STATISTICAL APPROACHES FOR THE ANALYSIS, MEASUREMENT, AND MODELING OF RFID SYSTEMS

By

Liyan Wang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2018

ABSTRACT

STATISTICAL APPROACHES FOR THE ANALYSIS, MEASUREMENT, AND MODELING OF RFID SYSTEMS

By

Liyan Wang

The goal of this thesis is to develop statistical and learning algorithms for the analysis, measurement, and modeling of wireless networking(Radio frequency identification systems). Next, I will give a brief overview of those topics.

Radio frequency identification (RFID) systems are widely used in logistic, supply chain industry and inventory management. RFID is already in use in multiple industries and for various purposes. The device in your car that lets you zoom by in the fast lane at a tollbooth, while deducting a dollar amount from your account, is an example of RFID technology in everyday use. Mostly, existing RFID systems are primarily used to identify the RFID tags present in a tag population(*e*.g., tracking a specific tag from a tag population) while identifying some specific tags is a critical operation, it is usually very time consuming and is not desired or nessary in some situations. For instance, if the objective is to determine whether any of the tags are missing(*e*.g., to detect some items according to a consignment), the first thing to do is to identify all tags' ID and then compare with the original record to determine if there is any tags are missing. Definitely, the whole process will be very slow if we have a very large tag population. In this thesis, I present novel statistical algorithms to enable fast and new applications in RFID systems. For example, detecting the missing tags in a large tag population with high accuracy while using the existing infrastructure of RFID systems which is already deployed in industry. More pacifically, I present my work on designing statistical algorithms for estimation the number of missing tags in a population of RFID tags, for detecting and identifying the missing tags from a population of RFID tags. The key distinction of my work compared to prior art is that my methods are compliant with EPCGlobal Class 1 Generation 2 (C1G2) RFID standard. It is critical for RFID methods to

be compliant with the C1G2 standard since the commercially available of-the-shelf RFID equipment follows the C1G2 standard. A method which does not comply with the C1G2 standard cannot be deployed on the existing installations of RFID systems because it requires custom hardware, which will cost a lot. In an RFID-enabled warehouse, there may be thousands of tagged items that belong to different categories, *e.g.*, different places of origin or different brands [207]. Each tag attached to an item has a unique ID that consists of two fields: a category ID that specifies the category of the attached object, and a member ID that identifies this object within its category. As a manager of the warehouse, one may desire to timely monitor the product stock of each category. If the stock of a category is too high, it may indicate that this product category is not popular, and the seller needs to adjust the marketing strategy (*e.g.*, lowering prices to increase sales). On the contrary, if the stock of a category is too low, the seller should perform stock replenishment as soon as possible. Manual checking is laborious and of low time-efficiency. You cannot imagine how difficult it is for a manager to manually count the number of items in each category that may be stacked together or placed on high shelves. Hence, it is desirable to exploit the RFID technique to quickly obtain the number of tagged items in each category. A multi-category RFID estimation protocol should satisfy three additional requirements. First, it should be standard compliant; otherwise, it will be difficult to be deployed. Second, it should preserve the privacy of tags by not reading their member IDs. Third, it should work with both a single-reader and multiple-reader environments. As the communication range between a tag and a reader is limited, a large population of tags is often covered by multiple readers whose regions often overlap.

To my parents,
for their support and encouragement.

ACKNOWLEDGEMENTS

Working towards a Ph.D. has been a deeply enriching and rewarding experience. Looking back, many people have helped shape my journey. I would like to extend them my thanks.

- First and foremost, I would like to thank to my PhD advisor, Prof. Alex X. Liu, for supporting me during five past years. He is the one of the smartest people I know. My work would not have been possible without his constant guidance, his unwavering encouragement, his many insights, and his exceptional resourcefulness. For all of this, Alex, thank you.
- I would also like to thank the rest of my thesis committee Profs. Yiying Tong, Xiaoming Liu and Susan Selke for their helpful research advice and suggestion in general.
- I would also like to thank Dr. Muhammad Shahzad, thank you for numerous insightful suggestions and discussion on my projects. I will remember how you encourage me to do research and think about the solution. Never give up and never stop thinking.
- Thanks NSF for supporting my Ph.D. .
- My great thanks to Michigan State University and the Department of Computer Science and Engineering for providing me financial support to attend the conferences.
- Many thanks to my colleagues in Systems and Security Lab at Michigan State University. In particular, I would like to thank Muhammad Zubair Shafiq, Kamran Ali, Ali Munir, and Faraz Ahmed for your friendship.
- I must say that I owe my great time in Michigan State University to all of my fabulous friends. It is simply not feasible to list all of them here. I would like to thank them all for their friendship and support.
- I must say thanks to my great parents. Without their encouragement, I even don't have a choice to pursue my PhD in U.S. Also, I love you more than I can say.
- I must sat thanks to my great friend Heidi. You gave me a lot suggestions and advices to help me face with a lot of problems in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Challenges and Motivation

In this thesis I present my work on measurement, modeling, design, and analysis of RFID systems. For RFID systems, I present my work on probabilistic network measurements. My research is focus on the modeling, design, and analysis of probabilistic measurement schemes for radio frequency identification (RFID) systems. It deserves to be specially noted that I present my work on designing statistical algorithms for estimating the number of missing tags in a population of RFID tags, for optimizing the RFID estimation protocol, and for identifying missing tags from a population of RFID tags. The main difference between the prior art and my work is that my approaches are compliant with the EPCGlobal Class 1 Generation 2 (C1G2) RFID standard.

It is quite important for RFID schemes to be compliant with the C1G2 standard because the commercially available off-the-shelf RFID equipment follows the C1G2 standard. complying with C1G2 standard is an important factor in determining the schemas which can be promoted in the commercial applications. A scheme that does not comply with the C1G2 standard cannot be deployed on the existing installations of RFID systems since it requires custom tags or readers, which will cost a lot.

1.2 Contributions

This thesis takes an in-depth look at the following research problems.

1.2.1 RFID Missing Tag Searching [55, 219]

We address the fundamental problem of estimating RFID missing tag population size, which is needed in many applications such as consignment identification, warehouse monitoring,

and privacy sensitive RFID systems. We propose a new scheme for estimating missing tag population size named *RFID Tag Searching Protocol*(RTSP) The technique is based on the average number of 1s in the receiving frames and the location of 1 in the receiving frames while using the standardized framed slotted Aloha protocol. RTSP is significantly faster than prior schemes. For example, given a required confidence interval of 0.1% and the number of missing tags is 500, the tag population is 5000, RTSP takes 15 seconds to search the tags whereas the fastest priori tag identification protocol needs 40 seconds.(TH)

1.2.2 RFID Multi-category Tag Estimation [115]

We concern the practically important problem of multi-category RFID estimation: given a set of RFID tags, we want to quickly and accurately estimate the number of tags in each category. However, almost all the existing RFID estimation protocols are dedicated to the estimation problem on a single set, regardless of tag categories. ART, the faster estimation protocol which is based on the average run-length of 1s in the bit string received using the standardized framed slotted Aloha protocol does not consider any tag categories. A feasible solution is to separately execute the existing estimation protocols on each category. The execution time of such a serial solution is proportional to the number of categories, and cannot satisfy the delay-stringent application scenarios. Simultaneous RFID estimation over multiple categories is desirable, hence, this paper proposes an approach called Simultaneous Estimation for Multi-category RFID systems (SEM). SEM exploits the Manchester-coding mechanism, which is supported by the ISO 18000-6 RFID standard, to decode the combined signals, thereby simultaneously obtaining the reply status of tags from each category. As a result, multiple bit vectors are decoded from just one physical slotted frame. Built on our SEM, many existing excellent estimation protocols can be used to estimate the tag cardinality of each category in a simultaneous manner. To ensure the predefined accuracy, we calculate the variance of the estimate in one round, as well as the variance of the average estimate in multiple rounds. To find the optimal frame size, we propose an efficient binary search-based algorithm. To address

significant variance in category sizes, we propose an Adaptive Partitioning (AP) strategy to group categories of similar sizes together and execute the estimation protocol for each group separately. Compared with the existing protocols, our approach is much faster, meanwhile satisfying the predefined estimation accuracy. For example, with 20 categories, the proposed SEM+AP is about 7 times faster than prior estimation schemes.

Moreover, our approach is the only one whose normalized estimation time (*i*.e., time per category) decreases as the number of categories increases.

1.2.3 RFID Valued Missing Tag Detection [119]

As I mentioned before, most methods are not considering the missing tags. The only methods to estimate the missing tag is using the identification protocol to identify all the tags then figure out all the missing tags. However, mostly, there is no necessary to identify all the tags. Moreover, sometimes people just care about the values tags(tags attached to the values items). In this scenario we just have to detect and identify all the missing tag with the value equal to or greater than the threshold which can be set by users. With this fundamental problem for different tag population size and different values of each tag, our approach have to meet different accuracy requirements based on the users definitions. Our exploratory analysis uncovers several statistically significant findings that have important implications for software development and deployment.

Based on the Single-one Manchester coding method we can easily classify the different valued tags into groups. For example, tags with value greater than 100 will be in group 1 which demand the higher accuracy requirement while group 2 with lower accuracy is consist of tags whose value are smaller than 100. After grouping we can use different estimation protocol to satisfy the different accuracy requirements. Apparently, detecting missing expensive tags with higher accuracy will need more time.

CHAPTER 2

RFID MISSING TAGS SEARCHING

2.1 Introduction

2.1.1 Motivation and Problem Statement

RFID doesn't provide much value on its own, but with RFID many companies can develop a lot applications which can make a lot profit. RFID systems can help to connect all the things into network which enable companies to communicate, educate, sell, entertain and distribute the products. As the RFID tags are cheaper and cheaper, RFID enables companies to do many different things. As we known, RFID is used to identify objects or even people by attaching a tag which including all the information we can track. Its advantage is no human intervention. Tags can be read by a reader and the information(*e.g.*, the tag ID, the reader's ID and the time the tag was read) can be transmitted to computers in real time. One of the most common uses of RFID is asset tracking. Companies attach tags to the asset to prevent from stealing. RFID has been used in manufacturing plants for more than a decade. RFID technology can track parts and work in process and to reduce defects while increasing throughput and managing the production of different versions of the products. RFID technology has been used in closed loop supply chains for years. Companies distribute RFID system to increase throughput, reduce shipping error, costs and save labor costs. Most retailers such as BestBuy, Metro, Target, Tesco, Wal-mart and Amazon are in the forefront of RFID adoption. Retailer are currently focus on improving supply chain efficiency and making sure products are on the right shelf when customers want to purchase. RFID technology is catching on as a convenient payment mechanism. One of the most popular uses of RFID is to pay for road tolls without stopping. Using as an electronic key to control the access to office or buildings is one of important application of RFID. There are many other innovative uses

for RFID. Locating children at theme parks, combining RFID tags with temperature loggers, motion sensors, radiation sensors to achieve more important goal. Object tracking using RFID since it is very convenience and low cost to deploy [33, 208, 20]. Since RFID tags can attach to most items, they are able to support for localization, aiming to pinpoint objects in 3D space referring as 3D positioning [200, 40, 144, 131]. Antitheft can be very useful in large warehouse since the cost of commercial RFID tags is negligible compared to the value of the products to which they are attached.(*e.g.*, just 5 cents per tag [21]). An RFID system consists of tags, readers and servers. A tag is a microchip with an antenna in a compact package with the limited computing power and communication range. There are three types of tags [87, 104]: (1) passive tags, which are powered up by harvesting the radio frequency energy from readers (as they do not have their own power sources) and have communication range often less than 10 meters; (2) active tags, which have their own power sources and transmitter thus have relatively longer communication range; (3) Battery-Assisted passive (BAP) tags, which have the internal power source to power on, and energy transferred from the reader to backscatter and have the moderate communication range. Active RFID systems typically operate in the ultra-high frequency (UHF) band and offer a range of up to 100 meters or more. In general, active tags are used on large objects, such as rail cars, big reusable containers, and other assets that need to be tracked over long distances. Passive RFID system with the passive tags which required strong signal from readers to power on. Because passive tags do not require a power source or transmitter, and only require a tag chip and antenna, they are cheaper, smaller, and easier to manufacture than active tags. While most passive RFID tags use the energy from the RFID reader's radio signal to power on the tag's integreted chip and backscatter to the reader, BAP tags use an integrated power source (usually a battery) to power on the chip, so all of the captured energy from the reader can be used for backscatter. Unlike transponders, BAP tags do not have their own transmitters.

An RFID reader has a dedicated power source with significant computing power. An RFID reader's function is to interrogate RFID tags. The means of interrogation is wireless

Table 2.1 RFID systems with different tags

	Passive RFID	**Active RFID**	**Battery-Assisted Passive RFID**
Tag Power Source	Energy transfer from the reader via RF	Internal to tag	Tag uses internal power source to power on, and energy transferred from the reader via RF to backscatter
Tag battery	No	Yes	Yes
Availability of Tag Power	Only within field of reader	continuous	only within field of reader
Required Signal Strength from Reader to Tag	very high(must power the tag)	very low	Moderate (does not need to power tag, but must power backscatter)
Available Signal Strength from Tag to Reader	Low	High	Moderate
Communication Rage	Short range(up to 10m)	Long range(100m or more)	Moderate range(up to 100m)
Data transfer	Ability to read and transfer sensor values only when tag is powered by reader	Ability to continuously monitor and record sensor input	Ability to read and transfer sensor values only tag receives RF signal from reader

and because the distance is relatively short; line of sight between the reader and tags are not necessary. A reader contains an RF module, which acts as both a transmitter and receiver of radio frequency signal. The transmitter consist of an oscillator to create the carrier frequency; a modulator to impinge data commands upon this carrier signal and an amplifier to boost the signal enough to awaken the tag. The receiver has a demodulator to extract the returned data and also contains an amplifier to strengthen the signal for processing. A microprocessor forms the control unit, which employs an operating system and memory to filter and store the data. The data is now ready to be sent to the network. It transmits a query to a set of tags and the tags respond over a shared wireless medium. RFID reader types are fixed, mobile or handhold units. Which type to use is governed by the application or environment in which they will be utilized. Fixed readers are often used for large-scale deployments;

installed in portals at dock doors and conveyor belts to capture inventory or for tracking parts, tools and equipment. Fixed RFID readers require access to a ground power source and usually connect to the network by cables such as RS-232 or USB. Mobile RFID readers come into play for hard to reach areas where it would be difficult to install a fixed reader. Their robustness is beneficial when it comes to mounting them on moving vehicles such as forklifts. When self-contained, with their own battery and antennas, their wireless communication allows them to connect to a network from a trolley or cart. Handhold RFID readers are light, compact and ruggedly built to withstand being mishandled. By tethering a cable to be reader, you can assure yourself of having constant power and communication to the network. Because mobility is usually more important, most have wireless capability with integrated antennas and a rechargeable battery.

This chapter concerns the fundamental problem of estimating the size of a missing tag cardinality of a given large tag set with high speed and high estimation accuracy. This is needed in many applications such as tag identification, privacy sensitive RFID systems, theft detecting and warehouse monitoring. In missing tag detection and identification protocols, which detect all the missing tags from an unknown tag population, the size of the missing tag population is estimated at the start to guide the identification process. For example, as tag identification protocols which are based on the framed slotted Aloha protocol (standardized in EPCGlobal Class-1 Generation-2 (C1G2) RFID standard and inlemanted in commercial RFID systems), missing tag estimation is often used to calculate the optimal frame size. In privacy sensitive RFID systems, for example, those systems used in museums for continuously monitoring the number of visitors in different areas of a museum to plan the guided trips efficiently, readers may not have the permission to identify human individuals. In warehouse with RFID-based monitoring systems, managers sometimes need to estimate the number of sold products quickly in order to estimate the stock of products or detection of employee theft. It is very straight forward to use the tag identification protocols to accurately measure the missing tag population. However, it will be very slow if we use identification protocols.

For example, for a population of 5k tags called A, we have a another known tag set with size 1k which we named it B. We want to know how many tag are missing in 5k A tags comparing with this 1k B tags. The missing tags are the tag which only exist in B not in A. This is very important to estimate those missing tag immediately. Now we formally define the tag search problem. Let A represent the set of IDs of tags that we want to search for in a population. We know exactly which IDs are present in set A. Let B represent the set of IDs of tags in the population in which we search for tags in set A. Let C represent the set of IDs of those tags that are present in both sets A and B. We do not have any prior knowledge about the IDs in sets B or C, however, we do know that $C \subseteq A$ and $C \subseteq B$. Let $\tilde{C}$ represent the set of IDs in C that the tag search protocol returns, where $C \subseteq \tilde{C} \subseteq A$. As most application can tolerate a small error in determining the IDs in set C, for a required confidence interval of β, our objective is to design a tag search protocol that uses a set of readers to quickly generate set $\tilde{C}$ such that $|\tilde{C}| - |C| \leq \beta|C|$. Confidence interval β represents the maximum tolerable fraction of tags in A that are not in C but are declared as members of C by RTSP. Additionally, a tag search protocol should work in single as well as multiple-reader environments, and should be compliant with the C1G2 standard.

2.1.2 Proposed Approach

For the problem of searching tags with IDs in set A in population with IDs in set B, there is a seemingly obvious solution based on RFID tag collection protocol. Execute an RFID tag collection protocol to first collect IDs of all tags in set B and then compare them with the IDs in set A. This will identify all tags in set A that are present in set B. This solution works; however, it is too slow. For example, our experimental results show that even the fastest existing tag collection protocol TH [176] is 2 times slower than our scheme. Slow searching of RFID tags may have unbearable consequences in time critical applications especially when tag search has to be performed for hundreds of thousands of different kinds of products such as in case of Amazon warehouses for balancing inventory. Furthermore, this solution can not

be used in settings where readers are not allowed to read the IDs of tags in set B due to privacy reasons. An example of such a setting is a multi-tenant warehouse, where one tenant may not permit readers of other tenants to read the IDs of its tags.

In this paper, we propose an identification tag search protocol called RFID Tag Search Protocol (RTSP) that can quickly identify tags in set C while ensuring that the requirement $|\tilde{C}| - |C| \leq \beta|C| \geq \alpha$ is satisfied. RTSP uses the frame slotted Aloha protocol specified in the C1G2 standard as its MAC layer communication protocol. In Aloha protocol, the reader first tells the tags a frame size f and a random seed number R. Later in the paper, we will see how a simple use of seed number R will make it straightforward to handle multiple readers with overlapping regions. Each tag within the transmission range of the reader then uses f, R, and its ID to select a slot in the frame by evaluating a hash function $h(f, R, ID)$ whose result is uniformly distributed in $[1, f]$. Each tag has a counter initialized with the slot number it chose to reply. After each slot, the reader first transmits an end of slot signal and then each tag decrements its counter by one. In any given slot, all the tags whose counters equal 1 respond with a random sequence called RN16. If no tag replies in a slot, it is called an *empty slot*. If one or more tags reply in a slot, it is called a *nonempty slot*. As per the C1G2 standard, tags do not transmit their IDs unless the reader specifically asks them to do so. In RTSP, reader checks if a slot is empty or nonempty using the RN16 sequence and never asks tags to transmit their IDs. This preserves the privacy in settings where a reader is not allowed to read IDs of tags in set B. C1G2 standard provisions this functionality of not asking the tags for their IDs.

To identify the tags in set C, *i*.e., the tags in set A that are present in population of set B, RTSP executes multiple Aloha frames with different seeds. In each frame, each tag uses the seed for that frame to select its slot. As RTSP already knows the IDs of all tags in set A, it pre-computes which tags in A will select which slots in the frames. Thus, it knows which slot in the frame must be nonempty if a certain ID in A is present in B. When a reader executes a frame, RTSP compares the response in each slot of that frame with the corresponding slot

in the pre-computed frame. RTSP continues and executes n frames with different seeds. At the end of n frames, for any given tag in A, if RTSP observes that the n slots this tag was supposed to respond in the n executed frames all turned out to be non-empty, it marks that tag in A to be present in B. If, however, RTSP observes that any of the n slots this tag was supposed to respond in the n executed frames turned out to be empty, it marks that tag in A as absent in B. The value of n is chosen such that in executing n frames, for any given tag in A that is not present in B, with a high probability, RTSP will see at least one slot in one of the n frames, which is 1 in pre-computed frame due to this tag but 0 in the executed frame.

Our proposed protocol works with multiple readers with overlapping regions. To handle multiple readers, RTSP uses a central controller for all readers to use same values of frame size f and seed R across all readers. When a reader transmits seed R_i in its i^{th} frame, it does not generate R_i on its own, rather it uses the i^{th} seed R_i issued by the central controller. That is, each reader generates the same sequence of seeds in consecutive frames. As all readers use the same seed R_i in the i^{th} frame, the slot number that a particular tag chooses in the i^{th} frame of each reader covering this tag is the same *i*.e., $h(f, R_i, ID)$ evaluated by the tag results in same value for each reader. Once a reader completes its frame, it sends the responses to the central controller. The controller applies logical OR operator on all the i^{th} frames from all readers and gets a single i^{th} frame as if returned by one reader covering the entire tag population. The controller repeats this process until it has n ORed frames and then determine which tags in A are present in B.

2.1.3 Limitation of Prior Art

There are two types of RFID tag search protocols: *estimating tag search protocols* that estimate the cardinality of set C [175] and *identification tag search protocols* that identify the IDs of tags in set C [176, 174, 179, 120]. Estimating tag search protocol is faster but does not return the IDs in set C. Identification tag search protocols return the IDs in set C but are comparatively slower. Both approaches have their merits. In fact, they are complementary to

each other, and should be used together. For example, an estimating tag search protocol can be used to determine if any tags in set A are present in set B, and if true, an identification tag search protocol should be invoked to identify the tags in C.

There are two major limitations of existing protocols. First, they can not achieve arbitrarily small confidence interval. Second, except KCTP (which does not return the IDs in set C), none of the existing protocols is compliant with the EPCGlobal Class 1 Generation 2 (C1G2) RFID standard [82] because they require the tags to receive, interpret, and act either according to pre-frame Bloom Filters or according to protocol specific parameters. Such functionalities are not provisioned in the C1G2 standard because tags, especially the passive ones, do not have enough computational power. It is important for an RFID protocol to be compliant with the C1G2 standard because the cheap commercially available off-the-shelf (COTS) tags follow the C1G2 standard. A protocol that is not compliant with the C1G2 standard will require custom tags, which will not only cost more but will also work only in limited settings. For example, if an airline uses a protocol and tags that are non-compliant with the C1G2 standard, it may be able to track its baggage at its home airport but not at the airports in rest of the world, which support only the C1G2 compliant tags.

2.2 RFID Tag Searching Background

2.2.1 RFID Tag Searching Motivation and Problem Statement

As the cost of commercial RFID tags, which is as low as 5 cents per tag [160], RFID system has become negligible compared to the prices of the products to which they are attached, RFID systems have been increasingly used in various applications such as supply chain management [94], indoor localization [213, 149], 3D positioning [195], object tracking [141], inventory control, electronic toll collection, and access control [60, 140]. For example, Walmart uses RFID tags to track expensive clothing merchandize [161] and Honeywell Aerospace uses RFID tags to track its products from birth to repair and retirement [185]. As we mentioned before, an RFID system consists of tags and readers. A tag is a microchip with an integrated

antenna in a compact package that has limited computing power and communication range. There are two types of tags: passive tags and active tags. Passive tags do not have their own power source, are powered up by harvesting the radio frequency energy from readers, and have communication ranges often less than 20 feet. Active tags have their own power sources and have relatively longer communication ranges. A reader has a dedicated power source with a significant amount of computing power. RFID systems mostly work in a query-response fashion where a reader transmits queries to a set of tags and the tags respond with their IDs over a shared wireless medium. Based on those RFID systems, we address the fundamental problem of RFID tag searching which can be stated as, *given a set of known tag IDs and a population of RFID tags with unknown IDs, where the tags may be passive or active, we want to know which tag IDs are in the tag population, i.e., search in a population of unknown tags for a set of known IDs.* Searching tags with unknown IDs has many applications such as products recall, inventory balancing, and stock verification. For product recall, if a manufacturer suspects that some of its products, which have already been distributed in different warehouses, are defective, they can use a tag searching protocol to quickly locate defective products, where the known tag IDs are defective products and the tag population are the products in a warehouse. For inventory balancing, if a large retailer, such as Amazon, wants to balance the quantity of different products among its warehouses across the country to reduce shipping time and costs, they can use a tag searching protocol to determine the quantity of any given product in each warehouse and then balance the quantity among warehouses accordingly, where the known tag IDs are the ones in inventory and the tag population are the ones in a warehouse. For stock verification, if a large retailer wants to check the quantity of each requested product sent to it in a large consignment, they can use a tag searching protocol to determine whether the consignment contains all requested products, where the known tag IDs are the ones that they are expecting and the tag population are the ones in the consignment. In this report, we use the three terms, a tag, a tag ID, and the product that a tag is attached to, interchangeably.

The tag searching problem can be formally defined as: *Given a set A, which is a set of known tag IDs, a set B, which is a population of RFID tags with unknown IDs, a required confidence interval β, a tag searching protocol outputs $\tilde{C}$ so that $C \subseteq \tilde{C} \subseteq A$ and $|\tilde{C}| - |C| \leq \beta|C|$, where $C = A \cap B$.* Confidence interval β represents the maximum tolerable fraction of tags in A that are not in C but are declared as members of C by a tag searching protocol. A tag searching protocol should satisfy three additional requirements:

- First, it should comply with the EPCGlobal Class 1 Generation 2 (C1G2) RFID standard [82]. Otherwise, it will be extremely difficult to be practically deployed because commercial RFID readers and tags are typically C1G2 compliant.
- Second, it should preserve the privacy of the RFID tags in set B by not reading their tag IDs. Many RFID tag searching applications need to satisfy this privacy requirement. For example, if a policeman searches for some items with known tag IDs in a private house with a population of tags with unknown tag IDs, the home owner may prefer not to read the IDs of all tags in the house.
- Third, it should work with both a single-reader and multiple-reader environments. As the communication range between a tag and a reader is limited, a large population of tags is often covered by multiple readers whose regions often overlap.

In this report, a protocol called RFID Tag Searching Protocol (RTSP) is proposed to solve RFID tag searching problem, which satisfies the following four requirement: (1) C1G2 compliance, (2) arbitrary accuracy, *i*.e., $C \subseteq \tilde{C} \subseteq A$ and $|\tilde{C}| - |C| \leq \beta|C|$ for any required confidence interval β, (3) privacy preserving, and (4) multiple-reader capability.

To satisfy the requirement of C1G2 compliance, RTSP uses the frame slotted Aloha protocol specified in the C1G2 standard as its MAC layer communication protocol. In Aloha, the reader first tells the tags a frame size f and a random seed number R. Each tag within the transmission range of the reader then uses f, R, and its ID to select a slot in the frame by calculating a hash function $h(f, R, ID)$ whose result is uniformly distributed in $[1, f]$. Each tag has a counter initialized with the slot number that it chose to reply. After each slot,

the reader first transmits an end of slot signal and then each tag decrements its counter by one. In any given slot, all the tags whose counters equal 1 respond with a random sequence called RN16. If no tag replies in a slot, it is called an *empty slot*. If one or more tags reply in a slot, it is called a *nonempty slot*. Using 0 to denote an empty slot and 1 to denote a nonempty slot, after we execute the Aloha protocol on a population A of tags using frame size f and random seed R, we obtain a binary array of f bits, denoted as $\mathbb{S}(A, f, R)$.

To satisfy the requirement of arbitrary accuracy, RTSP executes n runs of the Aloha protocol where each run uses a different seed. For the i^{th} run with frame size f and random seed R_i, RTSP executes the Aloha protocol on both sets A and B, and thus obtains two binary arrays $\mathbb{S}(A, f, R_i)$ and $\mathbb{S}(B, f, R_i)$. Note that RTSP executes the Aloha protocol on A virtually as it knows all tag IDs in A. After n runs, for each tag ID $t \in A$, if for all $1 \leq i \leq n$, we have $\mathbb{S}(A, f, R_i)[h(f, R_i, t)] = \mathbb{S}(B, f, R_i)[h(f, R_i, t)]$, (i.e., for all n runs, the two bits corresponding to tag t in both $\mathbb{S}(A, f, R_i)$ and $\mathbb{S}(B, f, R_i)$ are 1), then RTSP outputs $t \in \tilde{C}$. Clearly RTSP satisfies $C \subseteq \tilde{C} \subseteq A$. RTSP chooses a value of n so that $|\tilde{C}| - |C| \leq \beta|C|$.

To satisfy the requirement of privacy preserving, RTSP checks if a slot is empty or nonempty using the RN16 sequence and never asks tags to transmit their IDs. In C1G2, tags do not transmit their IDs unless the reader specifically asks them.

To satisfy the requirement of multi-reader capability, RTSP uses a central controller for all readers to use the same values for frame size f and seed R across all readers. When a reader transmits seed R_i in its i^{th} frame, it does not generate R_i on its own, rather, it uses the i^{th} seed R_i issued by the central controller. Thus, for a tag $t \in B$ that is covered by multiple readers, it chooses the same slot $h(f, R_i, t)$ for all readers. Once a reader completes its frame, it sends its binary array to the central controller. The controller applies the bit-wise logical OR operation on the binary arrays returned from all readers. The resulting binary array is the same as if there is one reader that covers all tags. RTSP uses this binary array to compute $\tilde{C}$.

The key novelty of RTSP is that it statistically guarantees to achieve any required

accuracy and complies with the C1G2 standard. The key technical depth of RTSP lies in its mathematical development to guarantee the arbitrary required accuracy and to minimize tag searching time. The key advantages of RTSP over prior tag searching protocols are that RTSP can achieve arbitrarily high accuracy and RTSP complies with the C1G2 standard. RTSP is easy to deploy because it neither requires modification to tags nor to the communication protocol between tags and readers. RTSP can be implemented as a software module on readers. We have extensively evaluated the performance of RTSP. Our results show that for a scenario with $|A| = 5000$, $|B| = 5000$, and $|C| = 500$, and a required confidence interval of 0.1%, RTSP takes 15 seconds to search the tags whereas the fastest prior tag identification protocol (TH [176]) takes 22 seconds.

2.3 Review of Related Research

To the best of our knowledge, there are only three identification tag search protocols [221, 54, 217] and one estimating tag search protocol [116]. By identification, we mean those protocols identify all the tags IDs firstly, then give the result by comparing with the IDs we want to search. And none of them satisfy all four requirements simultaneously. Next, I review these identification and estimating tag search protocols in this chapter.

2.3.1 Identification Tag Search Protocols

Zheng and Li proposed the first RFID tag search protocol namely CATS [221]. CATS works in two phases. In the first phase, a server first constructs a Bloom Filter by applying multiple hash functions in conjunction with a random seed on each tag ID in set A. Second, an RFID reader broadcasts the bit array of Bloom Filter generated by the server along with the random seed to all tags in the population B. Third, on receiving the broadcast, each tag constructs a Bloom Filter using the seed and its own ID. Fourth, if a tag finds that all the bits it has set to 1 in its local Bloom Filter are also set to 1 in the Bloom Filter array broadcasted by the reader, it considers itself as a candidate tag that the reader is searching for and thus stays

awake to participate in the next round; otherwise, it sleeps and does not participate in future rounds. The server and reader repeat the process of generating and transmitting Bloom Filter arrays with different seeds until most of the tags that are left awake are those that the reader is searching for. In the second phase, the reader executes standard Aloha protocol to identify the tags that are awake. Unfortunately, C1G2 compliant tags can not interpret or generate Bloom Filters, which makes CATS non-compliant with the C1G2 standard.

Chen *et* al. proposed another tag search protocol called ITSP, which is an improved version of CATS [54]. The authors of ITSP realized that the Bloom Filter array that CATS uses may be much larger than 96-bits. Therefore, they proposed to segment the implementation of Bloom Filter into small arrays. The major difference between CATS and ITSP is that in CATS, in the first phase, a reader transmits a single Bloom Filter array all at once, whereas in ITSP, reader only transmit a segment of the Bloom Filter to shrink the candidate set. In addition to using segmented Bloom Filters, authors also proposed to observe the empty and non-empty slots in the second phase and compare them against pre-computed frames to further filter out any tags not in B that were not filtered out by the Bloom Filters. Unfortunately, C1G2 compliant tags can not interpret or generate Bloom Filters, which makes ITSP non-compliant with the C1G2 standard.

Zhang *et* al. proposed another tag search protocol called TSM [217]. TSM extends CATS for use with multiple readers. It first executes CATS using each reader and then aggregates results from all readers to identify the tags in A that are present in B. Unfortunately, due to similar reasons as for CATS, TSM is also non-compliant with the C1G2 standard. In contrast, our proposed protocol RTSP is C1G2 compliant.

2.3.2 Estimating Tag Search Protocols

Liu *et* al. proposed Basic Key tag Counting protocol ($B-KC$ to count the number of tags in A that are present in B [116]. In stead of observing the whole time frame, reader in $B-KC$ just need to focus on the singleton slots. $B-KC$ first pre-computes a frame using IDs in set

A and then executes a frame on population B to determine how many times the slots that were 1 in the pre-computed frame turned out to be 1 in the executed frame. It then uses the number of such slots to obtain the estimate of the number of tags in A that are present in B. $B - KC$ falls short because it can only estimate the number of tags in A that are present in B, but it can not determine exactly which tags of A are present in B. Another fast estimation scheme is ART. Executing time of ART which is providing in [96] is provably independent of the tag population size. In contrast, our proposed protocol $RTSP$ can identify such tags.

2.4 Proposed Research

In this chapter, I explain the proposed research methods to address the problem of RFID tags searching.

2.4.1 Architecture

For searching RFID tags, RTSP uses a central controller connected with a set of readers that cover the area where the tags in set B are located. The use of a central controller ensures that all readers use consistent values of frame sizes and seeds when executing frames, which helps in efficiently aggregating and processing information returned by the readers. The readers use the standardized frame slotted Aloha protocol to communicate with tags and never ask the tags to transmit their IDs. The use of multiple readers with overlapping coverage regions introduces following two problems: (1) scheduling the readers such that no two readers with overlapping regions transmit at the same time, and (2) alleviating the effect of some tags responding to multiple readers due to overlap in the coverage region of those readers. For the first problem, the controller uses one of the several existing reader scheduling protocols [188] to avoid reader-reader collisions. For the second problem, we propose solution in Section 2.5.1. RTSP does not require any modifications to tags or readers. It only requires the readers to receive system parameters from the controller and communicate the responses in the frames back to the controller.

2.4.2 C1G2 Compliance

RTSP does not require any modifications to tags or readers. It only requires the readers to receive the frame size, persistence probability, and seed number from the controller and communicate the responses in the frames back to the controller. Persistence probability p is the probability with which a tag decides whether it will participate in a frame or not before selecting a slot in that frame. Later in the paper, we will show how we use p to handle frame sizes that exceed the C1G2 specified upper limit of 2^{15}. Such large frame sizes are required when the size of tag population is large and required confidence interval β is small. With the use of p, the reader reduces the number of tags that participate in each frame, which in turn reduces the optimal frame size at the expense of increased number of frames. As the C1G2 standard does not specify the use of p, COTS tags do not support it. To avoid making any modifications to tags, in RTSP, the reader implements p by announcing a frame size of f/p but terminating the frame after the first f slots, which can be done as per the C1G2 standard.

2.4.3 Communication Channel

We assume that the communication channel between readers and tags is reliable *i*.e., tags correctly receives queries from the readers and the readers correctly detect transmission of RN16 sequence in a slot if one or more tags in the population transmit in that slot. If the channel is unreliable, the solution proposed in [176] can be easily adapted for use with RTSP.

2.4.4 Formal Development Assumption

To make the formal development tractable, we assume that instead of picking a single slot to transmit at the start of i^{th} frame of size f, a tag independently decides to transmit in each slot of the frame with probability $1/f$ regardless of its decision about previous or forthcoming slots. Vogt first used this assumption for the analysis of Aloha protocol for RFID and justified

its use by recognizing that this problem belongs to a class of problems called *occupancy problem*, which deals with the allocation of balls to urns [193]. Ever since, the use of this assumption has become a norm in the formal analysis of all Aloha based RFID protocols [175, 193, 218].

The implication of this assumption is that a tag can end up choosing more than one slots in the same frame or even not choosing any at all, which is not in accordance with the C1G2 standard that requires a tag to pick exactly one slot in a frame. However, this assumption does not create any problems because the expected number of slots that a tag chooses in a frame is still one. The analysis with this assumption is, therefore, asymptotically the same as that without this assumption [39]. Bordenave *et* al. further explained in detail why this independence assumption in analyzing Aloha based protocols provides results just as accurate as if all the analysis was done without this assumption [39]. This independence assumption is made only to make the formal development tractable. In all our simulations, a tag chooses exactly one slot at the start of a frame. Table 2.2 lists the symbols used in this paper.

2.5 RFID Tag Search Protocol

2.5.1 Protocol Description

To search which tags in set A are present in the population B, in RTSP, the central controller executes n Aloha frames using the RFID readers. There are five steps involved in executing each frame. First, before executing any frame i, the controller calculates the optimal values of frame size f_i, persistence probability p_i, and generates a random seed number R_i. Second, as the controller knows the IDs in set A, it *pre-computes* which tag in A will choose which slot in the i^{th} frame, *i*.e., it virtually executes the Aloha protocol on set A and obtains the binary array $\mathbb{S}(A, f_i, R_i)$. Thus, the controller knows which bits in the binary array $\mathbb{S}(B, f_i, R_i)$ resulting from executing i^{th} frame on population B should be 1 if all the tags in A were present and a single reader covered the entire population. Third, it provides each reader with the parameters f_i, p_i, and R_i and asks each of them to execute the i^{th} frame using these

Table 2.2 Symbols used in the paper

Symbol	Description
A	set of tag IDs to be searched
B	set of tag IDs in RFID tag population
C	tag IDs in set A that are present in B
$\tilde{C}$	IDs of A returned by RTSP to be present in B
β	required confidence interval
f_i	frame size for round i
f_{op}	optimum value of frame size
T_s	duration of each slot in frame
n	Number of times frames are repeated
n_{op}	optimum value of n
p	persistence probability
R_i	random seed for i^{th} frame
$h(f, R, ID)$	unform hash function in $[1, f]$
P_{fp}	false positive probability
X_{ij}	indicator random variable for j^{th} slot in i^{th} frame to be $1 \leftrightarrow 1$
$\mathcal{N}_i^{11}$	random variable for # of $1 \leftrightarrow 1$ slots in i^{th} frame
S	Total number of execution slots
$E[.]$	Expected value

parameters. The motivation behind using the same values of f_i, p_i, and R_i across all readers for the i^{th} frame is to enable RTSP to work with multiple readers with overlapping regions. As all readers use the same values of f_i, p_i, and R_i in the i^{th} frame, the slot number that a particular tag chooses in the i^{th} frame of each reader covering this tag is the same *i*.e., $h(\frac{f_i}{p_i}, R_i, ID)$ evaluated by the tag results in same value for each reader. Fourth, each reader executes the frame on its turn as per the reader scheduling protocol and sends the responses in the frame back to the controller. Fifth, after the controller has received the i^{th} frame of each reader, it applies logical OR operator on all the received i^{th} frames and obtains the resultant bit array $\mathbb{S}(B, f_i, R_i)$. This resultant bit array $\mathbb{S}(B, f_i, R_i)$ is same as if generated by a single reader covering all the tags. After obtaining n bit arrays $\mathbb{S}(B, f_i, R_i)$ for $1 \leq i \leq n$, for each tag t in A, the controller checks whether $\mathbb{S}(A, f_i, R_i)[h(\frac{f_i}{p_i}, R_i, t)] = \mathbb{S}(A, f_i, R_i)[h(\frac{f_i}{p_i}, R_i, t)]$ for all n frames, *i*.e., for all n frames, the two bits corresponding to tag t in both $\mathbb{S}(A, f_i, R_i)$ and $\mathbb{S}(B, f_i, R_i)$ are 1, then RTSP declares that tag t is present in population B. Note that

RTSP can have false positives, *i*.e., it can declare a tag in set A to be present in population B, when it actually is not. Apparently, based on the design of RTSP we know that RTSP does not have false negatives.

2.5.2 Estimating Number of Tags in Set C

Recall from the previous section that before executing any frame i, the controller calculates the optimal values of frame size f_i and persistence probability p_i. To calculate these optimal values for i^{th} frame, the controller needs estimate of $|C|$ at start of the i^{th} frame, which it obtains using the responses from the tag population in the previous $i-1$ frames. We represent the estimate of $|C|$ at the start of i^{th} frame by $|\tilde{C}_i|$. As the controller executes more and more frames, *i*.e., as i increases, the estimate $|\tilde{C}_i|$ asymptotically becomes equal to $|C|$. Next, we present a method to estimate the value of $|C|$ at start of any frame i.

The intuition behind our estimation method is that as the number of tags in set C increase, the number of pairs of bits in $\mathbb{S}(A, f_i, R_i)$ and $\mathbb{S}(B, f_i, R_i)$ that are 1 also increase. We represent a bit that is 1 in both $\mathbb{S}(A, f_i, R_i)$ and $\mathbb{S}(B, f_i, R_i)$ by $1\leftrightarrow1$. The number of $1\leftrightarrow1$ bits for any given frame is a function of $|C|$ and can, therefore, be used to estimate the value of $|C|$. Next, we derive an expression that relates the number of $1\leftrightarrow1$ bits with the value of $|C|$, *i*.e., we derive an expression for $E[\mathcal{N}_i^{11}]$ as a function of $|C|$, where $\mathcal{N}_i^{11}$ is random variable for number of $1\leftrightarrow1$ bits in pair of bit arrays *i*.e., $\mathbb{S}(A, f_i, R_i)$ and $\mathbb{S}(B, f_i, R_i)$. To derive the expression for $E[\mathcal{N}_i^{11}]$, we need the probability that any given bit in a pair of bit arrays is $1\leftrightarrow1$. We calculate this probability in the following lemma.

Lemma 1. *Let A be the set of IDs of tags that we want to search for in a population. Let B be the set of IDs of tags in the population in which we search for tags in set A. Let C be the set of IDs of those tags that are present in both sets A and B. Let X_{ij} be an indicator random variable for the event that the j^{th} bit in i^{th} bit array pair is a $1\leftrightarrow1$ bit. For frame size f_i and persistence probability p_i, the probability distribution of X_{ij} is given by the following*

equation.

$$P\left\{X_{ij}=1\right\}=1-(1-\frac{p_i}{f_i})^{|A|}-(1-\frac{p_i}{f_i})^{|B|}+(1-\frac{p_i}{f_I})^{|A|+|B|-|C|} \tag{2.1}$$

Proof. Probability that any given bit j in a bit array pair is a 1↔1 bit can be obtained by first calculating the probability that this bit is not a 1↔1 bit, and then subtracting it from 1. The j^{th} bit is not 1↔1 when one of the following three cases happens.

1. None of the tags in set A select the j^{th} slot in pre-computed frame *i*.e., j^{th} bit in $\mathbb{S}(A, f_i, R_i)$ is 0, and none of the tags in population B select the j^{th} slot in corresponding executed frame *i*.e., j^{th} bit in $\mathbb{S}(B, f_i, R_i)$ is 0. We represent this event by an indicator random variable Y_{00}. The probability distribution of Y_{00} is given by the following equations.

$$\begin{aligned} P\left\{Y_{00}=1\right\} &= \left(1-\frac{p}{f}\right)^{|A-C|}\left(1-\frac{p}{f}\right)^{|C|}\left(1-\frac{p}{f}\right)^{|B-C|} \\ &= \left(1-\frac{p}{f}\right)^{|A|+|B|-|C|} \end{aligned} \tag{2.2}$$

2. One or more tags in set $A-C$ select the j^{th} slot in pre-computed frame *i*.e., j^{th} bit in $\mathbb{S}(A, f_i, R_i)$ is 1, and none of the tags in population B select the j^{th} slot in corresponding executed frame *i*.e., j^{th} bit in $\mathbb{S}(B, f_i, R_i)$ is 0. We represent this event by an indicator random variable Y_{10}. The probability distribution of Y_{10} is given by the following equations.

$$\begin{aligned} P\left\{Y_{10}=1\right\} &= \left(1-\left(1-\frac{p}{f}\right)^{|A-C|}\right)\left(1-\frac{p}{f}\right)^{|C|}\left(1-\frac{p}{f}\right)^{|B-C|} \\ &= \left(1-\left(1-\frac{p}{f}\right)^{|A-C|}\right)\left(1-\frac{p}{f}\right)^{|B|} \end{aligned} \tag{2.3}$$

3. None of the tags in set A select the j^{th} slot in pre-computed frame *i*.e., j^{th} bit in $\mathbb{S}(A, f_i, R_i)$ is 0, and one or more tags in population $B-C$ select the j^{th} slot in corresponding executed frame, *i*.e., *i*.e., j^{th} bit in $\mathbb{S}(B, f_i, R_i)$ is 1. We represent this event by an indicator random variable Y_{01}. The probability distribution of Y_{01} is given by the following equations.

$$P\{Y_{01}=1\} = \left(1-\frac{p}{f}\right)^{|A-C|}\left(1-\frac{p}{f}\right)^{|C|}\left(1-\left(1-\frac{p}{f}\right)^{|B-C|}\right)$$
$$= \left(1-\left(1-\frac{p}{f}\right)^{|B-C|}\right)\left(1-\frac{p}{f}\right)^{|A|} \tag{2.4}$$

The probability distribution of X_{ij} is given by the following equation.

$$P\left\{X_{ij}=1\right\} = 1 - P\{Y_{00}=1\} - P\{Y_{10}=1\} - P\{Y_{01}=1\} \tag{2.5}$$

Substituting the expressions for the probability distributions of Y_{00}, Y_{10}, and Y_{01} from Equations (2.2), (2.3), and (2.4), respectively, into Equation (2.5) and simplifying, we get Equation (1). □

Following theorem derives the expression for $E[\mathcal{N}_i^{11}]$ as a function of $|C|$.

Theorem 2. *Let A be the set of IDs of tags that we want to search for in a population. Let B be the set of IDs of tags in the population in which we search for tags in set A. Let C be the set of IDs of those tags that are present in both sets A and B. Let $\mathcal{N}_i^{11}$ be the random variable for the number of $1\leftrightarrow1$ slots in a pair of bit arrays of size f_i each. When persistence probability is p_i, the expected value of $\mathcal{N}_i^{11}$ is given by the following equation.*

$$E[\mathcal{N}_i^{11}] = f_i \times \left(1-(1-\frac{p_i}{f_i})^{|A|}-(1-\frac{p_i}{f_i})^{|B|} + (1-\frac{p_i}{f_i})^{|A|+|B|-|C|}\right) \tag{2.6}$$

Proof. It is straight forward to see that $\mathcal{N}_i^{11} = \sum_{j=1}^{f_i} X_{ij}$. As $\left\{X_{i1}, X_{i2}, \ldots, X_{if_i)}\right\}$ forms a set of identically distributed random variables, $E[\mathcal{N}_i^{11}]$ is given by

$$E[\mathcal{N}_i^{11}] = E[\sum_{j=1}^{f_i} X_{ij}] = f_i \times E[X_{ij}]$$

As expected value of an indicator random variable equals its probability of being 1, $E[X_{ij}] = P\left\{X_{ij}=1\right\}$. Substituting the value of $E[X_{ij}]$ in the equation above with the value of $P\left\{X_{ij}=1\right\}$ from Equation (2.5), we get the equation for $E[\mathcal{N}_i^{11}]$ in theorem statement. □

Figure 2.1 plots $E[\mathcal{N}_i^{11}]$ as a function of $|C|$ using Equation (2.6). This figure is obtained using $|A| = 200$, $|B| = 300$, $f_i = 300$ and $p_i = 1$. We observe from this figure that $E[\mathcal{N}_i^{11}]$ is a monotonically increasing function of $|C|$. To estimate the value of $|C|$, let $\tilde{\mathcal{N}}_i^{11}$ represent the observed value of number of $1 \leftrightarrow 1$ bits for i^{th} pair of bit arrays. Replacing $E[\mathcal{N}_i^{11}]$ in Equation (2.6) with $\tilde{\mathcal{N}}_i^{11}$ and solving for $|C|$ gives an estimate of $|C|$. This estimate is obtained by utilizing the information from the i^{th} frame only. While this estimate may not be accurate, if we use the information from more frames, the estimate will become more accurate. Specifically, we leverage the well known statistical result that the variance in the observed value of a random variable reduces by x times if we take the average of x observations of that random variable. Therefore, to obtain the estimate $|\tilde{C}_i|$ of $|C|$ at the start of the i^{th} frame, we obtain an estimate from each of the previous $i-1$ frames and take their average. Solving Equation (2.6) for $|C|$ and averaging over past $i-1$ frames, the formal expression for $|\tilde{C}_i|$ becomes

$$|\tilde{C}_i| \approx |A| + |B| + \frac{\sum_{l=1}^{i-1} \frac{f_l}{p_l} \ln \left\{ \frac{E[\mathcal{N}_l^{11}]}{f_l} - 1 + e^{-\frac{p_l}{f_l}A} + e^{-\frac{p_l}{f_l}B} \right\}}{i-1} \tag{2.7}$$

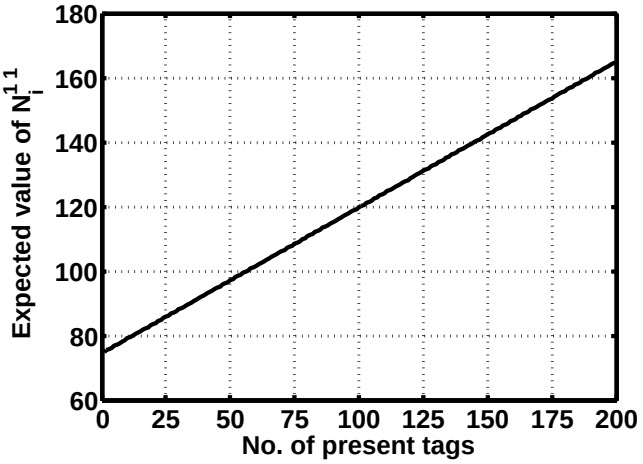


Figure 2.1 Expected value of number of $1 \leftrightarrow 1$ slots vs. $|C|$

Finally, note that the controller obtains this estimate without executing any additional frames. It gets this estimate from the frames it was already executing to search for tags.

2.6 Parameter Optimization

In this section, we will derive equations that the controller uses at the start of i^{th} frame to calculate the optimal values of frame size f_i and persistence probability p_i to minimize the execution time of RTSP while ensuring that its actual confidence interval is less than the required confidence interval. At the start of i^{th} frame, the controller uses the estimate $|\tilde{C}_i|$ along with the values of $|A|$, $|B|$, and β to calculate the optimal values of f_i and p_i. Before asking the readers to execute the i^{th} frame, the controller also calculates the maximum number of frames that it should execute, represented by n_i. Recall from Section 2.5.2 that as the number of executed frames increase, the estimate of $|C|$ becomes more accurate. Consequently, n_i, f_i, and p_i asymptotically become equal to constants n, f, and p, respectively. When the estimate of $|C|$ changes by less than 2 in 10 consecutive frames, the controller considers the estimate to be close enough to $|C|$.At this point, the controller calculates the values of n_i, f_i, and p_i one last time and puts $f = f_i$, $p = p_i$, and $n = n_i$, and uses these fixed values of f and p to execute subsequent frames until the total number of frames executed since the first frame become equal to n. For the first frame, *i*.e., when $i = 1$, the controller uses $n_1 = \infty$, $f_1 = \max\{|A|, |B|\}$, and $p_1 = 1$. The choices of the values of n_1, f_1, and p_1 are arbitrary and do not really matter because as the controller executes more frames, number of frames, frame size, and persistence probability converge to constants n, f, and p, respectively.

In subsequent calculation of n_i, f_i, and p_i, we will drop the subscript i to make the presentation simple. Next, we first derive the expression for false positive probability *i*.e., probability with which RTSP declares a tag in set A to be present in population B, when it actually is not. Second, using the expression for false positive probability, we derive a *confidence condition* that the values of n, f, and p must satisfy to ensure that the observed confidence interval is smaller than the required confidence interval β, *i*.e., the requirement $|\tilde{C}| - |C| \leq \beta|C|$ is satisfied. Third, we derive a *duration condition*, which the values of f and p must satisfy to ensure that the execution time of RTSP is minimized. The controller

solves these two conditions simultaneously to obtain the optimal values of n, f, and p. Last , we will describe our strategy to bring the value of f within limit when the optimal value of the frame size exceeds the C1G2 specified upper limit of 2^{15}.

2.6.1 False Positive Probability

A false positive occurs when all the bits that a particular tag in A that is not present in B selects in the n bit arrays $\mathbb{S}(A, f_i, R_i)$ turn out to be nonempty in $\mathbb{S}(B, f_i, R_i)$ because some other tags in the population made those bits 1. Lemma 3 gives the expression to calculate the false positive probability.

Lemma 3. *Let B be the set of IDs of tags in the population in which we search for tags. With persistence probability p, frame size f, and number of frames n, the false positive probability, P_{fp}, is given by the following equation.*

$$P_{fp} = \left[1 - \left(1 - \frac{p}{f}\right)^{|B|}\right]^n \tag{2.8}$$

Proof. Consider a tag t such that $t \in A \wedge t \notin B$. The probability that the slot tag t selects in the i^{th} pre-computed frame *i*.e., the bit it selects in $\mathbb{S}(A, f_i, R_i)$ is selected by at least one tag in population B in $\mathbb{S}(B, f_i, R_i)$ is $1 - (1 - \frac{p}{f})^{|B|}$. The probability that all n bits tag t selects in the n bit arrays $\mathbb{S}(A, f_i, R_i)$ are also selected by some other tags in population B in corresponding bit arrays $\mathbb{S}(B, f_i, R_i)$ is $[1 - (1 - \frac{p}{f})^{|B|}]^n$, which is the expression for false positive probability given in Equation (2.8). □

Figure 2.2 shows the theoretically calculated false positive probability from Equation (2.8) represented by the solid line and experimentally observed values of false positive probability represented by the dots. To obtain this figure, we use $f = 600$, $p = 1$, and $n = 10$. Each dot represents the false positive probability calculated from 200 runs of simulation. We observe that the theoretically calculated values match perfectly with experimentally observed values, showing that our independence assumption that we stated in Section 2.4.4 does not cause the theoretical analysis to deviate from practically observed values.

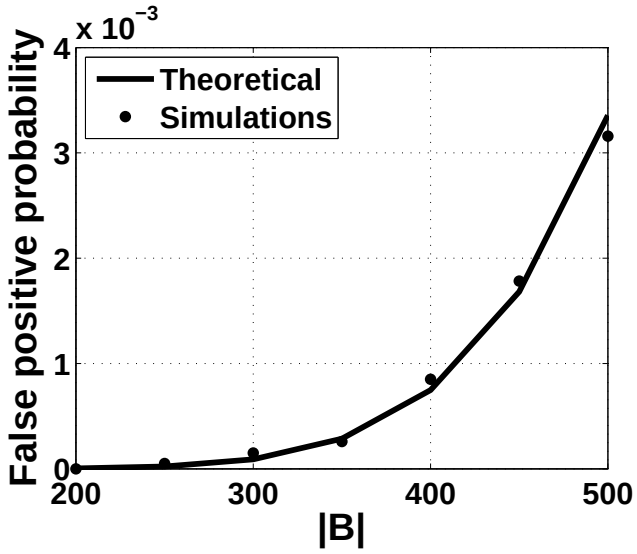


Figure 2.2 Comparison of theoretical and experimental P_{fp}

2.6.2 Confidence Condition

Theorem 4 states the confidence condition that the values of n, f, and p must satisfy to achieve the required confidence interval β.

Theorem 4. *Let A be the set of IDs of tags that we want to search for in a population. Let B be the set of IDs of tags in the population in which we search for tags in set A. Let C be the set of IDs of those tags that are present in both sets A and B. To ensure that RTSP satisfies the requirement $|\tilde{C}| - |C| \leq \beta|C|$, the controller must use the values for number of frames n, frame size f, and persistence probability p that satisfy the confidence condition given in the following equation.*

$$n = \frac{\ln\left(\frac{\beta \times |\tilde{C}|}{|A| - |\tilde{C}|}\right)}{\ln\left(1 - (1 - \frac{p}{f})^{|B|}\right)} \tag{2.9}$$

Proof. Let $E[|\tilde{C}|]$ represent the number of tags that RTSP declares as belonging to set C after executing n frames of size f with persistence probability p. Replacing $|\tilde{C}|$ in $|\tilde{C}| - |C| \leq \beta|C|$ by $E[|\tilde{C}|]$, the reliability requirement is given in the following equation.

$$E[|\tilde{C}|] - |C| \leq \beta|C| \tag{2.10}$$

Next, we derive the expression for $E[|\tilde{C}|]$. Recall from Section 2.5.1 that RTSP can have false positives, but it cannot have false negatives *i*.e., it will always identify the tags of A present in B and in addition, it may also declare some tags in A that are not in B to be present in B. Thus, $E[|\tilde{C}|] = |C| + (|A - C|) \times P_{fp}$. As $C \subseteq A$, thus, $E[|\tilde{C}|] = |C| + (|A| - |C|) \times P_{fp}$ Substituting this value of $E[|\tilde{C}|]$ into Equation (2.10), we get the following equation.

$$|C| + (|A| - |C|) \times P_{fp} - |C| \leq \beta|C|$$

Substituting the value of P_{fp} from Equation (2.8) into equation above and rearranging, we get

$$n \leq \frac{\ln\left(\frac{\beta \times |C|}{|A| - |C|}\right)}{\ln\left(1 - (1 - \frac{p}{f})^{|B|}\right)}$$

As we do not know the exact value of $|C|$, rather we know the estimate $|\tilde{C}|$ of $|C|$, replacing $|C|$ in this equation with $|\tilde{C}|$ and using the largest value for n to ensure that confidence requirement is always met, we get Equation (2.9) in theorem statement. □

2.6.3 Duration Condition

Theorem 5 states the duration condition that the values of f and p must satisfy to minimize the execution time of RTSP.

Theorem 5. *Let A be the set of IDs of tags that we want to search for in a population. Let B be the set of IDs of tags in the population in which we search for tags in set A. Let C be the set of IDs of those tags that are present in both sets A and B. To ensure that the execution time of RTSP is minimum, the controller must use the values for frame size f and persistence probability p that satisfy the duration condition given in the following equation.*

$$p \times |B| = f \times \left(1 - e^{\frac{p}{f}|B|}\right) \times \ln\left\{1 - e^{-\frac{p}{f}|B|}\right\} \tag{2.11}$$

Proof. Execution time is directly proportional to the total number of slots because the duration of each slot is the same, typically $300\mu s$ for Philips I-Code RFID reader [171]. Let

S represent the total number of slots. Thus, $S = f \times n$. To ensure that RTSP achieves the required confidence interval, we use the value of n from Equation (2.9). Thus,

$$S = \frac{f \ln\left(\frac{\beta \times |\tilde{C}|}{|A| - |\tilde{C}|}\right)}{\ln\left(1 - (1 - \frac{p}{f})^{|B|}\right)} \tag{2.12}$$

Figure 2.3 plots S as a function of f using Equation (2.12). This figure is made using $|A| = 100$, $|B| = 100$, $|\tilde{C}| = 52$, $p = 1$, and $\beta = 0.05$. We observe from this figure that S is a convex function of f. Therefore, optimum value of f exists, represented by f_{op}, that minimizes the total number of slots S. To find optimal value of f, we differentiate Equation (2.12) with respect to f and equate the resulting expression to 0, which results in the following expression.

$$\left[\ln\left(\frac{\beta \times |\tilde{C}|}{|A| - |\tilde{C}|}\right)\right]\left[p|B| - f\left(1 - e^{\frac{p}{f}|B|}\right)\ln\left\{1 - e^{-\frac{p}{f}|B|}\right\}\right] = 0$$

Note that $\ln\left(\frac{\beta \times |\tilde{C}|}{|A| - |\tilde{C}|}\right) \neq 0$, which means that

$$p|B| - f\left(1 - e^{\frac{p}{f}|B|}\right)\ln\left\{1 - e^{-\frac{p}{f}|B|}\right\} = 0$$

Rearranging the equation above, we get the duration condition in the theorem statement.

□

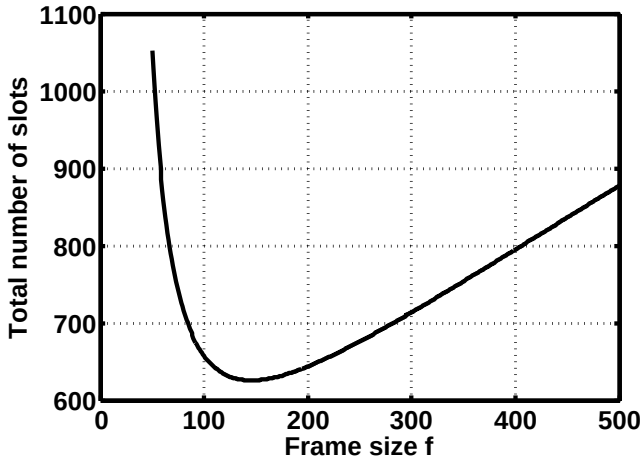


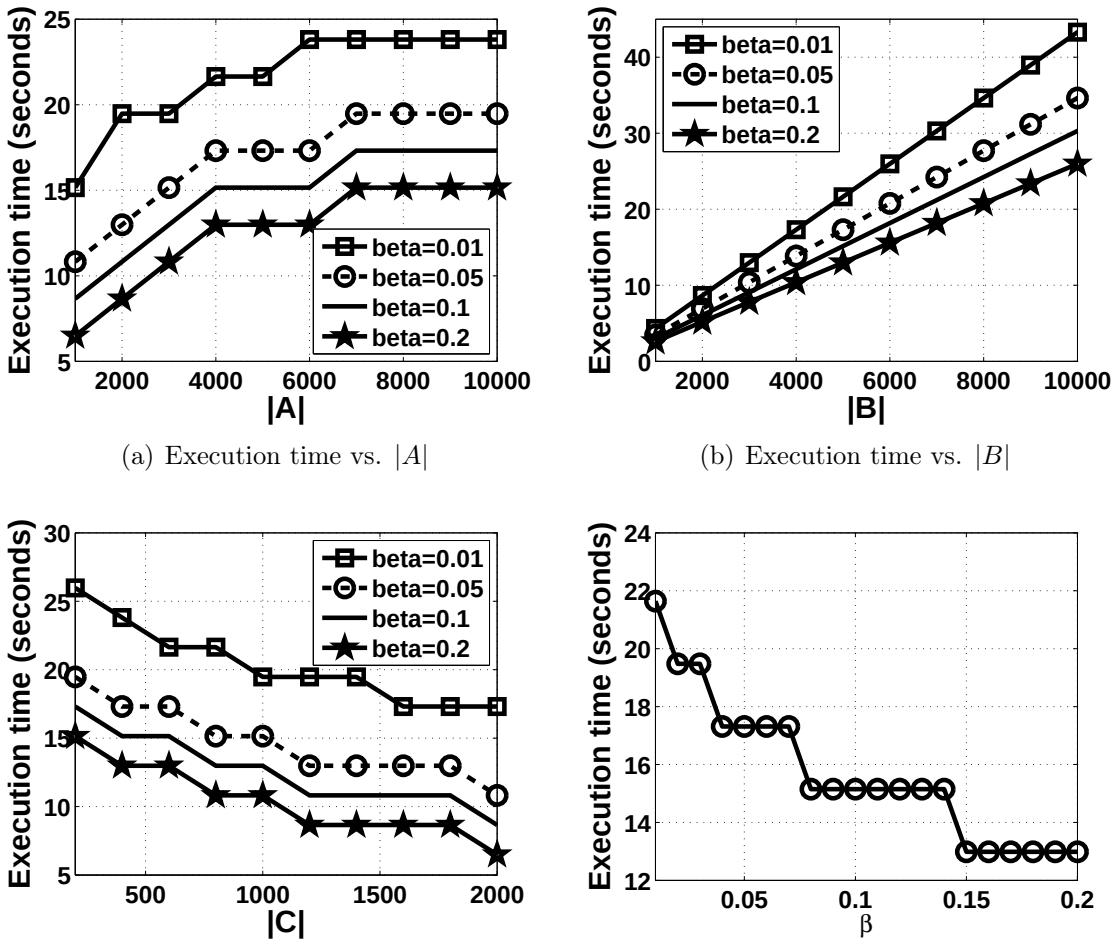
Figure 2.3 Total number of slots S vs. frame size f

The controller solves Equations (2.9) and (2.11) simultaneously using $p = 1$ and gets the optimal values of n and f represented by n_{op} and f_{op}, respectively. Next, we study the effect of $|A|$, $|B|$, $|C|$, and β on execution time of RTSP.

Execution Time vs. $|A|$*:* Intuitively, as the number of tags in A increases, the execution time of RTSP should increase because greater number of tags in A imply higher chances of false positives. Thus, to ensure that the number of false positives stay small enough so that the required confidence interval is achieved, RTSP executes more frames, *i*.e., the value of n_{op} increases, which increases the overall execution time. Figure 2.4(a) confirms our intuition. This figure plots the expected execution time of RTSP for multiple values of $|A|$ while fixing $|B|$ at $X5000$ and $|C|$ at 500. We calculated the execution time as $n_{\text{op}} \times f_{\text{op}} \times T_s$, where T_s is the time of each slot and is equal to $300\mu s$ as per the specifications of Philips I-Code RFID reader [171]. We observe from Figure 2.4(a) that as the number of tags in A increase, the execution time of RTSP increases.

Execution Time vs. $|B|$*:* Intuitively, as the number of tags in B increases, the execution time of RTSP should increase because greater number of tags in B also imply higher chances of false positives. Thus, to ensure that the number of false positives stay small enough so that the required confidence interval is achieved, RTSP increases the frame size, *i*.e., the value of f_{op} increases according to Equation (2.11), which increases the overall execution time. Figure 2.4(b) confirms our intuition. This figure plots the expected execution time of RTSP for multiple values of $|B|$ while fixing $|A|$ at 5000 and $|C|$ at 500. We observe from Figure 2.4(b) that as the number of tags in B increase, the execution time of RTSP increases.

Execution Time vs. $|C|$*:* Intuitively, as the number of tags in C increase, the execution time of RTSP should decrease because greater number of tags in C means RTSP has greater margin of error *i*.e., $\beta|C|$. Thus, RTSP reduces the value of n_{op}, which decreases the overall execution time. Figure 2.4(c) confirms our intuition. This figure plots the expected execution time of RTSP for multiple values of $|C|$ while fixing $|A|$ at 5000 and $|B|$ at 5000. We observe from Figure 2.4(c) that as the number of tags in C increase, the execution time of RTSP



(a) Execution time vs. $|A|$

(b) Execution time vs. $|B|$

(c) Execution time vs. $|C|$

(d) Execution time vs. β

Figure 2.4 Expected execution times of RTSP

decreases.

Execution Time vs. β: Intuitively, as the required confidence interval β increases, the execution time of RTSP should decrease because larger required confidence interval means RTSP has greater margin of error. Thus, RTSP reduces the values of n_{op}, which decreases the overall execution time. Figure 2.4(d) confirms our intuition. This figure plots the expected execution time of RTSP for different values of β while fixing $|A|$ at 5000, $|B|$ at 5000, and $|C|$ at 500. We observe from Figure 2.4(d) that as the required confidence interval increases, the execution time of RTSP decreases.

2.6.4 Handling Large Frame Sizes

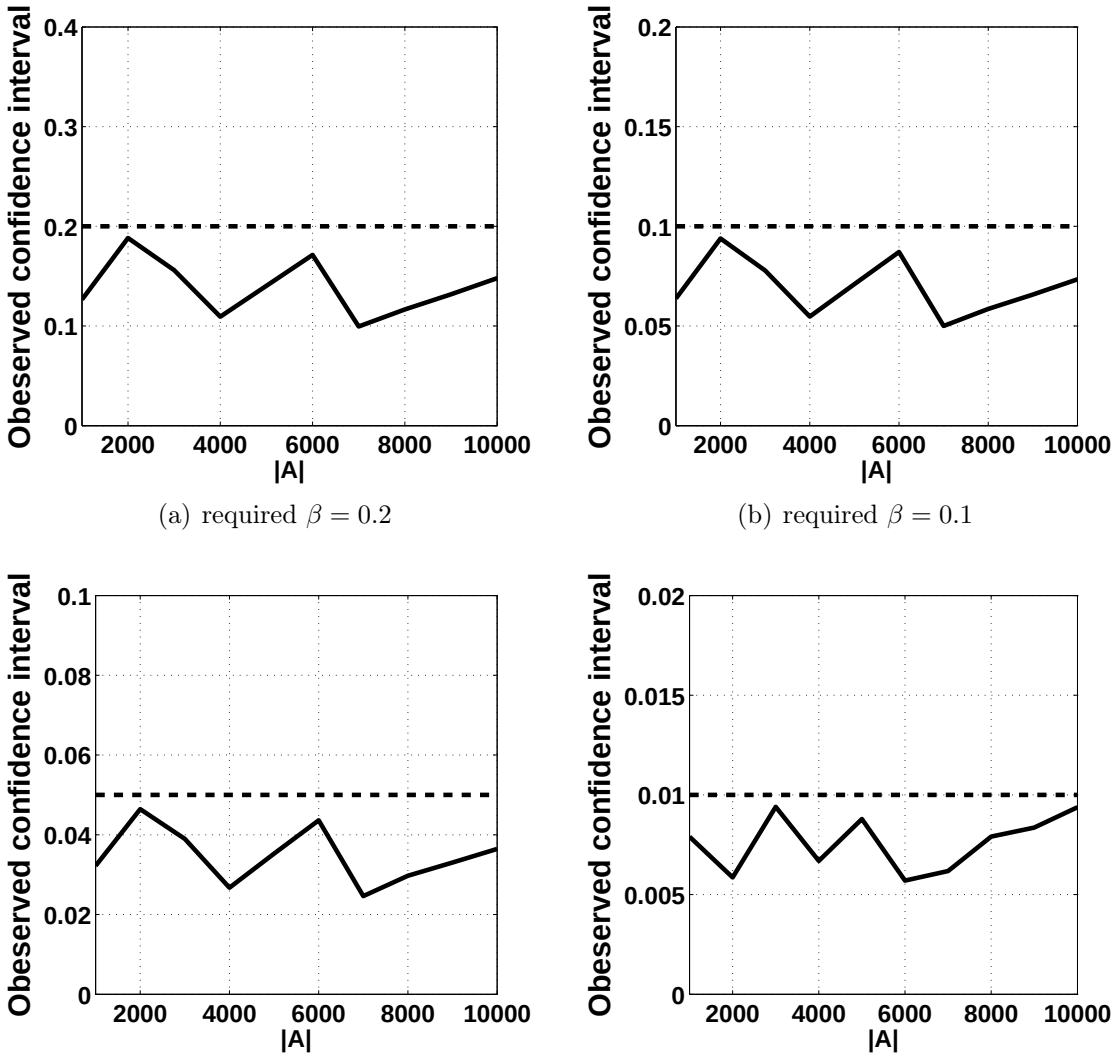
For large populations and/or small required confidence interval, it is possible for the value of f_{op} to exceed the C1G2 specified upper limit of 2^{15}. Next, we describe how we use p to bring the frame size within limits. Bringing the frame size within limits comes at a cost of increased number of slots; greater than the minimum value of S that would have been achieved if the controller could use $f_{\text{op}} > 2^{15}$.

When we decrease the value of p, the number of tags that participate in a frame decreases. Therefore, the required value of f also decreases. Participation by fewer tags means that participation by the tags belonging to both the sets A and B decreases. This increases the chances that a given tag in A that is present in B will not select any slot in a given pre-computed frame, which means that chances of identifying its presence decrease. Therefore, the overall uncertainty in identifying tags in A increases. To reduce this uncertainty, the value of n increases when p decreases to achieve the required confidence interval.

We use these two observations to reduce the value of f whenever $f_{\text{op}} > 2^{15}$. When $f_{\text{op}} > 2^{15}$, the controller uses $f = f_{\max} = 2^{15}$ in Equation (2.9), which leaves two unknowns, p and n, in the resulting equation. The controller solves the resulting equation simultaneously with Equation (2.11) to get new values of p and n. The new value of p is less than 1 and the new value of n is greater than n_{op} (we represent n with n_{op} when we use $f = f_{\text{op}}$ to calculate it). The controller uses these new values of n and p along with $f = f_{\max}$ to pre-compute the bit array $\mathbb{S}(A, f_i, R_i)$. Although the total number of slots $S = f_{\max} \times n > f_{\text{op}} \times n_{\text{op}}$, this is still the smallest under the constraints that the required confidence interval is achieved and the frame size does not exceed $f_{\max}$.

2.7 Performance Evaluation

We simulated and implemented RTSP in Matlab. We also implemented the fastest existing tag identification protocol, TH [176], to compare the execution time of RTSP with it. We choose tag ID length of 64 bits as specified in the C1G2 standard. Note that the distributions



(a) required $\beta = 0.2$

(b) required $\beta = 0.1$

(c) required $\beta = 0.05$

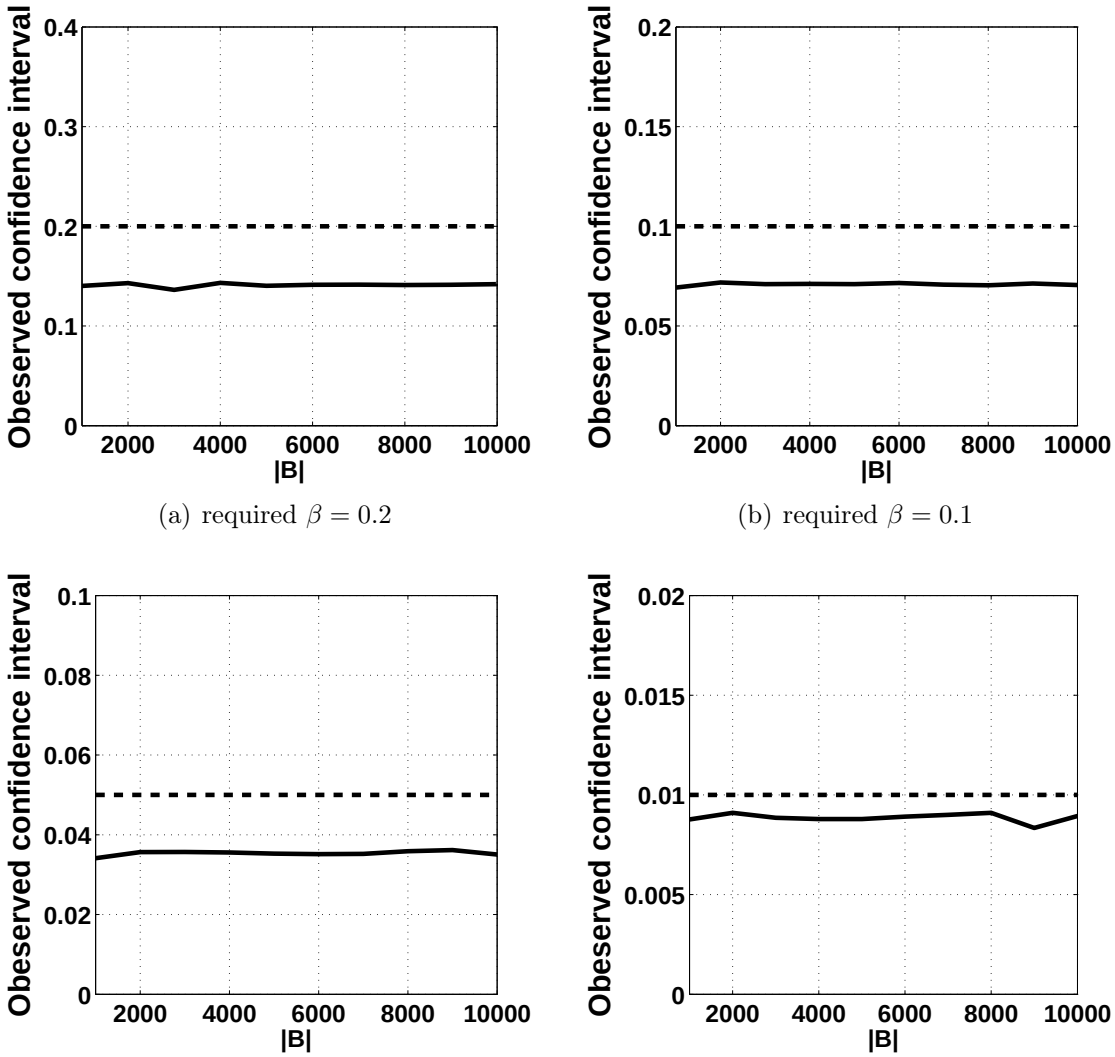
(d) required $\beta = 0.01$

Figure 2.5 Observed confidence interval vs. $|A|$ when $|B| = 5000$, and $|C| = 500$

of the IDs of tags in A and B do not matter because RTSP is independent of ID distributions. Next, we first evaluate the accuracy of RTSP and then compare its execution time with the execution time of TH. All results reported in this section are obtained from averaging over 200 independent runs of RTSP.

2.7.1 Accuracy

To evaluate the accuracy of RTSP, we study whether it achieves the required confidence interval for different values of $|A|$, $|B|$, and $|C|$. If we recall the how we calculate the



(a) required $\beta = 0.2$

(b) required $\beta = 0.1$

(c) required $\beta = 0.05$

(d) required $\beta = 0.01$

Figure 2.6 Observed confidence interval vs. $|B|$ when $|A| = 5000$, and $|C| = 500$

parameters we will know that RTSP can ensure the confidence interval with any combinations of $|A|$, $|B|$, and $|C|$.

2.7.1.1 Observed Confidence interval vs. $|A|$

Our experimental results show that RTSP always achieves the required confidence interval regardless of the size of set A. Figures 2.5(a), 2.5(b), 2.5(c), and 2.5(d) plot the actual confidence interval RTSP achieved for different sizes of set A when the required values of confidence interval are $\beta = 0.2$, $\beta = 0.1$, $\beta = 0.05$, $\beta = 0.01$, respectively. To plot these

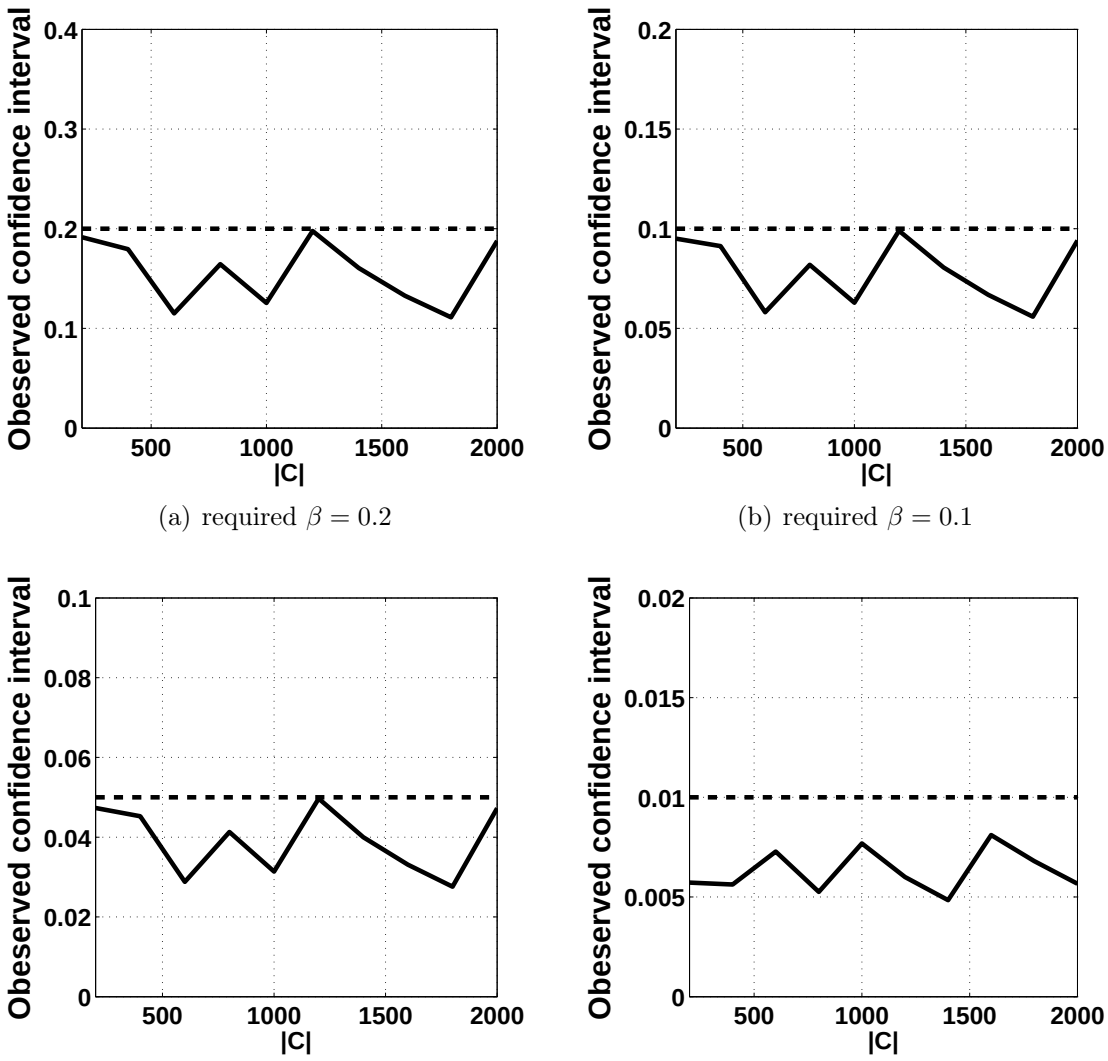
figures, we fixed number of tags in set B at 5000 and number of tags in A that are in B, *i*.e., number of tags in set C at 500. The dashed horizontal line in each of these figures shows the required value of confidence interval and the solid line shows the observed values of confidence interval achieved by RTSP. We observe from these figures that the observed values of confidence interval are always smaller than the required values of confidence interval.

2.7.1.2 Observed Confidence interval vs. $|B|$

Our experimental results show that RTSP always achieves the required confidence interval regardless of the number of tags in population B. Figures 2.6(a), 2.6(b), 2.6(c), and 2.6(d) plot the actual confidence interval RTSP achieved for different sizes of set B when the required values of confidence interval are $\beta = 0.2$, $\beta = 0.1$, $\beta = 0.05$, $\beta = 0.01$, respectively. To plot these figures, we fixed number of tags in set A at 5000 and number of tags in set C at 500. We observe from these figures that the solid lines are always below their corresponding dashed lines, which means that the actual values of confidence interval are always smaller than the required values of confidence interval.

2.7.1.3 Observed Confidence interval vs. $|C|$

Our experimental results show that RTSP always achieves the required confidence interval regardless of the number of tags in set C. Figures 2.7(a), 2.7(b), 2.7(c), and 2.7(d) plot the actual confidence interval RTSP achieved for different sizes of set C when the required values of confidence interval are $\beta = 0.2$, $\beta = 0.1$, $\beta = 0.05$, $\beta = 0.01$, respectively. To plot these figures, we fixed number of tags in sets A and B at 5000 each. Again, we observe from these figures that the solid lines are always below their corresponding dashed lines, which means that RTSP always achieves the required confidence interval.



(a) required $\beta = 0.2$

(b) required $\beta = 0.1$

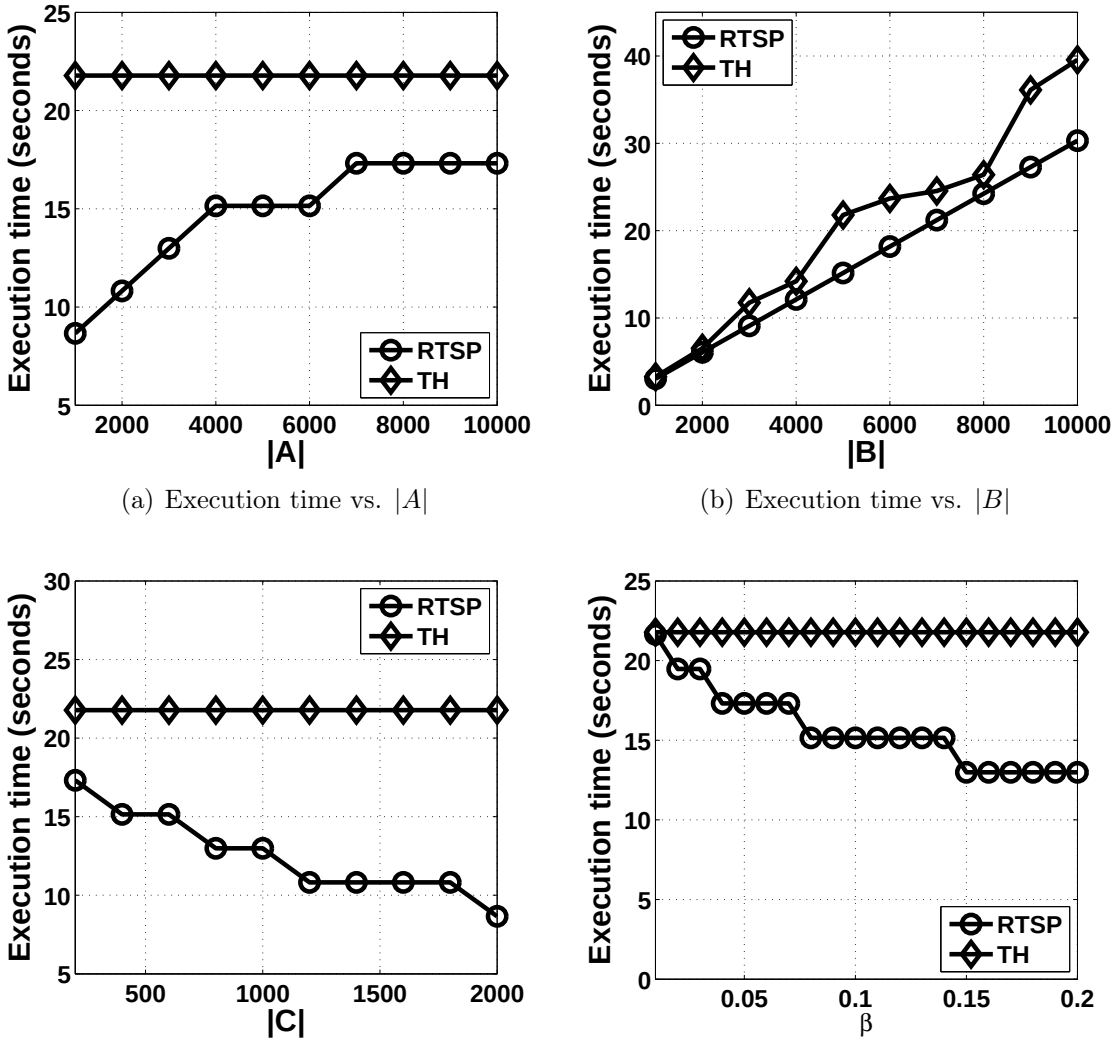
(c) required $\beta = 0.05$

(d) required $\beta = 0.01$

Figure 2.7 Observed confidence interval vs. $|C|$ when $|A| = 5000$, and $|B| = 5000$

2.7.2 Execution Time

Execution time of RTSP is smaller than TH. Fig. 2.8(a) plots the execution times of TH and RTSP vs. $|A|$ for $\beta = 0.1$, $|B| = 3000$, and $C = 500$. We observe from this figure that RTSP is up to 77.27% faster compared to TH. Similarly, Fig. 2.8(b) plots the execution times vs. $|B|$ for $\beta = 0.1$, $|A| = 1000$, and $|C| = 500$ and Fig. 2.8(c) plots the execution times vs. $|C|$ for $\beta = 0.1$, $|A| = 5000$, and $|B| = 5000$. Again, we observe from these figures that RTSP is always faster compared to TH. Finally, Fig. 2.8(d) plots the execution times



(a) Execution time vs. $|A|$

(b) Execution time vs. $|B|$

(c) Execution time vs. $|C|$

(d) Execution time vs. β

Figure 2.8 Comparison of execution times of RTSP and TH

vs. β for $|A| = 5000$, $|B| = 5000$, and $|C| = 500$. We observe from this figure that RTSP is faster compared to TH as long as required confidence interval is greater than 0.01. When the required confidence interval is less than 0.01, TH is faster. Thus, if privacy is not a concern, a user should use TH to search for tags whenever $\beta < 0.01$. If, however, privacy is a concern, then the user should use RTSP regardless of the value of β.

2.8 Future Work

In this section, I provide an overview of the planned future work.

2.8.1 Valued Tag Detection and Identification

As we shown before, RTSP can be used to detect and identify some tags we are caring about. However, when we talk about the missing tag detection we are not mean all the missing tags which maybe very cheap. For example, given two items A and B attached with tags(we will just name as tag A and B), value of item A is $1k$ and item B is just one dollar. Apparently, those two tag should be detected with different accuracy and confidence. We want to design a protocol which can detect and identify tag A with 99.99% accuracy comparing with tag B with 90% accuracy. However, RFID reader have no idea about value(the value can be anything else not just measure by money) of the each tag which means we may need to design a method to help reader to understand the value of each tags. I am working on this problem and one idea is to use one hot coding to categories all the tags into different categories according to their values.

CHAPTER 3

RFID MULTI-CATEGORY TAGS ESTIMATION

3.1 Introduction

In this section I will briefly introduce the background and problem statement of RFID multi-category problem. The design of our scheme will also mention in this section.

3.1.1 Background and Problem Statement

Radio Frequency Identification (RFID) has been widely used in many applications such as inventory management [98, 122, 109, 55, 113, 110, 53], object tracking [212, 225, 114], and localization [209, 210, 180]. A typical RFID system consists of readers, tags, and a back-end server. The back-end server controls the reader to interrogate a set of tags, and the tags respond with their IDs over a shared wireless medium. A tag is a microchip with an antenna in a compact package that has limited computing power and communication ranges. The reader communicates with the tags via wireless channel to identify or monitor the tagged objects. There are two types of RFID tags: passive tags, which do not have their own power sources and are powered up by harvesting the radio frequency energy from readers, and active tags, which have their own power sources. In an RFID-enabled warehouse, there may be thousands of tagged items that belong to different categories, *e.g.*, different places of origin or different brands [207]. Each tag attached to an item has a unique ID that consists of two fields: a category ID that specifies the category of the attached object, and a member ID that identifies this object within its category. As a manager of the warehouse, one may desire to timely monitor the product stock of each category. If the stock of a category is too high, it may indicate that this category of products are not popular, and the manager needs to adjust the marketing strategy (*e.g.*, lowering prices to increase sales). On the contrary, if the stock of a category is too low, the manager should perform stock replenishment as

soon as possible. Manual checking is laborious and of low time-efficiency. You can imagine how difficult it is for a manager to manually count the number of items in each category that may be stacked together or placed on high shelves. Hence, it is desirable to exploit the RFID technique to quickly obtain the number of tagged items in each category. We concerns the practically important problem of multi-category RFID estimation: given a set of RFID tags, we want to quickly and accurately estimate the number of tags in each category. However, almost all the existing RFID estimation protocols are dedicated to the estimation problem on a single set, regardless of tag categories. A feasible solution is to separately execute the existing estimation protocols on each category. The execution time of such a serial solution is proportional to the number of categories, and cannot satisfy the delay-stringent application scenarios. Simultaneous RFID estimation over multiple categories is desirable, hence, this paper proposes an approach called Simultaneous Estimation for Multi-category RFID systems (SEM). SEM exploits the Manchester-coding mechanism, which is supported by the ISO 18000-6 RFID standard, to decode the combined signals, thereby simultaneously obtaining the reply status of tags from each category. As a result, multiple bit vectors are decoded from just one physical slotted frame. Built on our SEM, many existing excellent estimation protocols can be used to estimate the tag cardinality of each category in a simultaneous manner. To ensure the predefined accuracy, we calculate the variance of the estimate in one round, as well as the variance of the average estimate in multiple rounds. To find the optimal frame size, we propose an efficient binary search-based algorithm. To address significant variance in category sizes, we propose an Adaptive Partitioning (AP) strategy to group categories of similar sizes together and execute the estimation protocol for each group separately. Compared with the existing protocols, our approach is much faster, meanwhile satisfying the predefined estimation accuracy. For example, with 20 categories, the proposed SEM+AP is about 7 times faster than prior estimation schemes. Moreover, our approach is the only one whose normalized estimation time (*i*.e., time per category) decreases as the number of categories increases. Radio Frequency Identification (RFID) has been widely used

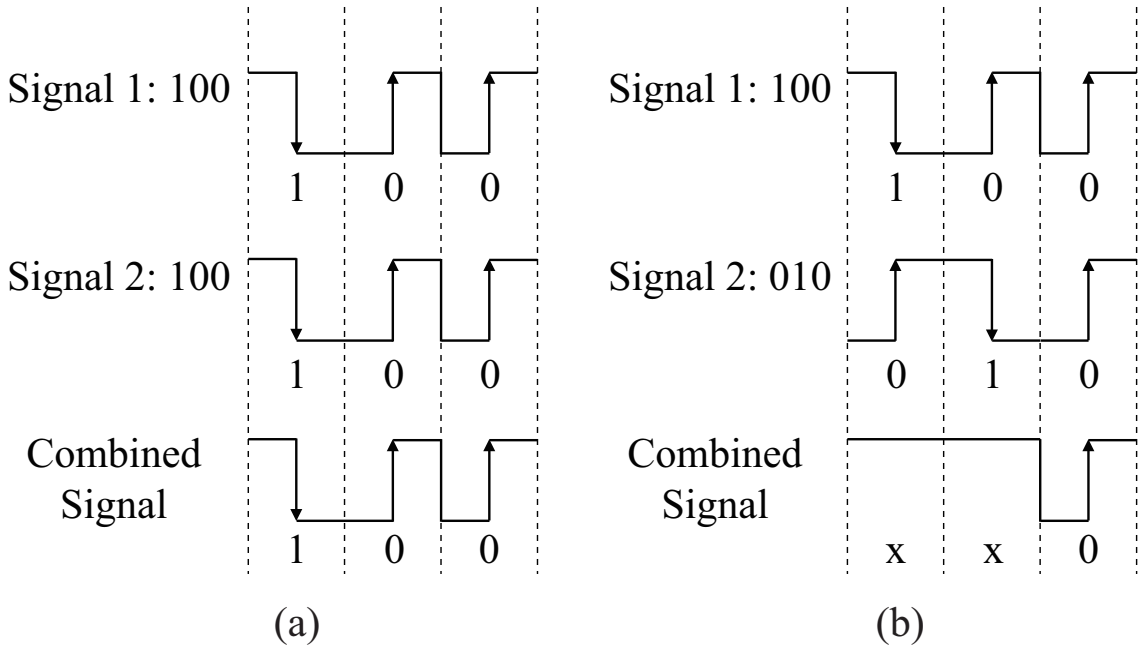


Figure 3.1 Single-one Manchester Coding

in many applications such as inventory management [98, 122, 109, 55, 53], object tracking [222, 212, 225], and localization [209, 210, 180].

This paper formulates and addresses the practical problem of *multi-category RFID estimation*. Given a set of RFID tags with λ categories denoted by $C_1, C_2, \cdots, C_\lambda$, whose cardinalities are denoted by $n_1, n_2, \cdots, n_\lambda$, respectively, a confidence interval $\alpha \in (0, 1]$, and a required reliability $\beta \in [0, 1)$, we want to estimate the number of tags in each category using one or more readers such that for each $1 \leq i \leq \lambda$, we have $P\{|\hat{n_i} - n_i| \leq n_i\alpha\} \geq \beta$, where $\hat{n_i}$ is the estimate of n_i.

3.1.2 Proposed Approach

In this paper, we propose an approach called Simultaneous Estimation for Multi-category RFID systems (SEM). At the start of SEM, we inject a so-called single-one string ($\mathbb{SO}$ string for short) into each tag. Given λ categories, the $\mathbb{SO}$ string injected into the tag belonging to the i-th category is a vector of λ bits where exactly the i-th bit is 1 and all other bits are 0s. For example, given 3 categories, the $\mathbb{SO}$ strings are 100, 010, and 001, respectively.

100 is injected into the tags of the first category; 010 is injected into the tags of the second category; 001 is injected into the tags of the third category. Such a string injecting operation can be easily implemented as follows. The reader uses `SELECT` command [59] to activate the tags in a specific category while keeping the other tags inactive. Then, the reader broadcasts the corresponding $\mathbb{SO}$ string, and the active tags record the received string in their memories. The RFID tags respond to the reader's query with the $\mathbb{SO}$ strings that are modulated by Manchester coding mechanism. When querying two tags, which are in the i-th category and the j-th category, respectively, if $i = j$, then the reader obtains a vector of λ bits where exactly the i-th bit is 1 and all other $\lambda - 1$ bits are 0s; if $i \neq j$, then the reader obtains a vector of λ bits where exactly the i-th bit and the j-th bit are collisions and all other $\lambda - 2$ bits are 0s. Specifically, as illustrated in Figure 3.1, 1 is encoded as a falling edge and 0 is encoded as a rising edge in the Manchester coding. If all tags transmit 0 (or 1) at the same time, the reader can successfully recover the bit as 0 (or 1); otherwise, the reader will detect a bit collision x. Thus, from the bit vector that the reader obtains, we know exactly which categories of tags responded in this slot. Note that Manchester coding is supported by the RFID standard ISO 18000-6 [22] for detecting bit-level collisions [56, 92]. Many excellent literature [225, 91] makes use of the bit-level synchronization to address RFID application problems.

SEM is based on the standard Framed Slotted Aloha protocol [97] for MAC layer communication. First, the RFID reader initializes a slotted time frame by broadcasting a binary request $\langle \delta, f \rangle$, where δ is a random seed and f is the frame size (*i*.e., the number of slots in the forthcoming frame). Each tag randomly chooses a slot in the frame to reply its $\mathbb{SO}$ string. Specifically, each tag initializes its slot counter $sc = H(ID, \delta) \mod f$, which follows a uniform distribution within $[0, f - 1]$. The reader broadcasts the `QueryRep` command at the end of each slot to inform every tag to decrement its slot counter sc by 1. In each slot, a tag responds to the reader once its slot counter sc becomes 0. At the end of each frame, the reader obtains an array of f ternary strings where each ternary string has λ bits and each bit

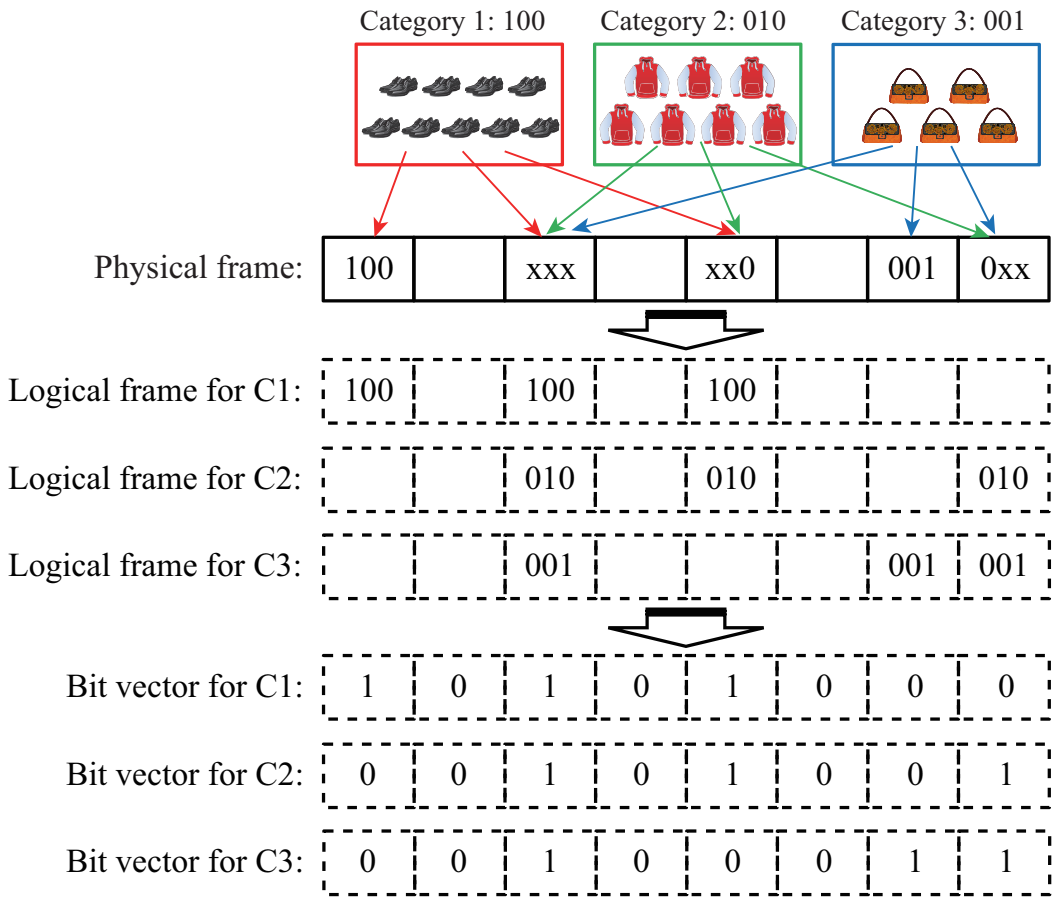


Figure 3.2 From one physical frame to $\lambda = 3$ logical frames

has a value of 0, 1, or x. We call this array a *physical frame*. For the λ-bit ternary string t_i of the i-th slot, for each $1 \leq j \leq \lambda$, if $t_i[j] = 0$, then there is no tag in category C_j that responded in the i-th slot; if $t_i[j] = 1$, then only tags in category C_j responded in the i-th slot; if $t_i[j] = x$, then more than one tag responded in the i-th slot: at least one in category C_j and the remaining not in category C_j. Thus, from this physical frame, we can obtain λ *logical frames*, one for each category, where the logical frame for category C_i is the same as the physical frame that the reader could obtain if the tag population only contains the tags in category C_i. Figure 3.2 shows an example of obtaining λ logical frames from a physical frame. For example, in the third slot, the ternary string xxx is the collision result of three types of single-one strings: 100, 010, and 001.

We now zoom into the logical frame for category C_i. For each slot, we either have the $\mathbb{SO}$

string or nothing. By denoting the slot containing $\mathbb{SO}$ string with 1 and the slot that is empty with 0, we can obtain a bit vector with f bits. Figure 3.2 shows three bit vectors that we obtain. Based on the bit vectors obtained by SEM, many excellent tag estimation protocols, as summarized in Table **??**, can be used to simultaneously estimate the tag cardinality of each category. For example, Enhanced Zero-Based estimator (EZB) [89] relies on an important intuition: the fewer tags are, the more empty slots will appear in the frame. Thus, EZB can exploit the number of empty slots in a frame to conduct the tag estimation. Here, we could use the number of 0s in each bit vector as the input of EZB to estimate the number of tags in the corresponding category. Besides EZB, many existing protocols such as FNEB [78] that leverages the index of the first non-empty slots in the frame, LoF [155] that makes use of the length of continuous non-empty slots, ART [177] that exploits the average run length of non-empty slots, can be built on our SEM to achieve simultaneous estimation over multiple categories.

Using our SEM approach, previous RFID estimation protocols can be significantly accelerated when facing the multi-category estimation problem. In the following, we use a numerical example to show this point. Let t_γ represent the duration of a slot for transmitting γ-bit data and is given by $\tau_w + \gamma \times \tau_b$, where τ_w is the waiting time and τ_b is the time for transmitting one bit. Typically, $\tau_w = 302us$ and $\tau_b = 18.8us$ [215, 156]. As there are λ categories, in SEM each slot contains λ-bit single-one string, *i*.e., $\gamma = \lambda$. Thus, the time cost of an SEM frame is $f(\tau_w + \lambda \times \tau_b)$. On the other hand, the time cost of executing a frame of existing protocol once for each category is $\lambda f(\tau_w + \tau_b)$, where in this case $\gamma = 1$ because in existing protocols, each slot needs to carry only a single bit to indicate empty or non-empty. By comparing the execution time of a frame of SEM and existing protocols for all categories, it is easy to see that the number of slots executed by SEM are much smaller than the total number of slots executed by existing protocols. For example, when $\lambda = 30$, SEM is almost 11 times faster than the existing estimation protocols.

3.1.3 Challenges and Proposed Solutions

The first key challenge is to guarantee the required estimation accuracy specified by confidence interval $\alpha \in (0, 1]$ and required reliability $\beta \in [0, 1)$ for all categories. As the estimation based on one round of SEM has an inherent variance due to the probabilistic nature, we execute multiple rounds of SEM to reduce the variance of the estimate of each category. To ensure that SEM achieves the required accuracy, we first calculate the variance of the estimate for one round and the variance of the average estimate in multiple rounds. Then, we use statistical methods to find the minimum number of rounds that can achieve the required accuracy.

The second key challenge is to choose an optimal frame size f that minimizes the estimation time. The key factor that affects estimation time is f. We show that the execution time is a convex function with respect to the frame size, which means that the estimation time is long when the frame size is too small or too large. To find the optimal frame size, we propose an efficient binary search-based algorithm.

The third key challenge is to deal with categories that vary significantly in size. To minimize the estimation time, categories with small sizes demand a small frame size, whereas, categories with large sizes demand a large frame size. To address this issue, we propose an Adaptive Partitioning (AP) to group categories of similar sizes together and execute SEM for each group separately. Although this introduces more times of executing SEM, the estimation time for each group is well optimized as the categories in each group have similar sizes. Such a hybrid strategy has a smaller estimation time in comparison with the two extreme strategies of estimating each category separately and estimating all categories together. As we do not know category sizes in advance, we adaptively partition the categories based on the execution of previous rounds.

3.1.4 Novelty and Advantage over Prior Art

The key technical novelty of this paper lies in proposing an Single-one Manchester coding-based approach called SEM, built on which traditional tag estimation protocols can be used to address the multi-category RFID estimation in a simultaneous manner. The key technical depth of this paper is in the mathematical development of SEM in addressing the three technical challenges of guaranteeing accuracy, choosing frame sizes, and partitioning categories. The key advantage of our approach over prior art is that SEM can decode multiple bit vectors from just one physical frame to simultaneously estimate the tag cardinality of each category. Compared with the prior separate estimation methods, our SEM approach significantly reduces the number of physical slots, and thus achieves much better time-efficiency. For example, for an RFID system with 20 categories, our SEM+AP uses 2 seconds whereas the state-of-the-art ART protocol takes 14 seconds [177]. It represents that our SEM+AP is 7x faster than ART. As the number of categories increases, the normalized estimation time of our approach decreases, whereas, that of prior estimation protocols does not.

The rest of the paper is organized as follows. In Section 3.2, we review the related work. In Section 3.3.1, we present our SEM approach in detail along with its analysis. In Section 3.4, we describe how to calculate the optimal values for system parameters to minimize the estimation time of SEM while achieving the required reliability. In Section 3.5, we describe how SEM adaptively partitions the categories into comparable sizes to reduce the estimation time. In Section 3.6, we present results from our extensive evaluation of the proposed approach and its comparison with the existing protocols. Finally, in Section 3.7, we conclude the paper.

3.2 Related Work

At the infancy stage of RFID research, the academic communities have paid much attention to the exact tag identification problem [97, 178], which is to exactly identify the tag IDs within the interrogation range of an RFID reader. Generally, there are two types of

tag identification protocols: Aloha-based protocols and Tree-based protocols. Their basic principles are presented as follows. Fundamentally, the Aloha-based protocol is a kind of Time Division Multiple Access (TDMA) mechanism. A tag ID can be successfully identified in a slot when only one tag responds in this slot. As for tree-based protocols, the reader broadcasts a 0/1 string to query the tags. A tag responds with its ID once it finds that the queried string is the prefix of its ID. A reader identifies a tag ID when only one tag responds. Although RFID identification protocols can be used to obtain the exact tag IDs, it is a well-recognized fact that the tag identification protocols are slow because their execution time is proportional to the number of tags. For some purposes like stock monitoring, it is not efficient to execute the tag identification protocols because we only need to know the approximate number of tags instead of exact tag IDs.

Another direction of research on RFID systems is targeted at the cardinality estimation of RFID tag populations. Kodialam *et* al. proposed the first set of cardinality estimation schemes, USE and UPE, which use the number of empty or collision slots to estimate population sizes [88]. Similarly, Zheng *et* al. proposed Probabilistic Estimation Tree (PET) to estimate cardinalities for tree-based RFID systems [220]. Shahzad *et* al. proposed ART, which uses the average run length of non-empty slots for cardinality estimation [177]. Li *et* al. proposed Maximum Likelihood Estimator (MLE), which looks at the energy aspect of cardinality estimation [101]. Liu *et* al. studied the problem of key tag population tracking [121]. Gong *et* al. investigated INformative Counting (INC) to estimate the number of counterfeit tags whose IDs are not stored in a database [74]. For privacy reason, RFID estimation with the presence of blocker tag is investigated in [118].

The above literature assumes that all tags within an RFID system belong to the same category. However, in practical scenarios, tags are usually classified into different categories according to brands. In recent years, the researchers have shifted some attention to the interesting problems raising in the multi-category RFID systems. Sheng *et* al. addressed the problem of identifying categories whose cardinalities are above a given threshold [182]. They

proposed the Group Testing (GT) scheme, that rapidly eliminates the groups containing small-sized categories. Luo *et* al. claimed that the GT protocol is not suitable for RFID systems in which the sizes of a large number of categories are above a threshold, because each group has a high probability of containing a large-size category, and thus is difficult to eliminate. To accommodate this situation, they proposed an efficient Threshold-Based Classification (TBC) Protocol [128] that obtains multiple logical bitmaps from a single time frame. Each bitmap is used to approximate the tag cardinality of a category. The categories whose cardinalities are obviously above (or below) the given threshold can be rapidly eliminated. Unfortunately, GT and TBC protocols can only identify the categories with sizes greater than a threshold, but cannot estimate sizes of individual categories. The work closest to ours, focusing on multi-category RFID system, is Ensemble Sampling (ES) [207], which exploits the number of singleton slots occupied by each category in a time frame to estimate the tag cardinality in each category. ES can only distinguish three types of slots: empty slot, singleton slot, and collision slot. For collision slot, ES only knows two or more tags responded in this slot, and nothing else. How to make full use of the information in each type of slots especially that in the collision slots is the key to achieve better time efficiency. The proposed SEM exploits the Single-one Manchester coding string, and could know which categories of tags responded in a collision slot. From a single physical frame, it can derive multiple logical frames, and each servers the tag cardinality estimation for a category.

3.3 Proposed Research

In this section, I explain the proposed research methods to address the problem of multi-category RFID tags estimation.

3.3.1 SEM: Estimator and Variance

To estimate the number of tags in each category, SEM executes multiple Aloha frames. At the end of each frame, it obtains a bit vector for each category. Based on the obtained bit

vectors, SEM can perform any estimator mentioned before to simultaneously estimate the number of tags of each category. Here, for the purpose of clarity, we let SEM exploit the most classical estimator in EZB [89]. Note that, if more advanced estimators such as ART [177] or SRC [51] are used, the performance of SEM can be further improved. An insight behind EZB: the fewer tags are there, the more empty slots appear in the frame. Hence, we make use of the number of 0s in each bit vector to perform the estimation. The estimate obtained from the number of 0s observed in a single bit vector is not accurate due to the variance associated with the estimation process. Thus, instead of executing a single round, SEM executes k rounds and obtains k estimates of the number of tags in that category. It then calculates the average of those k estimates to obtain the fine-grained estimate. Next, we first formally derive the estimator that SEM uses to estimate the size of any given category, using the number of 0s in the bit vector corresponding to that category as input. Then, we derive the expression for variance of the estimator, which we will use in Section 3.4 to determine the values of system parameters to ensure that SEM achieves the required reliability in the minimum possible time. Table 3.1 summarizes the main notations used in this paper.

3.3.2 Estimator of SEM

Based on such a logical frame, many previous literature proposed excellent schemes to estimate the tag cardinality. Murali Kodialam et al. proposed to use the number of empty slots to perform the estimation [88]. Muhammad Shahzad et al. proposed to use average run length of non-empty slots for estimation [177]. As illustrated in Fig. 3.2, in an arbitrary *bit vector* for category C_i, we know the number of 0s. Intuitively, the more tags there are in category C_i, the less 0s there are in the bit vector. In fact, there is a *monotonically functional relationship* between the number of 0s in the bit vector and the tag cardinality n_i. Later, we will present a rigorous analysis to derive the functional relationship. Hence, we could use the number of 0s observed in the bit vector to estimate the tag cardinality n_i [88]. Note that, such an empty-slot-based estimator was proposed in [88]. We do not claim novelty on such

Table 3.1 Main notations used in the paper.

Notations	Descriptions
λ	# of tag categories under estimation.
C_i	category ID, $i \in [1, \lambda]$.
n_i	tag cardinality of category C_i.
α	required confidence interval.
β	required reliability.
$\hat{n_i}$	estimate of n_i.
δ	random number.
f	broadcast frame size.
f'	executed frame size.
$H(\cdot)$	uniform hash function.
τ_w	waiting time in a slot.
τ_b	time for transmitting 1-bit data from tag to reader.
t_γ	slot length that transmits γ-bit data. $t_\gamma = \tau_w + \gamma\tau_b$.
$p_{i,0}$	probability that a bit in the bit vector is 0.
$N_{i,0}$	# of 0s in the bit vector corresponding to category C_i.
$\hat{n_{i,j}}$	estimate for category C_i of the j^{th} frame.
$A_k(\hat{n_i})$	averaged result of k rounds of estimation for category C_i. $A_k(\hat{n_i}) = \frac{1}{k}\sum_{j=1}^{k} \hat{n_{i,j}}$
ℓ	the parameter of ℓ-sigma method. $\ell = 0,\ 1,\ 2,$ or 3.

an estimation idea. The novelty of this paper lies in proposing *single-one machester coding* to parallelize the estimation processes of multiple tag categories. The frame size should be no more than 512 in practice [177, 178] (the detailed reasons can be found in literature [178]). A solution used in [177] is to *initialize* a long frame with length of f, but terminate the frame after the first f' slots. Let n_i represent the number of tags in category C_i. Let f represent the number of slots that the reader broadcasts at the start of the frame. We call f the broadcast frame size. Let $p_{i,0}$ represent the probability that any bit in the bit vector of category C_i is 0. Formally, for large values of f, the probability $p_{i,0}$ is given by the following equation.

$$p_{i,0} = \left(1 - \frac{1}{f}\right)^{n_i} \approx e^{-\frac{n_i}{f}} \tag{3.1}$$

In the above equation, such an approximation is usually made in previous literature [119, 177].

Let the reader terminate the frame after executing f' slots, where $f' \leq f$. We call f' the executed frame size. Let $N_{i,0}$ be the random variable for number of 0s observed in the first f' bits of the bit vector of category C_i. As the probability for any bit to be 0 is $p_{i,0}$, the random variable $N_{i,0}$ follows binomial distribution $Binom(f', p_{i,0})$. Thus, the expected value of $N_{i,0}$ is given by the following equation.

$$E(N_{i,0}) = f' \cdot p_{i,0} = f' e^{-\frac{n_i}{f}} \tag{3.2}$$

Solving Eq. (3.2) for n_i, we get the following equation.

$$n_i = -f \ln \left\{ \frac{E(N_{i,0})}{f'} \right\} \tag{3.3}$$

This equation shows that for fixed given values of f and f', n_i is a monotonically decreasing function of $E(N_{i,0})$. Thus, we can estimate the value of n_i by substituting $E(N_{i,0})$ in the equation above by the observed value of $N_{i,0}$ from the logical frame of category C_i. Recall that $N_{i,0}$ represents the number of 0s observed from the bit vector of category C_i. Substituting $N_{i,0}$ for $E(N_{i,0})$ in Eq. (3.3), we get the estimator $\hat{n_i}$ of n_i as follows.

$$\hat{n_i} = -f \ln \left\{ \frac{N_{i,0}}{f'} \right\} \tag{3.4}$$

3.3.3 Variance of SEM

The following lemma calculates the variance in the estimator derived in Eq. (3.4).

Lemma 1. Let f and f' be the broadcast and executed frame sizes, respectively, and n_i be the number of tags in category C_i. The variance in the estimate $\hat{n_i}$ of n_i is given by the following equation.

$$Var(\hat{n_i}) = \frac{f^2}{f'} \left(e^{\frac{n_i}{f}} - 1 \right) \tag{3.5}$$

Proof. According to Eq. (3.4), $\hat{n_i}$ is a function of the random variable $N_{i,0}$. Thus, we express $\hat{n_i}$ as $\phi(N_{i,0})$. The Taylor's series expansion of $\phi(N_{i,0})$ around $E(N_{i,0})$ is given by the following equation.

$$\hat{n_i} = \phi(N_{i,0}) = \phi(\eta) + \frac{\partial \phi}{\partial N_{i,0}}(N_{i,0} - \eta),$$

where $\frac{\partial \phi}{\partial N_{i,0}}$ is the first-order derivative. Taking the expectation of both sides of the equation above, we have:

$$E[\hat{n_i}] = E[\phi(\eta)] + \frac{\partial \phi}{\partial N_{i,0}} E[(N_{i,0} - \eta)] = \phi(\eta)$$

The variance of $\hat{n_i}$ can now be calculated using the following expression.

$$Var(\hat{n_i}) = E[\hat{n_i} - E(\hat{n_i})]^2 = \left[\frac{\partial \phi}{\partial N_{i,0}}\right]^2 Var(N_{i,0}) \tag{3.6}$$

As required by the equation above, we next calculate the first-order derivative $\frac{\partial \phi}{\partial N_{i,0}}|_{N_{i,0}=\eta}$ and the variance $Var(N_{i,0})$.

$$\frac{\partial \phi}{\partial N_{i,0}} = -f \times \frac{f'}{N_{i,0}} \times \frac{1}{f'}$$

Replacing $N_{i,0}$ by $\eta = E(N_{i,0}) = f' e^{-\frac{n_i}{f}}$ in the equation above, we get:

$$\frac{\partial \phi}{\partial N_{i,0}}|_{N_{i,0}=\eta} = -\frac{f}{f'} e^{\frac{n_i}{f}}$$

As $N_{i,0} \sim Binom(f', p_{i,0})$, the variance $Var(N_{i,0})$ is given by the following equation.

$$Var(N_{i,0}) = f' p_{i,0}(1 - p_{i,0}) = f' e^{-\frac{n_i}{f}} \left(1 - e^{-\frac{n_i}{f}}\right) \tag{3.7}$$

Substituting the expressions of $\frac{\partial \phi}{\partial N_{i,0}}|_{N_{i,0}=\eta}$ and $Var(N_{i,0})$ into Eq. (3.6), we get the variance of $\hat{n_i}$ as given in Eq. (3.5) in the lemma statement. □

3.4 SEM: Parameter Optimization

In this section, we will derive equations that the controller uses at the start of i^{th} frame to calculate the optimal values of frame size f and f' and number of frames k to minimize the execution time of SEM while ensuring that its actual confidence interval is less than the required confidence interval. Recall from Section 3.3.1 that SEM executes k frames to estimate the number of tags in each category. Next, we first derive the expression to calculate the value of k, which ensures that SEM achieves the required reliability. After that, we derive expressions to calculate broadcast frame size f and executed frame size f', which ensure that the execution time of SEM is the minimum.

3.4.1 Number of Frames k

Let $\hat{n_{i,j}}$ represent the estimate of the number of tags in category C_i obtained from the j^{th} frame. Let $A_k(\hat{n_i})$ represent the average of the k estimates obtained from the k frames, *i*.e., $A_k(\hat{n_i}) = \frac{1}{k}\sum_{j=1}^{k} \hat{n_{i,j}}$. In what follows, Theorem 1 calculates the value of k which ensures that the average estimate satisfies the required reliability.

Theorem 1. Given required confidence interval α, required reliability β, broadcast frame size f, and executed frame size f', the average estimate $A_{k_i}(\hat{n_i})$ of the number of tags in category C_i satisfies the requirement $P\{|A_{k_i}(\hat{n_i}) - n_i| \leq n_i\alpha\} \geq \beta$ when the average is obtained from k_i frames, where k_i satisfies the following equation.

$$k_i \geq \left(\frac{fZ_\beta}{\alpha n_i}\right)^2 \left(\frac{e^{\frac{n_i}{f}} - 1}{f'}\right) \tag{3.8}$$

Proof. As SEM uses different seeds for each frame, the k_i frames are independent of each other. According to the central limit theorem, $\frac{A_{k_i}(\hat{n_i}) - E[A_{k_i}(\hat{n_i})]}{\sqrt{Var[A_{k_i}(\hat{n_i})]}}$ is a random variable that follows the standard normal distribution. Let us represent this random variable by $\aleph$. As $\aleph$ follows a standard normal distribution, for any required reliability β, there exists a number Z_β such that

$$P(-Z_\beta \leq \aleph \leq Z_\beta) = \beta \tag{3.9}$$

The requirement $P\{|A_{k_i}(\hat{n_i}) - n_i| \leq n_i\alpha\} \geq \beta$ can be written as below.

$$P\left\{\frac{(1-\alpha)n_i - E[A_{k_i}(\hat{n_i})]}{\sqrt{Var[A_{k_i}(\hat{n_i})]}} \leq \aleph \leq \frac{(1+\alpha)n_i - E[A_{k_i}(\hat{n_i})]}{\sqrt{Var[A_{k_i}(\hat{n_i})]}}\right\} \geq \beta \tag{3.10}$$

Comparing Eqs. (3.9) and (3.10), SEM will achieve the required reliability when the following conditions hold.

$$\begin{cases} \dfrac{(1-\alpha)n_i - E[A_{k_i}(\hat{n_i})]}{\sqrt{Var[A_{k_i}(\hat{n_i})]}} \leq -Z_\beta \\ \dfrac{(1+\alpha)n_i - E[A_{k_i}(\hat{n_i})]}{\sqrt{Var[A_{k_i}(\hat{n_i})]}} \geq Z_\beta, \end{cases} \tag{3.11}$$

Next we calculate the expectation and variance of $A_{k_i}(\hat{n_i})$.

$$E[A_{k_i}(\hat{n_i})] = \frac{1}{k_i}\sum_{j=1}^{k_i} E(\hat{n_{i,j}}) = n_i$$

$$Var[A_{k_i}(\hat{n_i})] = \frac{1}{k_i^2}\sum_{j=1}^{k_i} \frac{f^2}{f'}(e^{\frac{n_i}{f}}-1) = \frac{f^2}{k_i f'}(e^{\frac{n_i}{f}}-1) \tag{3.12}$$

Substituting the expressions for $E[A_{k_i}(\hat{n_i})]$ and $Var[A_{k_i}(\hat{n_i})]$ into either of the two inequalities in Eq. (3.11) and rearranging, we get the inequality in Eq. (3.8). □

3.4.2 Frame Sizes f and f'

For the given values of f' and f, Theorem 1 calculates the number of frames that SEM must execute to achieve the required reliability. Next we optimize the values of the executed and broadcast frame sizes to ensure that the estimation time of SEM is minimized.

Let T_i represent the minimum execution time needed by category C_i, t_λ represent the duration of each slot, and t_ξ represent the time that the reader takes to transmit the ξ-bit parameters for frame initialization. Thus, $T_i = k_i \times (t_\xi + f' \times t_\lambda) = \frac{(fZ_\beta)^2}{f'(\alpha n_i)^2}(e^{\frac{n_i}{f}} - 1) \times (t_\xi + f' \times t_\lambda)$. Let T represent the execution time of SEM for all categories, which should be equal to the longest execution time among all minimum execution times for the λ categories. It is easy to see that T_i is a monotonically decreasing function of f' because its first derivative $\frac{\partial T_i}{\partial f'} = -\frac{Z_\beta^2}{\alpha^2 n_i^2} f^2 (e^{\frac{n_i}{f}} - 1)\frac{t_\xi}{f'^2}$ is always negative. Therefore, SEM always sets f' to its maximum value. According to $C1G2$, the executed frame size f' should be no more than 512 due to practical reasons [177]. Meanwhile, executed frame size f' should also be smaller than the broadcast frame size f. Briefly, we set f' as $\min\{512, f\}$. Next, we will show that T_i is a convex function of n_i. To prove convexity, a sufficient and necessary condition is that the second-order derivative of T_i with respect to n_i is always larger than 0. The following equation calculates the second-order derivative of T_i with respect to n_i.

$$\frac{\partial^2 T_i}{\partial n_i^2} = \frac{f^2 Z_\beta^2 (t_\xi + f' t_\lambda)}{f' \alpha^2}\left[e^{\frac{n_i}{f}}\left(\frac{1}{f^2 n_i^2} - \frac{4}{f n_i^3} + \frac{6}{n_i^4}\right) - \frac{6}{n_i^4}\right] \tag{3.13}$$

For simplicity, we substitute $(\frac{1}{f^2 n_i^2} - \frac{4}{f n_i^3} + \frac{6}{n_i^4})$ with Φ. Note that $\Phi = (\frac{1}{f^2 n_i^2} - \frac{4}{f n_i^3} + \frac{6}{n_i^4}) \geq 2\sqrt{\frac{1}{f^2 n_i^2} \times \frac{6}{n_i^4}} - \frac{4}{f n_i^3} = \frac{2\sqrt{6}-4}{f n_i^3} > 0$. Furthermore, using the fourth-order Taylor series expansion of $e^{\frac{n_i}{f}}$, we know that $e^{\frac{n_i}{f}} > 1 + \frac{n_i}{f} + \frac{n_i^2}{2f^2} + \frac{n_i^3}{6f^3} + \frac{n_i^4}{24f^4}$. Then, Eq. (3.13) can be written as the following inequality.

$$\frac{\partial^2 T_i}{\partial n_i^2} > \frac{f^2 Z_\beta^2 (t_\xi + f' t_\lambda)}{f' \alpha^2} \left[\left(1 + \frac{n_i}{f} + \frac{n_i^2}{2f^2} + \frac{n_i^3}{6f^3} + \frac{n_i^4}{24f^4} \right) \Phi - \frac{6}{n_i^4} \right]$$

Substituting the value of Φ in the inequality above and simplifying, we get $\frac{\partial^2 T_i}{\partial n_i^2} > \frac{f^2 Z_\beta^2 (t_\xi + f' t_\lambda)}{f' \alpha^2} (\frac{2}{f n_i^3} + \frac{1}{12 f^4} + \frac{n_i^2}{24 f^6}) > 0$. As this second-order derivative is always greater than 0, T_i is a convex function of n_i. Let C_x and C_y be the categories with the fewest and the most number of tags, respectively, among all λ categories. Let n_x and n_y be the number of tags in the categories C_x and C_y, respectively. By the property of convex function, the maximum value of T_i lies at one of the two boundary points, *i*.e., (n_x, T_x) or (n_y, T_y). Thus, $T = \max\{T_x, T_y\}$. Minimizing the overall time T is equivalent to minimizing $\max\{T_x, T_y\}$. Formally, we need to solve the following optimization problem to find out the optimal values of f' and f to minimize $\max\{T_x, T_y\}$.

$$\begin{aligned}
&\text{Minimizing } \max\{T_x, T_y\} \\
s.t.\ & f' \in [1, 512] \\
& f' \leq f \\
& C_x \text{ is the smallest category under estimation} \\
& C_y \text{ is the largest category under estimation} \\
& T_i = \frac{(f Z_\beta)^2}{f' (\alpha n_i)^2} \left(e^{\frac{n_i}{f}} - 1 \right) \times \left(t_\xi + f' \times t_\lambda \right) \\
& i = x \text{ or } y
\end{aligned} \tag{3.14}$$

In the optimization problem formulated in Eq. (3.14), the executed frame size f' should be no more than 512 due to practical reasons [177]. It is easy to enumerate each possible value of f' to find the optimal one because of its small value range. However, f has a large value

range, and the enumeration method is not suitable when optimizing its value. Therefore, we investigate how to quickly optimize the value of f in the following. We first show that $\max\{T_x, T_y\}$ is a convex function of f, which means that there is a value of f for which the execution time of SEM is the minimum. Then, we describe a simple binary search-based method to determine the optimal value of f. The second-order derivative of T_i with respect to f is given by the following equation.

$$\frac{\partial^2 T_i}{\partial f^2} = \frac{Z_\beta^2(t_\xi + f' t_\lambda)}{\alpha^2 n_i^2 f'} \left[e^{\frac{n_i}{f}} \left(\frac{n_i^2}{f^2} - \frac{2n_i}{f} + 2 \right) - 2 \right] \tag{3.15}$$

For simplicity, we substitute $\frac{n_i^2}{f^2} - \frac{2n_i}{f} + 2$ with Ψ. Note that $\Psi = \frac{n_i^2}{f^2} - \frac{2n_i}{f} + 2 = (\frac{n_i}{f} - 1)^2 + 1 > 0$. Substituting $e^{\frac{n_i}{f}}$ with its fourth-order Taylor series in Eq. (3.15) and simplifying, we have the following inequality.

$$\frac{\partial^2 T_i}{\partial f^2} > \frac{Z_\beta^2(t_\xi + f' t_\lambda)}{\alpha^2 n_i^2 f'} \left(\frac{n_i^3}{3f^3} + \frac{n_i^4}{4f^4} + \frac{n_i^5}{12f^5} + \frac{n_i^6}{24f^6} \right) > 0 \tag{3.16}$$

As the second order derivative of T_i, with respect to f, is always greater than 0, T_i is a convex function of f. Thus, T_x and T_y are both convex functions of f. Consequently, $\max\{T_x, T_y\}$ is also a convex function of f.

Leveraging this convexity of $\max\{T_x, T_y\}$ with respect to f, SEM uses a fast binary-searching algorithm to find the optimal value of f. Given a $f' \leq 512$, SEM first initializes f_{low} to f', and f_{high} to $3n_y$. We have observed through simulations that $3n_y$ is a good upper bound on the size of broadcast frame. Second, SEM calculates the first-order derivative of $\max\{T_x, T_y\}$ at $\frac{f_{low}+f_{high}}{2}$. If this derivative is less than 0, it updates f_{low} to $\frac{f_{low}+f_{high}}{2}$; otherwise, it updates f_{high} to $\frac{f_{low}+f_{high}}{2}$. SEM recursively performs this search until $f_{low} = f_{high}$, at which point it stops and returns the value of f as $f = f_{low} = f_{high}$.

3.4.3 Dynamic Parameter Adjusting

To calculate the optimal values of system parameters, our proposed methods assume that SEM already knows the size of each category apriori. However, the category sizes are unknown

apriori and are actually the quantity we need to estimate. Next, we present how to obtain rough estimates of category sizes, which are then used to calculate the optimal values of system parameters.

Before executing the first frame, SEM sets the size of the smallest category to n_{min} and the largest category to n_{max}, where n_{min} and n_{max} are the lower and upper bounds on category sizes, respectively, and are provided by the system administrator. Using n_{min} and n_{max} as inputs, we calculate the broadcast frame size f using the binary search-based method proposed above. Note that our binary search based method is not sensitive to the rough values of n_{min} and n_{max} because the system parameter values converge to their near optimal values after only a few frames. After executing $\kappa > 1$ frames, we get average estimate $A_\kappa(\hat{n_i})$ for each category C_i. This $A_\kappa(\hat{n_i})$ is used to calculate the number of required frames, and should be repeated using Eq. (3.8).

3.4.4 Avoiding Premature Termination

As we calculate the number of times the frames are executed (*i*.e., k_i) using the estimated value $A_\kappa(\hat{n_i})$, which is not very accurate when κ is small, the value of k_i may be smaller than what it should be. Consequently, SEM may stop after executing fewer frames than it should have executed causing the estimated size of category C_i do not satisfy the required reliability. In other words, the estimation process for category C_i is terminated too early, which we call *premature termination*. As k_i is a monotonically increasing function of n_i, instead of substituting n_i with $A_\kappa(\hat{n_i})$, SEM substitutes n_i with $A_\kappa(\hat{n_i}) + \ell \cdot \sqrt{Var[A_\kappa(\hat{n_i})]}$ to calculate the value of k_i. The variance of $A_\kappa(\hat{n_i})$ was calculated in Eq. (3.12). According to the famous three-sigma rule [183], $\ell = 3$ should be large enough. We name this method of calculating k_i as the ℓ-*sigma* method. Through extensive simulations in Section 3.6, we show that our ℓ-sigma method is highly effective against premature termination.

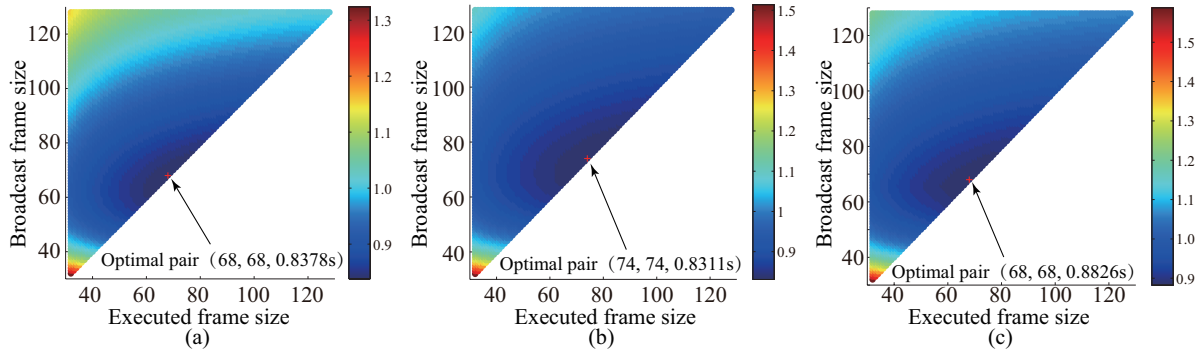


Figure 3.3 Separate estimation vs. simultaneous estimation in a balanced RFID system that contains two categories C_1 and C_2 with sizes of 100 and 110 tags respectively. $(\alpha, \beta) = (5\%, 95\%)$. (a) SEM on C_1. (b) SEM on C_2. (c) SEM on C_1 and C_2.

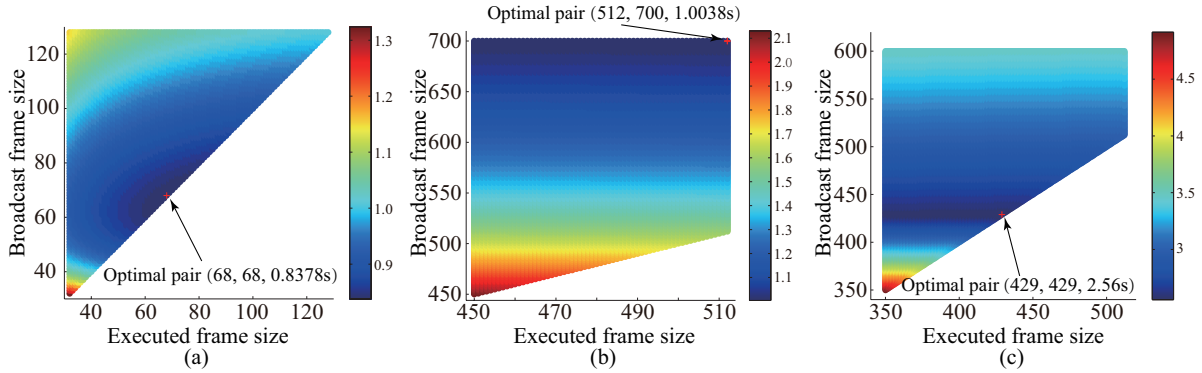


Figure 3.4 Separate estimation vs. simultaneous estimation in an unbalanced RFID system that contains two categories C_1 and C_2 with sizes of 100 and 2000 tags respectively. $(\alpha, \beta) = (5\%, 95\%)$. (a) SEM on C_1. (b) SEM on C_2. (c) SEM on C_1 and C_2.

3.5 SEM: Adaptive Partitioning

Until now, we have described how SEM executes multiple frames for all categories simultaneously, and estimates the sizes of the categories. This strategy works well only when all categories are *balanced*, *i*.e., sizes of all categories are similar. When the categories are *unbalanced*, *i*.e., sizes of categories are very different, simultaneously estimating sizes of all categories adversely affects the performance of SEM. Next, we discuss the two scenarios of balanced and unbalanced categories, respectively.

3.5.1 Category Types Analysis

3.5.1.1 Balanced Categories

We first consider an RFID system that consists of two categories C_1 and C_2 with similar sizes of 100 and 110 tags, respectively. Fig. 3.3(a) and (b) respectively show the minimal execution time of SEM when it is separately executed on tags in category C_1 and C_2. In the figure, the optimal pair, *e.g.*, $(68, 68, 0.8378s)$, means that the optimal values of both f' and f for SEM are 68, and the corresponding minimum execution time is 0.8378s. Note that the minimum time SEM takes to solely estimate the number of tags in category C_1 is 0.8378s. And the time for category C_2 is 0.8311s. Clearly, the total time of SEM when executed separately for each category is 1.6689s. In contrast, as shown in Fig. 3.3 (c), the minimum time SEM takes to simultaneously estimate the number of tags in both categories C_1 and C_2 is just 0.8826s, which is much smaller than the time SEM takes to estimate the number of tags in the categories separately. Thus, simultaneous estimation performs much better than separate estimation method in such a balanced RFID system.

3.5.1.2 Unbalanced Categories

Fig. 3.4(a) and (b) plot the minimum execution times of SEM for two categories C_1 and C_2 with quite different sizes of 100 and 2000 tags, respectively. The minimum time SEM takes to estimate the number of tags in categories C_1 and C_2 separately are 0.8378s and 1.0038s, resulting in the total time of 1.8416s. In contrast, as shown in Fig. 3.4(c), the minimum time SEM takes to simultaneously estimate the number of tags in both categories is 2.56s, which is much larger than the time SEM takes to separately estimate the number of tags in the categories. This happens because, for the unbalanced categories, it is hard to find a pair of parameters $\langle f, f' \rangle$ that simultaneously fit categories with large and small sizes. Thus, separate estimation performs better in the scenario of unbalanced categories.

From the above case studies of balanced and unbalanced categories, we conclude that

when the category sizes are unbalanced, SEM should first partition categories into groups such that the sizes of categories in the same group are comparable and then simultaneously estimate the sizes of categories in individual groups. This will reduce the overall estimation time of SEM. Next, we describe how SEM partitions categories into groups.

3.5.2 Adaptive Partitioning

At start, SEM assumes that all categories belong to the same group. Without loss of generality, it assumes that all categories are arranged in a list $L_{1,\lambda}$ in ascending order, *i*.e., $L_{1,\lambda} = \langle n_1, n_2, .., n_\lambda \rangle$, and for any i, $j \in [1, \lambda]$, if $i < j$, we have $n_i \leq n_j$. As aforementioned, the sizes of the smallest and largest categories in a group, *i*.e., n_1 and n_λ in this case, determine the estimation time of SEM. We represent the minimum time of SEM on a group that has the smallest category size n_i and the largest category size n_j by $T_{i,j}$. Recall that the estimation time of SEM is minimum when the values of n, f', and f are calculated as described in Section 3.4.

SEM partitions the group represented by list $L_{x,y} = \langle n_x, .., n_y \rangle$ into two groups represented by lists $L_{x,s} = \langle n_x, .., n_s \rangle$ and $L_{s+1,y} = \langle n_{s+1}, .., n_y \rangle$, where the value of s should satisfy the following two conditions.

1. $T_{x,s} + T_{s+1,y} \leq T_{x,y}$
2. $\forall z \in [x, y-1]$, $T_{x,s} + T_{s+1,y} \leq T_{x,z} + T_{z+1,y}$

SEM recursively applies this partitioning method on groups starting with $x = 1$ and $y = \lambda$ and continues until for a given group represented by list $L_{x,y}$, there is no $s \in [x, y-1]$ that satisfies the first condition. Fig. 3.5 shows an example where SEM partitions a large unbalanced group represented by the list $\langle n_1, n_2, n_3, n_4, n_5, n_6 \rangle$ into several small balanced groups represented by the lists $\langle n_1, n_2 \rangle$, $\langle n_3, n_4 \rangle$, and $\langle n_5, n_6 \rangle$. Note that, in Fig. 3.5, $T_{x,x}$ (*e*.g., $T_{1,1}$) means the minimum estimation time of SEM on a group that contains *just one* category C_x (*e*.g., C_1). After obtaining the small balanced groups, SEM takes one balanced group at a time and estimates the sizes of categories in that group simultaneously.

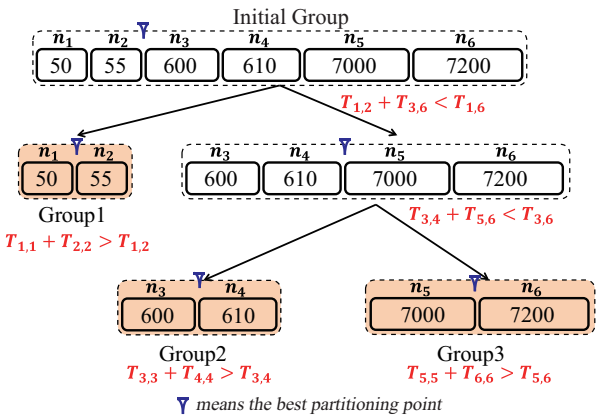


Figure 3.5 Example of Adaptive Partitioning (AP): an initial unbalanced group is partitioned into 3 balanced groups.

Just like in the calculation of optimal system parameters, adaptive partitioning also needs the size of each category apriori. At the very beginning, we do not know the number of tags in each category at all. Hence, for the first round of SEM, we let all the categories be in the same group. After the first round of estimation, SEM uses the method proposed in Section 3.4.3 to obtain the rough estimates of category sizes to guide the group partitioning process, and to find the optimum values of broadcast frame size f and executed frame size f' for each group. If for any category, the estimate $A_\kappa(\hat{n_i})$ achieves the required reliability after κ frames, SEM removes category C_i from the list $L_{1,\lambda}$. Before executing each frame, SEM first updates the list $L_{1,\lambda}$ by removing the categories for which the required reliability has been achieved, and then partitions them into groups. The estimation process terminates when all categories achieve the required reliability. What we should clarify is that the proposed Adaptive Partitioning method is a heuristic algorithm, and does not ensure to return the optimal grouping result.

3.5.3 Discussion about SEM

3.5.3.1 Multi-reader Estimation

Due to the limited communication range, a single RFID reader cannot cover a large area. Thus, multiple RFID readers are frequently deployed. SEM uses one of the many existing reader-scheduling protocols [211] to schedule which reader transmits and receives at what time. All readers always send the same commands and relay the data they receive to a back-end server. Thus, these readers essentially work like a logical big reader. SEM works seamlessly in single as well as multi-reader environments.

3.5.3.2 Bit Synchronization

Katabi *et* al. reported in [197] that the synchronization offset for commercial RFID tags is normally no more than 1us. Recall that transmitting each bit from a tag to a reader requires 18.8us. Hence, the 1us offset is only about 5.3% of a bit duration. In other words, the signal offset does not have much negative impact on SEM. Hence, like many top level RFID literature [225, 91], we also assume that the signals of each tag is well synchronized on bit level.

3.6 Performance Evaluation

In this section, we conduct extensive simulations on a large scale multi-category RFID system to evaluate the performance of SEM. We evaluate SEM in a variety of scenarios both with and without adaptive partitioning. We implemented SEM and 5 existing state-of-the-art RFID cardinality estimation schemes, namely Maximum Likelihood Estimator (MLE) [101], Enhanced Zero Based estimator (EZB) [89], Unified Probabilistic Estimator (UPE) [88], Average Run-based Tag estimation (ART) [177], and Ensemble Sampling (ES) [207]. Following the simulation strategy used by these state-of-the-art cardinality estimation schemes, we assume that the communication channel is error-free and a single reader covers

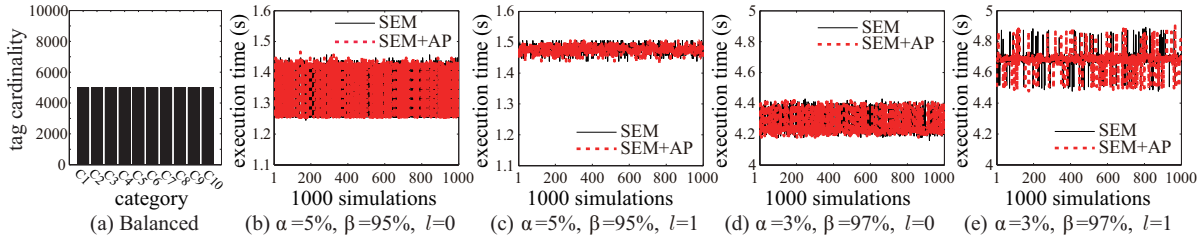


Figure 3.6 Comparing SEM+AP with SEM for balanced category sizes. Each category has the same size of 5000 tags.

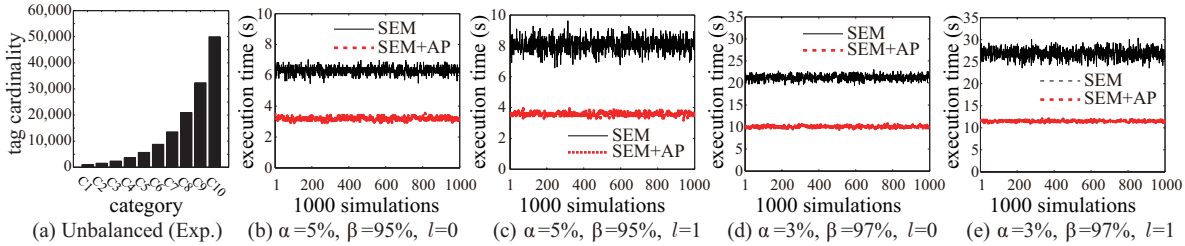


Figure 3.7 Comparing SEM+AP with SEM for unbalanced category sizes. The cardinalities of 10 categories are exponentially distributed.

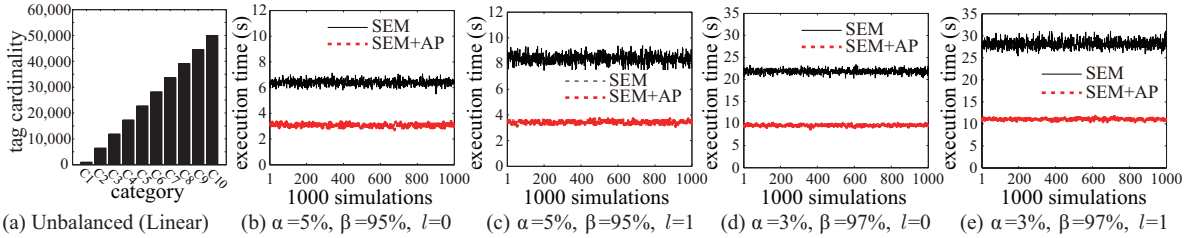


Figure 3.8 Comparing SEM+AP with SEM for unbalanced category sizes. The cardinalities of 10 categories are linearly distributed.

all tags. Recall that the execution of SEM in multi-reader scenario is same as that in the single reader scenario.

3.6.1 Evaluation Metrics

We evaluate SEM on two important metrics: (1) actual reliability, which is the percentage of times the relative errors in the estimates calculated by SEM are less than α, and (2) execution time, which is the time SEM takes to estimate the cardinalities of all tags in each category. We run each simulation 1000 times and use the results from these 1000 simulations to calculate the values of the performance metrics. Before evaluating these metrics, we first evaluate the effectiveness of our adaptive partitioning strategy. In the rest of this section,

we use SEM+AP to denote SEM with adaptive partitioning and simply SEM to denote it without partitioning.

3.6.2 Adaptive Partitioning

To evaluate the improvement in execution time due to adaptive partitioning, we simulate an RFID system containing a tag population with 10 categories. We conduct simulations for two accuracy requirements, *i*.e., $\alpha = 5\%, \beta = 95\%$ and $\alpha = 3\%, \beta = 97\%$ and two settings of ℓ, *i*.e., $\ell = 0$ and $\ell = 1$. We conduct simulations for both balanced and unbalanced categories.

3.6.2.1 Balanced Categories

In this case, each category has the cardinality of 5000 tags, as shown in Figure 3.6(a). Figures 3.6(b) through 3.6(e) show the execution times of both SEM+AP and SEM for the two accuracy requirements and the two settings of ℓ from 1000 independent runs of simulations. We observe from these figures that the execution time of SEM+AP and SEM are almost the same. This is because the frame size calculated by SEM is appropriate for all categories. This means that there is no need to partition the list $L_{1,10}$ into multiple groups. In fact, when SEM applies the adaptive partitioning algorithm on these 10 categories, the categories are not divided into multiple groups; rather, they are returned in a single group only.

3.6.2.2 Unbalanced Categories

In this case, the category sizes vary from 1000 tags to 50000 tags. We pick the category sizes from two different distributions: exponential distribution as shown in Figure 3.7(a) and linear distribution as shown in Figure 3.8(a). Figures 3.7(b) through 3.7(e) and 3.8(b) through 3.8(e) show the execution times of both SEM+AP and SEM for the two accuracy requirements and the two settings of ℓ from 1000 independent runs of simulations. We observe from these figures that the execution time of SEM+AP is 50% smaller than the execution time of SEM. For example, the execution times of SEM+AP and SEM with $\ell = 0$, $\alpha = 5\%$, and

$\beta = 95\%$ are approximately 3.2s and 6.3s, respectively. We make similar observations about SEM+AP and SEM for other settings of ℓ, α, and β when the categories are unbalanced. The underlying reason is that SEM+AP first adaptively partitions an unbalanced group into multiple balanced groups and then finds proper broadcast frame sizes for each group, which significantly reduces the execution time.

3.6.3 Actual Reliability

Recall that actual reliability is the percentage of times the estimates for any category C_i lie in the range $[(1-\alpha)n_i, (1+\alpha)n_i]$, where n_i is the actual cardinality of category C_i. We independently repeat each simulation scenario 1000 times and calculate the actual reliability from those 1000 estimation results. Figure 3.9 plots the actual reliability of SEM+AP in a balanced RFID system for the two accuracy requirements and the two settings of ℓ when the cardinalities of the categories are those shown in Figure 3.6(a). We observe from these two figures that the actual reliability achieved by SEM+AP for each category is higher than the required reliability β.

Figures 3.10 and 3.11 plot the actual reliability of SEM+AP in the unbalanced RFID system for the two accuracy requirements and the two settings of ℓ when the cardinalities of the categories are those shown in Figures 3.7(a) and Figures 3.8(a), respectively. We observe from these figures that SEM+AP with $\ell = 0$ sometimes does not satisfy the required reliability. This happens due to the premature termination, discussed in Section 3.4.4. However, with $\ell = 1$, the actual reliability of SEM+AP is always higher than the required reliability β in all scenarios. This further shows that our ℓ-sigma method with $\ell = 1$ is very effective in alleviating premature termination.

3.6.4 Execution Time

We evaluate the execution time of SEM+AP and present its side-by-side comparison with the execution times of five existing estimation protocols, namely MLE, EZB, UPE, ART,

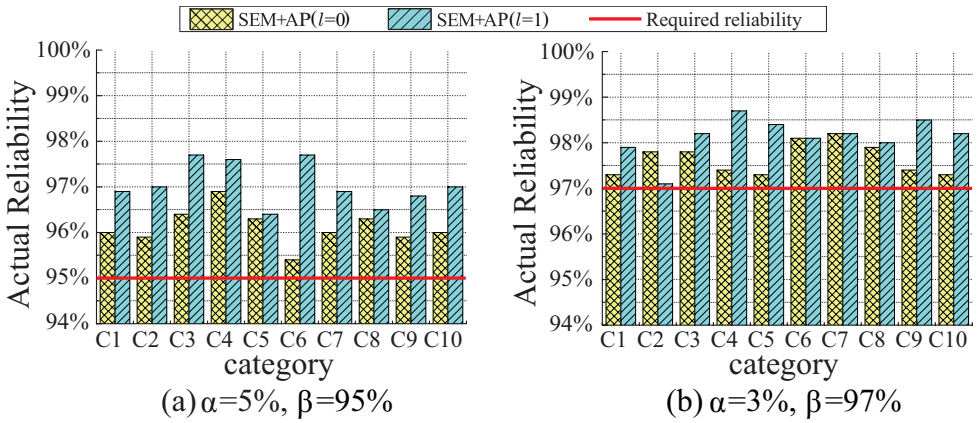


Figure 3.9 Actual reliability of SEM+AP for balanced categories.

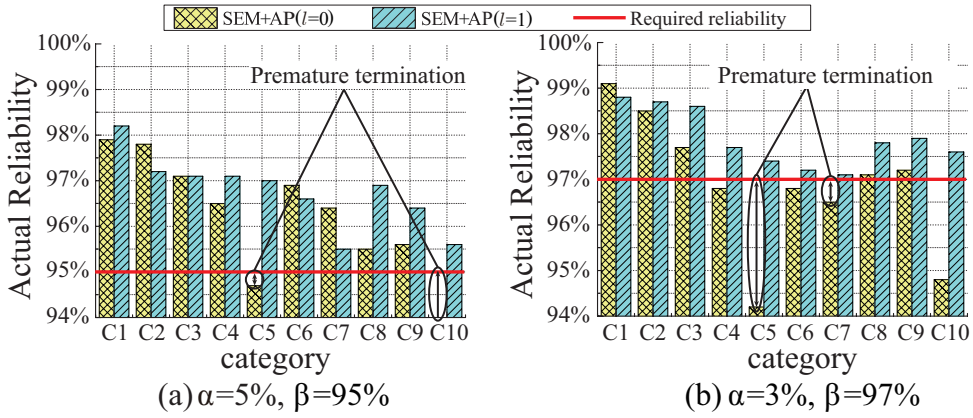


Figure 3.10 Actual reliability of SEM+AP for unbalanced (exponential) categories.

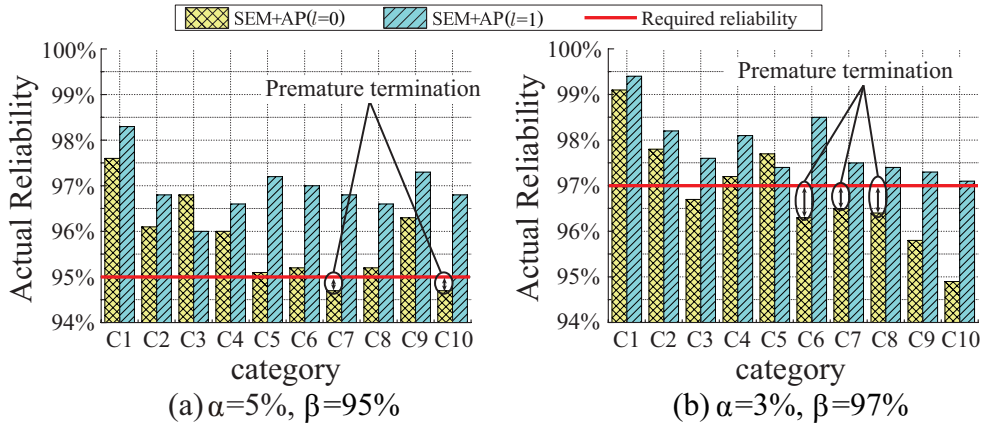


Figure 3.11 Actual reliability of SEM+AP for unbalanced (linear) categories.

and ES. We use these existing estimation protocols to separately estimate the cardinality of each category one by one, except for ES that simultaneously estimates the cardinalities of the top-k largest categories. We set $\mathcal{K}$ in ES equal to the total number of categories. We change the number of categories in tag populations from 1 to 20 and pick category sizes from two distributions: a non-uniform distribution to generate balanced categories and a uniform distribution to generate unbalanced categories. Next we present the execution time of SEM and existing protocols for the balanced and unbalanced categories.

3.6.4.1 Balanced Categories

In this case, for each value of number of categories, we pick the sizes of categories from the distribution shown in Figure 3.12(a). For example, the probability corresponding to 10000 tags is 0.25, which means that an arbitrary category has a 25% likelihood of being assigned a cardinality of 10000 when simulating the RFID system. Since the cardinalities with non-zero probabilities are within a relatively small range ($[8000, 12000]$), all categories will have similar cardinalities, resulting in a balanced categories scenario. Figure 3.12(b) plots the normalized average execution times of SEM+AP and existing protocols. Normalized execution time is calculated by dividing the execution time with the number of categories. We observe from this figure that SEM+AP is the only protocol whose average execution time per category decreases as the number of categories increases. Furthermore, SEM+AP is significantly faster compared with the prior estimation protocols. For example, with 20 categories, the average time per category of the fastest existing protocol, *i*.e., ART, is about 0.7 seconds, whereas that of our SEM+AP is just about 0.10 seconds, which is nearly 7 times faster than ART.

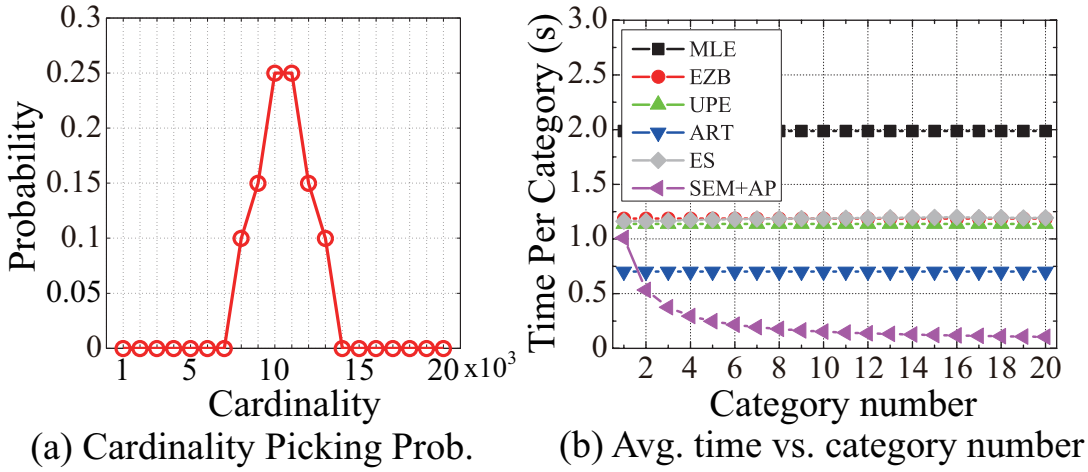


Figure 3.12 Execution time of SEM+AP ($\ell = 1$) and prior protocols for balanced categories. $\alpha = 5\%$, $\beta = 95\%$.

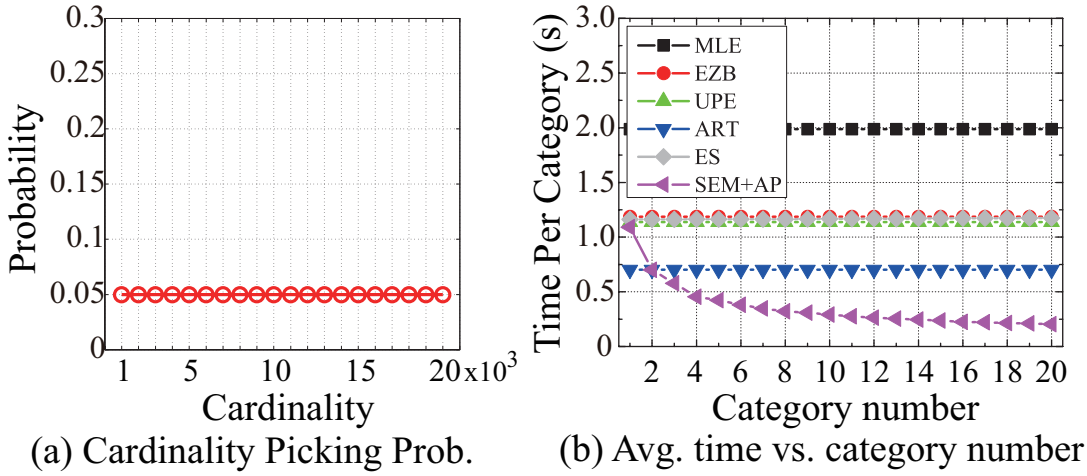


Figure 3.13 Execution time of SEM+AP ($\ell = 1$) and prior protocols for unbalanced categories. $\alpha = 5\%$, $\beta = 95\%$.

3.6.4.2 Unbalanced Categories

In this case, for each value of number of categories, we pick the sizes of categories from the distribution shown in Figure 3.13(a). Since the cardinalities with non-zero probabilities are in a relatively wide range ($[1000, 20000]$), different categories will have different cardinalities, resulting in an unbalanced categories scenario. Figure 3.12(b) plots the normalized average execution times of SEM+AP and existing protocols. We make two important observations. First, the average execution time of the existing protocols is almost the same as that

in the scenario of balanced categories. This is because the execution times of existing protocols only depend on the required accuracy and are independent of tag population sizes [177, 101, 89, 88, 51]. Thus, as long as the number of categories does not change, the execution time of the existing protocols does not change. Second, our SEM+AP protocol is persistently several times faster than all prior protocols for unbalanced categories as long as the number of categories is greater than 2.

3.7 Conclusion

In this paper, we make the following three key contributions. First, we formally defined the practically important problem of multi-category RFID estimation and proposed an Single-one Manchester coding-based approach called SEM. Our SEM approach could decode multiple bit vectors from a single physical frame, thereby achieving simultaneous estimation over multiple categories. Second, we propose the optimization technique of adaptive partitioning called AP to address the issue that category sizes may have large variances. The key idea is to group categories of similar sizes together and execute our SEM approach for each group separately. Third, we conducted extensive simulations to evaluate the proposed approaches. The simulation results show that our optimized SEM+AP approach can satisfy the predefined estimation accuracy while significantly outperforming all prior schemes, in terms of execution time. Moreover, we find that our SEM is the only approach whose normalized estimation time decreases as the number of categories increases. Many excellent estimation protocols dedicated to single-set estimation can be built on our SEM+AP to achieve fast and simultaneous estimation in multi-category RFID systems.

CHAPTER 4

FUTURE WORK

In this section, I provide an overview of the planned future work.

4.1 Missing Tag Detection and Identification

After briefly explain of my RFID work there is another question come out. Even RTSP can detect and identify some specific tags we want to know but RTSP cannot tell anything more than the ID of tag. Since previously we haven't make any change of the entire RFID systems which means we can not do some operations. However, as the RFID tags are designed to be more powerful we can do more work about that.

4.2 Motivation and Problem Statement

RFID systems with active tags are widely used in various application, such as indoor localization, object tracking , work-in-process tracking, supply chain management [143], [141], [142] etc. A typical RFID system consists of three elements: a large number of tags, RFID readers and a back-end data server. RFID tags are labeled in designated objects where each tag has a small size of memory to store its unique ID and some other information (e.g., product price, expiry date, personal information, etc). A reader has a dedicated power source with significant computing power. It transmits a command to query a set of tags, and the tags respond over a shared wireless medium. A data server which stores all tags information such as tag ID, brands or values of tagged items will do some computing based on the information from reader.

An interesting application of RFID is to detect and identify missing items in a large storage [126]. Consider a warehouse that stores a large number of commercial products. Some of the items might be expensive and some of the items might be inexpensive. Now assume a scenario where some of the items are stolen from the warehouse. In such situations, it is

important to not only identify the stolen items but also detect these events in timely fashion. However, most of the existing missing tag detection protocols spend a lot time to detect such events. These protocols also do not consider the value of the missing items and treat all the tags as having same value, [99], [181], [216], [61], [151]. However, this is untrue in the real world, RFID missing tag detection protocol is expected to be $unfair$. The reason is that the missing tag event of expensive products should be detected successfully with a high probability. As in this scenario, managers may want to detect and identify the expensive tags with a higher accuracy like 99.99%. But for other inexpensive ones, they may just want to detect with the accuracy above 90%.

This project will address this fundamental problem of achieving unfairness in RFID missing tag detection while minimizing identification time. Specifically, given a tag population of known size, with IDs and values of known distribution in the data server, and a required detection probability β, design a missing tag detection and identification protocol that minimizes the execution time under the constraint that the accuracy of detecting and identifying the expensive missing tag event is no less than β. The protocol should be compliant with the prevalent EPCGlobal Class 1 Generation 2 (C1G2) RFID standard.

4.3 Limitations of Prior Art

To the best of our knowledge, no prior work has been targeted on developing a unfair RFID missing tag detection protocol that can achieve any required accuracy. Existing RFID missing tag detection protocols mostly focus on minimizing detection time and energy [126], thus are not suitable to the real world because in these protocols, all non-singleton tag has to transmit equal number of times on average. In contrast, our objective is to minimize the detection time with total value of missing items constraint.

There are two types of missing tag detection and identification protocols: probabilistic [126], [186], [123] and deterministic [112], [99], [181], [216]. The probabilistic protocols are faster but only report the event that some tags are missing, without pinpointing exactly

which ones. The deterministic protocols return IDs of all missing tags but are comparatively slower. Both protocols are have their merits, and they are complementary to each other. For example, a probabilistic detection protocol should be used to detect a missing tag event and once detected positive, a deterministic protocol should be invoked to identify all the missing tags. There are two key limitations of existing protocols. The first limitation is that all existing protocols don't consider the value of items attached with tags, which is not a realistic assumption. Here is an example, suppose in a shopping mall which uses RFID readers to monitor only expensive merchandize, the reader receive response from tags of inexpensive merchandize as well. Existing protocols can not handle the presence of inexpensive tags because they pick up unexpected slot in Aloha frames resulting in unexpected false positive. The second key limitation of the missing tags detection and identification protocols except TRP, none of them is compliant with the EPCGloable Class 1 Generation 2 (C1G2) RFID standard. Most of those protocols require the manufactures to put random bit vectors in tags that tags use to calculate specialized hash functions. They also require the tags to be able to receive and interpret "pre-vector" and/or "post-vector" frames to select slots in frames. Such functionalities are not provisioned in the C1G2 standard. A protocol which is not compliant with the C1G2 RFID standard will lead to difficulty in deploying RFID system in different scenarios.

4.4 Solution Directions

For the problem of RFID missing tag detection with different values, there are three seemingly straightforward solutions based on previous work. The first solution is to execute a tag collection protocol repeatedly to collect IDs of all tags compare with IDs stored in the data server to detect and identify the missing tags. This method works; however, the long execution time is not appropriate for some scenarios. For example, in the airport luggage check out, we want to detect the missing tags as soon as possible.

The second solution is to first execute missing tag detection protocol on expensive tags

and then run the protocol on inexpensive tags. This can be achieved by using bloom filter to deactivate the inexpensive tags when executing the protocol with a higher accuracy requirement. After that, the inexpensive tags can be activated and expensive tags will not participate, running the same protocol again with lower accuracy requirement to minimize the total execution time. This method pays much attention to the expensive tags however it's not compliant with C1G2 RFID standard.

The third solution is to run our searching protocol RTSP. Since the data server has IDs of all tags which are supposed to be in the tag population. After running RTSP, we will know tags present in the population, therefore, the missing tag can be detected and identified. However, RTSP only can detect and identify the missing tags. It doesn't show any information about the value of items attached with tags.

In this project, we will develop a new protocol based on a sampled method to detect and identify the expensive tags with a higher accuracy requirement which can be assigned before the execution.

The missing tag detection and identification problem can be defined as: Given a tag set X with known IDs, a threshold value λ, a required accuracy α. Let A denote the event that reader reports a missing tag alert after running our detection protocol. m missing tags with values $\{x_1, x_2, ..., x_m\}$ and $\{x_1 \geq x_2 \geq ... \geq x_m\}$. Neither m or $\{x_i, i = 1..m\}$ is known. Our detection protocol should meet the following probability:

$$P\{A|\sum_{i=1}^{m} \geq \lambda\} \geq \alpha$$

Given a tag set with known tag IDs, a required confidence interval β, a required accuracy α and a threshold value λ. m missing tags $x_1, x_2, ..., x_m$ are unknown. If the total value of missing tags are larger or equal to λ, a missing tag detection and identification protocol outputs $\widetilde{y_1}, \widetilde{y_2}, ..., \widetilde{y_k}, .., \widetilde{y_s}$ so that top-k expensive tags can be identified with the probability greater or equals to $\alpha_1, \alpha_2, ..., \alpha_k$ and the missing tag event can be detected with the probability $P\{\frac{\sum_{i=1}^{m} x_i - \lambda}{\lambda} \leq \beta\} \geq \alpha$.

In our protocol, before each run, the data server will calculate a hash function to maximize the number of expensive tags who will respond in the coming frame. Meanwhile, this hash function will try to minimize the number of inexpensive tags who will participate in the frame. After the reader broadcast all the parameters, the data server will compare the frame collected by the reader with the logical frame it computes and deactivate those tags which are present and identify missing tags through empty slots which are supposed to be nonempty based on its computation. After several runs, all the missing tags can be detected.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Common Vulnerabilities and Exposures, http://cve.mitre.org/.

[2] Community Emergency Response Teams, http://www.cert.org/.

[3] Forum for Incident Response and Security Teams, http://www.first.org/cvss.

[4] http://www.centreforaviation.com/news/share-market/2010/06/17/hong-kong-airport-sets-new-cargo-traffic-record-fedex-sees-surging-asian-exports/page1.

[5] http://www.impinj.com/resources/about-rfid/.

[6] http://www.informationweek.com/rsa-unveils-rfid-tag-blocker/d/d-id/1023433?

[7] http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24LE1.

[8] http://xACMlpdp.sourceforge.net/.

[9] IBM Internet Security Systems, http://www.iss.net/.

[10] Managing more than 50,000 inbound freight containers and 30,000 outbound trailers annually is a logistical nightmare. but nyk logistics has found a truckload of savings by using an rfid yard-management system.

[11] National Vulnerability Database, http://nvd.nist.gov/.

[12] Organization for the Advancement of Structured Information Standards (OASIS), http://www.oasis-open.org.

[13] Secunia, http://secunia.com/.

[14] Security Focus, http://www.securityfocus.com/.

[15] Sun's XACML Implementation, http://sunxACMl.sourceforge.net.

[16] The Open Source Vulnerability Database, http://osvdb.org/.

[17] Vupen Security, http://www.vupen.com/english/.

[18] www.rfidsupplychain.com.

[19] University of California, Berkeley. Mica2 schematics. http://webs.cs.berkeley.edu/tos/hardware /design/ORCAD_FILES/MICA2/6310-0306-01ACLEAN.pdf, March 2003.

[20] Rfid insider roadmap. `http://blog.atlasrfidstore.com/learn-rfid/`, 2017.

[21] Rfid tags and readers store. `https://www.atlasrfidstore.com/`, 2017.

[22] Standard ISO 18000-6. Information technology automatic identification and data capture techniques-radio frequency identification for item management air interface. *Part 6. Parameters for Air interface communications at 860-960 MHZ*, 2003.

[23] Martin Abadi and Leslie Lamport. Composing specifications. *ACM Transactions on Programming Languages and Systems*, 15(1):73–132, 1993.

[24] Milton Abramowitz and Irene A. Stegun, editors. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover Publications, 1964.

[25] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of of 20th International Conference of on Very Large Data Bases*, pages 487–499, 1994.

[26] Omar H. Alhazmi and Yashwant K. Malaiya. Quantitative vulnerability assessment of systems software. In *Proceedings of Annual Reliability and Maintainability Symposium*, pages 615–620, 2005.

[27] Omar H. Alhazmi and Yashwant K. Malaiya. Prediction capabilities of vulnerability discovery models. In *Proceedings of Annual Reliability and Maintainability Symposium*, pages 86–91, 2006.

[28] Ross Anderson. Why information security is hard – an economic perspective. In *Proceedings of 17th Annual Computer Security Applications Conference of* , pages 358–365, 2001.

[29] Ross Anderson. Security in open versus closed systems – the dance of boltzmann, coase and moore. In *Proceedings of Open Source Software: Economics, Law, and Policy Confocuce*, June 2002.

[30] Ashish Arora, Chris Forman, Anand Nandkumar, and Rahul Telang. Competition and patching of security vulnerabilities: An empirical analysis. *Information Economics and Policy*, 22(2):164–177, 2010.

[31] Ashish Arora, Ramayya Krishnan, Rahul Telang, and Yubao Yang. An empirical analysis of software vendors?patch release behavior: Impact of vulnerability disclosure. *Information Systems Research*, 21(1):115–132, 2010.

[32] Ashish Arora, Anand Nandkumar, and Rahul Telang. Does information security attack frequency increase with vulnerability disclosure? an empirical analysis. *Information Systems Frontiers*, 8(8):350–362, 2006.

[33] E. Aydin, R. Oktem, Z. Dincer, and I. K. Akbulut. Study of an rfid based moving object tracking system. In *2007 1st Annual RFID Eurasia*, pages 1–5, Sept 2007.

[34] Michael Backes, Thomas R. Gross, and Guenter Karjoth. Tag identification system, 2008.

[35] Elisa Bertino, Sushil Jajodia, and Pierangela Samarati. A flexible authorization mechanism for relational data management systems. *ACM Transactions on Information Systems*, 17(2):101–140, 1999.

[36] N. Bhandari, A. Sahoo, and S. Iyer. Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency. *Proc. of IEEE ICIT*, 2006.

[37] B. A. Bjerke, J. G. Proakis, K. Y. M. Lee, and Z. Zvonar. A comparison of gsm receivers for fading multipath channels with adjacent and co-channel interference. *IEEE Journal on Selected Areas in Communications*, 18(11):2211–2219, Nov 2000.

[38] Piero Bonatti, Sabrina De Capitani di Vimercati, and Pierangela Samarati. An algebra for composing access control policies. *ACM Transactions on Information and System Security (TISSEC)*, 5(1):1–35, 2002.

[39] Charles Bordenave, David McDonald, and Alexandre Proutiere. Performance of random medium access control, an asymptotic approach. In *Proceedings of ACM SIGMETRICS*, 2008.

[40] M. Bouet and A. L. dos Santos. Rfid tags: Positioning principles and localization techniques. In *2008 1st IFIP Wireless Days*, pages 1–5, Nov 2008.

[41] Mehran Bozorgi, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond heuristics: Learning to classify vulnerabilities and predict exploits. In *Proceedings of of 16th International Conference of on Knowledge discovery and data mining*, pages 105–114, 2010.

[42] Richard P. Brent. *Algorithms for minimization without derivatives*. Prentice-Hall, 2002.

[43] Hilary K. Browne, William A. Arbaugh, John McHugh, and William L. Fithen. A trend analysis of exploitations. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 214–231, 2001.

[44] Michael Buettner, Richa Prasad, Alanson Sample, Daniel Yeager, Ben Greenstein, Joshua R. Smith, and David Wetherall. RFID sensor networks with the Intel WISP. In *Proceedings of 6th ACM Conference of on Embedded Networked Sensor Systems*, pages 393–394, 2008.

[45] Miguel Bustillo. Wal-mart radio tags to track clothing. *The Wall Street Journal*, 2010.

[46] Y. Liu C. Qian, H. Ngan and L. Ni. Cardinality estimation for large-scale rfid systems. *IEEE Transactions on Parallel and Distributed Systems*, 22(9):1441–1454, 2011.

[47] John I. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Transactions on Information Theory*, 25:505–515, 1979.

[48] Bogdan Carbunar, Murali Krishna Ramanathan, Mehmet Koyuturk, Christoph Hoffmann, and Ananth Grama. Redundant reader elimination in RFID systems. In *Proceedings of IEEE Communications Society Conference of on SECON*, pages 576–580, 2005.

[49] Auto ID Center. *Draft Protocol Specification for a 900 MHz Class 0 Radio Frequency Identification Tag*, 2–10 2007.

[50] Jae-Ryong Cha and Jae-Ryong Cha. Dynamic framed slotted aloha algorithms using fast tag estimation method for RFID system. In *Proceedings of 3rd IEEE Consumer Communications and Networking Conference of* , 2006.

[51] Binbin Chen, Ziling Zhou, and Haifeng Yu. Understanding RFID Counting Protocols. *Proc. of ACM MobiCom*, 2013.

[52] Min Chen and Shigang Chen. ETAP: Enable Lightweight Anonymous RFID Authentication with O(1) Overhead. *Proc. of IEEE ICNP*, 2015.

[53] Min Chen and Shigang Chen. Identifying State-free Networked Tags. *Proc. of IEEE ICNP*, 2015.

[54] Min Chen, Wen Luo, Zhen Mo, Shigang Chen, and Yuguang Fang. An efficient tag search protocol in large-scale RFID systems. In *Proceedings of IEEE INFOCOM*, 2013.

[55] Min Chen, Wen Luo, Zhen Mo, Shigang Chen, and Yuguang Fang. An Efficient Tag Search Protocol in Large-Scale RFID Systems With Noisy Channel. *IEEE/ACM Transactions on Networking*, 24(2):703–716, 2016.

[56] Yuan-Hsin Chen, Shi-Jinn Horng, Ray-Shine Run, Jui-Lin Lai, Rong-Jian Chen, Wei-Chih Chen, Yi Pan, and Terano Takao. A Novel Anti-Collision Algorithm in RFID Systems for Identifying Passive Tags. *IEEE Transactions on Industrial Informatics*, 6(1):105–121, 2010.

[57] Sandy Clark, Stefan Frei, Matt Blaze, and Jonathan Smith. Familiarity breeds contempt: The honeymoon effect and the role of legacy code in zero-day vulnerabilities. In *Proceedings of 26th International Annual Computer Security Applications Conference of* , pages 251–260, 2010.

[58] Robert Dorfman. The detection of defective members of large populations. *Annals of Mathematical Statistics*, 14:436–440, 1943.

[59] EPC EPCglobal. Radio-frequency identity protocols class-1 generation-2 uhf rfid protocol for communications at 860 mhz–960 mhz version 1.0. 9. *K. Chiew et al./On False Authenticationsfor C1G2 Passive RFID Tags*, 65, 2004.

[60] Klaus Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. Wiley, 2010.

[61] B. Firner, P. Jadhav, Y. Zhang, R. Howard, W. Trappe, and E. Fenson. Towards continuous asset tracking: Low-power communication and fail-safe presence assurance. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, pages 1–9, June 2009.

[62] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

[63] Robert C Francis, James P McGee, Robert A Sainati, Richard L Sheehan Jr, and Sai-Kit K Tong. Object tracking and management system and method using radio-frequency identification tags, July 29 2003. US Patent 6,600,418.

[64] Stefan Frei, Thomas Duebendorfer, Gunter Ollmann, and Martin May. Understanding the web browser threat: Examination of vulnerable online web browser populations and the "insecurity iceberg". Technical report, Eidgen
"ossische Technische Hochschule Z
"urich, 2008.

[65] Stefan Frei, Martin May, Ulrich Fiedler, and Bernhard Plattner. Large-scale vulnerability analysis. In *Proceedings of 2006 SIGCOMM workshop on Large-Scale Attack Defense*, pages 131–138, September 2006.

[66] Stefan Frei, Bernhard Tellenbach, and Bernhard Plattner. 0-day patch exposing vendors (in) security performance. In *Proceedings of Black Hat Technical Security Conference of* , volume 14, 2009.

[67] Karsten Fyhn, , Rasmus Melchior Jacobsen, Petar Popovski, and Torben Larsen. Fast capture – recapture approach for mitigating the problem of missing rfid tags. *IEEE Transactions on Mobile Computing*, 11(3):518–528, 2012.

[68] Xingxin Gao, Zhe Xiang, Hao Wang, Jun Shen, Jian Huang, and Song Song. An approach to security and privacy of RFID system for supply chain. *Proc. of IEEE CEC-East*, 2004.

[69] Joaquin Garcia-Alfaro, Michel Barbeau, and Evangelos Kranakis. Analysis of threats to the security of EPC networks. In *Proceedings of 6th Annual Communication Networks and Services Research Conference of* , pages 67–74, 2008.

[70] Sylvie Ghez, Sergio Verdu, and Stuart C. Schwartz. Stability properties of slotted aloha with multipacket reception capability. *IEEE Transactions on Automatic Control*, 33:640–649, 1988.

[71] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, and Philip Levis. CTP: Robust and efficient collection through control and data plane integration. Technical report, Univ. of Southern California, UC Berkeley, MIT CSAIL, Stanford University, 2008.

[72] Simon Godik and Tim Moses. eXtensible Access Control Markup Language (XACML) Version 2. *Standard, OASIS. February*, 2005.

[73] Wei Gong, Kebin Liu, Xin Miao, and Haoxiang Liu. Arbitrarily Accurate Approximation Scheme for Large-Scale RFID Cardinality Estimation. *Proc. of IEEE INFOCOM*, 2014.

[74] Wei Gong, Kebin Liu, Xin Miao, Qiang Ma, Zheng Yang, and Yunhao Liu. Informative Counting: Fine-grained Batch Authentication for Large-Scale RFID Systems. *Proc. of ACM Mobihoc*, 2013.

[75] Wei Gong, Kebin Liu, Xin Miao, Qiang Ma, Zheng Yang, and Yunhao Liu. Informative Counting: Fine-grained Batch Authentication for Large-Scale RFID Systems. *Proc. of ACM MobiHoc*, 2013.

[76] Mohamed G. Gouda and Xiang-Yang Alex Liu. Firewall design: Consistency, completeness, and compactness. In *Proceedings of 24th International Conference of on Distributed Computing Systems*, pages 320 – 327, 2004.

[77] Dirk Hahnel, Wolfram Burgard, Dieter Fox, Ken Fishkin, and Matthai Philipose. Mapping and localization with RFID technology. *Proc. of IEEE ICRA*, 2004.

[78] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu. Counting RFID Tags Efficiently and Anonymously. *Proc. of IEEE INFOCOM*, 2010.

[79] Hao Han, Bo Sheng, Chiu C. Tan, Qun Li, Weizhen Mao, and Sanglu Lu. Counting RFID tags efficiently and anonymously. In *Proceedings of IEEE INFOCOM*, 2010.

[80] Peter Adam Hoeher, Sabah Badri-Hoeher, Wen Xu, and Claudiu Krakowski. Single-antenna co-channel interference cancellation for tdma cellular radio systems. *IEEE Wireless Communications*, 12(2):30–37, 2005.

[81] Don R. Hush and Cliff Wood. Analysis of tree algorithms for RFID arbitration. In *Proceedings of IEEE International Symposium on Information Theory*, 1998.

[82] EPCGlobal Inc. *Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz*. EPCGlobal Inc, 1.2.0 edition, 2008.

[83] Rasmus Jacobsen, Karsten Fyhn Nielsen, Petar Popovski, and Torben Larsen. Reliable identification of rfid tags using multiple independent reader sessions. In *Proceedings of IEEE International Conference of on RFID*, pages 64–71, 2009.

[84] Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical report, Digital Equipment Corporation, 1984.

[85] Sushil Jajodia, Pierangela Samarati, V. S. Subrahmanian, and Eliza Bertino. A unified framework for enforcing multiple access control policies. In *Proceedings of ACM SIGMOD International Conference of on Management of data*, pages 474–485, 1997.

[86] Ari Juels, Ronald L Rivest, and Michael Szydlo. The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. *Proc. of ACM CCS*, 2003.

[87] M Ayoub Khan, Manoj Sharma, and Ha Prabhu R. A survey of rfid tags.

[88] Murali Kodialam and Thyaga Nandagopal. Fast and reliable estimation schemes in RFID systems. In *Proceedings of 12th International Conference of on Mobile Computing and Networking*, pages 322–333, 2006.

[89] Murali Kodialam, Thyaga Nandagopal, and Wing Cheong Lau. Anonymous Tracking using RFID tags. *Proc. of IEEE INFOCOM*, 2007.

[90] Murali Kodialam, Thyaga Nandagopal, and Wing Cheong Lau. Anonymous tracking using RFID tags. In *Proceedings of IEEE INFOCOM*, 2007.

[91] Linghe Kong, Liang He, Yu Gu, Min-You Wu, and Tian He. A Parallel Identification Protocol for RFID Systems. *Proc. of IEEE INFOCOM*, 2014.

[92] Yuan-Cheng Lai, Ling-Yen Hsiao, Hong-Jie Chen, Ching-Neng Lai, and Jian-Wei Lin. A Novel Query Tree Protocol with Bit Tracking in RFID Tag Identification. *IEEE Transactions on Mobile Computing*, 12(10):2063–2075, 2013.

[93] Ching Law, Kayi Lee, and Kai-Yeung Siu. Efficient memoryless protocol for tag identification. In *Proceedings of 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2000.

[94] Chun Hee Lee and Chin-Wan Chung. Efficient storage scheme and query processing for supply chain management using RFID. In *Proceedings of ACM Conference of on Management of data*, pages 291–302, 2008.

[95] S. Lee, S. Joo, and C. Lee. An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification. *Proc. of IEEE MobiQuitous*, 2005.

[96] Su-Ryun Lee, Sung-Don Joo, and Chae-Woo Lee. An enhanced dynamic framed slotted aloha algorithm for rfid tag identification. In *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 166–172, July 2005.

[97] Su Ryun Lee, Sung Don Joo, and Chae Woo Lee. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. In *Proceedings of 2nd Annual International Conference of on Mobile and Ubiquitous Systems: Networking and Services*, pages 166–172, 2005.

[98] T. Li, S. Chen, and Y. Ling. Efficient Protocols for Identifying the Missing Tags in a Large RFID System. *IEEE/ACM Transactions on Networking*, 21(6):1974–1987, 2013.

[99] Tao Li, Shigang Chen, and Yibei Ling. Identifying the missing tags in a large rfid system. In *Proceedings of the Eleventh ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '10, pages 1–10, New York, NY, USA, 2010. ACM.

[100] Tao Li, Shigang Chen, and Yibei Ling. Identifying the Missing Tags in a Large RFID System. *Proc. of ACM MobiHoc*, 2010.

[101] Tao Li, Samuel Wu, Shigang Chen, and Mark Yang. Energy efficient algorithms for the RFID estimation problem. In *Proceedings of IEEE INFOCOM*, 2010.

[102] Tao Li, Samuel Wu, Shigang Chen, and Mark Yang. Generalized Energy-Efficient Algorithms for the RFID Estimation Problem. *IEEE/ACM Transactions on Networking*, 20(6):1978–1990, 2012.

[103] Chieh-Jan Mike Liang, Jie Liu, Liqian Luo, Andreas Terzis, and Feng Zhao. RACNet: A high-fidelity data center sensing network. In *Proceedings of 7th ACM Conference of on Embedded Networked Sensor Systems*, pages 15–28, 2009.

[104] Convergence Systems Limited. `http://www.csl-rfid.com/`.

[105] Alex X. Liu, Fei Chen, JeeHyun Hwang, and Tao Xie. XEngine: A Fast and Scalable XACML Policy Evaluation Engine. In *Proceedings of of International Conference of on Measurements and Modeling of Computer Systems SIGMETRICS*, pages 265–276, 2008.

[106] Alex X. Liu, Fei Chen, and JeeHyun Hwang Tao Xie. XEngine: A Fast and Scalable XACML Policy Evaluation Engine. In *Proceedings of of International Conference of on Measurements and Modeling of Computer Systems SIGMETRICS*, pages 265–276, 2008.

[107] Alex X. Liu and Mohamed G. Gouda. Diverse firewall design. *IEEE Transactions on Parallel and Distributed Systems*, 19(9):1237 – 1251, September 2007.

[108] Haoxiang Liu, Wei Gong, Lei Chen, Wenbo He, Kebin Liu, and Yunhao Liu. Generic Composite Counting in RFID Systems. *Proc. of IEEE ICDCS*, 2014.

[109] J. Liu, B. Xiao, K. Bu, and L. Chen. Efficient Distributed Query Processing in Large RFID-enabled Supply Chains. *Proc. of IEEE INFOCOM*, 2014.

[110] Jia Liu, Bin Xiao, Shigang Chen, Feng Zhu, and Lijun Chen. Fast RFID grouping protocols. *Proc. of IEEE INFOCOM*, 2015.

[111] Tianci Liu, Lei Yang, Qiongzheng Lin, and Yunhao Liu. Anchor-free Backscatter Positioning for RFID Tags with High Accuracy. *Proc. of IEEE INFOCOM*, 2014.

[112] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu. Completely pinpointing the missing rfid tags in a time-efficient way. *IEEE Transactions on Computers*, 64(1):87–96, Jan 2015.

[113] Xiulong Liu, Keqiu Li, Geyong Min, Kai Lin, Bin Xiao, Yanming Shen, and Wenyu Qu. Efficient Unknown Tag Identification Protocols in Large-Scale RFID Systems. *IEEE Transactions on Parallel and Distributed Systems*, 25(12):3145–3155, 2014.

[114] Xiulong Liu, Keqiu Li, Geyong Min, Yanming Shen, Alex X. Liu, and Wenyu Qu. A Multiple Hashing Approach to Complete Identification of Missing RFID Tags. *IEEE Transactions on Communications*, 62(3):1046–1057, 2014.

[115] Xiulong Liu, Keqiu Li, Heng Qi, Bin Xiao, and Xin Xie. Fast Counting the Key Tags in Anonymous RFID Systems. *Proc. of IEEE ICNP*, 2014.

[116] Xiulong Liu, Keqiu Li, Heng Qi, Bin Xiao, and Xin Xie. Fast counting the key tags in anonymous RFID systems. In *Proceedings of IEEE ICNP*, 2014.

[117] Xiulong Liu, Keqiu Li, Heng Qi, Bin Xiao, and Xin Xie. Fast Counting the Key Tags in Anonymous RFID Systems. *Proc. of IEEE ICNP*, 2014.

[118] Xiulong Liu, Bin Xiao, Keqiu Li, Alex X. Liu, Jie Wu, Xin Xie, and Heng Qi. RFID Estimation with Blocker Tags. *IEEE/ACM Transactions on Networking*, in press, 2016.

[119] Xiulong Liu, Bin Xiao, Keqiu Li, Jie Wu, Alex X. Liu, Heng Qi, and Xin Xie. RFID Cardinality Estimation with Blocker Tags. *Proc. of IEEE INFOCOM*, 2015.

[120] Xiulong Liu, Bin Xiao, Keqiu Li, Jie Wu, Alex X. Liu, Heng Qi, and Xin Xie. RFID Cardinality Estimation with Blocker Tags. *Proc. of IEEE INFOCOM*, 2015.

[121] Xiulong Liu, Xin Xie, Keqiu Li, Bin Xiao, Jie Wu, Heng Qi, and Dawei Lu. Fast Tracking the Population of Key Tags in Large-scale Anonymous RFID Systems. *IEEE/ACM Transactions on Networking*, in press, 2016.

[122] Xuan Liu, Shigeng Zhang, Bin Xiao, and Kai Bu. Flexible and Time-Efficient Tag Scanning with Handheld Readers. *IEEE Transactions on Mobile Computing*, pp(99):1–1, 2015.

[123] W. Luo, S. Chen, T. Li, and S. Chen. Efficient missing tag detection in rfid systems. In *INFOCOM, 2011 Proceedings IEEE*, pages 356–360, April 2011.

[124] W. Luo, S. Chen, T. Li, and S. Chen. Efficient Missing Tag Detection in RFID Systems. *Proc. of IEEE INFOCOM*, 2011.

[125] W. Luo, S Chen, T. Li, and Y. Qiao. Probabilistic Missing-tag Detection and Energy-Time Tradeoff in Large-scale RFID Systems. *Proc. of ACM MobiHoc*, 2012.

[126] Wen Luo, Shigang Chen, Tao Li, and Yan Qiao. Probabilistic missing-tag detection and energy-time tradeoff in large-scale rfid systems. In *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '12, pages 95–104, New York, NY, USA, 2012. ACM.

[127] Wen Luo, Yan Qiao, and Shigang Chen. An Efficient Protocol for RFID Multigroup Threshold-based Classification. *Proc. of IEEE INFOCOM*, 2013.

[128] Wen Luo, Yan Qiao, Shigang Chen, and Min Chen. An Efficient Protocol for RFID Multigroup Threshold-Based Classification Based on Sampling and Logical Bitmap. *IEEE/ACM Transactions on Networking*, 24(1):397–407, 2016.

[129] Michael R. Lyu. *Handbook of Software Reliability Engineering*. McGraw-Hill, Inc. Hightstown, NJ, USA, 1996.

[130] Chandan Maity, Ashutosh Gupta, and Mahua Maity. Timing analysis of passive UHF RFID-EPC c1g2 system in dynamic frame. *Contemporary Computing*, pages 216–227, 2009.

[131] J. Maneesilp, C. Wang, H. Wu, and N. F. Tzeng. Rfid support for accurate 3d localization. *IEEE Transactions on Computers*, 62(7):1447–1459, July 2013.

[132] Gaia Maselli, Chiara Petrioli, and Claudio Vicari. Dynamic tag estimation for optimizing tree slotted aloha in RFID networks. In *Proceedings of 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 315–322, 2008.

[133] Peter Mell, Karen Scarfone, and Sasha Romanosky. *A Complete Guide to the Common Vulnerability Scoring System Version 2.0.* National Institute of Standards and Technology, June 2007.

[134] Katina Michael and Luke McCathie. The pros and cons of RFID in supply chain management. *Proc. of IEEE ICMB*, 2005.

[135] Jihoon Myung and Wonjun Lee. An adaptive memoryless tag anti-collision protocol for RFID netwroks. In *Proceedings of IEEE INFOCOM*, 2005.

[136] Jihoon Myung and Wonjun Lee. Adaptive splitting protocols for RFID tag collision arbitration. In *Proceedings of 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 202–213, 2006.

[137] Jihoon Myung and Wonjun Lee. Adaptive Splitting Protocols for RFID Tag Collision Arbitration. *Proc. of ACM MobiHoc*, 2006.

[138] V. Namboodiri and L. Gao. Energy-Aware Tag Anti-Collision Protocols for RFID Systems. *Proc. of IEEE PerCom*, 2007.

[139] Vinod Namboodiri and Lixin Gao. Energy-aware tag anticollision protocols for RFID systems. In *Proceedings of 5th IEEE International Conference of on Pervasive Computing and Communications*, pages 23–36, 2007.

[140] Badri Nath, Franklin Reynolds, and Roy Want. RFID technology and applications. *Proceedings of IEEE Pervasive Computing*, 5:22–24, 2006.

[141] Aditya Nemmaluri, Mark D. Corner, and Prashant Shenoy. Sherlock: Automatically locating objects for humans. In *Proceedings of International Conference of on Mobile Systems, Applications, and Services*, 2008.

[142] T. J. Ng, M. M. Wong, J. B. Zhang, and O. P. Gan. Rfid for mro work in progress tracking. In *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, pages 4779–4784, Nov 2006.

[143] L. M. Ni, Yunhao Liu, Yiu Cho Lau, and A. P. Patil. Landmarc: indoor location sensing using active rfid. In *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 407–415, March 2003.

[144] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil. Landmarc: Indoor location sensing using active RFID. *Wireless networks*, 10:701–710, 2004.

[145] Qun Ni, Elisa Bertino, and Jorge Lobo. D-Algebra for Composing Access Control Policy Decisions. In *Proceedings of of 4th International Symp. on Information, Computer, and Communications Security*, pages 298–309, 2009.

[146] Qun Ni, Elisa Bertino, and Jorge Lobo. D-Algebra for Composing Access Control Policy Decisions. In *Proceedings of of 4th International Symp. on Information, Computer, and Communications Security*, pages 298–309, 2009.

[147] Jeongyeup Paek and Ramesh Govindan. RCRT: Rate-controlled reliable transport for wireless sensor networks. In *Proceedings of 5th International Conference of on Embedded Networked Sensor Systems*, pages 305–319, 2007.

[148] Lei Pan and Hongyi Wu. Smart trend-traversal: A low delay and energy tag arbitration protocol for large RFID systems. In *Proceedings of IEEE INFOCOM*, 2009.

[149] Apostolia Papapostolou and Hakima Chaouchi. Rfid-assisted indoor localization and the impact of interference on its performance. *Journal of Network and Computer Applications*, 34(3):902 – 913, 2011. RFID Technology, Systems, and Applications.

[150] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of 2nd International Conference of on Embedded Networked Sensor Systems*, pages 95–107, 2004.

[151] P. Popovski, K. Fyhn, R. M. Jacobsen, and T. Larsen. Robust statistical methods for detection of missing rfid tags. *IEEE Wireless Communications*, 18(4):74–80, August 2011.

[152] Saiyu Qi, Yuanqing Zheng, Mo Li, Li Lu, and Yunhao Liu. COLLECTOR: A Secure RFID-Enabled Batch Recall Protocol. *Proc. of IEEE INFOCOM*, 2014.

[153] Chen Qian, Yunhuai Liu, Hoilun Ngan, and Lionel M. Ni. ASAP: Scalable identification and counting for contactless RFID systems. In *Proceedings of 30th IEEE International Conference of on Distributed Computing Systems*, pages 52–61, 2010.

[154] Chen Qian, Hoilun Ngan, and Yunhao Liu. Cardinality estimation for large-scale RFID systems. In *Proceedings of 6th Annual IEEE International Conference of on Pervasive Computing and Communications*, pages 30–39, 2008.

[155] Chen Qian, Hoilun Ngan, Yunhao Liu, and Lionel M. Ni. Cardinality Estimation for Large-scale RFID Systems. *IEEE Transactions on Parallel and Distributed Systems*, 22(9):1441–1454, 2011.

[156] Yan Qiao, Shigang Chen, Tao Li, and Shiping Chen. Energy-efficient Polling Protocols in RFID Systems. *Proc. of ACM Mobihoc*, 2011.

[157] Eric Rescorla. Security holes... who cares. In *Proceedings of 12th USENIX Security Symposium*, 2003.

[158] Eric Rescorla. Is finding security holes a good idea? *IEEE Security and Privacy*, 3(1):14–19, Januray 2005.

[159] John Rice. *Mathematical statistics and data analysis*. Cengage Learning, 2006.

[160] Mark Roberti. A 5-cent breakthrough. *RFID Journal*, 5(6), 2006.

[161] Mark Roberti. Wal-mart relaunches EPC RFID effort, starting with men's jeans and basics. *RFID Journal*, 2010.

[162] L. G. Roberts. Aloha Packet System with and without Slots and capture. *ACM SIGCOMM Computer Communication Review*, 5(2):28–42, 1975.

[163] Walter A. Rosenkrantz and Don Towsley. On the instability of slotted ALOHA multiaccess algorithm. *IEEE Transactions on Automatic Control*, 28:994–996, 1983.

[164] Walter A. Rosenkrantz and Donald Towsley. On the instability of slotted aloha multiaccess algorithm. *IEEE Transactions on Automatic Control*, 28(10):994–996, 1983.

[165] Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, Elsevier, 9th edition, 2007.

[166] Mark Schilling. Understanding probability: Chance rules in everyday life. *The American Statistician*, 60(1):97–98, 2006.

[167] Bruce Schneier. Cryptogram September 2000-Full Disclosure and the Window of Exposure.

[168] Frits C Schoute. Dynamic Frame Length ALOHA. *IEEE Transactions on Communications*, 31(4):565 – 568, 1983.

[169] Guido Schryen. A comprehensive and comparative analysis of the patching behavior of open source and closed source software vendors. In *Proceedings of 5th International Conference of on IT Security Incident Management and IT Forensics*, pages 153–168, 2009.

[170] E. Eugene Schultz, David S. Brown, and Thomas A. Longstaff. *Responding to Computer Security Incidents: Guidelines for Incident Handling*. Lawrence Livermore National Laboratory, Livermore, CA, 1990.

[171] Philips Semiconductors. *SL2 ICS11 I.Code UID Smart Label IC Functional Specification Datasheet http://www.advanide.com/datasheets/sl2ics11.pdf*, 2004.

[172] Philips Semiconductors. I-CODE Smart Label RFID Tags. *http://www.nxp.com/acrobat_ download/other/identification/SL092030.pdf*, Jan 2004.

[173] Vahid Shah-Mansouri and Vincent W.S. Wong. Cardinality estimation in RFID systems with multiple readers. In *Proceedings of IEEE Global Communications Conference of* , 2009.

[174] Vahid Shah-Mansouri and Vincent W.S. Wong. Cardinality estimation in RFID systems with multiple readers. *IEEE Transactions on Wireless Communications*, 10(5):1458–1469, 2011.

[175] Muhammad Shahzad and Alex X Liu. Every bit counts: fast and scalable RFID estimation. In *Proceedings of ACM MobiCom*, 2012.

[176] Muhammad Shahzad and Alex X. Liu. Probabilistic optimal tree hopping for RFID identification. In *Proceedings of ACM SIGMETRICS*, 2013.

[177] Muhammad Shahzad and Alex X. Liu. Fast and Accurate Estimation of RFID Tags. *IEEE/ACM Transactions on Networking*, 23(1):241–254, 2015.

[178] Muhammad Shahzad and Alex X. Liu. Probabilistic Optimal Tree Hopping for RFID Identification. *IEEE/ACM Transactions on Networking*, 23(3):796–809, 2015.

[179] Muhammad Shahzad, M. Zubair Shafiq, and Alex X. Liu. A large scale exploratory analysis of vulnerability life cycles (extended version). Technical report, Michigan State University, `www.msu.edu/~shahzadm/ICSE2012/shahzad2011vulnsTR.pdf`, 2011.

[180] Longfei Shangguan, Zimu Zhou, Xiaolong Zheng, Lei Yang, Yunhao Liu, and Jinsong Han. ShopMiner: Mining Customer Shopping Behavior in Physical Clothing Stores with Passive RFIDs. *Proc. of ACM SenSys*, 2015.

[181] Bo Sheng, Qun Li, and Weizhen Mao. Efficient continuous scanning in rfid systems. In *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, pages 1010–1018, Piscataway, NJ, USA, 2010. IEEE Press.

[182] Bo Sheng, Chiu C. Tan, Qun Li, and Weizhen Mao. Finding Popular Categories for RFID Tags. *Proc. of ACM Mobihoc*, 2008.

[183] Nikolaj V Smirnov, Igor? V Dunin-Barkovskij, and Wolfgang Richter. *Mathematische Statistik in der Technik*. Dt. Verlag d. Wiss., 1963.

[184] David Eugene Smith. *A source book in mathematics*. Courier Dover Publications, 2012.

[185] Claire Swedberg. Honeywell aerospace tags parts for airbus. *RFID Journal*, 2013.

[186] C. C. Tan, B. Sheng, and Q. Li. How to monitor for missing rfid tags. In *Distributed Computing Systems, 2008. ICDCS '08. The 28th International Conference on*, pages 295–302, June 2008.

[187] Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall, 2002.

[188] ShaoJie Tang, Jing Yuan, Xiang-Yang Li, Guihai Chen, Yunhao Liu, and JiZhong Zhao. Raspberry: A stable reader activation scheduling protocol in multi-reader RFID systems. In *Proceedings of IEEE ICNP*, 2009.

[189] Rahul Telang and Sunil Wattal. An empirical analysis of the impact of software vulnerability announcements on firm stock price. *IEEE Transactions on Software Engineering*, 33(8):544–557, 2007.

[190] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[191] Géraldine Vache. Vulnerability analysis for a quantitative security evaluation. In *Proceedings of 3rd International Symp. on Empirical Software Engineering and Measurement*, 2009.

[192] Ehsan Vahedi, Vahid Shah-Mansouri, Vincent W. S. Wong, Ian F.Blake, and Rabab K. Ward. Probabilistic Analysis of Blocking Attack in RFID Systems. *IEEE Transactions on Information Forensics and Security*, 6(3):803 – 817, 2011.

[193] Harald Vogt. Efficient object identification with passive RFID tags. *Pervasive Computing*, 2414:98–113, 2002.

[194] James Waldrop, Daniel W. Engels, and Sanjay E. Sarma. Colorwave: A MAC for RFID reader networks. In *Proceedings of IEEE Wireless Communications and Networking*, pages 1701–1704, 2003.

[195] Chong Wang, Hongyi Wu, and Nian-Feng Tzeng. RFID-based 3-D positioning schemes. In *Proceedings of IEEE INFOCOM*, 2007.

[196] Fei Wang, Bin Xiao, Kai Bu, and Jinshu Su. Detect and Identify Blocker Tags in Tree-based RFID Systems. *Proc. of IEEE ICC*, 2013.

[197] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and Reliable Low-power Backscatter Networks. *Proc. of ACM SIGCOMM*, 2012.

[198] Roy Want. An introduction to rfid technology. *IEEE Pervasive Computing*, 5(1):25–33, 2006.

[199] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. *Security in Pervasive Computing*, pages 50–59, 2004.

[200] J. Werb and C. Lanzl. Designing a positioning system for finding things and people indoors. *IEEE Spectrum*, 35(9):71–78, Sep 1998.

[201] D. Wijesekera and S. Jajodia. A propositional policy algebra for access control. *ACM Transactions on Information and System Security (TISSEC)*, 6(2):286–325, 2003.

[202] Duminda Wijesekera and Sushil Jajodia. A propositional policy algebra for access control. *ACM Transactions on Information and System Security (TISSEC)*, 6(2):286–325, 2003.

[203] Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. *Weka: Practical Machine Learning Tools and Techniques with Java Implementations*. Citeseer, 1999.

[204] Yan Wu, Harvey Siy, and Robin Gandhi. Empirical results on the study of software vulnerabilities: NIER track. In *Proceedings of 33rd International Conference of on Software Engineering*, pages 964–967, 2011.

[205] Qingjun Xiao, Bin Xiao, and Shigang Chen. Differential Estimation in Dynamic RFID Systems. *Proc. of IEEE INFOCOM*, 2013.

[206] Qingjun Xiao, Bin Xiao, and Shigang Chen. Differential Estimation in Dynamic RFID Systems. *Proc. of IEEE INFOCOM*, 2013.

[207] Lei Xie, Hao Han, Qun Li, Jie Wu, and Sanglu Lu. Efficient Protocols for Collecting Histograms in Large-Scale RFID Systems. *IEEE Transactions on Parallel and Distributed Systems*, 26(9):2421–2433, 2015.

[208] L. Yang, J. Cao, W. Zhu, and S. Tang. Accurate and efficient object tracking based on passive rfid. *IEEE Transactions on Mobile Computing*, 14(11):2188–2200, Nov 2015.

[209] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. Tagoram: Real-Time Tracking of Mobile RFID Tags to High Precision Using COTS Devices. *Proc. of ACM MobiCom*, 2014.

[210] Lei Yang, Yi Guo, Tianci Liu, Cheng Wang, and Yunhao Liu. Perceiving the Slightest Tag Motion beyond Localization. *IEEE Transactions on Mobile Computing*, 14(11):2363–2375, 2015.

[211] Lei Yang, Jinsong Han, Yong Qi, Cheng Wang, Tao Gux, and Yunhao Liu. Season: Shelving Interference and Joint Identification in Large-Scale RFID Systems. *Proc. of IEEE INFOCOM*, 2011.

[212] Lei Yang, Qiongzheng Lin, Xiang-Yang Li, Tianci Liu, and Yunhao Liu. See Through Walls with COTS RFID System. *Proc. of ACM MobiCom*, 2015.

[213] Guang yao Jin, Xiao yi Lu, and Myong-Soon Park. An indoor localization mechanism using active rfid tag. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, volume 1, pages 4 pp.–, June 2006.

[214] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.

[215] Yafeng Yin, Lei Xie, Sanglu Lu, and Daoxu Chen. Check out the Rules: Towards Time-Efficient Rule Checking over RFID Tags. *Mobile Networks and Applications*, 19(4):524–533, 2014.

[216] R. Zhang, Y. Liu, Y. Zhang, and J. Sun. Fast identification of the missing tags in a large rfid system. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*, pages 278–286, June 2011.

[217] Shigeng Zhang, Xiaoxian He, Hong Song, and Daqiang Zhang. Time efficient tag searching in multiple reader RFID systems. In *Proceedings of Green Computing and Communications (GreenCom)*. IEEE, 2013.

[218] Bin Zhen, Mamoru Kobayashi, and Masashi Shimizu. Framed ALOHA for multiple RFID objects identification. *IEICE Transactions on Communications*, 88:991–999, 2005.

[219] Yuanqing Zheng and Mo Li. Fast Tag Searching Protocol for Large-Scale RFID Systems. *Proc. of IEEE ICNP*, 2011.

[220] Yuanqing Zheng and Mo Li. PET: Probabilistic Estimating Tree for Large-scale RFID Estimation. *IEEE Transactions on Mobile Computing*, 11(11):1763–1774, 2012.

[221] Yuanqing Zheng and Mo Li. Fast tag searching protocol for large-scale RFID systems. *IEEE/ACM Transactions on Networking (TON)*, 21(3):924–934, 2013.

[222] Yuanqing Zheng and Mo Li. Fast Tag Searching Protocol for Large-Scale RFID Systems. *IEEE/ACM Transactions on Networking*, 21(3):924–934, 2013.

[223] Yuanqing Zheng and Mo Li. ZOE: Fast Cardinality Estimation for Large-Scale RFID Systems. *Proc. of IEEE INFOCOM*, 2013.

[224] Yuanqing Zheng and Mo Li. ZOE: Fast Cardinality Estimation for Large-Scale RFID Systems. *Proc. of IEEE INFOCOM*, 2013.

[225] Yuanqing Zheng and Mo Li. P-MTI: Physical-Layer Missing Tag Identification via Compressive Sensing. *IEEE/ACM Transactions on Networking*, 23(4):1356–1366, 2015.

[226] Zongheng Zhou, Himanshu Gupta, Samir R. Das, and Xianjin Zhu. Slotted scheduled tag access in multi-reader RFID systems. In *Proceedings of IEEE International Conference of on Network Protocols*, pages 61–70, 2007.