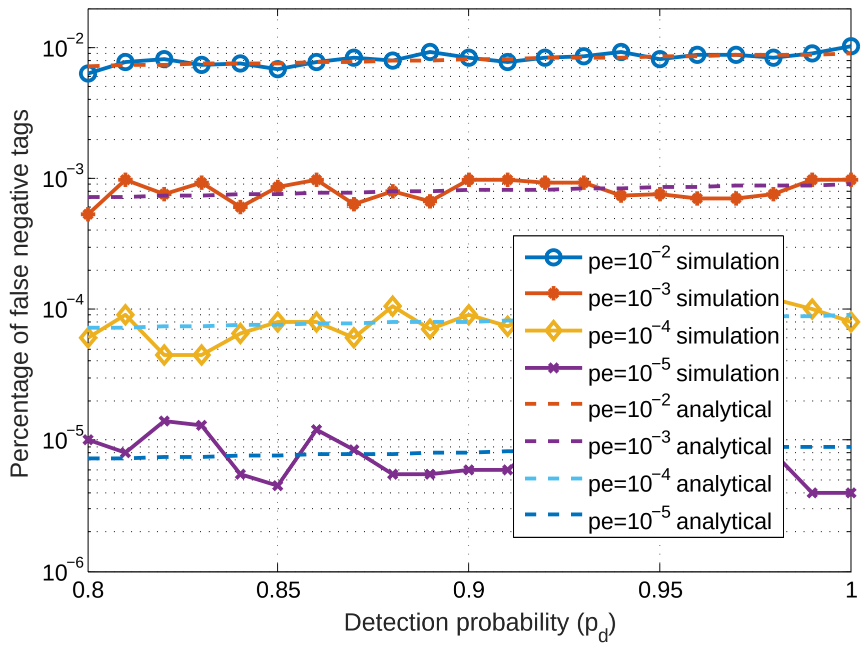
Similarly, the percentage of false negative errors among all the missing tags is

$$p_{fn} = \frac{\frac{2m(n-m)}{n^2}p_{fn}^e\lceil\frac{n}{2}\rceil}{m} \approx \frac{n-m}{n}p_dp_e. \tag{3.27}$$

Moreover, Fig. 3.8 gives both the simulation and analytical results of the percentages of false positive and false negative tags. In the figure, $n = 2000$,



(a)



(b)

Figure 3.8: Percentage of detection errors when $n = 2000$, $m = 200$, $P_c = 1\%$: (a) Percentage of false positive errors vs. detecting probability; (b) Percentage of false negative errors vs. detecting probability.

$m = 200$, $p_c = 1\%$, p_d ranges from 0.8 to 1, and p_e is in $[10^{-2},\ 10^{-3},\ 10^{-4},\ 10^{-5}]$. The figure shows that the simulation results coincide with the analytical values very well. As can be observed from Fig. 3.8(a), the percentage of false positive tags is affected by both p_d and p_e. A snapshot of $0.8 \leq p_d \leq 0.9$ shows that the percentage of false positive tags is slightly higher than $1-p_d$. The additional errors are caused by capture effect and transmission errors. Fig. 3.8(b) demonstrates that the percentage of false negative errors is almost the same with the bit error probability.

3.5 Evaluation

In this section, we evaluate the performance of the proposed PCMTI protocol and compare it with some comparative benchmarkers. The performance of PCMTI is evaluated with the optimal impact factor obtained in Section 3.4, i.e., $\alpha_{\text{opt}} = 0.8735$. The number of tags ranges from 50 to 10,000. The simulations are conducted in Matlab with each instance of the simulation repeated 100 times to obtain the average experimental results.

3.5.1 Simulation Configurations

Parameters settings: Since the reader's request message in each frame usually contains long message bits, we divide it into 96-bit segments as in previous work [17–20]. Based on Philips I-Code [83], the time for 1-bit short response message transmission is $t_s = 0.4$ ms, and the time for transmitting a 96-bit tag ID (or segment) $t_{id} = 2.4$ ms. The data rate is about $96/2.4 = 40$ kbps [17]. Thus, the time for transmitting the one data symbol in Manchester coding is $t_p = 0.425$ ms. Since unreliable channels affect the comparative benchmarks [19, 20] almost the same, the communications channels between the reader and tags are assumed to be error-free as in the literature [7, 8, 15, 19, 20].

Comparative works: In the simulation, we first compare the simulation results of the proposed PCMTI protocol with the analytical ones obtained in Section 3.4. Then we compare the proposed protocol with some best-performing benchmarkers, such as the two hash protocol (THP) [18], the slot filter-based missing tag identification (SFMTI) protocol [19], and the probabilistic tag retardation-based (ProTaR) [20] protocol. THP is the most typical protocol in missing tag identification. SFMTI and ProTaR are the most recent representative protocols reported in the literature. It should be noted that ProTaR needs the *a prior* knowledge of the tag missing probability. For a fair comparison, the naive-ProTaR (NProTaR) protocol proposed in [20], which omits the tag missing probability, is used. Moreover, the time needed to generate the pseudo-identifiers (PIDs) is ignored as in [20].

3.5.2 Average Identification Time

Firstly, Fig. 3.9 depicts the average identification time for verifying a tag when $\mathcal{N}$ ranges from 50 to 10000. It can be concluded from this figure that the proposed PCMTI protocol takes about 0.32 ms for verifying one tag, which is much shorter time than the other protocols.

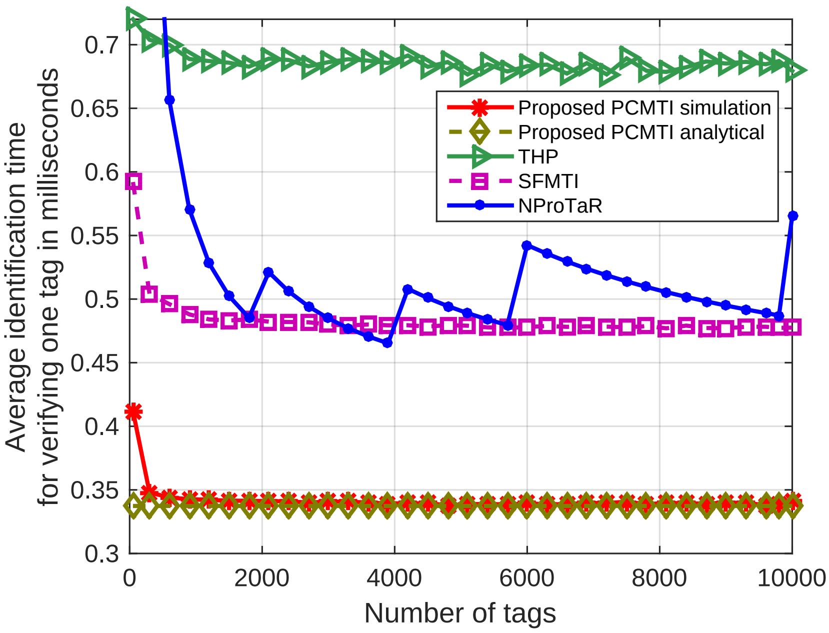


Figure 3.9: Simulation results of the comparative protocols: the average identification time for verifying a tag versus the number of tags.

Compared with the best missing tag identification protocol in the literature, i.e., the SFMTI protocol, PCMTI reduces the identification time by about 33%. By comparison, the THP protocol takes the longest to verify a tag. Because except the singleton slots, THP also executes the empty and collision slots, resulting in much more time than other comparative protocols.

It is also shown in Fig. 3.9 that NProTaR takes longer time than SFMTI, and the curve of the NProTaR protocol exhibits some fluctuations. This is because the length of the pseudo-identifier (PID) is a step function (see Table III in [20]). The length of the transmitted bit vector depends on the length of the PID. When $\mathcal{N}$ is in a distinct interval, such as $\mathcal{N} \in [2000,\ 4000)$, the transmitted bit vectors are of the same length, which takes almost the same time in the reader's request message. The overhead per tag is bigger when the number of tags is smaller. Therefore, the average identification time for verifying a tag for NProTaR decreases when $\mathcal{N}$ increases in each interval.

It can also be observed from Fig. 3.9 that the average identification time decreases when $\mathcal{N}$ increases from 0 to 1000. The reason is as follows. When the number of tags is small, the messages transmitted in the reader's request command at the beginning of a frame are also small. If the transmitted message is less than 96 bits, it still occupies a slot which takes t_{id} time to transmit. When the number of tags is small, this overhead on a per tag basis becomes notable. Therefore, the proposed PCMTI protocol takes longer to identify a tag in small-scale scenarios, i.e., $\mathcal{N} < 1000$, than in large-scale scenarios.

3.5.3 Reader Message Transmission

In this subsection, we compare the communications overhead at the reader in Figs. 3.10 and 3.11, which depict the simulation results on two performance metrics, i.e., the total message bits transmitted at the reader, and the average reader message transmission time for one tag identification.

It should be noted that in PCMTI simulation, the total number of bits in all the reader transmission messages is obtained by counting the number of bits transmitted in every $Query()$ command, which consists a $2f_i$-bit mask, a 16-bit frame length parameter, and $(\mathcal{C}_i^{2\text{cs}} + 1)$ hash seeds. More specifically, the bit counting process is given below,

$$\mathcal{M}^{\text{reader}} = \sum_{i=1}^{\mathscr{F}} 2f_i + 16 + 8(\mathcal{C}_i^{2\text{cs}} + 1), \tag{3.28}$$

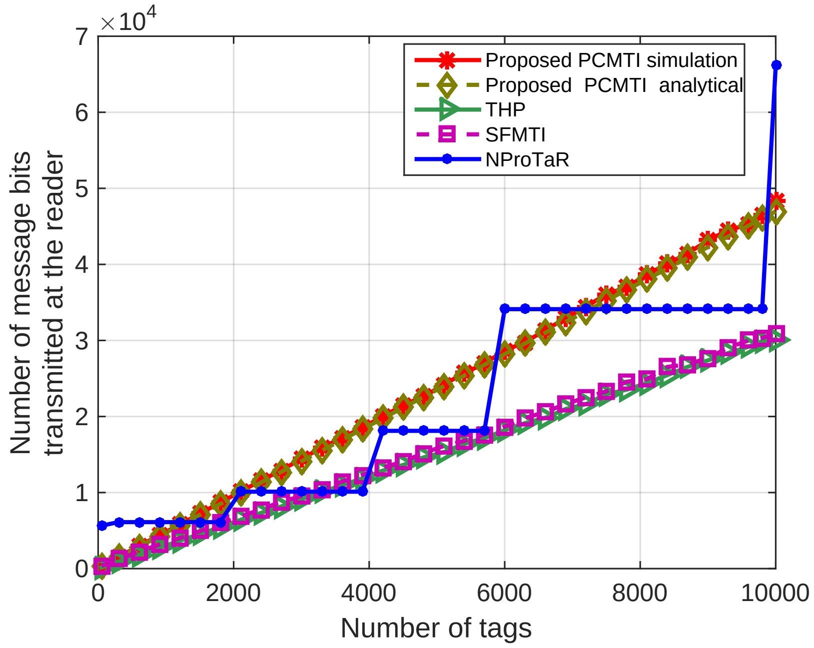


Figure 3.10: Number of message bits transmitted by the reader for n tag identification.

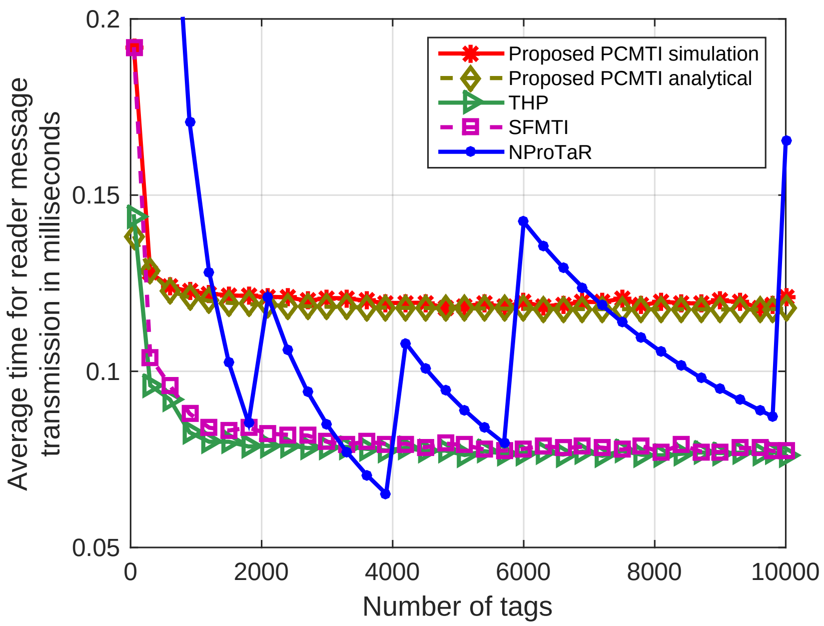


Figure 3.11: Average reader transmission time for one tag identification.

where $\mathscr{F}$ is the total number of frames, and each hash seed is of 8-bit length.

Fig. 3.10 plots the total message bits transmitted by the reader. It is observed that the simulation results match very well with their analytical counterparts. The proposed PCMTI protocol transmits more message bits at the reader than other protocols. The extra time cost is due primarily to the transmission of the second hash seed for each two-collision slot. THP and SFMTI protocols transmit almost the same number of message bits at the reader. The reader transmission messages of NProTaR still increase in a step-by-step manner. Fig. 3.11 gives the comparative results on the average reader message transmission time. As can be seen from the figure, it takes the proposed PCMTI about 0.12 ms to transmit a reader request message on average. The THP and SFMTI protocols take about 0.075 ms to transmit the reader request message for one tag identification. Compared with the best protocol SFMTI, the increased average time cost at the reader amounts to around 0.05 ms per tag. Although the proposed SFMTI protocol needs to transmit more message bits at the reader side, it will be shown that the increased transmission time of the reader command is much smaller than the saved transmission time of the tag responses in the following. Thus, the overall identification time of the proposed PCMTI protocol is much smaller than other comparative protocols.

3.5.4 Time Cost of Tag Responses

Then we compare the time costs of the tag response slots. Similarly, the comparison also consists of two aspects, i.e., the total number of short response slots executed and the average response time for one tag identification. Figs. 3.12 and 3.13 depict the total number of short response slots and the average tag response time, respectively.

As can be seen from Fig. 3.12, THP takes the most number of tag response slots. This is mainly because that THP executes all the singleton, collision and empty slots. Without having to execute any collision and empty slots, SFMTI and NProTaR require smaller numbers of response slots. In PCMTI, every two tags, which are assigned to two consecutive singleton slots, will reply in the same slot. With re-arranged response slots, only half of the singleton slots and the

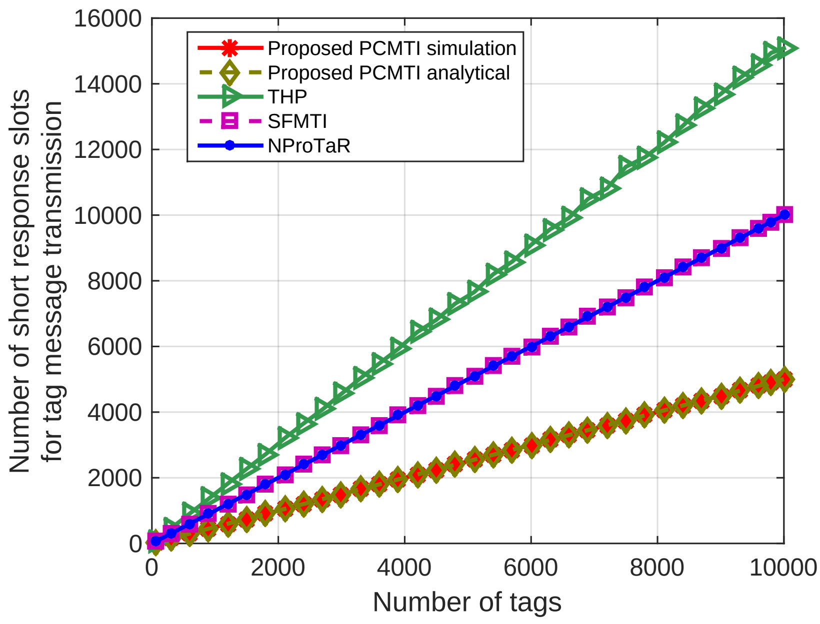


Figure 3.12: Number of short response slots for n tag identification.

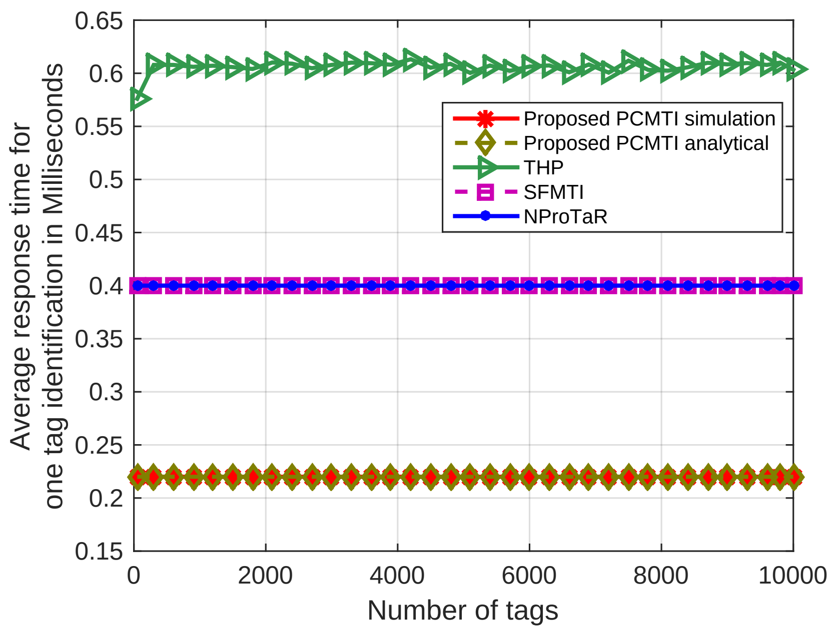


Figure 3.13: Average response time for one tag identification.

two-collision slots are executed in the interactive communications. Therefore, the proposed PCMTI protocol requires a much smaller number of response slots than other protocols. Fig. 3.13 shows that PCMTI takes 0.22 ms to verify a tag averagely. However, THP takes about 0.6 ms, while SFMTI and NProTaR take 0.4 ms for average tag response. Compared with such protocols, the proposed PCMTI protocol takes at least 0.18 ms less time for tag response message transmission.

3.6 Summary

In this chapter, the proposed PCMTI protocol is introduced and its performance is evaluated through theoretical analysis and simulation comparison. The new PCMTI protocol is proved to be capable of reducing the average identification time by 30% than the best missing tag identification protocol in the literature, making it more suitable for fast tag monitoring than the comparative protocols. In this chapter, the impact of unreliable channels is also analyzed. Although false positive and false negative errors cannot be entirely eliminated, we will try to reduce the percentage of such errors in our future work.

Chapter 4

Bit-detecting Protocol for Continuous Tag Recognition in Mobile RFID Systems

4.1 Introduction

In a mobile RFID system, such as a conveyor belt and large-scale warehouse environments, many tags move in and out of the system continuously, so that the reader has very limited time to recognise all the tags. When a new reading cycle begins, some tags recognised in the preceding reading cycle (termed *known tags*) may stay in the reader's range for two consecutive reading cycles. Meanwhile, some *unknown tags* may enter the current reading cycle. Note that a reading cycle is the recognition process in which the reader recognises all the tags within its reading range. Compared with static RFID systems, the reader has limited time to complete the reading process in mobile RFID systems, especially when the number of tags is large. We denote the identification problem of the mobile RFID system as continuous tag identification problem.

For continuous tag identification, two types of tags need to be recognised: the known and unknown tags. On the one hand, although the reader has the IDs for all the tags recognised in the preceding reading cycle, it does not know which tags moved out of its reading range in the current reading cycle. Therefore, the reader needs to check which tags remain in the reader's range (i.e., the known tags) and retrieve their IDs from the back-end database. Identifying known tags is of great importance, especially in large-scale warehouse management [8,17,21]. Since a tag may be missing due to management faults or theft, rapid updates of missing tags are of great significance. On the other hand, since the reader has no prior knowledge of the unknown tags, to obtain their IDs the reader usually sends requests to the tags. The tags that meet the reader's request reply to the reader [77]. If two or more tags reply simultaneously, a tag collision occurs, because all the tags share the same wireless channel. Tag collision not only increases the identification delay but also wastes bandwidth. Consequently, efficient tag anti-collision protocols are of great importance for unknown tag identification.

Therefore, effectively identifying known tags and rapidly recognising unknown tags in large-scale mobile systems are among the most challenging problems facing RFID tag identification. This chapter studies the important continuous tag identification problem in practical mobile RFID systems [10,60–63,86].

4.2 Contributions and Outline

In this chapter, we propose an efficient bit-detecting (EBD) protocol for continuous tag identification in mobile RFID systems. In the proposed protocol, two new methods, i.e., the known tag bit-monitoring and M-ary bit-detecting tree recognition methods, are developed to identify the known and unknown tags separately. The new bit-monitoring method can identify multiple known tags in one slot simultaneously, which greatly reduces the time for known tag identification. The new M-ary bit-detecting method reduces the number of collision slots and eliminates all the idle slots, which accelerates the identification process for unknown tag recognition. Armed with the two efficient methods, the proposed EBD protocol is demonstrated to outperform previous methods for continuous tag recognition through both theoretical analysis and simulation experiments.

To summarise, the major contributions of this part of work are threefold as follows:

- A new EBD protocol for continuous tag identification is proposed. Both theoretical analysis and simulation experiments are conducted to prove that the proposed EBD protocol outperforms the state-of-the-art protocols reported in the literature.

- In this protocol, two new methods are proposed to accelerate the identification process:
  - An efficient bit-monitoring method is proposed to detect the presence of known tags and to retrieve their IDs from the back-end database, which outperforms existing tag monitoring protocols;
  - An M-ary bit-detecting tree method is proposed to rapidly recognise unknown tags and to collect their IDs, which performs better than previous tag ID collecting protocols.
- The branch number M, which maximises the system performance of the proposed M-ary bit-detecting tree method, is optimised through theoretical analysis.

The remainder of this chapter is organized as follows. Section 4.3 presents the system model and defines some common symbols used in the paper. Section 4.4 describes the proposed EBD protocol in detail. In Section 4.5, the performance of the proposed protocol is analyzed theoretically. Section 4.6 evaluates the protocol through computer simulations. Finally, concluding remarks are drawn in Section 4.8.

4.3 System Model and Definition

In this section, we first introduce our system model and the objective of this work, followed by the definitions of common symbols used in this chapter.

4.3.1 System Model

In this work, we consider two consecutive reading cycles of a mobile RFID system, which is comprised of a reader and numerous tags as illustrated in Fig. 4.1. In this model, the reader is considered to have the ability to store (through connections with a back-end database) the information of already recognized tags as in [17,21,60,61,77]. The tags are capable of storing data for long time in their internal memory [60, 61]. In this model, the reader (or the database connected to it) maintains a list ls to record the known tags recognized in C_{i-1}. For easy of exposition, some common terms used in the system model are defined in the following.

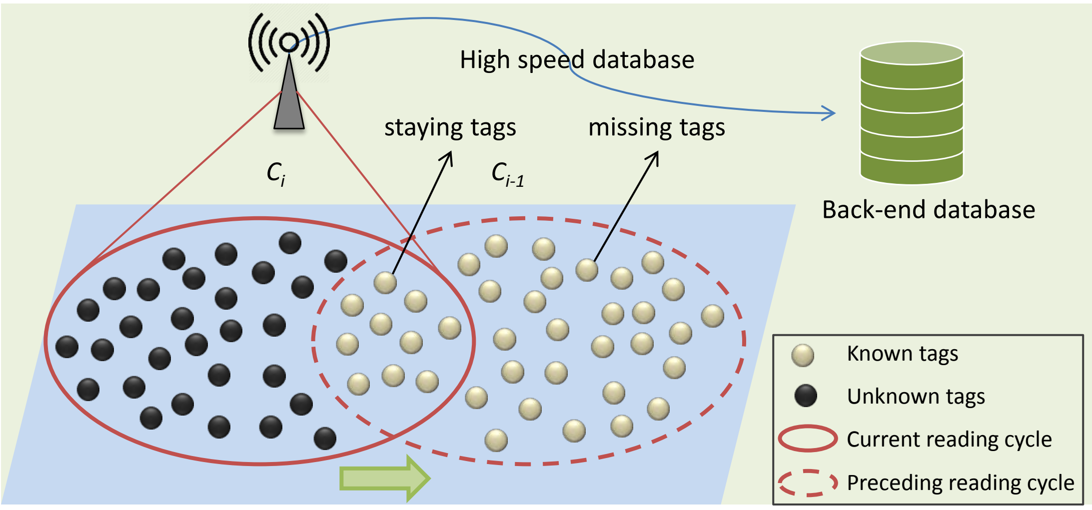


Figure 4.1: Recognition model of the mobile RFID systems.

- *Reading Cycle*: A reading cycle is defined as a single recognition process implemented by the reader to recognize all the tags within its reading range. Denote by C_i the i-th reading cycle.
- *Unknown tags in* C_i: The tags participate in C_i, but did not participate in C_{i-1} (the black dots in Fig. 4.1).
- *Known tags in* C_i: The tags participated in C_{i-1} (the white dots in Fig. 4.1). Generally, there are two types of known tags:
  - *Staying tags*: The tags participated in C_{i-1}, and also participate in C_i (the white dots within the reading area of C_i in Fig. 4.1).
  - *Missing tags*: The tags participated in C_{i-1}, but do not participate in C_i (the white dots outside the reading area of C_i in Fig. 4.1).

In this model, the reader (or the database connected to it) maintains a list ls to record the known tags recognized in C_{i-1}. Based on various applications, the following two typical mobile RFID applications, e.g., low-mobility tag monitoring and fast moving tag identification applications, are considered in this work.

- In low-mobility tag monitoring applications, such as warehouse management systems, the reader focuses on continuously monitoring all the tags within the range to detect missing tags timely and to identify unknown ones efficiently. Therefore, all the tags in the reader's reading range participate in every reading cycle. In such applications, tags' mobility is so slow that the tag set is assumed to remain unchanged during a reading cycle as in most previous work [44, 60–62, 74];

- In fast moving tag identification applications, such as conveyor belt systems, tags quickly pass through the reader's reading range. In such applications, the tag set always changes [10, 44, 63, 65, 86], and the reader concentrates on rapidly recognizing all the tags passing through its reading range. Therefore, the known tags will remain silent in the following reading cycles.

To be compatible with both types of applications, our *objectives* in each reading cycle are:

- Efficient detection of the presence of staying tags, and retrieving their IDs from the back-end database;

- Rapid collection of unknown tags' IDs.

4.3.2 Notation and Assumptions

It is assumed that the reader and all the tags have the ability to use Manchester coding to encode and decode messages. This is a reasonable assumption because both ISO/IEC 18000-6 and ISO/IEC 14443 standards define the use of Manchester coding in RFID systems. Besides, the tags are supposed to be capable of executing the following two simple functions, i.e., the detecting string generating function $DSG()$ and the bit counting function $CNT()$.

- $DSG(l, r)$ generates l bits string with the rth bit is "1" and other bits are all "0"s. It should be noted that *the bit position used in this paper starts from the left most bit.* For example, $DSG(5, 4) =$ "00010".

- $CNT(st, p, B)$ counts the number of "0"s (resp. "1"s) before the pth bit in string st, when B = "0" (resp. B = "1"). Note that if p is larger than the bit length of st, CNT counts the number of all the "B"s in st. Take st = "011010" as an example. We have $CNT(st,\ 4,\ \text{"0"}) = 1$, and $CNT(st,\ 7,\ \text{"1"}) = 3$.

Table 4.1: Symbol definitions

Symbol	Definition
ID (resp. RID)	Tag's (resp. Reader's) unique ID.
RI	Recognition index to denote that the tag is the RI-th tag recognized by the reader.
ls	List maintained by the reader to record the $(ID,\ RI)$ values for the recognized tags.
Tc	Recognized tag counter maintained by the reader and tags to record the number of tags identified.
Ac	Allocated slot counter maintained by the tags to decide in which slot the tag will reply to the reader.
mds (resp. rds)	Bit detecting string generated by a tag in the KTBM phase (resp. the MBTR phase).
mfs (resp. rfs)	Feedback string generated by the reader in the KTBM phase (resp. the MBTR phase).
$rs,\ rn$	Random numbers generated by the tags.
$L(resp.M)$	Length of the mds (resp. rds) string.

Finally, we assume that the communications channel between the reader and tags are error-free as in the literature [26, 44, 52]. The main reason is that a passive RFID system has a very limited reading range, usually within several meters. Most RFID systems are installed in a warehouse or on the conveyor belt, where the communications is so good that the channel errors between the reader and tags are negligible. Some common symbols used in our protocol are summarized in Table 4.1.

4.4 Proposed Efficient Bit-detecting Protocol

In this section, we propose an efficient bit-detecting (EBD) protocol for continuous tag recognition in mobile RFID systems. In EBD, the reader first distinguishes known tags from unknown tags, and then recognizes all the tags in two separate phases, namely the known tag bit-monitoring (KTBM) and M-ary bit-detecting tree recognition (MBTR) phases. In KTBM, a new bit-monitoring algorithm is developed to identify the known tags with few number of slots. In MBTR, a novel M-ary bit-detecting algorithm is proposed to predict and eliminate the idle slots.

Before describing the proposed protocol, we introduce two important counters, i.e. the allocated slot counter Ac and the recognized tag counter Tc. The Ac counter is maintained by the tags to decide when to reply. Only when a tag's $Ac = 0$, it replies to the reader. The Tc counter is maintained by both the reader and the tags in order to record the number of recognized tags. To record the recognition order, a tag sets $RI = Tc + 1$ when it is recognized by the reader.

4.4.1 Preparation: Distinguish Known Tags from Unknown Ones

To distinguish the known tags from the unknown ones, the unique reader ID RID and the index of the reading cycle C_i is used. After a tag is recognized, the tag stores (RID, C_i, RI) in its memory, and the reader records (ID, RI) of the recognized tag in a table list ls. At the beginning of a new reading cycle, the reader first activates all the tags, and informs them of its reader ID and the index of the current reading cycle. The tag determines whether it is a known or unknown tag by comparing them with the stored information. If the stored RID is the same with the received reader ID and the index of the reading cycle is right after the stored index of the reading cycle index, the tag is a known tag. It will participate in the KTBM phase. Otherwise, the tag is an unknown tag, and will participate in the MBTR phase.

4.4.2 Phase 1: Known Tag Bit-monitoring (KTBM)

In KTBM, a new bit-monitoring algorithm is developed to effectively identify known tags, in which the reader and tag record the recognition index when a tag is recognized in the preceding reading cycle. In the current reading cycle, the known tags reply one bit message to inform their presence in sequence according to their RI values. Then the reader identifies the staying tags by retrieving their IDs from the back-end database.

In KTBM, the known tags can reply only one bit to the reader one after another according to their RI values. To further reduce the identification time, the reader divides the n known tags into $\lceil \frac{n}{L} \rceil$ groups, where $\lceil \rceil$ is the ceil function. With a specially designed mds string, the reader verifies the existence of L tags simultaneously in each slot.

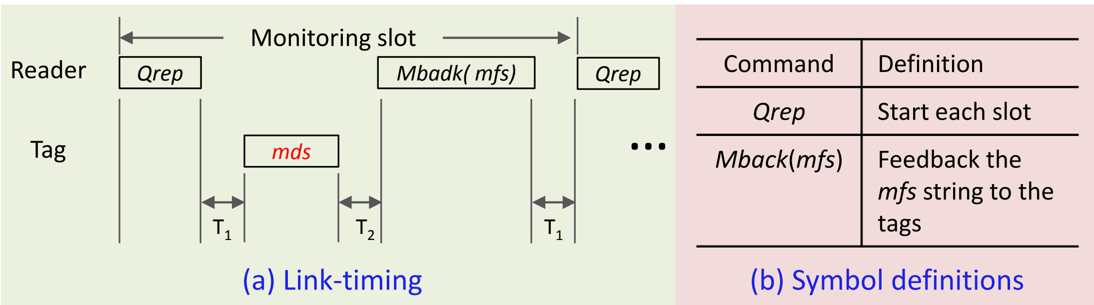


Figure 4.2: Tag identification in the KTBM phase.

In detail, the KTBM phase starts with the reader's $Monit(L)$ command. After receiving this command, all the known tags initiate $Tc = 0$, and calculate the values of Ac and mds by setting $Ac = \lceil \frac{RI}{L} - 1 \rceil$, and $mds = DSG\big(L,\ [(RI - 1)\ mod\ L] + 1\big)$. Then the reader begins every slot with a $Qrep$ command. Next, the tag with $Ac = 0$ replies mds to the reader. After receiving the tags' responses, the reader identifies the staying tags and feeds back a mfs string to update the RIs of the recognized tags and Tcs of the remaining tags. Fig. 4.2 gives the link-timing and symbol definitions of the KTBM phase, where T_1 is the time taken from the reader transmission to a tag response, and T_2 is the reader response time required if a tag is to demodulate the interrogator signal as defined in [77].

Moreover, Fig. 4.3 also gives an example of the identification process. In it, the table on the top gives the $(ID,\ RI)$s of 8 known tags in list ls stored at the reader before the recognition process, where the red items refer to the missing tags. The mds strings generated by the tags and the decoded messages at the reader are shown in the middle. The table on the bottom illustrates the new list ls_{new} generated at the end of the KTBM phase.

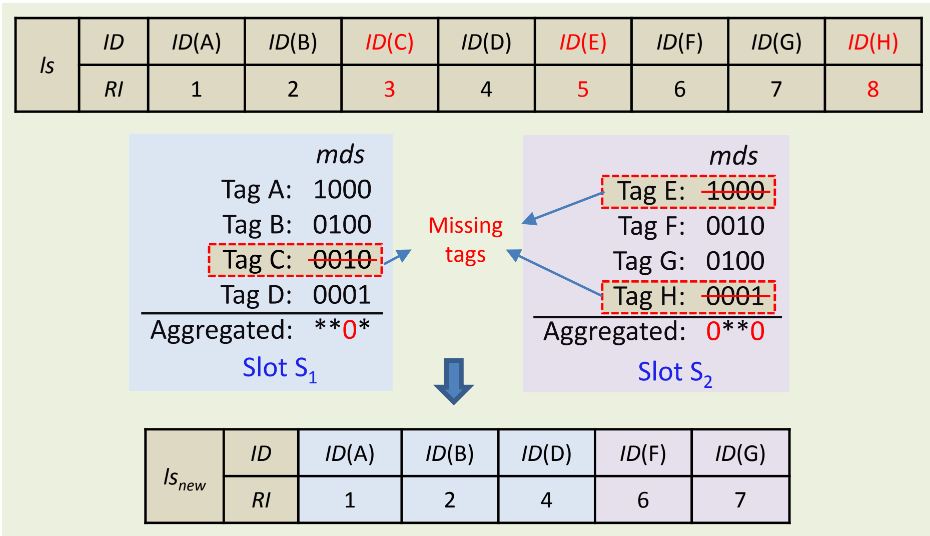


Figure 4.3: Example of the KTBM recognition phase.

Assume tags C, E, and H are missing, and $L = 4$. After receiving the $Monit(L)$ command, the staying tags set $Ac(A) = Ac(B) = Ac(D) = 0$ and $Ac(F) = Ac(G) = 1$. The generated mds strings are given in the middle of Fig. 3. In the first slot S_1, tags A, B, and D reply their mds strings simultaneously. At the reader side, the aggregated message is "$**0*$", where "$*$" indicates the colliding bit. Since the 3rd bit is "0", the reader determines that the corresponding tag (i.e., tag C), is missing. The reader updates ls by deleting tag C and assigning new RIs for tags A, B, and D as shown in the bottom table of Fig. 4.3. Then the reader updates $Tc = Tc + CNT(mfs,\ L+1, \text{"1"})$ to ensure that the RIs of the identified tags in the next slot starts from $Tc+1$. For example, in slot S_2 in Fig. 4.3, tag F's new recognition index is $RI = 4$.

To synchronize the information at the tags, the reader broadcasts the $Mback(mfs)$ command at the end of each slot, where mfs is obtained by converting all the colliding bits in the decoded message to "1". With this information, the tag with $Ac = 0$ updates $RI = Tc + CNT(mfs,\ [(RI-1)\ mod\ L],\ \text{"1"}) + 1$, stores ($RID,\ C_i,\ RI$) in its memory, and remains silent until the end of C_i. The remaining tags update $Ac = Ac - 1$ and $Tc = Tc + CNT(mfs,\ L+1, \text{“1”})$. More specifically, the pseudo-codes of the KTBM phase at the reader and tags are given in Algorithms 4 and 5.

Algorithm 4 KTBM phase at the reader

Initialize $ls_{new} = NULL$, $Tc = 0$,
Broadcast $Monit(L)$.
for x=1:$\lceil n/L \rceil$ **do**
  Broadcast $Qrep$ to start slot S_x and wait for tag responses. Upon receiving the superimposed tag responses, the reader decodes the received message as dm.
  for $p = 1 : L$ **do**
    if $dm(p)$ is a collision bit **then**
      Set $RI = Tc + CNT(dm,\ p+1,\ \text{“1”})$, and $ls_{new} = QueueIn(ls_{new},\ (ID_{p+L\cdot(x-1)},\ RI))$[1].
    end if
  end for
  Set $Tc = Tc + CNT(mfs,\ \text{“1”},\ L+1)$.
  Generate mfs by converting all the collision bits in dm to be “1”, and broadcast $Mback(mfs)$.
end for

4.4.3 Phase 2: M-ary Bit-detecting Tree Recognition (MBTR)

In MBTR, a novel bit-detecting algorithm is proposed to rapidly identify unknown tags, in which the reader splits tags with an M-ary tree structure, and skips the idle slots with the aid of a specially designed rds string in the response message. More specifically, a “0” bit in the ith position of rds indicates that there will be no tag responses in the ith slot. Then the reader and tag will skip this slot. By comparison, a colliding or “1” bit indicates that there will be tag responses. Thus, the corresponding slot will not be skipped. In general, Fig. 4.4

[1] $dm(p)$ is the p-th bit of dm. $QueueIn(Q,\ a)$ is the operation to put a into queue Q. $ID_{p+L\cdot(x-1)}$ is the ID of the $[p + L\cdot(x-1)]$-th tag in ls.

Algorithm 5 KTBM phase at the tags

1: After receiving $Monit(L)$, a known tag sets

2: $Tc = 0$, $p = [(RI - 1)\ mod\ L]\ + 1$,
$mds = DSG(L,\ p)$, $Ac = \lceil RI/L - 1 \rceil$.

3: After receiving $Qrep$, a known tag checks:

4: **if** $Ac == 0$ **then** Reply mds.

5: **end if**

6: After receiving $Mback(mfs)$, a tag checks:

7: **if** $Ac == 0$ **then**

8: Set $RI = Tc + CNT(mfs,\ p + 1,\ \text{“1”})$.
Record $(RID,\ C_i,\ RI)$ in its internal memory.
Remain silent until the next reading cycle.

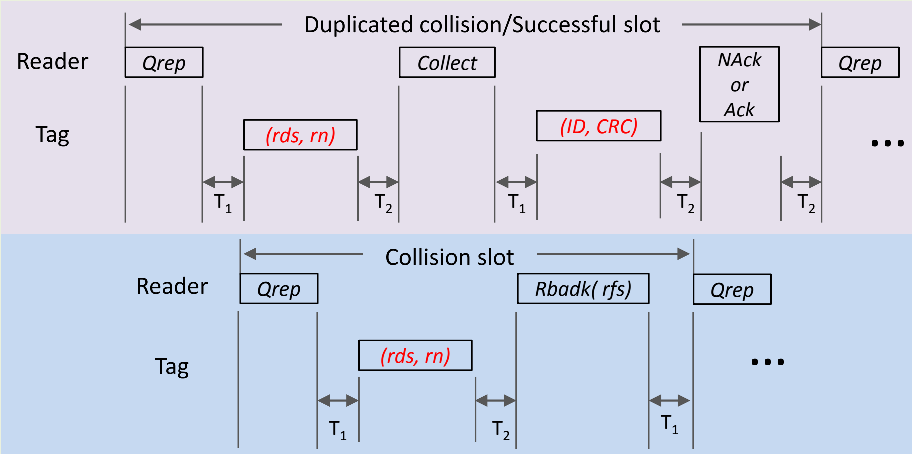
9: **else** $Ac = Ac - 1$, $Tc = Tc + CNT(mfs,\ L + 1,\ \text{“1”})$.

10: **end if**

illustrates the link-timing and the commands of the MBTR phase, where T_1 is the time from the reader transmission to a tag response, and T_2 is the reader response time required if a tag is to demodulate the interrogator signal as defined in [77].

More specifically, the MBTR phase begins with the reader's $Recog(\alpha,\ M)$ command, where α is the number of tags identified in the KTBM phase. Note that the unknown tags did not participate in the KTBM phase. In order to be consistent, initiating the unknown tags' $Tc = \alpha$ guarantees that the recognized tag's RI begins from $\alpha + 1$ in the MBTR phase. After receiving this command, all the remaining tags set $Ac = 0$ and $Tc = \alpha$. Then the reader broadcasts $Qrep$ to start each slot as shown in Fig. 4.4(a). After receiving $Qrep$, the tag with $Ac = 0$ generates two random numbers $rs,\ rn$ and a detecting string rds, where $rs \in [1, M]$, $rn \in [0, 2^{10} - 1]$, and $rds = DSG(M,\ rs + 1)$. Here, rs means that the tag plans on replying again in the rsth slot if the current slot is a collision slot. Then the tag replies $(rds,\ rn)$ to the reader.

In tags' responses, two types of slots, i.e., successful and collision slots as illustrated in Fig. 4.4(a), may occur. If the reader can successfully decode the received message, the *Collect* command is issued to request the tag ID as shown on the top of Fig. 4.4(a). If only one tag replies its ID, the reader can success-



(a) Link-timing

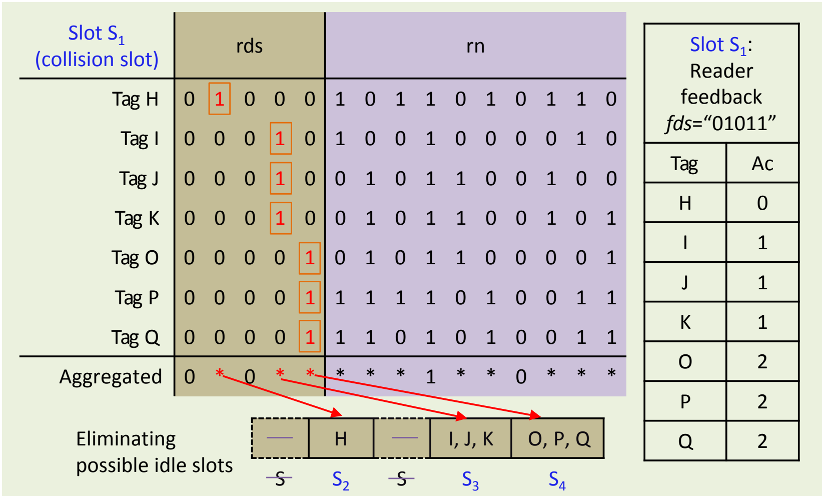
Command	Definition	Command	Definition
Qrep	Start each slot	Collect	Request tag's ID
Rback(*rfs*)	Feedback the *rfs* string to the tags	Ack (resp. NAck)	Inform tags of a successful (resp. duplicated collision) slot occurs

(b) Command definitions

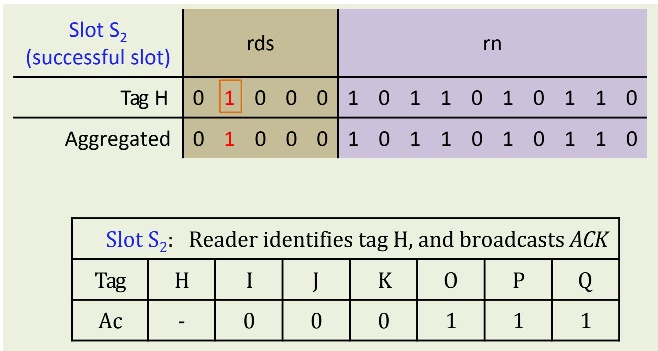
Figure 4.4: Tag identification in the MBTR phase.

fully identify the tag. Then, it broadcasts an *Ack* command to all the tags. If more than one tag reply, this is a duplicated collision slot caused by multiple tags transmitting the same $(rds,\ rn)$ message. Then the reader broadcasts an $NAck$ command. On the other hand, if the reader cannot successfully decode the $(rds,\ rn)$ string, this is a collision slot as illustrated at the bottom of Fig. 4.4(a). Then, the reader feeds back a rfs string by converting all the colliding bits to "1" in the aggregated message. In the feedback message, the tags will skip the idle slots by updating their Ac values.

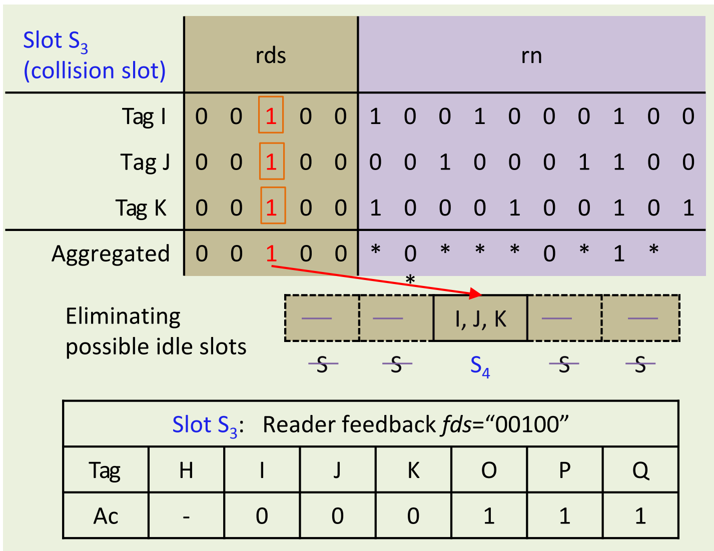
Besides, we also elaborate on the MBTR phase with an example demonstrated in Fig. 4.5. In the example, the parameters of the first three slots are given. In the example, slot S_1 in Fig. 4.5(a) is a collision slot caused by multiple tags replying with different $(rds,\ rn)$ messages. The table on the right shows the reader's feedback message and tags' Ac values at the end of this slot. Slot S_2 in Fig. 4.5 is a successful slot and the table at the bottom gives the tags' Ac values after this slot. Since tag H is recognized, it will remain silent until the end of the current reading cycle. Slot S_3 in Fig. 4.5(c) is a collision slot caused by multiple tags replying with the same rds but different rn messages. The table at



(a)



(b)



(c)

Figure 4.5: An example of the MBTR phase.

the bottom gives the tags' Ac values at the end of slot S_3. In this slot, all the tags keep their Ac values unchanged.

Assume that there are 7 unknown tags. After receiving the $Recog(\alpha,\ M)$ command, all the tags set $Ac = 0$ and $Tc = \alpha$. In the first slot S_1, all the tags will reply to the reader since their $Ac = 0$. The $(rds,\ rn)$ string of each tag is given in Fig. 4.5(a). After receiving the aggregated signal, the reader decodes it with Manchester decoding as "$0 * 0 * *$". From this message, the reader knows that there will be no tag responses in the 1st and 3rd slots after this slot, since the corresponding bits are "0"s. Then the reader sets fds = "01011" and feeds back $Rback(fds)$ to the tags. From the fds string, the tags know that a collision slot occurs. They update their Ac values as follows:

- A tag with $Ac = 0$ sets its $Ac = rs - CNT(rfs,\ rs,\ \text{"0"}) - 1$ to skip the idle slots. For instance, at the end of slot S_1, $Ac(\text{H}) = 2 - 1 - 1 = 0$, $Ac(\text{I}) = Ac(\text{J}) = Ac(\text{K}) = 4 - 2 - 1 = 1$, and $Ac(\text{O}) = Ac(\text{P}) = Ac(\text{Q}) = 5 - 2 - 1 = 2$ as shown in the right table of Fig. 4.5(a); and

- A tag with $Ac > 0$ sets $Ac = Ac + CNT(rfs,\ M + 1,\ \text{"1"}) - 1$ to reserve some slots for the colliding tags.

In slot S_2, since only tag H's $Ac = 0$, *the reader can decode the received message successfully* as shown in Fig. 4.5(b). Then the reader issues a *Collect* command to identify the tag. After successfully identifying tag H, the reader sets $Tc = Tc + 1$ and inserts $(ID(\text{H}),\ Tc)$ to ls_{new}. Then, the reader broadcasts *Ack* to all the tags, and the tags do the following operations.

- A tag with $Ac = 0$, such as tag H in slot S_2, sets $RI = Tc + 1$, and stores $(RID,\ C_i,\ RI)$ in its memory. Then the tag remains silent until the end of the current reading cycle;

- A tag with $Ac > 0$ updates $Tc = Tc + 1$ and $Ac = Ac - 1$.

If the tags receive an $NAck$ command, they will keep their Ac and Tc values unchanged. The reader repeats the recognition process in the following slots until there are no more tag responses. Then the reader replaces ls with ls_{new}, and terminates the current reading cycle C_i. The whole recognition process of

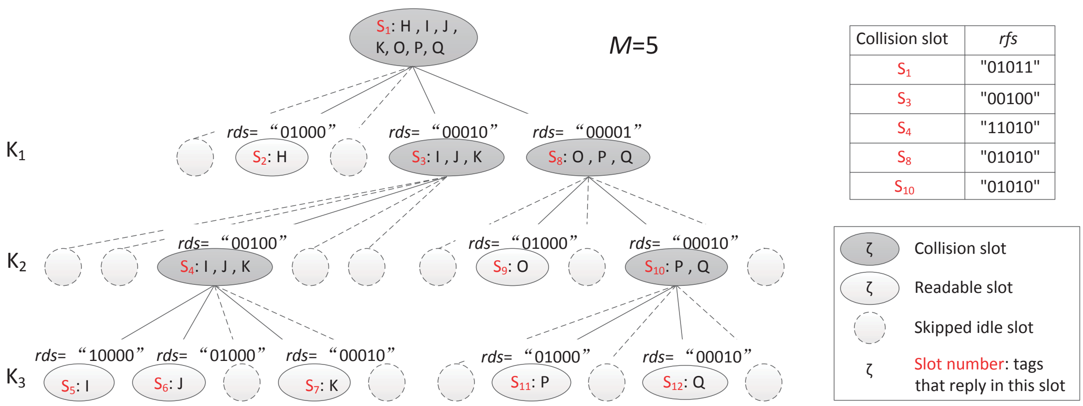


Figure 4.6: Identification tree structure of the MBTR phase.

the MBTR phase can also be represented by an M-ary tree structure as shown in Fig. 4.6, where the right corner of Fig. 4.6 gives the feedback messages of the collision slots. Table 4.2 illustrates the transmitted messages and counter values of each slot. As can be observed, only 13 slots are used to identify 10 unknown tags in the MBTR phase. Moreover, the pseudcodes of the MBTR phase at the reader and tags are given in Algorithms 3 and 4, respectively.

Algorithm 6 MBTR phase at the reader

Broadcast $Recog(\alpha,\ M)$.
while (1) **do**
    Broadcast $Qrep$ and wait for tag responses.
    if There is no tag response **then** Replace ls with ls_{new} and terminate C_i.
    else Decode the received message.
        if rds and rn can be correctly decoded **then** Broadcast $Collect$ command.
            if There is only one tag response **then** $Tc = Tc + 1$, $ls_{new} = QueueIn(ls_{new},\ (ID,\ Tc))$, then broadcast Ack.
            else Broadcast $NAck$.
            end if
        else Generate rfs by converting all the colliding bits in rds into "1"s, and broadcast $Rback(rfs)$.
        end if
    end if
end while

Table 4.2: Parameters and transmitted messages at the reader and tag sides in the MBTR example

		Tag							Reader		
		H	I	J	K	O	P	Q	received rds	Feedback	Tc (RI)
Initiate	Ac	0	0	0	0	0	0	0			5
1 (collision)	Ac	0	0	0	0	0	0	0	“0*0**”	*Rback*(“01011”)	5
	rs	2	4	4	4	5	5	5			
2 (successful)	Ac	0	1	1	1	2	2	2	“00100”	*Collect*, *Ack*	6
	rs	3									
3 (collision)	Ac	–	0	0	0	1	1	1	“00100”	*Rback*(“00100”)	6
	rs		3	3	3						
4 (collision)	Ac	–	0	0	0	1	1	1	“0*0**”	*Rback*(“01011”)	6
	rs		1	2	4						
5 (successful)	Ac	–	0	1	2	3	3	3	“00001”	*Collect*, *Ack*	7
	rs	1									
6 (successful)	Ac	–	–	0	1	2	2	2	“10000”	*Collect*, *Ack*	8
	rs			5							
7 (successful)	Ac	–	–	–	0	1	1	1	“01000”	*Collect*, *Ack*	9
	rs				4						
8 (collision)	Ac	–	–	–	–	0	0	0	“0*0*0	*Rback*(“01010”)	9
	rs					2	4	4			
9 (successful)	Ac	–	–	–	–	0	1	1	“00001”	*Collect*, *Ack*	10
	rs					1					
10 (collision)	Ac	–	–	–	–	–	0	0	“0*0*0”	*Rback*(“01010”)	10
	rs						2	4			
11 (successful)	Ac	–	–	–	–	–	0	1	“10000”	*Collect*, *Ack*	11
	rs						5				
12 (successful)	Ac	–	–	–	–	–	–	0	“01000”	*Collect*, *Ack*	11
	rs							4			
13 (Terminate)	Ac	–	–	–	–	–	–	–			

4.5 Performance Analysis

In this section, we analyze the performance of the proposed protocol in terms of the total number of slots, and the total identification time needed to recognize all the tags (including n known, and β unknown tags) in the i-th reading cycle C_i. Before presenting a detailed analysis, some symbols and parameters used in the analysis are defined in Table 4.3.

Algorithm 7 MBTR phase at the tags

After receive $Recog(\alpha,\ M)$, a tag sets $Tc = \alpha$, $Ac = 0$.
After receive $Qrep$ command, a tag checks:
if $Ac == 0$ **then** Generate random numbers rs, rn and detecting string rds, where $rs \in [1,\ M]$, $rn \in [0, 2^{10-1}]$, and $rds = DSG(M,\ rs+1)$. Then reply $(rds,\ rn)$.
end if
After receive $Collect$ command, a tag checks:
if $Ac == 0$ **then** Reply its ID.
end if
After receive $Rback(rfs)$
if $Ac == 0$ **then** $Ac = rs - CNT(rfs,\ rs,\ \text{“0”}) - 1$.
else $Ac = Ac + CNT(rfs,\ M+1,\ \text{“1”}) - 1$.
end if
After receive Ack command, a tag checks:
if $Ac == 0$ **then**
    Set $RI = Tc + 1$, record $tRID$, tRC, RI in its internal memory, and remain silent until the next reading cycle.
else $Tc = Tc + 1$, $Ac = Ac - 1$.
end if

In EBD, the number of message bits transmitted by a tag varies in different types of slots. Denote by t_{id} the time needed to transmit a 96-bit ID message and a 4-bit string is used to transmit the commands used in our protocol. As can be seen from the link-timing defined in Figs. 4.2 and 4.4, we have

$$T_{monit} = (8 + 2L + T_1 + T_2)t_{id}/96,$$

$$T_{succ} = (118 + M + 2T_1 + 2T_2)t_{id}/96,$$

$$T_{dupl} = (118 + 2M + 2T_1 + 2T_2)t_{id}/96,$$

$$T_{coll} = (18 + 2M + T_1 + T_2)t_{id}/96.$$

Table 4.3: Symbols and their definitions

Symbol	Definition
$MT(n)$	Time needed to monitor the presence of n known tags in KTBM phase.
T_{monit}	Time of a monitoring slot in KTBM phase.
$S_{monit}(n)$	Number of slots needed to monitor n known tags in KTBM phase.
$RT(\beta)$	Recognition time needed to recognize β unknown tags in MBTR phase.
T_{coll}	Time of a collision slot in MBTR phase.
T_{succ}	Time of a successful slot in MBTR phase.
T_{dupl}	Time of a duplicated collision slot in MBTR phase.
$S_{succ}(\beta)$	Number of successful slots needed in MBTR phase.
$S_{coll}(\beta)$	Number of collision slots costs in MBTR phase.
$S_{dupl}(\beta)$	Number of duplicated collision slots generated in MBTR phase.

In the KTBM phase, each known tag is represented by one bit position in the superimposed tag responses. So the number of slots needed to monitor all the known tags is $S_{monit}(n) = \lceil n/L \rceil$. Compared with existing work in the literature, this work uses only a few number of slots to monitor hundreds even thousands of known tags. The time needed to monitor n tags is given by

$$MT(n) = S_{monit}(n)T_{monit} = \left\lceil \frac{n}{L} \right\rceil \left(\frac{8 + 2L}{96}\, t_{id} + T_1 + T_2 \right). \tag{4.1}$$

In the MBTR phase, the identification process behaves like an M-ary tree, in which all the idle slots are eliminated as illustrated in Fig. 4.6. Since the identification process is stochastic, the identification time is an average value. To obtain the average identification time, we first analysis the average number of slots needed in the MBTR identification phase. In the M-ary tree, there are at most M^K slots at the Kth level. As every tag randomly chooses a slot at each level, the probability that i out of β tags choose the same slot at level K can be computed as per the following binomial distribution

$$P(i/\beta,\ K) = \binom{\beta}{i} p^i (1-p)^{\beta-i}, \tag{4.2}$$

where $p = M^K$. Considering a single slot at level K, the probabilities that no tag, only one tag, and more than one tags choose this slot are denoted by $P_{idle}(\beta,\ K)$, $P_{succ}(\beta,\ K)$, and $P_{coll}(\beta,\ K)$, respectively. From (4.2), we arrive at

$$P_{idle}(\beta,\ K) = (1 - \frac{1}{M^K})^{\beta}, \tag{4.3}$$

$$P_{succ}(\beta,\ K) = \frac{\beta}{M^K}\left(1 - \frac{1}{M^K}\right)^{\beta-1}, \tag{4.4}$$

$$\begin{aligned} P_{coll}(\beta,\ K) &= 1 - P_{idle} - P_{succ} \\ &= 1 - \left(1 - \frac{1}{M^K}\right)^{\beta} - \frac{\beta}{M^K}\left(1 - \frac{1}{M^K}\right)^{\beta-1}. \end{aligned} \tag{4.5}$$

From Section 4.4.3, we know that a duplicated collision slot, which costs almost the same time as a successful slot, may occur if multiple tags transmit the same $(rds,\ rn)$ simultaneously. Suppose there are i ($i \geq 2$) tags that will reply in a slot simultaneously. The probability that the i tags generate the same $(rds,\ rn)$ messages is

$$P_{same}(i) = \left(\frac{1}{M}\right)\left(\frac{1}{M}\right)^i\left(\frac{1}{2^{10}}\right)\left(\frac{1}{2^{10}}\right)^i = \left(\frac{1}{2^{10}M}\right)^{i-1}. \tag{4.6}$$

Based upon (4.2) and (4.6), the probability that a duplicated collision slot occurs at the Kth level is

$$\begin{aligned} P_{dupl}(\beta,\ K) &= \sum_{i=2}^{\beta} P(i/\beta, K) P_{same}(i) \\ &= \sum_{i=2}^{\beta} \binom{\beta}{i} \left(\frac{1}{M^K}\right)^i \left(1 - \frac{1}{M^K}\right)^{\beta-i} \left(\frac{1}{2^{10}M}\right)^{i-1}. \end{aligned} \tag{4.7}$$

Suppose K can be as large as ∞, the numbers of collision and duplicated collision slots in the MBTR phase can be obtained separately as in the following.

$$
\begin{aligned}
S_{coll}(\beta) &= \sum_{K=0}^{\infty} \sum_{i=0}^{M^K-1} P_{coll}(\beta,\ K) \\
&= \sum_{L=0}^{\infty} M^K \left\{ 1 - \left(1 - \frac{1}{M^K}\right)^{\beta} - \frac{\beta}{M^K}\left(1 - \frac{1}{M^K}\right)^{\beta-1} \right\},
\end{aligned} \tag{4.8}
$$

$$
\begin{aligned}
S_{dupl}(\beta) &= \sum_{K=0}^{\infty} \sum_{i=0}^{M^K-1} P_{dupl}(\beta,\ K) \\
&= \sum_{K=0}^{\infty} M^K \left\{ \sum_{i=2}^{\beta} \binom{\beta}{i} \left(\frac{1}{M^K}\right)^{i} \left(1 - \frac{1}{M^K}\right)^{\beta-i} \left(\frac{1}{2^{10}M}\right)^{i-1} \right\}.
\end{aligned} \tag{4.9}
$$

The total number of slots used in this phase is $S_{succ}(\beta) + S_{dupl}(\beta) + S_{coll}(\beta)$. Since the number of successful slots equals the number of unknown tags, average time needed to recognize β tags in the MBTR phase is

$$
\begin{aligned}
RT(\beta) =& T_{succ}S_{succ}(\beta) + T_{dupl}S_{dupl}(\beta) + T_{coll}S_{coll}(\beta) \\
=& \beta T_{succ} + T_{dupl}S_{dupl}(\beta) + T_{coll}S_{coll}(\beta).
\end{aligned} \tag{4.10}
$$

In fact, the value of $S_{dupl}(\beta)$ is very small. Taking the example in Section 4.4, where $M = 5$, we can obtain $S_{dupl}(\beta) \approx 6.065 \times 10^{-5}\beta$ through numerical method. Ignoring the influence of duplicated slots on the identification time of the MBTR phase, we have

$$
\begin{aligned}
RT(\beta) \approx& T_{succ}\beta + T_{coll}S_{coll}(\beta) \\
=& \beta\left(\frac{M+118}{96}\ t_{id} + 2T_1 + 2T_2\right) \\
&+ S_{coll}(\beta)\left(\frac{2M+18}{96}\ t_{id} + T_1 + T_2\right).
\end{aligned} \tag{4.11}
$$

As can be seen from (4.11), the average identification time of MBTR partly depends on the number of collision slots S_{coll}, which is affected by the branch number M. Table 4.4 lists some numerical values of S_{coll} with M ranging from 2 to 15. Since the analysis uses an infinite sum for the number of levels K, we calculate S_{coll} in (4.8) using the mathematical methods introduced in [12, 87].

From Table 4.4, one can infer that with the increase of the branch number M, the number of collision slots decreases, whereas the duration of each slot

Table 4.4: Relationship between M and S_{coll}

M	2	3	4	5	6	7	8
S_{coll}	1.4407β	0.9098β	0.7211β	0.615β	1.5594β	0.5139β	0.4772β
M	9	10	11	12	13	14	15
S_{coll}	0.4525β	0.4366β	0.4237β	0.4105β	0.3961β	0.3811β	0.3665β

increases. Therefore, M should be carefully selected to achieve the best time performance. To obtain the optimal value of M, Fig. 4.7 plots both the analytical and simulation results of the identification time in the MBTR phase. As in [19], the data rate between the reader and tags is 62.5 Kbps, the time used to transmit a 96-bit ID $t_{id} = 2.4$ milliseconds, and the reaction time T_1 and T_2 are ignored. In Fig. 4.7, the branch number M ranges from 2 to 16, and the number of unknown tags $\beta \in \{1000,\ 5000,\ 10000\}$. Fig. 4.7 demonstrates that the simulation results agree well with the analytical values. When $M = 4$, it costs MBTR the least time to identify all the unknown tags. Hereafter, we adopt $M = 4$ in the MBTR phase.

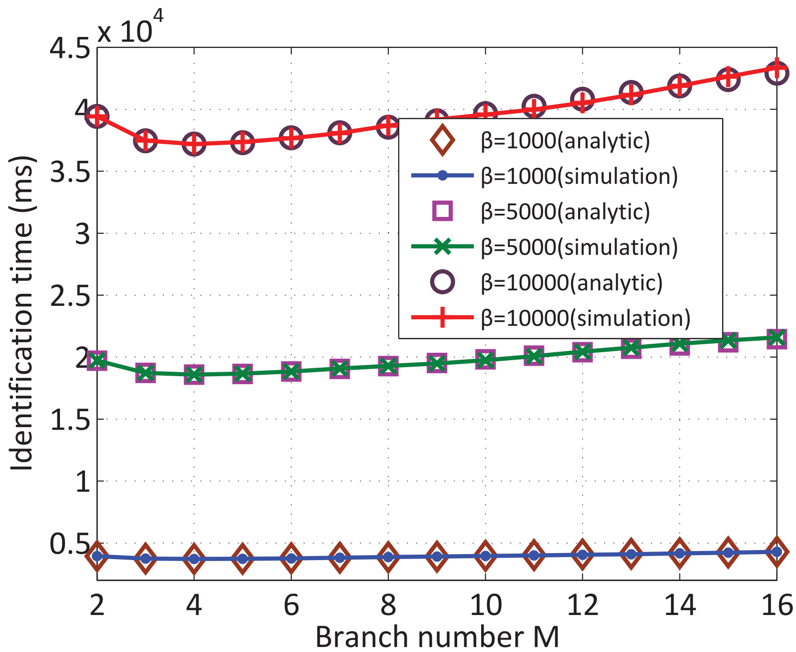


Figure 4.7: Identification time of the MBTR phase vs. M.

4.6 Simulation Results

In this section, we first evaluate the performance of the proposed EBD protocol in three aspects, namely the known tag monitoring time of the KTBM phase, the unknown tag recognition time of the MBTR phase, and the overall identification time to recognize all the known and unknown tags within the reader's reading range. In the simulation, we also compare our protocol with the most related and benchmark protocols. In the MBTR phase, the optimal branch number $M = 4$ is chosen.

4.6.1 System Settings

In the simulation, we consider a typical mobile RFID system, which is comprised of a reader and a lot number of RFID tags. Each tag has a unique ID with length of 96 bits. In this system, the reader maintains a list to record the IDs and the recognition indices of n known tags upon completing the reading cycle C_{i-1}. In C_i, α ($\alpha \in [0,\ n]$) staying tags are within the reader's reading range, and $n - \alpha$ tags are missing. Meanwhile, β unknown tags newly move into the reader's reading range. Our goal is to identify all the tags within the reader's range in the current reading cycle C_i as soon as possible.

Similar to [19], the data rate between the reader and tags is 62.5 Kbps, the time used to transmit a 96-bit tag ID $t_{id} = 2.4$ ms, and an empty slot takes 0.184 ms. The reaction time of the reader and tags are too small to be meaningful. The communications between the reader and the tags are assumed to be error-free as in the literature [26, 61]. The simulations are conducted using Matlab, and each test is averaged over 100 trials.

4.6.2 Evaluation and Comparison Configurations

The performance evaluation contains of three main parts, i.e., the known tag monitoring time, the unknown tag recognition time and the total time for moving tag identification. For a fair comparison, the performance of the proposed EBD protocol is compared with that of some benchmark protocols listed below.

- We compare the known tag monitoring time of EBD with that of the pair resolution blocking ABS (PRB) protocol [60], and the multi-pairing unknown tag identification (MUIP) protocol [62], which are the most related protocols. We also compare it with the best tag monitoring protocol in the literature, i.e., the slot filter-based missing tag identification (SFMTI) protocol [19]. Before the recognition process of SFMTI, we suppose the reader has successfully distinguished the known tags from the unknown ones.

- We compare the simulated unknown tag recognition time of EBD with the analytical value. Next, we compare it with some state-of-the-art tag ID collecting protocols, such as the multiframe maximum-likelihood (MFML) protocol[2] [44], and the adaptive assigned tree slotted Aloha (AdATSA) protocol [88].

To evaluate the overall identification time for mobile systems, different configurations are used for low-mobility tag monitoring and fast moving tag identifications as follows.

- In the low-mobility tag monitoring applications, all the tags in range will participate in C_i, and the tag set are assumed to be unchanged during the reading process in C_i. In this scenario, we compare the proposed EBD protocol with the most related PRB, MUIP protocols, and the combination of the best tag monitoring and tag ID collecting protocols, i.e., SFMTI+AdATSA.

- In the fast moving tag identification applications, the known tags will be muted in the following reading cycles. In such scenario, the performance of the proposed EBD protocol is compared with the most related AdATSA, MFML and PRB protocols.

4.6.3 Known Tag Monitoring Time

To evaluate the known tag monitoring time, the following three scenarios are used in the simulation:

[2]To show the best performance of the Aloha-based protocols, the prior knowledge of the tag number is given at the beginning of MFML.

a) Firstly, we evaluate the performance of the comparative protocols with an increasing number of known tags. In this scenario, we set $\beta = 2000$, $\alpha = n$, and n ranges from 0 to 10000;

b) Secondly, the impact of the number of staying tags on the performance of the comparative protocols is evaluated. For this scenario, we have $n = 10000$, $\beta = 2000$, and α ranges from 0 to 10000;

c) Finally, the impact of the number of unknown tags is measured. For this scenario, we have $n = 10000$, $\alpha = 2000$, and β ranges from 0 to 10000.

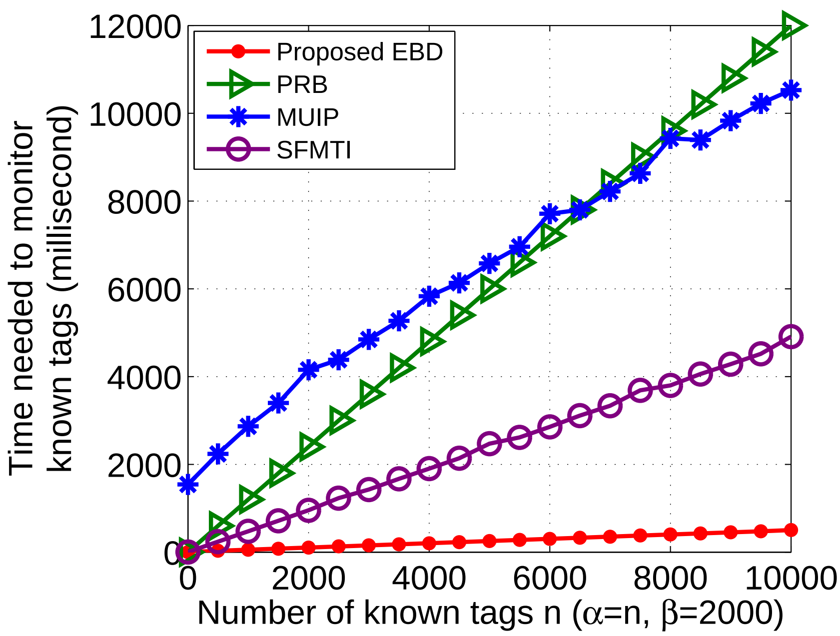


Figure 4.8: Known tag monitoring time vs. the number of known tags.

Firstly, Fig. 4.8 shows that the known tag monitoring time of all the four comparative protocols increases with the number of the known tags. In EBD, every known tag is represented by one bit position in the response message. As a result, it takes EBD $\lceil n/96 \rceil$ time slots to monitor the known tags, which is much fewer than other protocols. In SFMTI, the short message (such as 1 bit message) transmitted in each slot, and the slot reconciliation and filtering methods used in the recognition process greatly reduce the identification delay. Therefore, SFMTI takes less time than MUIP and PRB. However, SFMTI still generates many collision slots, so that it takes more time than the proposed EBD protocol.

Similar to SFMTI, MUIP also transmit short messages in each slot. However, the reader does not distinguish known tags from unknown tags before the recognition process. In the known tag monitoring phase, the unknown tags generate many collision slots resulting in increased identification delay. From Fig. 4.8, one can observe that when $\beta = 2000$, $n < 6500$, MUIP takes the most time to identify all the known tags. When $\beta = 2000$, $n > 6500$, the number of collision slots caused by the unknown tags becomes smaller, so that MUIP takes less time than PRB. In PRB, although the reader only needs $n/2$ slots to recognize all the known tags, the need to transmit the tags' IDs in each slot results in increased monitoring time. Therefore, PRB takes more time than EBD and SFMTI.

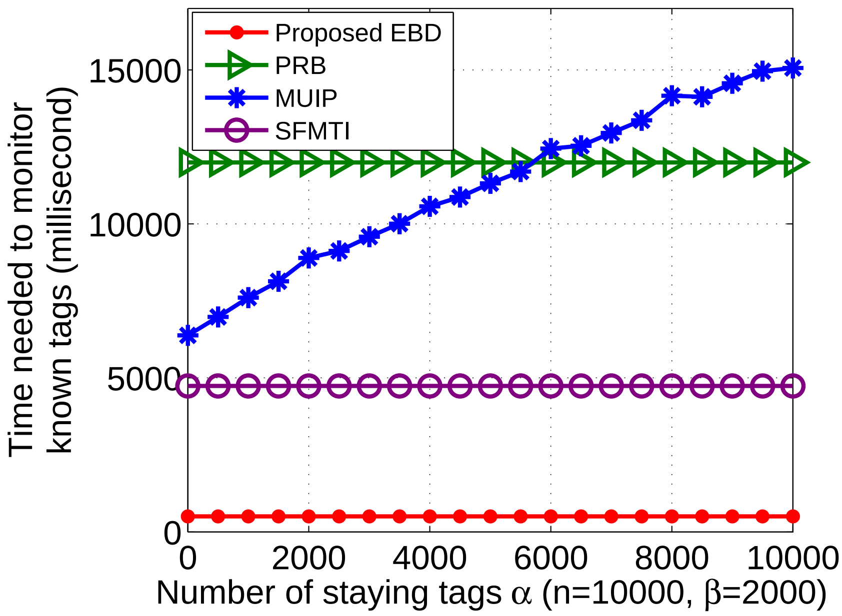


Figure 4.9: Known tag monitoring time vs. the number of staying tags.

Secondly, as can be observed from Figs. 4.9 and 4.10, the recognition times of EBD, PRB and SFMTI are constant when $n = 10000$. This is because EBD, PRB and SFMTI need to verify the state of all the known tags (including both staying and missing tags). Therefore, the known tag recognition time of them is only affected by the number of known tags. From these two figures, it can be concluded that the proposed EBD takes the least time to verify all the known

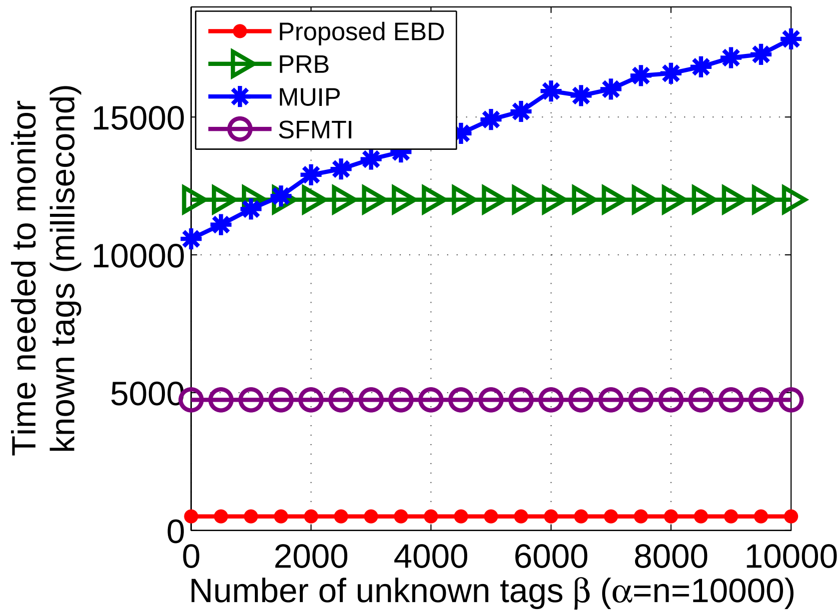


Figure 4.10: Known tag monitoring time vs. the number of unknown tags.

tags. Furthermore, the monitoring time of MUIP increases with the numbers of staying and unknown tags as shown in Figs. 4.9 and 4.10, respectively. Note that MUIP needs an extra missing tag monitoring protocol to detect the missing tags prior to the recognition process of MUIP in scenario b). Therefore, it takes MUIP more time in Fig. 4.9 than that in Fig. 4.8.

4.6.4 Unknown Tag Recognition Time

The simulation results for unknown tag identification of the comparative protocols are compared in Fig. 4.11. From this figure, one can observe that the simulation results of EBD are in good agreement with the corresponding analytic results. Compared with MFML and AdATSA, the MBTR phase of the proposed EBD protocol takes the least average time to identify a tag. Moreover, it is observed that the performance of EBD is more stable nearly independent of the number of unknown tags. The reasons are as follows.

Firstly, the M-ary tree structure and the bit-detecting mechanism used in the MBTR phase of EBD can not only reduce the message bits transmitted in each slot, but also reduce the number of slots needed in the identification process by eliminating the idle slots. As a result, it takes EBD less average time to identify

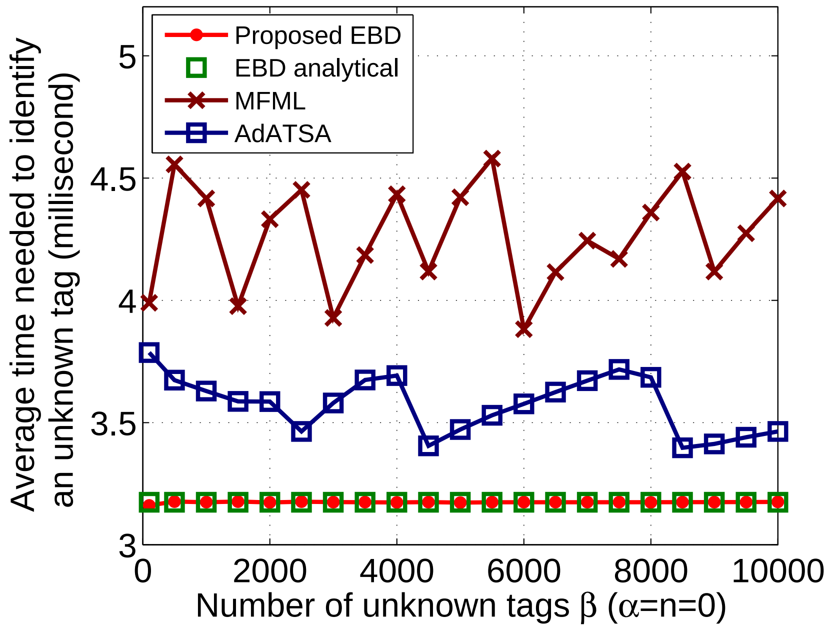


Figure 4.11: Average identification time for unknown tags vs. the number of unknown tags.

an unknown tag than MFML and AdATSA.

Secondly, with a constant branch number $M = 4$, the average time needed to identify an unknown tag using EBD is constant, which agrees with the analytical results in Section 4.5. On the other hand, MFML and AdATSA use the estimated frame length to achieve the optimal performance. Only when the actual number of unknown tags is close to the estimated frame length, these protocols show better performance. Therefore, the simulation curves of MFML and AdATSA fluctuate with the number of unknown tags.

4.6.5 Total Time for Moving Tag Identification

4.6.5.1 Tag Monitoring in Low-Mobility Applications

Firstly, when there are no leaving tags, the total identification time of all the comparative protocols increases with the number of known tags as can be seen from Fig. 4.12. Specially, the identification time of the proposed EBD protocol increases very slowly with n increased from 0 to 10000. This is because EBD uses one bit to represent a known tag in the known tag monitoring phase. When $n = 10000$, only 105 slots are needed to monitor all the known tags.

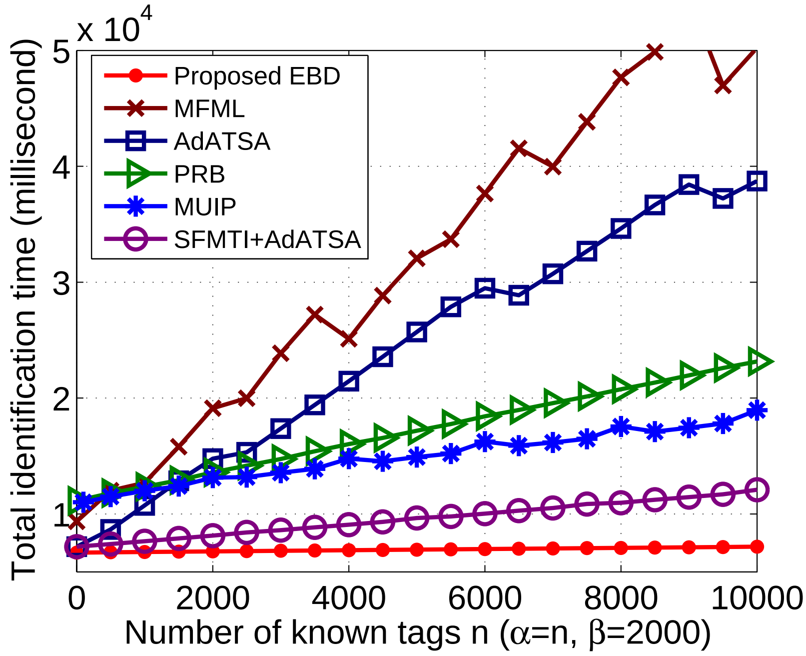


Figure 4.12: Total identification time vs. the number of known tags.

With *a prior* knowledge of the known tags, it takes EBD, PRB, MUIP and SFMTI+AdATSA much less time to recognize all the tags than MFML, and AdATSA, especially when n becomes large. However, when n is very small (i.e., $n < 1000$), it takes MUIP and PRB more time to recognize all the tags than AdATSA as shown in Fig. 4.12. This is because the basic binary tree and Aloha protocols used in the unknown tag recognition phases of PRB and MUIP are much less efficient than that in AdATSA. As a result, if the number of known tags is very small relative to the number of unknown tags, PRB and MUIP cost more time than AdATSA.

Secondly, Fig. 4.13 shows how the total identification time varies with the number of staying tags α. In this comparison scenario, it also takes the proposed EBD protocol the least time to identify all the tags compared with the other protocols. Note that the known tag monitoring times of EBD, PRB and SFMTI+AdATSA depend only on the number of known tags t. When n and β are invariant, the times needed to identify all the tags using EBD, PRB and SFMTI+AdATSA are constant as can be seen from Fig. 4.13.

Finally, Fig. 4.14 shows that the total identification time of all the comparative protocols increases with the number of unknown tags β. Similarly, the

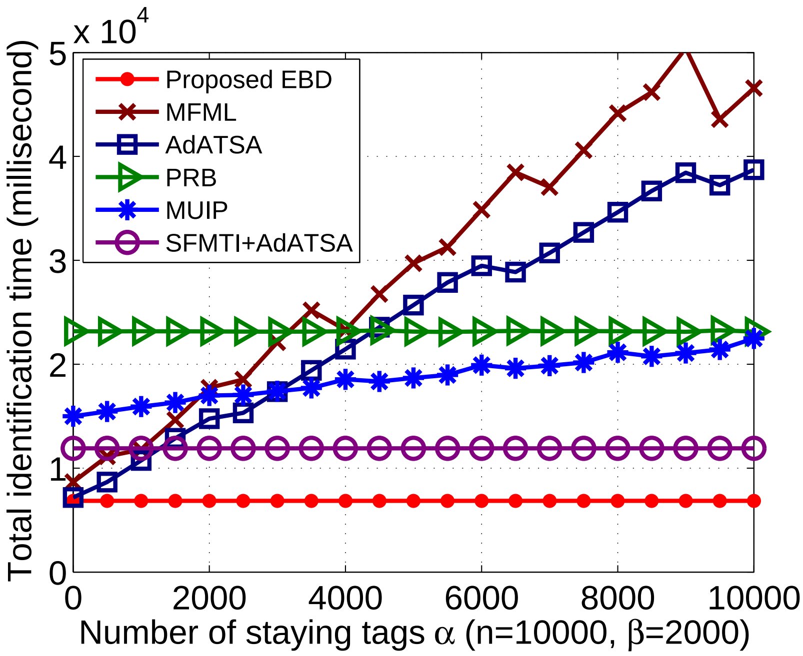


Figure 4.13: Total identification time vs. the number of staying tags.

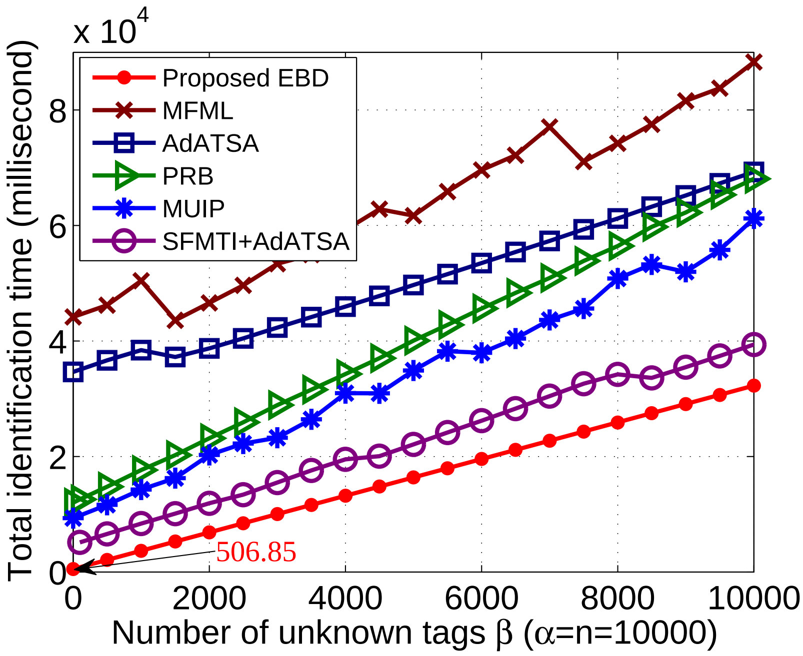


Figure 4.14: Total identification time vs. the number of unknown tags.

proposed EBD protocol always takes the least time to identify all the tags. Most notably, when $\beta = 0$, it takes EBD 506.85 milliseconds to monitor 10000 known tags, which is more efficient than all the other comparative protocols.

4.6.5.2 Tag Identification in High Mobility Applications

In the simulation, tags are uniformly distributed in the conveyor belt, and pass through the reader's three meters reading range with a speed of v meters/second. The tag density is d tags/meter. The location coordinates of all the tags are recorded and periodically updated during the identification process. In each slot, the corresponding tags, which meet the reader's requirement, reply to the reader only when their coordinates are within the reader's reading range. If a tag's coordinate is outside of the reading range, the tag will not participate in the following identification process. If a tag moves out of the reading range without being recognized by the reader, this tag is lost. The tag lost ratio is defined as the percentage of lost tags against all the tags passing through the reader's reading range. When the conveyor operates in a higher speed, the tags have less time to stay in the reader's reading range, resulting in a greater probability that a tag will not be read by the reader.

To evaluate the performance of tag identification protocols in mobile conveyor systems, two metrics, i.e., throughput and tag lost ratio, are used as in [65]. The throughput is the average number of tags identified per second. We evaluate the performance of the comparative protocols when the incoming tag flow vd ranges from 0 to 600 tags/second. The conveyor speed is set to be $v = 5,\ 10$ meters/second. The simulation results are given in Figs. 4.15 and 4.16.

As can be observed from Fig. 4.15, the proposed EBD protocol always has the highest throughput when the tag flow $vd \geq 200$. When $vd = 300$, EBD can recognize up to 280 tags/second, which is 12%, 56% and 87% higher than that of AdATSA, MFML, and PRB, respectively. When $vd > 300$, the throughput of EBD drops slightly with the increase of the tag flow, but always maintains above 270 tags/second. However, the throughput of AdATSA drops quickly after reaching its highest throughput. By comparison, the throughputs of MFML and PRB are around 175 and 150 when $vd \geq 200$, respectively.

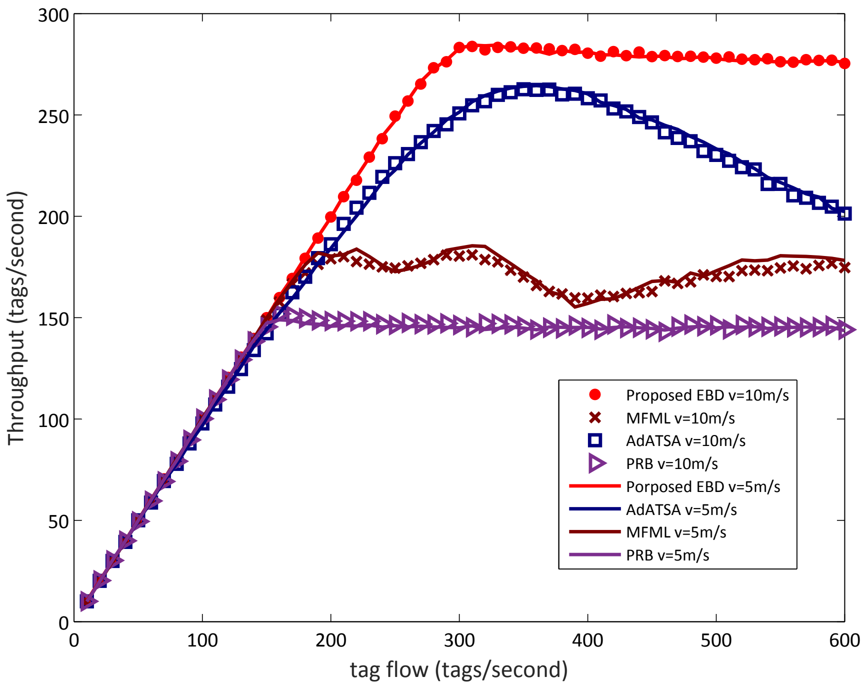


Figure 4.15: Tag identification for fast moving tag applications: throughput (number of tags identified per second) versus the incoming tag flow (number of tags pass through the reading range per second).

Secondly, Fig. 4.16 gives the comparison results of the tag lost ratio. As is shown, the proposed EBD protocol always has the least tag lost ratio compared with the other protocols. However, the PRB protocol has the highest tag lost ratio. When the tag flow $vd = 160$, the tag lost ratios of PRB and MFML are about 10^{-3} and 10^{-6}. However, the tag lost ratios of AdATSA and EBD are much less than 10^{-6}. When $vd = 260$, the tag lost ratios of EBD, AdATSA, MFML, and PRB are 10^{-7}, 10^{-3}, 0.2, and 0.3, respectively. Generally, the reliability of the proposed EBD protocol is much better than other comparative protocols.

In summary, the proposed EBD protocol outperforms all the other comparative protocols in mobile RFID systems. It can support a higher tag flow, while maintaining the highest throughput and the lowest tag lost ratio than its comparative counterparts.

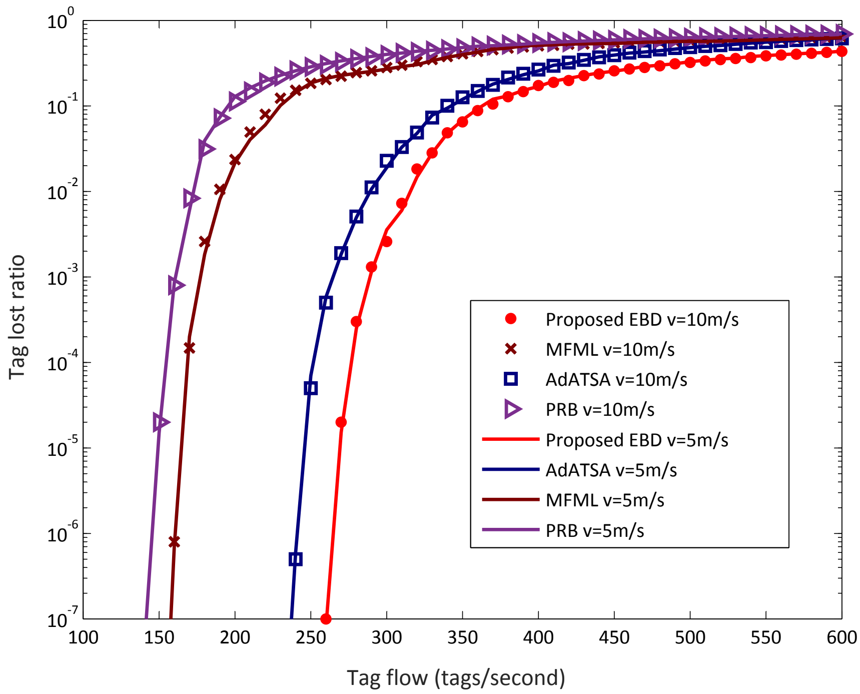


Figure 4.16: Tag identification for fast moving tag applications: tag lost ratio versus the incoming tag flow.

4.6.6 Computational Complexity

As is well known, low computational cost is of great significance to RFID tag identification applications, especially in large-scale RFID systems [44, 67]. The number of the floating point operations (FLOPs) at both the reader and tag sides is used as the chosen metric to compare the computational complexities of the proposed and comparative protocols.

In real world implementations, the FLOP depends strongly on the reader's digital signal processor. For fair comparison, we use the same values as in [44, 67], where a computational cost of 50 FLOPs is used for power, logarithm, and exponential operations, and 100 FLOPs for factorial operations. We also assume a computational cost of 300 and 1000 FLOPs for random number generation and hash operations, respectively. More specifically, Table 4 gives the computational cost for 1000 tag identification, where the known tag monitoring and unknown tag identification phases are considered separately.

For known tag monitoring, the proposed EBD protocol is compared with the most related PRB [60], MUIP [62] and SFMTI [19] protocols. As can be

Table 4.5: Computational costs at both the reader and tag sides

	Protocol	Reader's FLOP cost	Tag's FLOP cost
Known tag monitoring	Proposed EBD	1.40×10^4	6.49×10^2
	PRB [60]	1.51×10^3	5.01×10^2
	MUIP [62]	3.66×10^6	3.41×10^3
	SFMTI [19]	4.09×10^6	4.54×10^3
Unknown tag identification	Proposed EBD	1.38×10^4	6.53×10^3
	PRB [60]	7.00×10^3	9.00×10^3
	AdATSA [88]	2.50×10^4	4.56×10^3
	MFML [44]	3.40×10^5	7.42×10^4

observed from the left part of Table 4, the PRB and proposed EBD protocols have similar computational complexities at both the reader and tag sides. However, the complexities of the MUIP and SFMTI protocols are much larger, attributed to the complicated hash operations used in MUIP and SFMTI at both the reader and tag sides.

On the other hand, for unknown tag identification, the most related PRB [60], AdATSA [88], and MFML [44] protocols are used for comparison. As is shown on the right part of Table 4, the computational complexities at the tag side are almost the same for all the comparative protocols. However, at the reader side, PRB requires the least number of FLOPs. This is because that the reader uses only addition and subtraction operations in PRB. Due to extra bit operations, EBD and AdATSA are more computationally complex than PRB. MFML is shown to be more complex than the other comparative protocols. This is because the tag estimation process in MFML requires a large number of FLOPs.

Table 4.5 shows that PRB is of less time complexity than our protocol. However, the simulation results in Section 5 demonstrate that our protocol takes much less time for the communications between the reader and tags than the other comparative protocols. Since the time needed for the reader and tags' computational operations is far less than the time for communications, the overall time performance of our protocol is much better than that of PRB.

In summary, the computational complexities of the proposed EBD protocol at both the reader and tag sides are comparatively low. Thanks to higher identification efficiency, our protocol is able to outperform the other comparative protocols.

4.7 Effect of Detection Errors

In the paper, the communication channels between the reader and tags are assumed to be ideal as in the literature [26, 44, 52]. However, in realistic environments, detection errors may occur on the backscattered signals because of fading or capture effect. In passive RFID systems, detection errors are mainly caused by two reasons, i.e., transmission errors and unable to detect weak backscattering signals [63, 84, 85]. If detection errors occur, the original successful and collision slots will be transformed into other types of slots as illustrated in Fig. 4.17. The effect of each errors listed in the figure are discussed as follows.

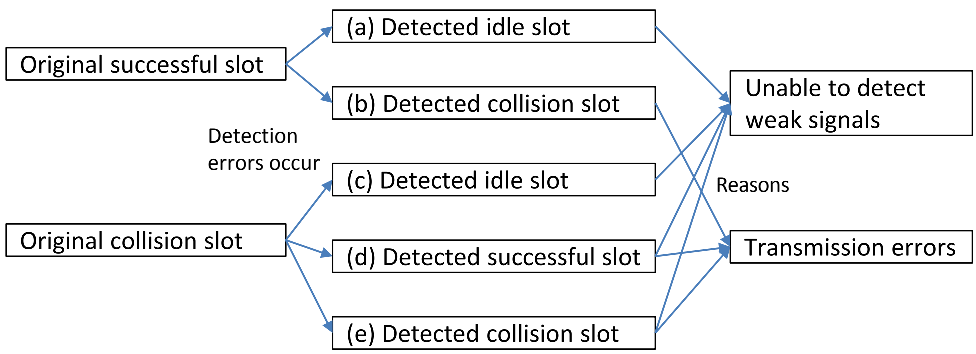


Figure 4.17: Slot transformation caused by detection errors.

- For some original successful slots: (a) if the tag's backscattering signal is too weak to detect, this slot will be transformed into an idle slot; (b) if the backscattered signal can be detected, but some transmission errors occur, the slot will be transformed into an collision slot. These errors increase the number of collision and idle slots, resulting in an increased identification delay.

- For some original collision slots: (c) if all the tags' response messages are lost, this slot will be transformed into an idle slot; (d) if the reader can successfully decode a tag's response message, this slot will be transformed into a successful slot, which improves identification efficiency. Finally, (e) if detection errors occur in some bit positions in an original collision slot, the detected slot is still a collision slot. Such a situation only reduces the identification efficiency of bit tracking protocols, such as AdATSA, OBTT, CwT, and so on. This is because these protocols need the bit information in collision slots to improve their performance [26,27,52,74,88]. However, this situation does not affect non-bit tracking protocols, such as the Aloha-based protocols, since they discard all the information in collision slots.

Generally speaking, situations (a)-(d) are common for most tag identification protocols. By contrast, the detection errors in situation (e) only affect bit tracking protocols, including ours. Nevertheless, our proposed protocol still outperforms all the previous work in imperfect channels.

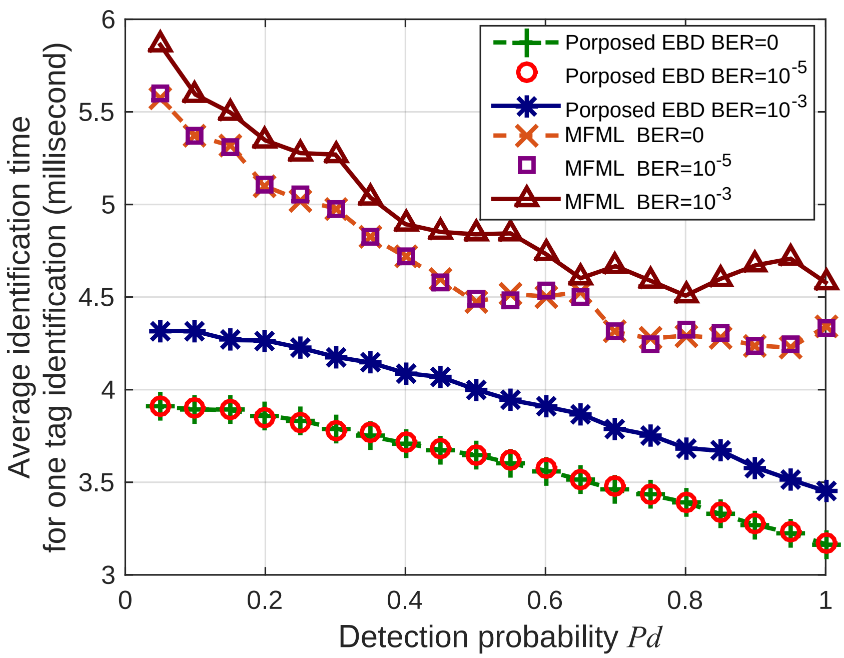


Figure 4.18: Effect of the detection errors: average identification time for one tag identification versus detection probability at various bit error rate.

To investigate the effect of detection errors, the performance of the proposed EBD protocol in imperfect channels is evaluated. The evaluation results are illustrated in Fig. 4.18, where the detection probability Pd ranges from 0.05 to 1, and the bit error rate BER $\in \{0,\ 10^{-5},\ 10^{-3}\}$. Moreover, the average identification time of the MFML protocol, one of the best non-bit tracking protocols, is also given for comparison. As can be observed from Fig. 4.18, the average identification time of both protocols increases when BER increases and Pd decreases. As illustrated, it takes EBD at least 20% less time than MFML under the same channel condition. It can be concluded that the proposed EBD protocol always exhibits a better performance than other comparative protocols even in imperfect channels.

4.8 Summary

In this chapter, we proposed an efficient bit-detecting protocol, dubbed the EBD protocol, to effectively monitor known tags and recognize unknown tags in large-scale mobile RFID systems. Theoretical analysis was carried out to determine the optimal branch number and evaluate the performance of EBD. Moreover, simulation results were also presented to demonstrate agreement between the analytical and simulation results and to verify that the proposed protocol performs better than the comparative protocols. Only large-scale warehouse management and conveyor belt applications are considered in this chapter, more diverse application, more diverse application scenarios with moving or mobile tags will be considered in the future.

Chapter 5

Time- and Energy-aware Collision Tree Protocol for Efficient Large-scale RFID Tag Identification

5.1 Introduction

Being able to provide a relatively easy and inexpensive way to collect data, portable RFID readers have gained increasing popularity in wide-ranging RFID applications. To maximise the reader's battery life, efficient tag identification protocols are of paramount importance in large-scale passive RFID systems. The reasons are twofold. On the one hand, the reader usually needs to identify thousands of passive tags without any prior knowledge, i.e., the so-called unknown tag identification problem. Since tags have very limited communications and computational capabilities, rapidly identifying all the tags is a challenging issue. Energy saving for portable readers is also a big concern because in passive RFID systems, the reader needs to provide energy not only for its own operations, but also for all the tags around it [74]. Generally speaking, the energy cost in such systems consists of two components: the energy needed for powering all the tags and for exchanging messages between the reader and tags. The former is related to the identification time, whereas the latter depends on the number of message bits transmitted. Therefore, to maximise the battery life, both the identification time and the number of message bits transmitted should be as low as possible.

Previous works concentrate either on saving energy for active RFID systems or on reducing identification times for passive RFID systems. They are not suitable for a passive system with a portable reader. In this chapter, we propose an efficient collision tree protocol to effectively reduce both the time and energy costs in passive RFID systems. In general, the new protocol makes use of the first $\log_2 M$ colliding bits in the aggregated message to effectively split colliding tags into M smaller groups. With the information of more colliding bits, the probability that a new group is occupied with only one tag increases, which reduces the overall number of slots and message bits transmitted by the tags. Moreover, the frame structure is also used to further reduce the number of message bits transmitted by the reader. In the frame structure, time is divided into frames and each frame consists of M slots. Since the query prefix is only transmitted at the beginning of each frame, and in each slot a short slot start command is used, the number of message bits transmitted by the reader is further reduced. With these techniques, our protocol can effectively reduce the time and energy costs.

To summarise, the major contributions of this work are threefold:

- A new M-ary collision tree (MCT) protocol is proposed to reduce the number of collision slots and transmission message bits. Due to its easy implementation, the new MCT protocol can reduce the average identification time and energy consumption by at least 16.12% and 15.73% respectively, compared with the benchmark protocols.

- A theoretical analysis is conducted to investigate the number of collision and total slots required for identifying n tags. Numerical results demonstrate that the proposed MCT protocol takes fewer collision slots than other comparative protocols.

- The impact of unreliable channels on the performance of our work is analysed. The time and energy costs of the proposed protocol with various detection and capture probabilities are evaluated.

The remainder of this chapter is organized as follows. Section 5.2 gives some preliminary information, such as the system model, Manchester coding method and the communication link-timing between the reader and tags. In Section 5.3, the proposed protocol is described in detail. Next, a theoretic analysis is given in Section 5.4, followed by the simulation and comparison results presented in Section 5.5. Section 5.6 discusses the effect of the practical environment. Finally, Section 5.7 draws concluding remarks.

5.2 Preliminary

In this chapter, we consider a typical large-scale RFID system consisting of one reader and a large number of passive tags. Prior to the identification process, the reader has no prior knowledge about the number of tags within its reading range. The reader also does not know the ID of any tag. This is a classical unknown tag identification problem that the reader has to collect all the tags' IDs without any prior knowledge. Since a passive tag has very limited communications and computational resources, the communications between the reader and tags employ the reader-talk-first model.

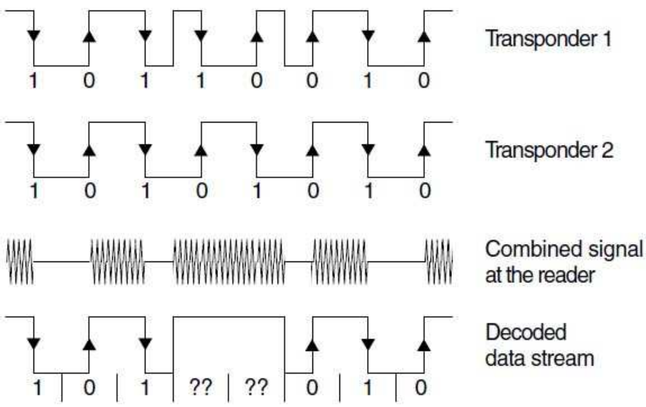


Figure 5.1: Example of Manchester coding, where ? indicates the invalid symbol.

In the system, Manchester coding is used to encode tag response messages, where a logic 0 and 1 are coded by a positive and negative transition in level within a bit window, respectively. If two (or more) transponders simultaneously transmit bits of different value then the positive and negative transitions of the received bits negate the effect of each other, which leads to an invalid symbol as shown in Fig. 5.1. Being capable of detecting the positions of colliding bits, Manchester coding has been widely used in a great number of RFID tag identification protocols to accelerate the recognition process [24, 25, 28, 52, 74], and the ISO/IEC 14443 standard also specifies this coding method for type A tags [89]. Since Manchester coding requires bit level synchronization among tag responses, the bit oriented anti-collision frame is defined to obtain the position of the colliding bit in the ISO/IEC 14443 standard. In such a frame, all the responding tag responses are synchronous.

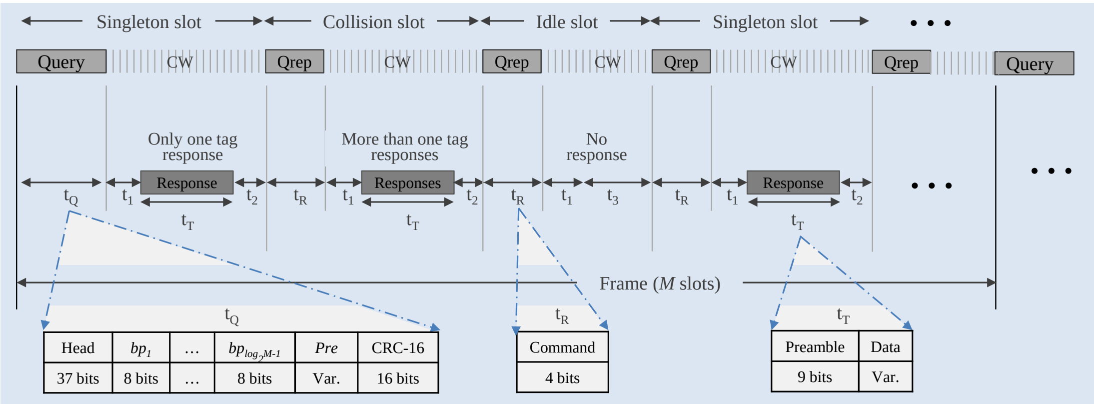


Figure 5.2: Illustration of the link timing between the reader and tags.

The link timing between the reader and tags is illustrated in Fig. 5.2, where the three tables at the bottom depict the structures of the messages transmitted. In the identification process, time is divided into multiple frames, and each of which consists of M slots. Each frame and slot start with the reader's *Query* and *Qrep* commands, respectively. Especially, the first slot in each frame starts automatically after the *Query* command. As is shown in Fig. 5.2, in a *Query* command 37 bits are used for the header, mask, command and address information [27, 76], $\log_2 M - 1$ bit position (BP) indicators of 8-bit length are used to inform tags of the positions of the first $\log_2 M - 1$ colliding bits, a 16-bit CRC is used to check the correctness of the transmit message, and the lengths of matching prefixes vary in different frames. A 4-bit message is used to transmit the *Qrep* command, and a 9-bit preamble accompanying the tag ID are used in a tag response message. t_Q and t_R are the time taken to transmit the *Query* and *Qrep* commands, respectively. t_T is the time needed to transmit the tag responses in each slot. t_1 is the time taken from reader transmission to tag response. t_2 is the interrogator response time required if a tag is to demodulate the interrogator signal. t_3 is the time a reader waits after t_1 if there is no tag response.

In each frame, all the tags stay in one of the following two states, i.e., the waiting and transmission states.

- A tag stays in the *waiting state* if it does not match the prefix transmitted in the *Query* command. In such state, the tag will not reply to the reader in the current frame.
- A tag transits into the *transmission state* when it matches the prefix transmitted in the *Query* command. In such state, the tag will reply to the reader in one of the slots in the current frame.

According to the number of tag responses, we define the following three types of slots.

- *Singleton slot*: A singleton slot is one, in which only one tag replies to the reader. In such a slot, the reader can successfully decode the received message, and identify the tag.
- *Collision slot*: A collision slot is one, in which more than one tag reply to the reader simultaneously. In a collision slot, the reader cannot successfully decode the received signal from the garbled waveforms.
- *Empty slot*: An empty slot is one, in which no tag replies. If the reader does not receive any message after waiting time t_3, the reader considers this slot as an empty one.

According to the link timing diagram in Fig. 5.2, the time and energy models are given as follows.

Time model: The time taken to identify n tags consists of the time required for transmitting all the reader's request commands and tags' responses in each slot. Denote by $C_c(n)$, $C_s(n)$, and $C_e(n)$ the numbers of collision, successful and empty slots, respectively. The time for identifying n tags is given as follows.

$$\begin{aligned} T(n) \quad = & \sum_{i=0}^{[C_c(n)+C_s(n)+C_e(n)]/M} \left[t_{Q_i} + (M-1)t_R\right] \\ & + \sum_{j=0}^{C_c(n)+C_s(n)} (t_1 + t_2 + t_{T_j}) + \sum_{i=0}^{C_e(n)} (t_1 + t_3), \end{aligned} \tag{5.1}$$

where t_{Q_i} refers to the time for transmitting the reader's *Query* message in the i-th frame, and t_{T_j} is the time for transmitting the tags' response message in the j-th non-empty (i.e., successful or collision) slot.

Energy model: In passive RFID systems, the energy of a tag is harvested through backscattering the reader's signal. Thus, the reader needs to send carrier waves (CWs) throughout the identification process with power P_{tx}. In the period of receiving tag response messages, the reader needs extra power P_{rx}. Therefore, the energy cost is given as follows

$$\begin{aligned} E(n) = & \sum_{i=0}^{[C_c(n)+C_s(n)+C_e(n)]/M} P_{tx}[t_{Q_i} + (M-1)t_R] \\ & + \sum_{j=0}^{C_c(n)+C_s(n)} \left[P_{tx}(t_1 + t_{T_j} + t_2) + P_{rx}t_{T_j}\right] \\ & + C_e(n)P_{tx}(t_1 + t_3). \end{aligned} \tag{5.2}$$

5.3 Proposed M-ary Collision Tree Protocol

In principle, the proposed M-ary collision tree (MCT) protocol identifies tags by constructing a new M-ary tree structure with less number of collision and empty slots. Firstly, Table 5.1 lists some symbols and commands used in this chapter. It should be noted that *bp* indicates that the *bp*-th bit in *pre* is a colliding bit and will not be used for prefix matching. $mPre$ and tID are the two parameters generated by the tags, and used to control tag response. If a tag has $mPre = tID$, it will reply to the reader in the current frame.

5.3.1 M-ary Collision Tree

Before describing our protocol, we introduce the new M-ary collision tree structure. In traditional query tree protocols, if a collision slot occurs the reader obtains the common prefix *pre* by recording the bit string until the first colliding bit of the aggregated message. Then the reader splits the colliding tags in two groups, and the common prefixes of the tags in these groups are $pre||0$ and $pre||1$, respectively. By continuously splitting the colliding tags into smaller groups, a binary collision tree is constructed. Since the traditional method generates many collision slots, we try to build up an M-ary collision tree structure to accelerate the identification process. In our method, the position of the first $\log_2 M$ colliding bits is used.

Table 5.1: Symbol definitions

Symbol	Definition
ID	Tag's unique ID, and $ID(i)$ indicates the value of the i-th bit.
pre	Prefix (of l bits length) transmitted by the reader at the beginning of each frame.
bp_i	Bit position of the i-th colliding bit in pre, where $i \in [1, log_2 M - 1]$.
$mPre$	Matching prefix used to control tag replies. It is obtained by deleting all the colliding bits (e.g. the bp_i-th bit) in pre.
tID	Tag's matching ID, which is obtained by deleting all the bp_i-th bit in $ID(1:l)$ [1]
Q	Query queue maintained by the reader to record the frame parameters.
F_i	Frame index, i.e., the i-th frame in the recognition process.
S_x	Slot index, i.e., the $(x+1)$-th slot in the current frame, where $x \in [0,\ M-1]$.
$Query()$	Frame start command broadcast by the reader to start each frame and to inform the tags of the frame parameters.
$Qrep$	Slot start command broadcast by the reader to start each slot within a frame except for the first one, because the first slot starts automatically and immediately after the $Query()$ command.

In detail, when a collision slot occurs, the reader maintains the bit string until the $\log_2 M$-th colliding bit of the aggregated message. Next, it substitutes all the

[1] Here, $st(i)$ indicates the ith bit of string st, and $st(i:j)$ denotes the bit string from $st(i)$ to $st(j)$. If $j < i$, it represents an empty string ϵ. If $j =$ "end", it returns the bit string from the ith bit to the end of st.

colliding bits with bit "1" and records the new string as *pre*. It also records the bit positions of the first $\log_2 M - 1$ colliding bits as $bp_i, i \in [1, \log_2 M - 1]$. With these information, the reader splits the colliding tags into M smaller groups. Take $M = 4$ as an example. Suppose the aggregated message in a collision slot is "100?11?01?0". Then we have pre = "100$\underline{1}$11" and $bp_1 = 4$. The colliding tags can be split into four groups, and their common prefixes are "100$\underline{0}$11$\underline{0}$", "100$\underline{0}$11$\underline{1}$", "100$\underline{1}$11$\underline{0}$" and "100$\underline{1}$11$\underline{1}$". Note that the underlined bit positions are the positions of the first two colliding bits in the aggregated message. Repeatedly splitting tags into smaller groups until there is at most one tag, the reader can identify all the tags with an M-ary collision tree structure.

5.3.2 Protocol Descriptions

At the beginning of MCT, the reader initiates the query queue $Q = \big\{(1,\ 2, ..., \log_2 M - 1,\ \underbrace{\text{"11...1"}}_{\log_2 M - 1})\big\}$, which means in the first frame $bp_i = i$ ($i \in [1,\ \log_2 M - 1]$) and *pre* is a bit string of $\log_2 M - 1$ "1"s. It should be noted that each element in Q consists of $\log_2 M - 1$ bp_is and a bit string of *pre*. Then the reader broadcasts $Init(M)$ to inform tags of the branch number M. Next, the reader identifies tags frame-by-frame. In each frame, the reader first obtains the frame parameters through $(bp_1, bp_2, ..., bp_{log_2 M-1}, pre) = QueueOut(Q)$ [2], and then broadcasts $Query(bp_1, bp_2, ..., bp_{log_2 M-1}, pre)$ to inform tags with these parameters. Note that bp_i, $i \in [1, log_2 M - 1]$, indicates that the i-th bit of *pre* is not used for prefix matching.

In our protocol, the *Query* command consists a field *pre* and several additional fields bp_i. Without transmitting the length information of *pre*, the tags cannot successfully separate *pre* and bp_i. To deal with this problem, we use 8 bits to transmit each bp_i. Then the tag obtains the value of bp_i one-by-one after the 37 bits head information as shown in Fig. 5.2. Because the number of *bp*s is $\log_2 M - 1$ in the *Query* command, by checking the strings after the 37 bits head information and the $8(\log_2 M - 1)$ bits *bp*s in the command, the tag obtains *pre*.

[2] $QueueOut(Q)$ is the operation to determine the first element, and then delete it from queue Q, and $InQueue(Q, a)$ is the operation of inserting element a into queue Q with a first-in-first-out manner.

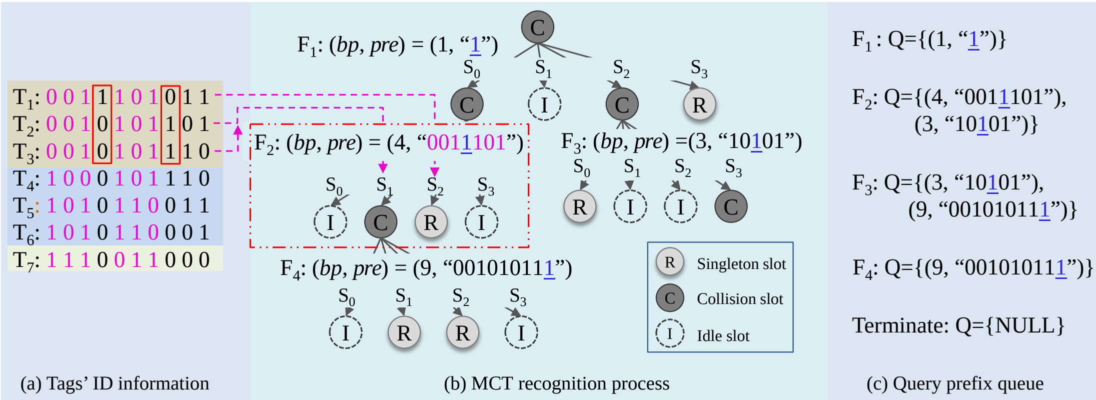


Figure 5.3: Example of the proposed MCT protocol, where $M = 4$ and the number of tags is 7.

Prefix matching at the tag side:

After receiving the reader's $Query(bp_1, bp_2, ..., bp_{log_2 M-1}, pre)$ command, the tags execute three steps as follows.

1) The tag obtains $mPre$ and tID by deleting all the bp_i-th bits in pre and $ID(1 : l)$, respectively. Note that l is the length of pre. Take $M = 4$ as an example. If $bp_1 = 4$ and $pre =$ "001$\underline{1}$101", then we have $mPre =$ "001101" and $tID = ID(1 : 3, 5 : 7)$;

2) The tag compares $mPre$ and tID, if they are the same the tag transits into the transmission state. Otherwise, it stays in the waiting state;

3) The tag in the transmission state converts $ID(bp_1, bp_2, ..., bp_{\log_2 M-1}, l+1)$ to a slot index S_x, and replies $ID(l + 2 : end)$ to the reader in the $(x + 1)$-th slot of the current frame. Follow the previous example, if the tag's $ID(4, 8) =$ "01", we have $x = 1$. Then the tag will reply in the second slot.

More specifically, Fig. 5.3 gives an example of the proposed protocol, where Fig. 5.3(a) gives the ID information of all the tags. Specially, we mark the tags' tIDs in frame F_2 with magenta colour. And the bits in the red boxes with solid lines are used to calculate the slot indices in frame F_2; Fig. 5.3(b) illustrates the tree structure of the identification process. The query parameters of each frame are given as $(bp,\ pre)$ pairs, and the slot index is given on the top of each circle. Since $M = 4$, we only need to transmit the bp value of the first colliding bit in

pre, i.e., the blue bits in pres; Fig. 5.3(c) gives the query queue Q maintained by the reader.

To elaborate on the above processes, we take the tags' operations in frame F_2 depicted in the red box with dashed lines in Fig. 5.3(b) as an example. In the example, tags' IDs are shown in Fig. 5.3(a), and the branch number $M = 4$. In frame F_2, $bp = 4$ and pre = "001$\underline{1}$101". By deleting the 4-th bit in pre and tags' IDs, we have $mPre$ = "001101", and $tID(T_1) = tID(T_2) = tID(T_3)$ = "001101", $tID(T_4) = tID(T_5) = tID(T_6)$ = "100101", $tID(T_7)$ = "111011", as marked with magenta colour in Figs. 5.3(a) and 5.3(b). Since only the tID values of tags T_1, T_2 and T_3 match with $mPre$, tags T_1, T_2 and T_3 will transit into the transmission state, and calculate their slot indices. Other tags will stay in the waiting state. By checking the 4-th and 8-th bits in tags IDs (i.e., the bits in red boxes in Fig. 5.3(a)), the slot indices of tags T_1, T_2 and T_3 are S_3, S_2 and S_2, respectively. Then tags T_2 and T_3 reply the last two bits of their IDs, e.g., "01" and "10", in the second slot, separately. And tag T_1 replies "11" in the third slot.

There is a special case. In the first frame, the reader broadcasts $Query(1,\ 2, ..., \log_2 M - 1,$ " $\underbrace{11...1}_{\log_2 M-1}$ "). Since the length of pre equals the number of bps, by deleting all the bp_i-th bits in pre and $ID(1:l)$, $mPre$ and tID become empty strings, i.e., $mPre = \epsilon$ and $tID = \epsilon$. In such case, all the tags have $mPre = tID$. Thus, they will transit into the transmission state and reply in the first frame. For example, in F_1 in Fig. 5.3(b) the frame parameters $(bp, pre) = (1,$ "1") indicate that there is only one colliding bit in pre. By deleting this bit in pre and $ID(1)$, all the tags obtain $mPre = tID = \epsilon$. Then, they all transit into the transmission state.

New prefix composing at the reader side:

In each slot, the reader receives the tags' responses. If there is no tag reply, an empty slot is declared. Otherwise, this is a non-empty (i.e., singleton or collision) slot. Then the reader decodes the received message with Manchester decoder as DM, and retrieves the maximum common prefix of each slot $comm$.

More specifically, the reader calculates the slot prefix $Spre = de2bi(x, log_2 M)$, and sets string $comm$ = "$pre(1 : bp_1{-}1)||Spre(1)||pre(bp_1{+}1 : bp_2{-}1)||Spre(2)||...$

$||pre(bp_{log_2 M-1} - 1 : l)||Spre(log_2 M)$" [3]. For example in fame F_2 of Fig. 5.3(b), $(bp, pre) = (4,$ "0011101"$)$. We have the maximum common prefixes of slots S_1 and S_2 are "001$\underline{0}$1011$\underline{1}$" and "001$\underline{1}$1010$\underline{0}$", respectively. Here, the underlined bits correspond to the bits in the red boxes with solid lines in Fig. 5.3(a). Then the reader identifies tags in the singleton slots and obtains frame parameters in the collision slots as follows.

- *Singleton slot:* If there is no colliding bits in DM, the reader identifies the tag and retrieves its ID as $comm||DM$. For example, in slot S_2 frame F_2 of Fig. 5.3(b), the decoded message $DM =$ "11". Then the reader retrieves the tag ID as "00111010"||"11", i.e., "0011101011".

- *Collision slot:* Denote by C_i, $i \in [1, log_2 M]$, the bit position of the i-th colliding bit in DM. Since the colliding bits in DM have invalid symbols, a bit "1" is used to replace each of them. Therefore, the reader sets $bp_i = l+1+C_i, i \in [1, \log_2 M-1]$ and obtains the new prefix until the $(\log_2 M-1)$-th colliding bit, e.g., $pre =$ "$comm||DM(1 : C_1 - 1)||1|| \ DM(C_1 + 1 : C_2 - 1)||1||...|| \ DM(C_{\log_2 M-1} : C_{\log_2 M} - 1)$". Then it inserts these parameters into Q, i.e., $Q = QueueIn\big(Q, (bp_1, bp_2, ..., bp_{\log_2 M-1}, \ pre)\big)$. For example, in slot S_1 frame F_2 of Fig. 5.3(b), tags T_2 and T_3 separately reply "01" and "10" to the reader simultaneously. Since the received message $DM =$ "??", we have $C_1 = 1$ and $C_2 = 2$. With $comm =$ "00101011", the reader obtains $bp_1 = 8 + 1 + C_1 = 9$ and $pre =$ "00101011"||"1".

Note that in slot S_3 of frame F_3 in Fig. 5.3(b), the common prefix is "101011" and the decoded message is "00?1". Since there is only one colliding bit and the ID of every tag is unique, it is easy to infer that there are two tag responses in the slot. Then, the reader can retrieve the IDs of the two colliding tags, i.e., "1010110001" and "1010110011". This is also known as a two-readable collision slot that exists in most query tree based protocols. The reader repeats the following frames until Q becomes empty.

Algorithms 8 and 9 give the pseudo-code of the proposed MCT protocol at the tag and reader sides, respectively. Note that in Algorithm 9, *Spre* is used to

[3] $de2bi(x, k)$ converts a decimal number x into k bits length binary string; $st_1||st_2$ concatenates strings st_1 and st_2.

Algorithm 8 MCT identification at the tags

After receiving $Query(bp_1, ..., bp_{\log_2 M-1}, pre)$, a tag obtains $l = length(pre)$, $mPre = pre(1 : bp_1 - 1, bp_1 + 1 : bp_2 - 1, bp_2 + 1, ..., l)$, and $tID = ID(1 : bp_1 - 1, bp_1 + 1 : bp_2 - 1, bp_2 + 1, ..., l)$.

The tag compares $mPre$ with tID.

if $tID == mPre$ **then**

  Transit into the transmision state, set S_x, $x = bin2dec(ID(bp_1, bp_2, ..., bp_{\log_2 M-1}, l + 1))$, then reply $ID(l + 2 : end)$ in slot S_x of the current frame.

else Stay in the waiting state.

end if

recover the common prefix of the current slot. For example in frame F_2 in Fig. 5.3(b), pre = "0011101 and $bp = 4$. The common prefixes of slots S_0, S_1, S_2, and S_3 in frame F_2 are "001$\underline{0}$1010$\underline{0}$", "001$\underline{0}$1010$\underline{1}$", "001$\underline{1}$1010$\underline{0}$", and "001$\underline{1}$1010$\underline{1}$", respectively. Moreover, $dec2bin(x,\ k)$ represents the operation of converting the decimal number x to a k-bit binary string; and $st_a||st_b||...||st_c$ stands for the operation of concatenating strings st_a, st_b,..., and st_c.

5.3.3 Implementation and Computational Complexity

In our protocol, the tags operations are quite simple such that the tags only need to execute operations of bit checking, string composing and number-to-string conversion. It is very easy to implement the proposed protocol on existing passive tags. Moreover, our protocol requires the use of Manchester coding in tag response messages. Since the ISO/IEC 14443 standard [89] also specifies Manchester coding to detect the position of colliding bit for type A tags, commercial RFID tags that support this standard can be easily reused to implement the proposed protocol, such as EM4305, TRF7960, TRF7964 and so on.

In the ISO/IEC 14443 standard, the traditional tree search protocol is used to identify unknown tags, in which the reader only needs to obtain the maximum common prefix until the first colliding bit in the aggregated message, and a tag needs to compare its ID with the received query prefix in each slot. By comparison, the proposed MCT protocol requires more operations in each slot,

Algorithm 9 MCT identification at the reader

1: **Initization:** Set $Q = \{(1,\ 2, ...,\ \log_2 M - 1,\ \text{"1"})\}$, $Spre = \epsilon$, and broadcast $Init(M)$;

2: Obtain $(bp_1, ...,\ bp_{\log_2 M-1},\ pre) = QueueOut(Q)$, and broadcast $Query(bp_1, ...,\ bp_{\log_2 M-1},\ pre)$.

3: **for** $x = 0 \to M - 1$ **do**

4: Set $Spre = dec2bin(x,\ \log_2 M)$, $l = length(pre)$, and $comm =$ "$pre(1 : bp_1 - 1)||Spre(1)||...||\ pre(bp_{\log_2 M-1} - 1 : l)||Spre(\log_2 M)$";

5: Receive tag responses in the current slot S_x:

6: **if** No tag response **then** An empty slot is declared;

7: **else** Decode the received message with Manchester decoding as DM.

8: **if** the reader can successfully decode the received message **then**

9: The reader recognizes the tag's ID as "$comm||DM$";

10: **else**

11: Record the bit positions of the first $\log_2 M$ colliding bits as $C_1,\ C_2, ...,\ C_{\log_2 M}$;

12: **if** there is only one colliding bit **then**

13: This is a two-readable collision slot. The reader recognizes the two tags' IDs as "$comm||DM(1 : C_1 - 1)||0||\ DM(C_1 + 1 : end)$" and "$comm||DM(1 : C_1 - 1)||1||\ DM(C_1 + 1 : end)$";

14: **else**

15: This is a collision slot. The reader sets $pre =$ "$comm||DM(1 : C_1 - 1)||1||\ DM(C_1 + 1 : C_2 - 1)||1||...||\ DM(C_{\log_2 M-1} : C_{\log_2 M} - 1)$", $bp_i = l + 1 + C_i$, $i \in [1, log_2 M - 1]$, and $Q = QueueIn\big(Q, (bp_1,\ bp_2, ...,\ bp_{\log_2 M-1},\ pre)\big)$.

16: **end if**

17: **end if**

18: **end if**

19: **end for**

20: **if** Q is not empty **then** Goto 2;

21: **else** Terminate the identification process.

22: **end if**

such as the operations of obtaining $mPre$ and tID and the conversion between a decimal number and a binary string. In spite of this, the overall computational complexity of our proposed protocol is still slight lower than that of the method used in the standard. This is because our protocol takes a fewer number of slots than the traditional method.

Table 5.2: Number of FLOPs needed for identifying n tags

		Reader's operations	Tag's operations
$n = 1000$	Proposed MCT($M = 4$)	2.79×10^5	2.71×10^4
	Proposed MCT($M = 8$)	2.76×10^5	2.29×10^4
	Tree search protocol [89]	2.80×10^5	4.14×10^4
$n = 3000$	proposed MCT($M = 4$)	2.79×10^5	2.71×10^4
	proposed MCT($M = 8$)	2.76×10^5	2.29×10^4
	tree search protocol [89]	2.80×10^5	4.14×10^4
$n = 5000$	proposed MCT($M = 4$)	1.41×10^6	1.63×10^5
	proposed MCT($M = 8$)	1.38×10^6	1.35×10^5
	tree search protocol [89]	1.42×10^6	2.53×10^5

More specifically, Table 5.2 lists the number of floating point operations (FLOPs) at the reader and tag side for identifying 1000 tags, where one FLOP is used for the addition, subtraction or multiplication, while two FLOPs are used for the operation of bit comparison. As is shown in Table 5.2, the computational complexity of the comparative protocols are almost the same. The proposed M-CT ($M = 8$) protocol needs a slight fewer number of FLOPs than the other two protocols, because the reduced slot number reduces the overall computational complexity of our protocol.

5.4 Performance Analysis

As can be observed from the time and energy models in equation (5.1) and (5.2), the time and energy costs of MCT depend highly on the numbers of successful, collision and empty slots. In this section, we will analytically derive the number of slots required in the proposed MCT protocol.

As it is well-known, in tree-based protocols larger values of branch numbers result in more empty and less collision slots [90]. Similarly, in MCT, with the increase of M, the number of empty slots increases and the number of collision slots decreases, which further reduces the time and energy costs. However, with a large M, the operation of converting multiple bits into slot information leads to an increased computational complexity at the tags. To balance between efficiency and complexity, $M = 4$ is a good trade-off. As such, the analytical results of the proposed MCT protocol are given with $M = 4$. Nevertheless, the results can be readily generalized to other values of M.

In MCT, the tags in each collision slot are divided into four subsets according to the first two colliding bits in the aggregated message. Denote the four subsets by $\mathbb{A}_j,\ j \in [0, 3]$. The event that i out of n tags are split into the same subset $\mathcal{A}_j$ follows a binomial distribution of $P(\mathcal{A}_j = i) = B(i, n, 1/4)$, where $i \in [0,\ n],\ j \in [0, 3]$, and $\mathcal{A}_j$ is the number of tags in set $\mathbb{A}_j$. Using Manchester code and the designed prefix match process in MCT, the colliding tags are grouped into one of the following two scenarios:

- Both $\mathbb{A}_0$ and $\mathbb{A}_3$ have at least one tag; and
- Both $\mathbb{A}_1$ and $\mathbb{A}_2$ have at least one tag.

This is because the splitting of the colliding tags is based on the first two colliding bits of their IDs. The tags with the values of their first two colliding bits being "00", "01", "10", or "11" will be split into subsets $\mathbb{A}_0,\ \mathbb{A}_1,\ \mathbb{A}_2$, or $\mathbb{A}_3$, respectively. In each position of the colliding bit, there should be at least one "0" and one "1" from two different tag responses. Therefore, the grouped subsets should have at least one tag in $\mathbb{A}_0$ and $\mathbb{A}_3$ (or $\mathbb{A}_1$ and $\mathbb{A}_2$), respectively. Let $\mathfrak{B}_1 = \big(\mathcal{A}_0 \geq 1\big) \cap \big(\mathcal{A}_3 \geq 1\big)$, and $\mathfrak{B}_2 = \big(\mathcal{A}_1 \geq 1\big) \cap \big(\mathcal{A}_2 \geq 1\big)$. The probability that i tags are grouped into $\mathbb{A}_j,\ j \in [0,\ 3]$ is $P\big(\mathcal{A}_j = i \big| \mathfrak{B}_1 \cup \mathfrak{B}_2\big)$.

Theorem 3 *The total number of slots needed to identify n tags is*

$$\mathcal{S}(n) = 1 + \sum_{j=0}^{3}\sum_{i=0}^{n-1} P\big(\mathcal{A}_j = i \quad \big|\mathfrak{B}_1 \cup \mathfrak{B}_2\big)\mathcal{S}(i), \tag{5.3}$$

where $\mathcal{S}(0) = \mathcal{S}(1) = 1$.

Proof 3 *At the beginning of MCT, the reader starts an initiation slot to check whether there are tags within its reading range. In the event of no tag replies, the reader will terminate the identification process. If there is one tag reply, the reader can identify this tag and then terminate the identification process. Therefore, when $i = 0$ or 1, we have $\mathcal{S}(0) = \mathcal{S}(1) = 1$. If there are more than one tag reply, a collision slot is detected. Then the reader recursively split the colliding tags into four subsets until there is at most one tag in each subset as discussed above. The processes for all the four subsets are almost identical. With recursive iteration, Theorem 3 is proved.*

Theorem 4 *When $n = 0$ or 1, the number of collision slots $C_c(0) = C_c(1) = 0$. When $n > 2$, the number of collision slots needed to identify n tags is*

$$C_c(n) = 1 + \sum_{j=0}^{3}\sum_{i=2}^{n-1} P\big(\mathcal{A}_j = i \quad \big|\mathfrak{B}_1 \cup \mathfrak{B}_2\big)C_c(i). \tag{5.4}$$

Proof 4 *Firstly, when $n = 0$ or 1, there is a successful or empty slot. Since there are no more than two tag responses in such a slot, the tag will not be divided in the following identification process. Therefore, we have $C_c(0) = C_c(1) = 0$. In a collision slot with $n = 2$, the two tags will be deterministically identified in the next four slots according to the positions of the first two colliding bits. Thus, we have $C_c(2) = 1$. When $n > 2$, the reader recursively divides the colliding tags into small groups in the same manner. Similar to Theorem 3, Theorem 4 is proved with the recursive method.*

To obtain $\mathcal{S}(n)$ and $C_c(n)$, we need the knowledge of $P\big(\mathcal{A}_j = i\big|\mathfrak{B}_1 \cup \mathfrak{B}_2\big)$. It follows from probability theory that

$$\begin{aligned} P & \quad \big(\mathcal{A}_j = i\big|\mathfrak{B}_1 \cup \mathfrak{B}_2\big) \\ = & \quad \frac{P\big(\mathfrak{B}_1 \cup \mathfrak{B}_2\big|\mathcal{A}_j = i\big)P\big(\mathcal{A}_j = i\big)}{P\big(\mathfrak{B}_1 \cup \mathfrak{B}_2\big)}. \end{aligned} \tag{5.5}$$

Lemma 5 *The probability that at least one tag is split into $\mathcal{A}_0$ and $\mathcal{A}_3$ (or $\mathcal{A}_1$ and $\mathcal{A}_2$) is*

$$P(\mathfrak{B}_1 \cup \mathfrak{B}_2) = 1 - \left(\frac{1}{2}\right)^{n-2} + \left(\frac{1}{4}\right)^{n-1}. \tag{5.6}$$

Proof 5 *It is well known, $P(A \cup B) = P(A) + P(B) - P(A \cap B)$. We have*

$$\begin{aligned} & P(\mathfrak{B}_1 \cup \mathfrak{B}_2) \\ = \; & P\big((\mathcal{A}_0 \geq 1) \cap (\mathcal{A}_3 \geq 1)\big) + P\big((\mathcal{A}_1 \geq 1) \cap (\mathcal{A}_2 \geq 1)\big) \\ & -P\big((\mathcal{A}_0 \geq 1) \cap (\mathcal{A}_3 \geq 1) \cap (\mathcal{A}_1 \geq 1) \cap (\mathcal{A}_2 \geq 1)\big). \end{aligned} \tag{5.7}$$

Then we calculate the probability of each term in (5.7)

$$\begin{aligned} P\big((& \mathcal{A}_0 \geq 1) \cap (\mathcal{A}_3 \geq 1)\big) \\ = \; & 1 - P\big((\mathcal{A}_0 = 0) \cup (\mathcal{A}_3 = 0)\big) \\ = \; & 1 - P\big(\mathcal{A}_0 = 0\big) - P\big(\mathcal{A}_3 = 0\big) + P\big((\mathcal{A}_0 = 0) \cap (\mathcal{A}_3 = 0)\big) \\ = \; & 1 - 2\left(\frac{3}{4}\right)^n + \left(\frac{1}{2}\right)^n. \end{aligned} \tag{5.8}$$

Similarly, we have

$$P\big((\mathcal{A}_1 \geq 1) \cap (\mathcal{A}_2 \geq 1)\big) = 1 - 2\left(\frac{3}{4}\right)^n + \left(\frac{1}{2}\right)^n. \tag{5.9}$$

$$\begin{aligned} P\big((\mathcal{A}_0 \; & \geq 1) \cap (\mathcal{A}_3 \geq 1) \cap (\mathcal{A}_1 \geq 1) \cap (\mathcal{A}_2 \geq 1)\big) \\ = \; & 1 - 4\big(\frac{3}{4}\big)^n - 6\big(\frac{1}{2}\big)^n + 4\big(\frac{1}{4}\big)^n. \end{aligned} \tag{5.10}$$

Plugging (5.8), (5.9), *and* (5.10) *into* (5.7), **Lemma 1** *is proved.*

Lemma 6 *The conditional probability*

$$\begin{aligned} P\big(\mathfrak{B}_1 \cup \; & \mathfrak{B}_2 \big| \mathcal{A}_j = i\big) \\ = \; & \begin{cases} 1 - 2\big(\frac{2}{3}\big)^n - \big(\frac{1}{3}\big)^n, & i = 0 \\ \\ 1 - 2\big(\frac{1}{3}\big)^{n-i}, & 1 \leq i \leq n-1 \end{cases}, \end{aligned} \tag{5.11}$$

where $j \in [0, \; 3]$.

Proof 6 *Firstly, we consider the situation in $\mathcal{A}_0$.*

When $i = 0$, its obvious that $P(\mathfrak{B}_1|\mathcal{A}_0 = 0) = P((\mathcal{A}_0 \geq 1) \cap (\mathcal{A}_3 \geq 1)|\mathcal{A}_0 = 0) = 0$. It follows

$$
\begin{aligned}
& P\big(\mathfrak{B}_1 \cup \ \mathfrak{B}_2 \big| \mathcal{A}_0 = 0\big) \\
= \quad & P\big((\mathcal{A}_1 \geq 1) \cap (\mathcal{A}_2 \geq 1)|\mathcal{A}_0 = 0)\big) \\
= \quad & 1 - P\big((\mathcal{A}_1 = 0) \cup (\mathcal{A}_2 = 0)|\mathcal{A}_0 = 0)\big) \\
= \quad & 1 - \frac{P\big(((\mathcal{A}_1 = 0) \cup (\mathcal{A}_2 = 0)) \cap (\mathcal{A}_0 = 0)\big)}{P(\mathcal{A}_0 = 0)} \\
= \quad & 1 - \frac{2\left(\frac{1}{2}\right)^n - \left(\frac{1}{4}\right)^n}{\left(\frac{3}{4}\right)^n} \\
= \quad & 1 - 2\left(\frac{2}{3}\right)^n + \left(\frac{1}{3}\right)^n. \qquad (5.12)
\end{aligned}
$$

When $1 \leq i \leq n-1$, we arrive at

$$
\begin{aligned}
& P\big(\mathfrak{B}_1 \cup \mathfrak{B}_2 \big| \mathcal{A}_0 = i\big) \\
= \ & P\big((\mathcal{A}_3 \geq 1) \cup ((\mathcal{A}_1 \geq 1) \cap (\mathcal{A}_2 \geq 1))|\mathcal{A}_0 = i)\big) \\
= \ & 1 - P\big((\mathcal{A}_3 = 0) \cap ((\mathcal{A}_1 = 0) \cup (\mathcal{A}_2 = 0))|\mathcal{A}_0 = i)\big) \\
= \ & 1 - P\big((\mathcal{A}_1 = 0) \cap (\mathcal{A}_3 = 0)|\mathcal{A}_0 = i)\big) \\
& -P\big((\mathcal{A}_1 = 0) \cap (\mathcal{A}_3 = 0)|\mathcal{A}_0 = i)\big) \\
& +P\big((\mathcal{A}_1 = 0) \cap (\mathcal{A}_3 = 0) \cap (\mathcal{A}_1 = 0) \cap (\mathcal{A}_3 = 0)|\mathcal{A}_0 = i)\big) \\
= \ & 1 - 2\left(\frac{1}{3}\right)^{n-i}. \qquad (5.13)
\end{aligned}
$$

According to the splitting condition, it can be said that the situation in each subset is nearly the same. Therefore, **Lemma 2** *is proved.*

Plugging (5.6) and (5.11) into (5.5), the probability that i tags are split into subset $\mathcal{A}_j,\ j \in [0,\ 3]$, for implementing the proposed MCT protocol is

$$
\begin{aligned}
& P\big(\mathcal{A}_j = i \quad \big| \mathfrak{B}_1 \cup \mathfrak{B}_2\big) \\
= \ & \begin{cases}
\dfrac{\left(1-2\left(\frac{2}{3}\right)^n - \left(\frac{1}{3}\right)^n\right)\left(\frac{3}{4}\right)^n}{1-4\left(\frac{1}{2}\right)^n + 4\left(\frac{1}{4}\right)}, & i = 0; \\[2ex]
\dfrac{\left(1-2\left(\frac{1}{3}\right)^{n-i}\right)\binom{n}{i}\left(\frac{1}{4}\right)^i\left(\frac{3}{4}\right)^{n-i}}{1-4\left(\frac{1}{2}\right)^n + 4\left(\frac{1}{4}\right)}, & 1 \leq i \leq n-1.
\end{cases} \qquad (5.14)
\end{aligned}
$$

From (5.3), (5.4), and (5.14), the numbers of the total and collision slots, i.e., $\mathcal{S}(n)$ and $\mathcal{C}(n)$, are obtained. This analytical method is readily generalized to obtain the number of slots for $M = 8,\ 16,\ \ldots$, etc.. Finally, Fig. 5.4 plots both the numerical and simulation results of the average number of slots $\mathcal{S}(n)/n$, and the average number of collision slots $C_c(n)/n$ for one tag identification with n ranging from 0 to 1000.

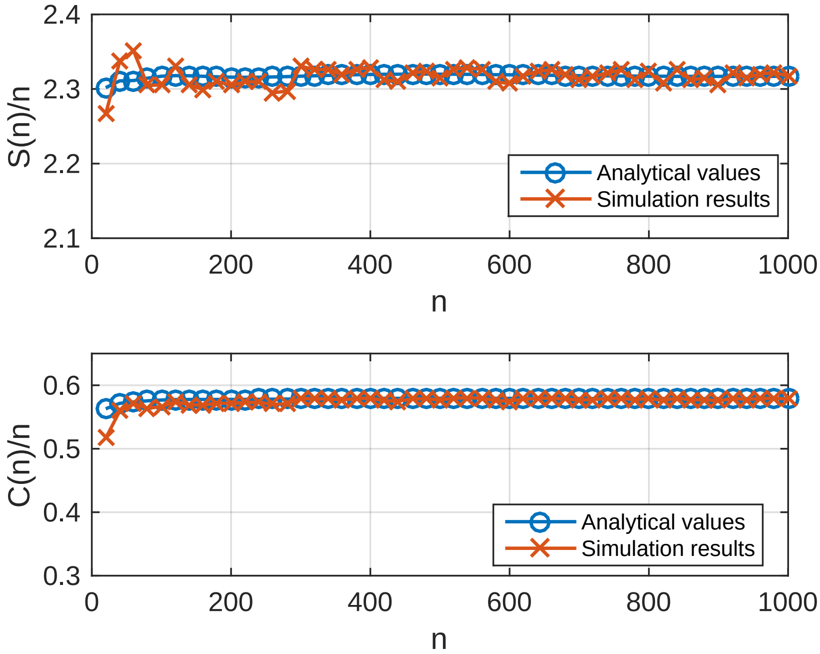


Figure 5.4: Analytical and simulation results of the slot number: (a) Average number of total slots for one tag identification $\mathcal{S}(n)/n$; (b) Average number of collision slots for one tag identification $C_c(n)/n$.

As can be observed from Fig. 5.4, the simulation results match well with their numerical counterparts, and we have $\mathcal{S}(n) \approx 2.31n$, $C_c(n) \approx 0.58n$. Since the number of successful slots $C_s(n) = n$ and $\mathcal{S}(n) = C_s(n) + C_s(n) + C_e(n)$, we have $C_e(n) \approx 0.73n$. Substituting the values of $C_s(n)$, $C_c(n)$ and $C_e(n)$ into (5.1) and (5.2), the average time and energy costs for one tag identification are obtained as follows

$$\begin{aligned} T(n)/n \approx\ & 0.58\bar{t}_Q + 1.73t_R + 1.58\bar{t}_T \\ & +2.31t_1 + 1.58t_2 + 0.73t_3, \end{aligned} \tag{5.15}$$

$$E(n)/n \approx P_{tx}\Big[\ 0.58\bar{t}_Q + 1.73t_R + 1.58\bar{t}_T + 2.31t_1 + 1.58t_2 + 0.73t_3\Big] + 1.58\bar{t}_T P_{rx}, \quad (5.16)$$

where t_1, t_2, t_3 and t_R depend on the system settings. $\bar{t}_Q$ and $\bar{t}_T$ are the average times for transmitting the reader commands and tag responses, respectively.

5.5 Simulation Results

In this section, we evaluate the performance of the proposed MCT protocol and compare it with some relevant benchmark protocols.

5.5.1 System Configurations

In the simulation, we consider a typical passive RFID system, which consists of a single reader and a large number of passive tags. The ID of each tag is unique and of 128 bits length. The tag IDs are uniformly distributed. The tag set are supposed to be unchanged during the identification process, and the reader has no knowledge on the number of tags and their IDs prior to the identification process. Since the practical environments have almost the same impact on the comparative protocols, the communication channels between the reader and tags are assumed to be ideal. All the tags' response signals are assumed to be correctly detected, and there is no capture effect as in the literature [25, 27, 28, 54, 74]. The effect of the detection errors and capture effect on the performance of our protocol will be discussed in Section 5.6.

The link timing between the reader and tags and the transmission messages follow the structure illustrated in Fig. 5.2. The time duration of the parameters in the link timing and the reader's transmission and receiving powers are given in Table 5.3, where L_{cmd} is the overhead length of the $Query()$ command, including the header, command, address, mask, BP, and CRC16 as shown in Fig. 5.2, K is the length of the tag ID and Dr indicates the data rate. In DPPS, CwT, and SPR, $L_{\text{cmd}} = 53$. In MCT, except for the common parameters, an extra BP parameter which indicates the position of the first colliding bit is needed in the $Query()$ command. Thus, $L_{\text{cmd}} = 61$ in MCT. Moreover, V_1 is the number of message bits in a reader request command, while V_2 is the number of message bits in a tag response.

Table 5.3: Parameters in the simulation

Parameter	Value	Parameter	Value
K	128 bits	Dr	160 kbps
$t_1,\ t_2$	25 us	t_3	12.5 us
t_Q	$(L_{\text{cmd}} + V_1)/Dr$	t_T	V_2/Dr
P_{tx}	825 mw	P_{rx}	125 mw

The performance of MCT is evaluated in four aspects, e.g., the numbers of message bits transmitted by the reader and tags, identification time, and energy cost. The simulation results of the algorithms MCT ($M = 4$ and 8) are given. The simulation results of some benchmark protocols, such as the collision tree (CT) [28], dual prefix probe scheme (DPPS) [27], collision window tree (CwT) [74], and parallel splitting with retrieve (PSR) [55] protocols, are given. They are the most relevant and state-of-the-art protocols in the literature. Among them, CT, PSR and DPPS are some of the best time efficient ones. CwT incurs the least energy consumption while maintaining a low identification delay. The comparison results are given in Figs. 5.5, 5.6, 5.7 and 5.8.

Moreover, to investigate the impact of the tag ID distributions, the time and energy costs of the proposed MCT ($M = 4$) protocol under the following five scenarios are evaluated, and the simulation results are given in Figs. 5.9 and 5.10.

- **Scenario S0:** the tag IDs are uniformly distributed;
- **Scenario S1:** the tag IDs are distributed continuously, and the variable bits are in the front part of the tag ID;
- **Scenario S2:** the tag IDs are selected from three subsets, and the variable bits of these subsets are separately in the front, middle and end parts of the tag ID. The IDs in each subset are distributed continuously;

- **Scenario S3:** the tag IDs are distributed normally, and the expectation is 2^{30}, the variance is 2^{10}; and

- **Scenario S4:** the tag IDs are normally distributed in three subsets, i.e., $N(2^{30}, 2^{10})$, $N(2^{70}, 2^{10})$ and $N(2^{100}, 2^{10})$.

5.5.2 Number of Transmitted Message Bits

Firstly, the average number of message bits transmitted by the reader is plotted in Fig. 5.5. As can be observed from the figure, the reader transmits the least number of message bits in the proposed MCT protocol than its comparative counterparts. The main reasons are in two aspects. On the one hand, by constructing a new M-ary collision tree structure, it takes MCT fewer number of slots to identify tags, which requires less number of message bits transmitted by the tags. On the other hand, instead of transmitting long *Query* command in every slot in the comparative protocols, our protocol only transmits the *Query* command at the beginning of each frame. In other slots, a 4 bits length *QueryRep* command is transmitted. Therefore, our protocol transmits much smaller number of message bits at the reader side.

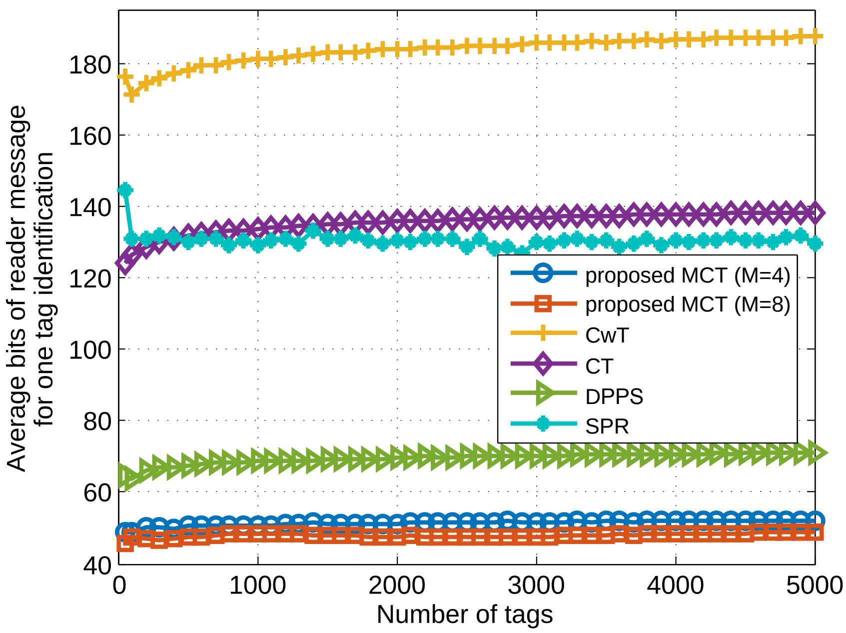


Figure 5.5: Average number of message bits transmitted by the reader for one tag identification.

Fig. 5.5 also demonstrates that CwT transmits the most number of message bits at the reader. This is mainly because that CwT needs a greater number of slots, including collision and go-on slots, than the other protocols. Since the reader needs to transmit a long *Query* message in each slot, it needs to transmit more message bits when using CwT than other protocols. With fewer numbers of slots, CT, SPR and DPPS transmit fewer numbers of message bits at the reader side than CwT. However, they still transmit more message bits than MCT. Overall, the proposed MCT algorithms ($M = 4$ and 8) are able to transmit at least 26.57% and 30.86% less numbers of message bits at the reader side than the other comparative protocols, respectively.

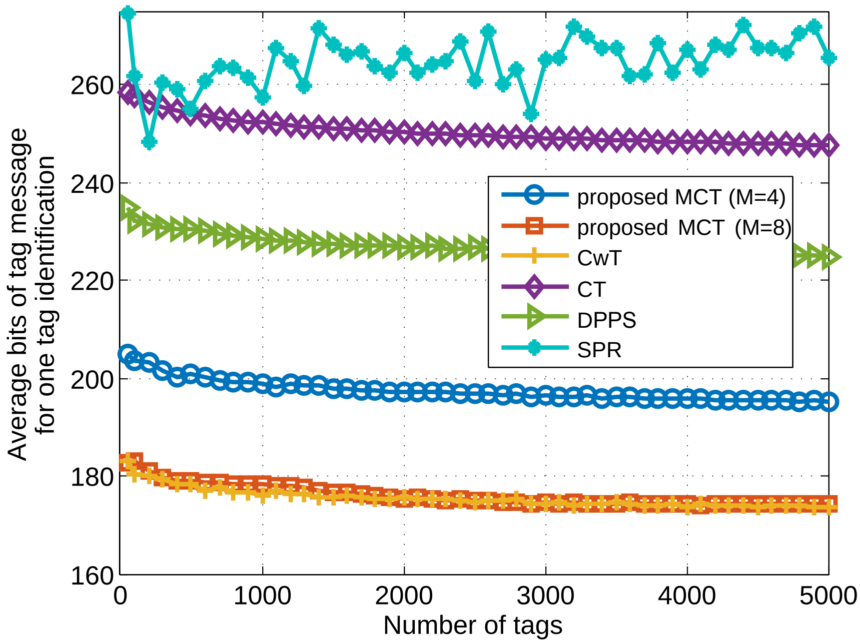


Figure 5.6: Average number of message bits transmitted by the tags for one tag identification.

Secondly, Fig. 5.6 depicts the average number of message bits transmitted in tag responses. As can be observed, the proposed MCT($M = 4$) and MCT ($M = 8$) transmit fewer number of message bits at the tag side than SPR, CT, and DPPS. This is because the use of the first $\log_2 M$ colliding bits in MCT effectively reduces the the number of collision slots, resulting in fewer number of message bits transmitted by the tags. So compared with SPR, CT, and DPPS, MCT ($M = 8$) transmits around 34.22%, 29.53%, 22.32% and MCT($M = 4$)

transmits around 23.37%, 20% and 11.11% less numbers of message bits in tag responses, respectively. Using the heuristic window structure, CwT transmits almost the same message bits with MCT($M = 8$). However, it needs to transmit much more message bits at the reader side as shown in Fig. 5.5. Therefore, the total number of message bits transmitted in CwT is larger than that in MCT ($M = 8$).

5.5.3 Time and Energy Costs

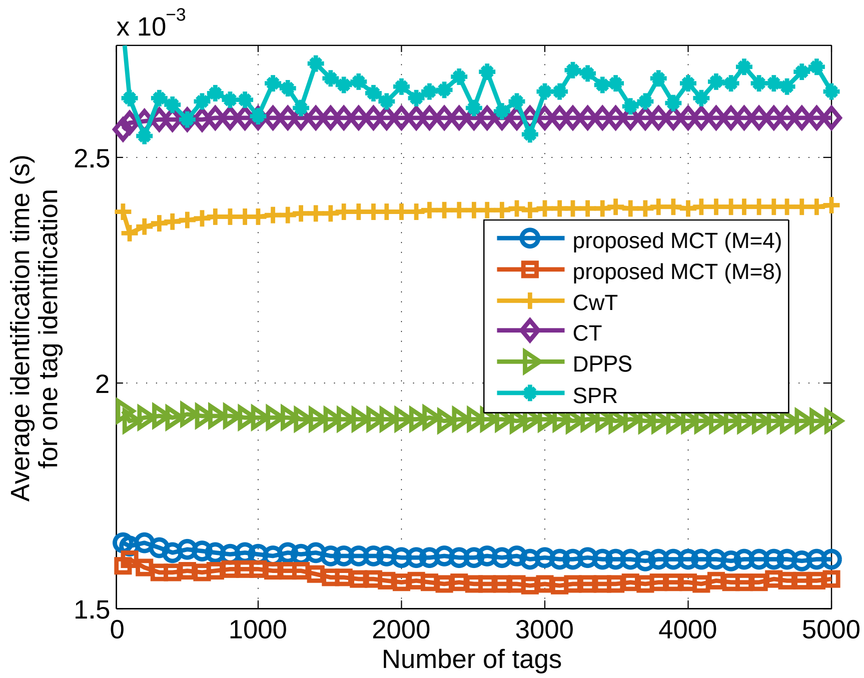


Figure 5.7: Average time in second for one tag identification.

Thirdly, Figs. 5.7 and 5.8 present the overall comparison results in terms of the average time and energy costs for one tag identification, respectively. As is observed, the proposed MCT protocol takes the least average time and energy cost to identify one tag. This is because MCT generates fewer number of slots and transmits fewer number of message bits at both the reader and tag sides. By comparison, due to the existence of many collision slots, CT and SPR take the most time to identify tags as opposed to all the other comparative protocols. Compared with the best protocol DPPS, the proposed MCT algorithms ($M = 4$ and 8) take 16.12% and 18.20% less average time for identifying one tag, respectively. Similarly, the simulation results in Fig. 5.8 show that the proposed MCT algorithms ($M = 4$ and 8) consume at least 15.73% and 20.22% less energy than the other benchmark protocols, respectively.

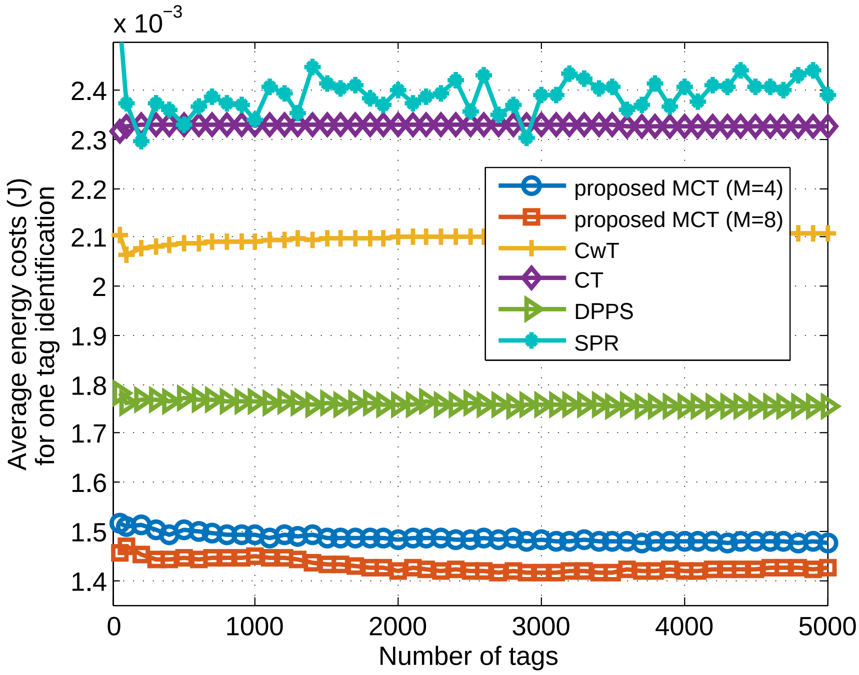


Figure 5.8: Average energy cost in joule for one tag identification.

As shown in Figs. 5.7 and 5.8, the time and energy costs of MCT ($M = 8$) is slightly smaller than those of MCT ($M = 4$). The main reason is that the number of idle slots increases drastically with the increase of M. Although the number of transmitted message bits in tag responses of MCT ($M = 8$) is about 10% less than that of MCT ($M = 4$) as shown in Fig. 5.6, the increased idle slots in MCT ($M = 8$) lead to increased time and energy costs. Therefore, the time and energy costs of the proposed MCT algorithms ($M = 4$ and 8) are very close. Since MCT ($M = 8$) requires more operations at the tags, MCT ($M = 4$) is more suitable for practical applications.

5.5.4 Impact of Tag ID Distributions

Finally, the performances of MCT($M = 4$) with various tag ID distributions are given in Figs. 5.9 and 5.10. As is shown, it takes the reader a shorter time and a smaller energy cost in the event of non-uniformly distributed tag IDs (i.e., S1, S2, S3, and S4) than the uniformly distributed tag ID scenario S0. The main reason is that in the non-uniformly distributed tag ID scenarios the variable bits in the tag IDs are more closely located. Thus the colliding tags can be more effectively occupied into smaller groups. Therefore, our protocol show better performance when the tag IDs are non-uniformly distributed.

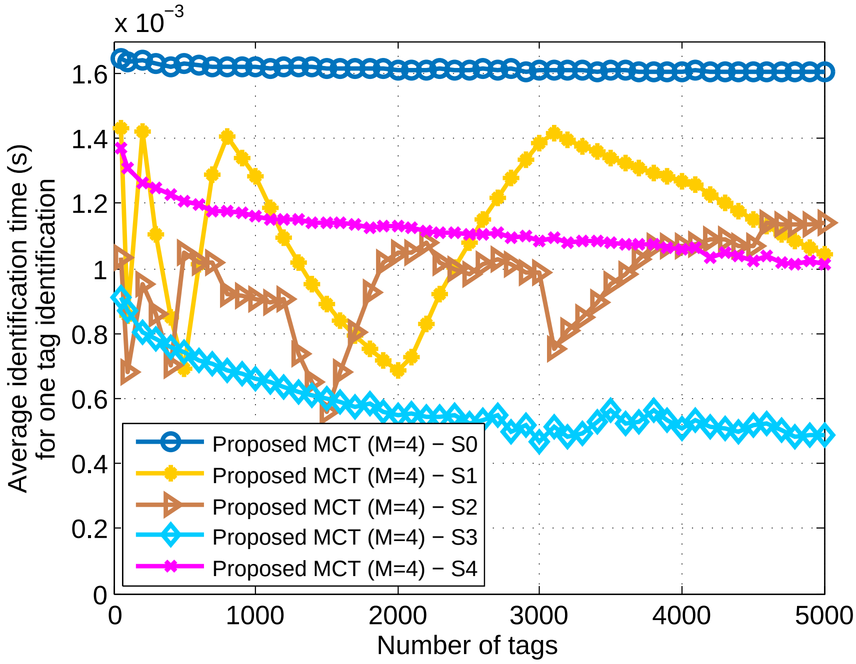


Figure 5.9: Average identification time for one tag identification under different tag ID distributions.

Moreover, when the tag IDs are continuously distributed, the time and energy costs fluctuate with the increase of the number of tags. This is mainly because the reader identifies the tags with the M-ary tree structure. If the tag IDs are continuously distributed, the constructed tree structure remains the same when the number of tags changes in a fixed set, e.g., $1000 \leq n \leq 3000$ in scenario $S1$. When the number of tags increases in each set, more tags occupy the slots of the constructed tree, resulting in an increased number of collision slots and a decreased number of empty slots. Thus, the identification time and energy costs increase first and then decrease during each tag set.

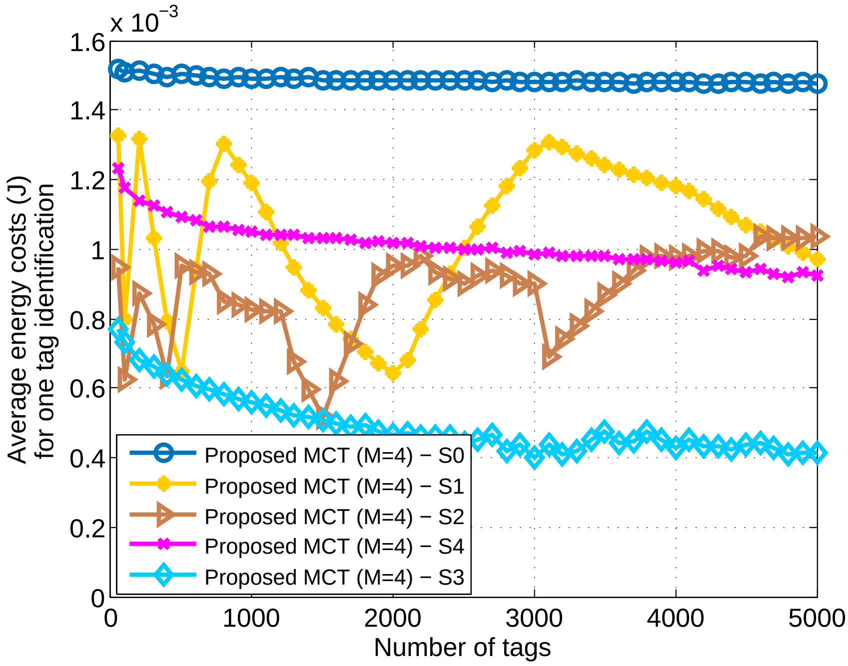


Figure 5.10: Average energy cost for one tag identification under different tag ID distributions.

5.6 Discussions

In this part, we discuss the impact of the practical environment on the proposed MCT protocol. In a practical environment, two main factors affect the performance of our protocol, i.e., detection errors and capture effect. Firstly, in a passive RFID system, tags' backscattering signals may not be detected by the reader, resulting in detection errors. Detection errors may turn an original singleton slot into an empty slot, and it may also turn an original collision slot into a singleton or empty slot. Secondly, since the distances between the tags and the reader are different, the strengths of the backscattered signals are different. When multiple tags reply to the reader simultaneously, it is possible that one tag's signal is much stronger than all the other signals, leading to the so-called capture effect. If the capture effect occurs, the original collision slot will be turned into a singleton slot, which accelerates the identification process.

More specifically, Figs. 5.11 give the slot transformations caused by detection errors and the capture effect. As can be observed, the detection errors increase the number of empty and collision slots. However, the capture effect accelerates the identification process by transferring some collision slots into singleton slots.

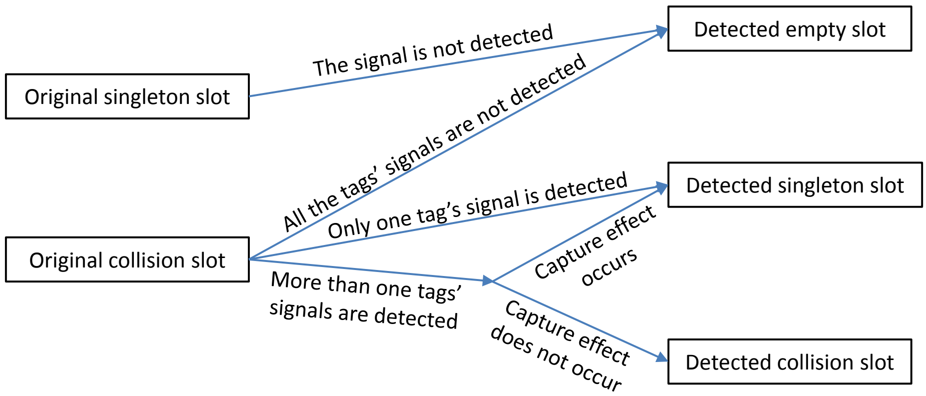


Figure 5.11: Slot transformation caused by detecting errors and capture effect.

Here, we do not consider the effect of transmission bit errors. Because when the distance between the reader and tags is smaller than 10m in indoor environment, the SNR of the passive RFID systems is greater than 15dB, and the bit error probability is smaller than 10^{-6} [85]. The transmission bit error is so small that it has almost no effect on the identification process.

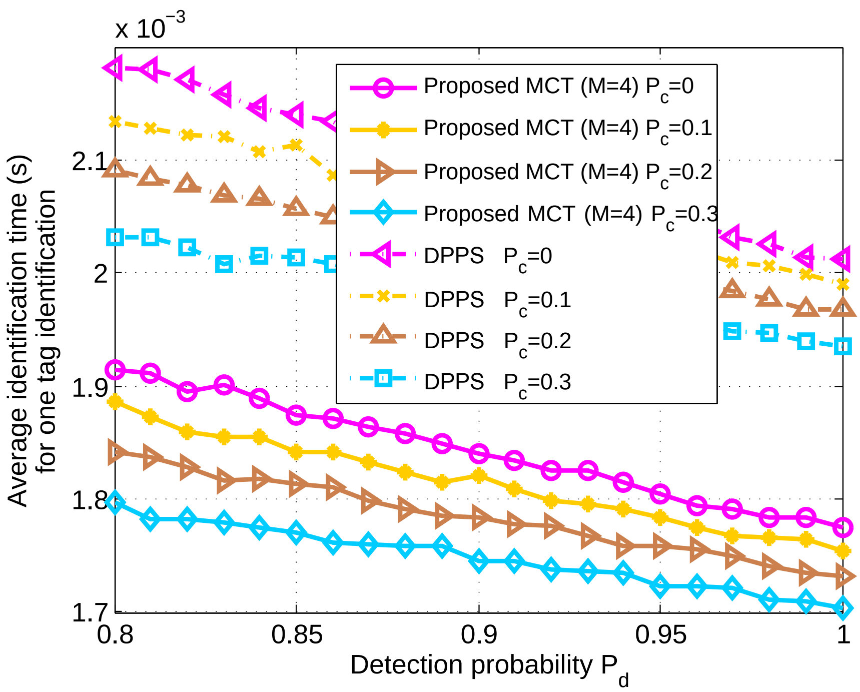


Figure 5.12: Average identification time for one tag identification when P_d and P_c vary.

Denote by P_d and P_c the probability that a tag's signal can be successfully detected and the probability that the capture effect occurs, respectively. Figs. 5.12 and 5.13 gives the time and energy performance with $P_d \in [0.8, 1]$ and

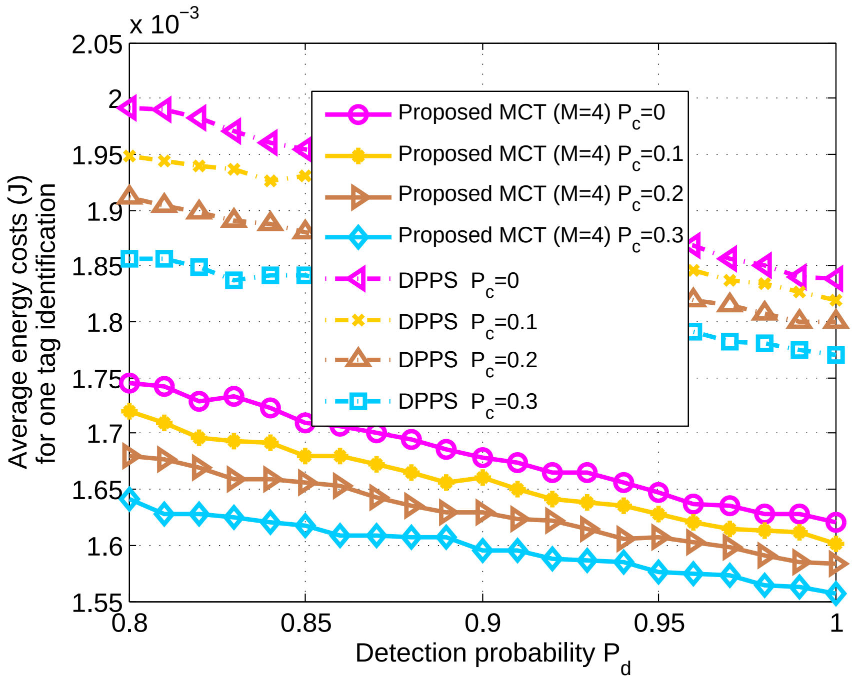


Figure 5.13: Average energy cost for one tag identification when P_d and P_c vary.

$P_c \in \{0, 0.1, 0.2, 0.3\}$. Note that when the detection error or capture effect occurs, the tags whose signals are not detected by the reader become hidden tags. Since the hidden tags are not recognized in the current reading cycle, the reader needs to implement multiple reading cycles of the identification process and frequently change the reader position to identify all the tags. As is demonstrated, both the time and energy costs decrease with the increases of P_d and P_c. For comparative purposes, the simulation results of the most related DPPS protocol under the same situations are also given. Similarly, the detection and capture probabilities have a similar impact on the performance of DPPS. The performance of our protocol always outperforms its DPPS counterpart.

5.7 Summary

In this paper, a new MCT protocol was proposed for time- and energy-efficient tag identification in large-scale passive RFID systems. Both theoretical analysis and simulations were conducted to demonstrate the high efficiency of our protocol. Finally, the performance of our protocol under the effects of various tag ID distributions and imperfect channel conditions (e.g., detection errors and capture effect) was also discussed and evaluated. However, there are still some issues to be addressed in future work. For example, in privacy concerned applications where tag IDs are secured, a randomly generated pseudo-ID may be used for the identification process.

Chapter 6

Conclusions and Future Work

This chapter concludes the work in the thesis and suggests some possible directions for future research.

6.1 Conclusions

This thesis focuses on improving upon the tag identification process in large-scale RFID systems. In consideration of various application environments, three efficient protocols are proposed to reduce the time and energy costs of the tag identification process.

Firstly, in Chapter 3 a time-efficient pair-wise collision-resolving protocol was proposed to effectively monitor known tags and to timely identify missing tags. Both theoretic analysis and simulation results were presented to prove the high time performance and low computational complexity of the proposed protocol. By comparing with the most relevant recent studies in the literature, the proposed protocol was proved to be more suited for large-scale RFID tag monitoring systems.

Secondly, in Chapter 4 a novel bit-detecting tag identification protocol was proposed for large-scale mobile RFID systems with moving tags. The time performances of our protocol in both low-mobility and high-mobility systems were analysed and compared with the best results reported in the literature. The effect of detection errors in practical environments was also analyzed. Simulation results were presented to demonstrate that the proposed protocol performs better than previous works.

Thirdly, in Chapter 5 a time- and energy-aware collision tree protocol was proposed for efficient tag identification in large-scale passive RFID systems with a portable reader. The theoretical time and energy performances of the proposed protocol were analysed. Simulation results were also compared to demonstrate the high efficiency of our protocol.

To summarize, through investigating the tag identification problems in three different types of RFID systems and proposing efficient tag identification protocols for each system, this thesis effectively solved the tag identification problems in large-scale RFID systems, and the proposed protocols greatly improve the performances of the tag identification process of these systems. The proposed protocols are of great significance in practical applications.

6.2 Future Work

This discussion concludes with the following recommendations for future work that are natural extensions of the research problems tackled in this thesis:

- With the widespread use of RFID technologies, many daily things can be tracked, monitored and connected. Information security and data privacy protection are two challenging issues since a lot of personal and private information can be collected automatically [91]. In such applications, tag IDs should be securely protected. Thus, a randomly generated pseudo-ID may be used for the identification process. Since most protocols rely on the uniqueness of tag IDs, a randomly generated ID will affect the identification process, which needs to be investigated in the future work.

- Manchester coding has the advantage of detecting the bit positions of colliding bits. To accelerate the identification process, most tag identification protocols are cross-layer designs which jointly consider media access control (MAC) and Manchester coding in the physical layer. Although existing standards and most commercial tags support Manchester coding, some tags may not. Therefore, some universal alternative methods should be considered in the future.

- With a limited tuning capability, tags in passive RFID systems are susceptible to interference from other RF signals. In practical applications, many factors influence the reliability of tag responses, e.g., the impedance mismatch between tag antenna and chip, multi-path fading, communication blind spots, and interference. [84]. The imperfections caused by these factors may greatly affect the identification process. In order to investigate the effect of such factors, experiments with real RFID devices can better bridge the gap for a wide deployment. Therefore, we aim to build an experimental platform to facilitate our future research.

In general, although RFID systems have been widely investigated in recent years, there are still some open problems relating to practical applications remaining to be solved, especially for some new application areas, e.g., motion tracking, orientation sensing, and RFID sensor systems. Therefore, we aim to tackle the aforementioned challenging issues to further advance our research.

References

[1] K. Finkenzeller, *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. Jhon Willey & Sons, 2010.

[2] M. Bolic, D. Somplot-Ryl, and I. Stojmenovic, *RFID systems: research trends and challenges*. John Willey & Sons, 2010.

[3] L. Zhu and T.-S. P. Yum, “A critical survey and analysis of rfid anti-collision mechanisms,” *IEEE Communications Magazine*, vol. 49, no. 5, pp. 214–221, May 2011.

[4] 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, Jul. 2006.

[5] D. K. Klair, K.-W. Chin, and R. Raad, “A survey and tutorial of rfid anti-collision protocols,” *IEEE Communications Surveys and Tutorials*, vol. 12, no. 3, pp. 400–421, Jul. 2010.

[6] C. C. Tan, B. Sheng, and Q. Li, “How to monitor for missing rfid tags,” in *Proc. IEEE ICDCS’08*, Beijing, China, Jun. 2008, pp. 295–302.

[7] W. Luo, S. Chen, T. Li, and S. Chen, “Efficient missing tag detection in rfid systems,” in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 356–360.

[8] C. C. Tan, B. Sheng, and Q. Li, “Efficient techniques for monitoring missing rfid tags,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, pp. 1882–1889, Jun. 2010.

[9] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast identification of the missing tags in a large rfid system," in *Proc. 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Salt Lake City, Utah, Jun. 2011, pp. 278–286.

[10] W. Zhu, J. Cao, H. C. B. Chan, X. Liu, and V. Raychoudhury, "Mobile rfid with a high identification rate," *IEEE Transactions on Computers*, vol. 63, no. 7, pp. 1778–1792, Jul. 2014.

[11] F. Schoute, "Dynamic fame length aloha," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 565–568, 1983.

[12] V. Namboodiri and L. Gao, "Energy-aware tag anticollision protocols for rfid systems," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 44–59, Jan. 2010.

[13] P. Popovski, K. Fyhn, R. M. Jacobsen, and T. Larsen, "Robust statistical methods for detection of missing rfid tags," *IEEE Wireless Communications*, vol. 18, no. 4, pp. 74–80, Aug. 2011.

[14] M. Shahzad and A. X. Liu, "Expecting the unexpected: fast and reliable detection of missing rfid tags in the wild," in *Proc. IEEE INFOCOM*, Hong Kong, China, Apr. 2015, pp. 1939–1947.

[15] Y. Zheng and M. Li, "P-mti: physical-layer missing tag identification via compressive sensing," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1356–1366, Aug. 2015.

[16] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large rfid system," in *Proc. ACM MobiHoc*, Chicago, lllinois, Sep. 2010, pp. 1–10.

[17] X. Liu, K. Li, G. Min, Y. Shen, A. Liu, and W. Qu, "A multiple hashing approach to complete identification of missing rfid tags," *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 1046–1057, Mar. 2014.

[18] T. Li, S. Chen, and Y. Ling, "Efficient protocols for identifying the missing tags in a large rfid system," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1974–1987, Dec. 2013.

[19] X. Liu, K. Li, G. Min, Y. Shen, A. Liu, and W. Qu, "Completely pinpointing the missing rfid tags in a time-efficient way," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 87–96, Jan. 2015.

[20] C. Shao, T. Kim, J. Yu, J. Choi, and W. Lee, "Protar: probabilistic tag retardation for missing tag identification in large-scale rfid system," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 513–522, Apr. 2015.

[21] W. Luo, S. Chen, Y. Qiao, and T. Li, "Missing-tag detection and energy-time tradeoff in large-scale rfid systems with unreliable channels," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1079–1091, Aug. 2014.

[22] S. Zhang, X. Liu, J. Wang, J. Cao, and G. M, "Energy-efficient active tag searching in large scale rfid systems," *Information Sciences*, vol. 317, no. 10, pp. 143–156, Oct. 2015.

[23] L. Yang, Y. Qi, J. Han, C. Wang, and Y. Liu, "Shelving interference and joint identification in large-scale rfid systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 11, pp. 3149–3159, Nov. 2015.

[24] Y. H. Chen, S. J. Horng, R. S. Run, J. L. Lai, R. J. Chen, W. C. Chen, Y. Pan, and T. Takao, "A novel anti-collision algorithm in rfid systems for identifying passive tags," *IEEE Transactions on Industrial Informatics.*, vol. 6, no. 1, pp. 105–121, Feb. 2010.

[25] Y.-C. Lai, L.-Y. Hsiao, H.-J. Chen, C.-N. Lai, and J.-W. Lin, "A novel query tree protocol with bit tracking in rfid tag identification," *IEEE Transactions on Mobil Computing*, vol. 12, no. 10, pp. 2063–2075, Oct. 2013.

[26] Y. C. Lai, L. Y. Hsiao, and B. S. Lin, "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.

[27] J. Su, Z. Sheng, G. Wen, and V. C. M. Leung, "A time efficient tag iden-

tification algorithm using dual prefix probe scheme (dpps)," *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 386–389, Mar. 2016.

[28] X. Jia, Q. Feng, and C. Ma, "An efficient anti-collision protocol for rfid tag identification," *IEEE Communications Letters*, vol. 14, no. 11, pp. 1014–1016, Nov. 2010.

[29] L. Pan and H. Wu, "Smart trend-traversal protocol for rfid tag arbitration," *IEEE Transactions on Wireless Communications*, vol. 10, no. 11, pp. 3565–3569, Nov. 2011.

[30] Y. Jiang and R. Zhang, "An adaptive combination query tree protocol for tag identification in rfid systems," *IEEE Communications Letters*, vol. 16, no. 8, pp. 1192–1195, Aug. 2012.

[31] H. Landaluce, A. Perallos, and I. J. G. Zuazola, "A fast rfid identification protocol with low tag complexity," *IEEE Communications Letters*, vol. 17, no. 9, pp. 1704–1706, Sep. 2013.

[32] Y. Cui and Y. Zhao, "Performance evaluation of multi-branch tree algorithm in rfid," *IEEE Transactions on Communications*, vol. 58, no. 5, pp. 1356–1364, May 2010.

[33] J. Choi, I. Lee, D.-Z. Du, and W. Lee, "Fttp: a fast tree traversal protocol for efficient tag identification in rfid networks," *IEEE Communications Letters*, vol. 14, no. 8, pp. 713–715, Aug. 2010.

[34] H. Guo, V. C. M. Leung, and M. Bolic, "M-ary rfid tag splitting with small idle slots," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 1, pp. 177–181, Jan. 2012.

[35] L. Zhu and T.-S. P. Yum, "The optimal reading strategy for epc gen-2 rfid anti-collision systems," *IEEE Transactions on Communications*, vol. 58, no. 9, pp. 2725–2733, Sep. 2010.

[36] C. Jiang, Y. Xu, and Q. Wang, "Cancellation strategy in dynamic framed slotted aloha for rfid system," in *Proc. IEEE Wireless Communications*

and Networking Conference (WCNC 2013), Shanghai, China, Apr. 2013, pp. 854–859.

[37] J. J. Alcaraz, J. VALES-Alonso, E. Egea-López, and J. García-Haro, "A stochastic shortest path model to minimize the reading time in dfsa-based rfid systems," *IEEE Communications Letters*, vol. 17, no. 2, pp. 341–344, Feb. 2013.

[38] W.-T. Chen, "A feasible and easy-to-implement anticollision algorithm for the epcglobal uhf class-1 generation-2 rfid protocol," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 485–491, Apr. 2014.

[39] J. Su, Z. Sheng, and D. Hong, "An efficient frame breaking policy for dynamic framed slotted aloha in rfid," *IEEE Communications Letters*, vol. 20, no. 4, pp. 692–695, Apr. 2016.

[40] W.-T. Chen, "Optimal frame length analysis and an efficient anticollision algorithm with early adjustment of frame length for rfid systems," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3342–3348, May 2016.

[41] Y. Chen, J. Su, and W. Yi, "An efficient and easy-to-implement tag identification algorithm for uhf rfid systems," *IEEE Communications Letters*, 2017, in print.

[42] I. Bratuz, A. Vodopivec, and A. Trost, "Resolving collision in epcglobal class-1 gen-2 system by utilizing the preamble," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5330–5339, Oct. 2014.

[43] H. Vogt, "Efficient object identification with passive rfid tags," in *Proc. the First International Conf. Pervasive Computing (Pervassive'02)*, Zrich, Switzerland, Aug. 2002, pp. 98–113.

[44] J. Vales-Alonso, V. Bueno-Delgado, E. Egea-Lopez, F. Gonzalez-Castano, and J. Alcaraz, "Multiframe maximum-likelihood tag estimation for rfid an-

ticollision protocols," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 487–496, Aug. 2011.

[45] B. Knerr, M. Holzer, C. Angerer, and M. Rupp, "Slot-wise maximum likelihood estimation of the tag population size in fsa protocols," *IEEE Transactions on Communications*, vol. 58, no. 2, pp. 578–585, Feb. 2010.

[46] E. Vahedi, V. S. Wong, and I. F. Blake, "Probabilistic analysis and correction of chen's tag estimation method," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 3, pp. 659–663, Jul. 2011.

[47] M. U. Farooq, M. Asif, S. W. Nabi, and M. A. Qureshi, "Optimal adjustment parameters for epc global rfid anticollision q-algorithm in different traffic scenarios," in *Proc. 10th Int. Conf. Frontiers of Inform. Technol.*, Islamabad, Pakistan, Dec. 2012, pp. 302–305.

[48] P. Šolić, J. Radić, and N. Rožić, "Linearized combinatorial model for optimal frame selection in gen2 rfid system," in *Proc. 2012 IEEE International Conf. RFID*, Orlando, FL, Apr. 2012, pp. 89–94.

[49] E. Felemban, "A revisit on modeling framed slotted aloha anti-collision protocol for rfid systems," in *2012 IEEE International Conf. RFID- Technologies and Applications (RFID-TA)*, Nice, France, Nov. 2012, pp. 315–318.

[50] M.-K. Yeh, J.-R. Jiang, and S.-T. Huang, "Adaptive splitting and presignaling for rfid tag anti-collision," *Computer Communications.*, vol. 32, no. 17, pp. 1862–1870, Nov. 2009.

[51] T. F. L. 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, Feb. 2011.

[52] L. Zhang, J. Zhang, and X. Tang, "Assigned tree slotted aloha rfid tag anti-collision protocols," *IEEE Transactions on Wireless Communications*, vol. 12, no. 11, pp. 5493–5505, Dec. 2013.

[53] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary tree slotted aloha for passive rfid tag anti-collision," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 19–31, Jan. 2013.

[54] S.-C. Tsai, Y.-M. Hu, C.-H. Chai, and J.-S. Li, "Efficient tag reading protocol for large-scale rfid systems with pre-reading," *Computer Communications*, vol. 88, pp. 73–83, Aug. 2016.

[55] H. Guo, C. He, N. Wang, and M. Bolic, "Psr: A novel high efficiency and easy-to-implement parallel algorithm for anticollision in rfid systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1134–1145, Jun. 2016.

[56] Y. Zheng and M. Li, "Pet: probabilistic estimating tree for large-scale rfid estimation," *IEEE Transactions on Mobile Computing*, vol. 11, no. 11, pp. 1763–1774, Nov. 2012.

[57] M. Bonuccelli and F. Martelli, "Instant collision resolution for tag identification in rfid networks," *Ad Hoc Networks*, vol. 5, no. 8, pp. 1220–1232, Nov. 2007.

[58] J. Myung, W. Lee, and T. Shih, "An adaptive memoryless protocol for rfid tag collision arbitration," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 1096–1101, Oct. 2006.

[59] 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, May 2006.

[60] Y.-C. Lai and C.-C. Lin, "Two blocking algorithms on adaptive binary splitting: single and pair resolutions for rfid tag identification," *IEEE Transactions on Networking*, vol. 17, no. 3, pp. 962–975, Jun. 2009.

[61] J.-S. Li and Y.-M. Huo, "An efficient time-bound collision prevention scheme for rfid re-entering tags," *IEEE Transactions on Mobile Computing*, vol. 12, no. 6, pp. 1054–1064, Jun. 2013.

[62] X. Liu, B. Xiao, S. Zhang, and K. Bu, "Unknown tag identification in large rfid systems: an efficient and complete solution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1775–1788, Jun. 2015.

[63] L. Xie, B.Sheng, C. C. Tan, H. Han, Q. Li, and D. Chen, "Efficient tag identification in mobile rfid systems," in *Proc. IEEE INFOCOM 2010*, San Diego, CA, Mar. 2010, pp. 1–9.

[64] J. J. Alcaraz, E. Egea-Lopez, J. Vales-Alonso, and J. Garcia-Haro, "Dynamic system model for optimal configuration of mobile rfid systems," *Computer Networks*, vol. 55, no. 1, pp. 74–83, Jan. 2011.

[65] J. J. Alcaraz, J. Vales-Alonso, and J. Garcia-Haro, "Rfid reader scheduling for reliable identification of moving tags," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 816–828, Jul. 2013.

[66] D. Klair, K.-W. Chin, and R. Raad, "On the energy consumption of pure and slotted aloha based rfid anti-collision protocols," *Computer Communications*, vol. 32, no. 5, pp. 961–973, Mar. 2009.

[67] P. Solic, J. Radic, and N. Rozic, "Energy efficient tag estimation method for aloha-based rfid systems," *IEEE Sensors Journal*, vol. 14, no. 10, pp. 3637–3647, Oct. 2014.

[68] T. Li, S. Wu, S. Chen, and M. Yang, "Energy efficient algorithms for the rfid estimation problem," in *Proc. IEEE INFOCOM 2010*, San Diego, CA, Mar. 2010, pp. 1–9.

[69] W.-J. Yoon and S.-H. Chung, "Iss-tca: an identified slot scan-based tag collection algorithm for performance improvement in active rfid systems," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 3, pp. 1662–1672, Mar. 2012.

[70] F. Vázquez-Gallego, J. Alonso-Zarate, P. Tuset-Peiró, and L. Alonso, "Energy performance of distributed queuing access in machine-to-machine networks with idle-to-saturation transitions," in *Proc. IEEE Globecom Workshop (GC Wkshps 2013)*, Atlanta, GA, Dec. 2013, pp. 867–872.

[71] F. Hessar and S. Roy, "Energy based performance evaluation of passive epc gen2 class1 rfid systems," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1337–1348, Apr. 2013.

[72] Y. Zhu, W. Jiang, Q. Zhang, and H. Guan, "Energy-efficient identification in large-scale rfid systems with handheld reader," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1211–1222, May 2014.

[73] L. Zhang and W. Xiang, "An energy- and time-efficient m-ary detecting tree rfid mac protocol," in *Proc. IEEE ICC 2015*, London, UK, Jun. 2015, pp. 2882–2887.

[74] H. Landaluce, A. Perallos, E. Onieva, L. Arjona, and L. Bengtsson, "An energy and identification time decreasing procedure for memoryless rfid tag anti-collision protocols," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4234–4247, Jun. 2016.

[75] P. Tuset-Peiro, F. Vázquez-Gallego, J. Alonso-Zarate, L. Alonso-Zarate, and X. Vilajosana-Guillen, "Demonstrating low-power distributed queuing for active rfid communications at 433 mhz," in *Proc. 2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, Toronto, Canada, May 2014, pp. 157–158.

[76] *ISO/IEC 18000-6, Information technology - Radio frequency identification for item management- part 6: parameters for air interface communications at 860 MHz to 960 MHz General*, Std., 2013. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics\.htm?csnumber=59644

[77] *EPCglobal C1G2, EPC^{TM} radio-frequency identity protocols generation-2 UHF RFID specification for RFID Air Interface protocol for communications at 860 MHz - 960 MHz*, Std., Version 2.0.1, 2015. [Online]. Available: http://www.gs1.org/sites/default/files/docs/epc/Gen2_Protocol_Standard\%.pdf

[78] L. Kong, L. He, Y. Gu, M.-Y. Wu, and T. He, "A parallel identification

protocol for rfid systems," in *Proc. INFOCOM 2014*, Toronto, Canada, Apr. 2014, pp. 154–162.

[79] R. Rivest, "The md5 message-digest algorithm," Apr. 1992. [Online]. Available: https://tools.ietf.org/pdf/rfc1321.pdf

[80] D. Eastlake and P. Jones, "Us secure hash algorithm 1 (sha1)," Tech. Rep., Sep. 2001. [Online]. Available: https://www.rfc-editor.org/rfc/pdfrfc/rfc3174.txt.pdf

[81] D. Eastlake and T. Hansen, "Us secure hash algorithms (sha and hmac-sha)," Tech. Rep., Jul. 2006. [Online]. Available: https://tools.ietf.org/html/rfc4634

[82] M. Abramowitz and A. Stegun, *Handbook of mathematical functions with formulas, graphs and mathematical tables.* Courier Corporation, Jun. 1964, vol. 55, no. 69.

[83] P. Semiconductors, "I-code smart label rfid tags," 2004. [Online]. Available: http://www.nxp.com/documents/data_sheet/SL092030.pdf

[84] A. Bekkali, S. Zou, A. Kadri, M. Crisp, and R. V. Penty, "Performance analysis of passive uhf rfid systems under cascaded fading channels and interference effects," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1421–1433, Mar. 2015.

[85] J. D. Griffin and G. D. Durgin, "Gains for rf tags using multiple antennas," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 2, pp. 563–570, Feb. 2008.

[86] V. Sarangan, M. R. Devarapalli, and S. Radhakrishnan, "A framework for fast rfid tag reading in static and mobile enviornments," *Computer Networks*, vol. 52, no. 5, pp. 1058–1073, Apr. 2008.

[87] M. A. Kaplan and E. Golko, "Analytic properties of multiple access trees," *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 255–263, Mar. 1985.

[88] L. Zhang, W. Xiang, and X. Tang, "An adaptive anti-collision protocol for large-scale rfid tag identification," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 601–604, Dec. 2014.

[89] *ISO/IEC 14443, Identification cards - contactless integrated circuit cards - proximity cards - Part 3: initialization and anticollision*, Std., 2016. [Online]. Available: https://www.iso.org/standard/70171.html

[90] D. Hush and C. Wood, "Analysis of tree algorithms for rfid arbitration," in *Proc. IEEE Int'l Symp. Informatin Theory*, Cambridge, MA, Aug. 1998, pp. 107–117.

[91] D. Li, H. Wu, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[92] D. M. Dobkin, *The RF in RFID: UHF RFID in practice.* Newness, 2012.

[93] R. Want, "An introduction to rfid technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, Mar. 2006.

[94] S. Chiu, I. Kipnis, M. Loyer, J. Rapp, D. Westbrg, J. Johansson, and P. Johansson, "A 900mhz uhf rfid reader transceiver ic," *IEEE Journal of Solid State Circuits*, vol. 42, no. 12, pp. 2822–2833, Dec. 2007.

[95] B. Kaliski, "The md2 message-digest algorithm," Apr. 1992. [Online]. Available: https://tools.ietf.org/html/rfc1319

[96] R. S. Khasgiwale, R. U. Adyanthaya, and D. W. Engels, "Extracting information from tag collisions," in *Proc. 2009 IEEE International Conf. RFID*, Orlando, FL, Apr. 2009, pp. 131–138.

[97] D. Shen, G. Woo, D. P. Reed, A. B. Lippman, and J. Wang, "Separation of multiple passive rfid signals using software defined radio," in *Proc. 2009 IEEE International Conf. RFID*, Orlando, FL, Apr. 2009, pp. 139–146.

[98] M. Zhang, T. Li, S. Chen, and B. Li, "Using analog network coding to improve the rfid reading throughput," in *Proc. IEEE 30th International*

Conf. Distributed Computing Systems (ICDCS 2010), Genoa, Italy, Jun. 2010, pp. 547–556.

[99] R. Kumar, L. L. Porta, G. Maselli, and C. Petrioli, "Interference cancellation based rfid tags identification," in *Proc. 14th ACM International Conf. Modeling, Analysis and Simulation of Wireless and Mobile Systems (ACM MSWiM 2011)*, Miami Beach, FL, Nov. 2011, pp. 111–118.

[100] K. Fyhn, R. Jacobsen, P. Popovski, A. Scaglione, and T. Larsen, "Multipacket reception of passive uhf rfid tags: a communication theoretic approach," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4225–4237, Sep. 2011.

[101] H. Duan, H. Wu, Y. Zeng, and Y. Chen, "Ls channel estimation and signal separation for uhf rfid tag collision recovery on the physical layer," *Sensors*, vol. 16, no. 4, pp. 442–454, Mar. 2016.

[102] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low power backscatter networks," in *Proc. ACM SIGCOMM 2012*, Helsinki, Finland, Aug. 2012, pp. 61–72.

[103] C. Angerer, R. Langwieser, and M. Rupp, "Rfid reader receivers for physical layer collision recovery," *IEEE Transactions Communications*, vol. 58, no. 12, pp. 3526–3537, Dec. 2010.

[104] G. Carroccia and G. Maselli, "Inducing collisions for fast rfid tag identification," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1838–1841, Oct. 2015.

[105] M. Shahzad and A. X. Liu, "Probabilistic optimal tree hopping for rfid identification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 796–809, Jun. 2015.

[106] Y.-C. Lai and C.-C. Lin, "Two couple-resolution blocking protocols on adaptive query splitting for rfid tag identification," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1450–1463, Oct. 2012.

[107] M. Buettner and D. Wetherall, "An expeirical study of uhf rfid performance," in *Proc. ACM MobiCom*, San Francisco, California, Sep. 2008, pp. 14–19.

[108] L. Xie, Q. Li, C. Wang, X. Chen, and S. Lu, "Exploring the gap between ideal and reality: an experimental study on comtinuous scanning with mobile reader in rfid systems," *IEEE Transactions on Mobile Computing*, vol. 14, no. 11, pp. 2272–2285, Nov. 2015.

[109] Z. G. Fan, S. Qiao, J. T. Huangfu, and L. X. Ran, "Signal descriptions and formulations for long range uhf rfid readers," in *Progress in Electromagnetics Research Symposium 2007 (PIERS 2007 Prague)*, Prague, Czech Republic, Aug. 2007, pp. 109–127.

[110] B. Li and J. Wang, "Efficient anti-collision algorithm utilizing the capture effect for iso 18000-6c rfid protocol," *IEEE Communications Letters*, vol. 15, no. 3, pp. 352–354, Mar. 2011.

[111] H. Wu and Y. Zeng, "Passive rfid tag anticollision algorithm for capture effect," *IEEE Sensors Journal*, vol. 15, no. 1, pp. 218–226, Jan. 2015.

[112] Y. Wang, H. Wu, and Y. Zeng, "Capture-aware estimation for large-scale rfid tags identification," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1274–1277, Sep. 2015.

[113] H. Salah, H. Ahmed, J. Robert, and A. Heuberger, "A time and capture probability aware closed form frame slotted aloha frame length optimization," *IEEE Communications Letters*, vol. 19, no. 11, pp. 2009–2012, Nov. 2015.

[114] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 505–515, Sep. 1979.

[115] G. Ehrlich, "Searching and sorting real numbers," *Journal of Algorithm.*, vol. 2, no. 1, pp. 1–12, Mar. 1981.

[116] M. Tamminen, "Analysis of n-trees," *Inf. Process. Lett.*, vol. 16, no. 3, pp. 131–137, Apr. 1983.

[117] W. Feller, *An introduction to probability theory and its applications*, 3rd ed. New York: Jhon Willey & Sons, 1968.