Fundamental study on improving RFID system operational efficiency

Zheng, Yuan Qing

2014

Zheng, Y. Q. (2014). Fundamental study on improving RFID system operational efficiency.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61878>

<https://doi.org/10.32657/10356/61878>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**FUNDAMENTAL STUDY ON
IMPROVING RFID SYSTEM
OPERATIONAL EFFICIENCY**

YUANQING ZHENG
SCHOOL OF COMPUTER ENGINEERING
2014

FUNDAMENTAL STUDY ON IMPROVING RFID SYSTEM OPERATIONAL EFFICIENCY

YUANQING ZHENG

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2014

Acknowledgments

I am deeply indebted to my PhD advisor, Prof Mo Li, for his great support and guidance throughout my study. He is an inspirational scientist in research, a great mentor along the way, and a great friend beside me. Without his consistent support and encouragement, it would be impossible for me to come up with the work presented in this dissertation. From the bottom of my heart, I am sincerely grateful for all you have done for me.

My special thanks go to Prof Wentong Cai, Prof Chengzheng Shun, Prof Xueyan Tang, Prof Bingsheng He, and all my colleagues in Parallel and Distributed Computing Center (PDCC), for creating excellent research environments. I would like to thank Irene Ng-Goh Siew Lai for her support along the way.

I would also love to thank all the family members of WANDS – Moye, Zhenjiang Li, Wan Du, Pengfei Zhou, Cheng Li, Jiajue Ou, Yuxiao Hou, Yaxiong Xie, Shiqi Jiang, and Man, Mibao, Fang, Xiaoxiao, Xiaoyu, as well as former members Saiyu Qi, Zhidan Liu, Wenwei Chen, and Zhiyuan Chen. It is truly my pleasure and privilege to be with you.

I would like to acknowledge the financial support of the Research Scholarship for my 4-year full-time studies at Nanyang Technological University, Singapore. I would also like to thank anonymous reviewers and editors for their insightful comments and valuable suggestions which contribute to improve the quality of this dissertation.

Finally, yet most importantly, my parents Jinnv Jin and Yinghao Zheng, and my wife Yan Han. Without your love and encouragement, I would never have come this far. This dissertation is dedicated to you.

Contents

List of Figures	iv
List of Tables	vii
Summary	viii
1 Introduction	1
1.1 Background	1
1.2 Review of Existing Research Works	2
1.2.1 Collision Arbitration	2
1.2.2 Cardinality Estimation	5
1.2.3 RFID Tag search	6
1.2.4 Missing tag identification	7
1.3 Dissertation Organization	8
2 System Models and Performance Metrics	9
2.1 System Models	9
2.2 Performance Metrics	11
2.2.1 Execution Time	11
2.2.2 Memory Overhead	12
2.2.3 Computational Overhead	13
3 Cardinality Estimation	14
3.1 Problem Description	15
3.2 PET: Probabilistic Estimating Tree based Cardinality Estimation	17
3.2.1 Basic Algorithm	17
3.2.2 Algorithm Analysis	20

3.2.3	General Protocol	24
3.2.4	$\mathcal{O}(\log\log n)$ Algorithm	26
3.2.5	Discussion	30
3.3	ZOE: Zero-One Estimator based Cardinality Estimation	33
3.3.1	Basic Algorithm	33
3.3.2	General Protocol	37
3.3.3	Parameter Setting	39
3.3.4	Reliable Estimation with Unreliable Channel	43
3.4	Evaluation	45
3.4.1	Simulation Setting	45
3.4.2	PET Investigation	47
3.4.3	ZOE Investigation	48
3.4.4	Performance Comparison	51
3.5	Summary	56
4	RFID Tag Search	57
4.1	Problem Description	60
4.2	CATS: Compact Approximator based Tag Search Protocol	62
4.2.1	Baseline protocol	62
4.2.2	Two-phase Tag Search Protocol	64
4.2.3	Joint Optimization	69
4.2.4	Cardinality Range Estimation	72
4.2.5	Discussion	75
4.3	Evaluation	77
4.3.1	Simulation Setting	77
4.3.2	CATS Investigation	78
4.4	Summary	83
5	Missing Tag Identification	84
5.1	Problem Description	86
5.2	P-MTI: Physical-layer Missing Tag Identification Protocol	87
5.2.1	P-MTI Rationale	88

5.2.2	Physical-layer Missing Tag Identification: A Basic Solution	90
5.2.3	Compressive Sensing based Recovery	92
5.2.4	Enhancement against Noisy Measurement	94
5.2.5	Operation Overview	95
5.2.6	Multiple Readers	96
5.2.7	Discussion	96
5.3	Implementation	98
5.3.1	Backscatter Signals	101
5.3.2	Backscatter Synchronization	103
5.3.3	Identifying Missing WISP Tag	104
5.4	Evaluation	104
5.4.1	Simulation Setup	105
5.4.2	P-MTI Investigation	105
5.4.3	Performance Comparison	108
5.5	Summary	110
6	Conclusion and Future Work	111
6.1	Conclusion	111
6.2	Future Work	112
	Appendix	114
A	Author's Publications related to this Dissertation	114

List of Figures

1.1	Aloha-based RFID identification [42]	3
1.2	Tree-based RFID identification [49]	4
3.1	Probabilistic Estimating Tree	18
3.2	Gray node on the estimating path	26
3.3	Protocol execution process	28
3.4	The distribution of random numbers. The left three subfigures depict the random number distribution in 100 estimation rounds. The right three subfigures depict the random number distribution in each estimation round. The skyline depicts $\mathcal{R}(\text{id}_i)_{\max}$ for 100 estimation rounds.	39
3.5	The denominator $f(\lambda)$ achieves maximum value at $\lambda \approx 1$.	41
3.6	Parameter setting process: (a) Fast convergence to the optimal threshold value with bisection search method; (b) When $e^{-\lambda} \rightarrow 0$ or $e^{-\lambda} \rightarrow 1$, the variance is very small.	43
3.7	Estimation Accuracy.	46
3.8	Standard deviation.	47
3.9	Normalized standard deviation.	47
3.10	Evaluation of ZOE protocol with different numbers of estimating rounds.	49
3.11	Evaluation of estimation accuracy under different error rates.	50
3.12	Performance comparison: (a) Protocol performance with different confidence interval ε , and the same error probability $\delta = 1\%$; (b) Protocol performance with different error probability δ and the same confidence interval $\varepsilon = 5\%$.	52
3.13	Cumulative distribution of estimation results.	53

3.14	Memory consumption in storing the random numbers: (a) with different confidence interval ε , and the same error probability $\delta = 1\%$; (b) with different error probability δ , and the same confidence interval $\varepsilon = 5\%$. . .	54
3.15	Accuracy comparison of different protocols under varying error rates. . .	55
4.1	RFID Tag Searching	58
4.2	Two-phase filtering: (1) Transmission of the compact approximator $BF_1(\mathbf{X})$ from backend server to the interrogation zone. The readers broadcast $BF_1(\mathbf{X})$ to the tags. The tags that pass $BF_1(\mathbf{X})$ remain in candidate tag set $\mathbf{Y}_C = \mathbf{Y} \cap BF_1(\mathbf{X})$; (2) The candidate tags $\mathbf{Y} \cap BF_1(\mathbf{X})$ respond in the second-phase communication. The responses forms $BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))$ for server examination.	67
4.3	Sensitivity analysis: L_1 against $ \mathbf{Y} $	71
4.4	Cardinality range estimation with $ \mathbf{Y} = 10000$: (a) Cumulative distribution of estimation results; (b) Estimation results with estimation rounds $n = 512, 1024$, and 2048	78
4.5	(a) $\frac{\alpha}{\beta}$ varies from $\frac{1}{16}$ to 16 , and the same false positive rate $P_{REQ} = 5\%$; (b) P_{REQ} varies from 1% to 10% , and the symmetric data rates $\frac{\alpha}{\beta} = 1$. . .	79
4.6	CATS investigation with varied $ \mathbf{X} $ from 0.5×10^5 to 32×10^5 and the fixed $ \mathbf{Y} = 5000000$: (a) Total transmission time (in <i>log scale</i>); (b) False positive fraction.	82
4.7	CATS investigation with varied $ \mathbf{Y} $ from 0.5×10^5 to 32×10^5 and the fixed $ \mathbf{X} = 500000$: (a) Total transmission time (in <i>log scale</i>); (b) False positive fraction.	82
5.1	Frame-slotted Aloha: 7 tags contend for 11 time slots.	88
5.2	Received signals at RFID reader: (a) Signals from tag 1; (b) Signals from tag 2; (c) Aggregated signals from the two tags.	89
5.3	Response signals from 7 tags mix in the air and aggregate at the reader. If tag i is missing, then its contribution to the aggregated physical layer symbols would be missing.	90

5.4	Testbed: 2 circular antennas are mounted to USRP N210. The USRP N210 is connected via GigE to a laptop which acts as an RFID reader. .	99
5.5	A WISP tag.	99
5.6	MTI process.	100
5.7	Received signal strength.	100
5.8	Initial offsets among tags.	101
5.9	Received signals at RFID reader: (a) Aggregated signals from 5 tags; (b) Aggregated signals from 4 tags; (c) Differential of the two measurements.	102
5.10	Ratio of successfully identified missing tags across different channel conditions.	102
5.11	Ratio of successfully identified missing tags with different number of missing tags.	103
5.12	Identified and missed tags with different number of missing tags, SNR=17dB, and $K_{\text{Max}}=20$	103
5.13	Execution time comparison for the single reader case.	106
5.14	Execution time comparison for the multiple reader case.	106
5.15	Execution time comparison with the different missing tag ratio.	106

List of Tables

3.1	Key notations in PET	19
3.2	Node classification	27
3.3	Key notations in ZOE	33
3.4	Total time slots needed for PET	48
3.5	Total time slots needed to meet the estimation accuracy requirement with different ε ($\delta = 1\%$).	51
3.6	Total time slots needed to meet the estimation accuracy requirement with different δ ($\varepsilon = 5\%$).	51
4.1	Parameter setting ($\frac{\alpha}{\beta} = 1$, $P_{REQ} = 0.05$, and $ \mathbf{Y} = 5000000$)	80
5.1	The error rates of classifying present tags as missing.	107

Summary

Radio Frequency Identification (RFID) technology has recently attracted dramatic attentions from the research and industrial communities. A typical RFID system consists of RFID readers, RFID tags, and the middleware software to support proper working of the system. An RFID tag with its unique ID is a small microchip capable of harvesting energy from reader interrogation, lightweight computation, wireless communication, which transmits data in response to RFID reader queries. RFID tags are usually attached to real objects for explicitly labelling the objects and an RFID reader can thus identify and itemize these objects by verifying the attached RFID tags. Due to the simple structure, small size, and low manufacturing cost of RFID tags, it provides us an economic and competitive method for massive object management. RFID technology is currently becoming ubiquitously available for a variety of applications, including inventory management, transportation and logistics, object identification and tracking.

Having been widely adopted across many applications, RFID technology is fast growing as a major element of Internet of Things (IoT) for building future pervasive computing environment. However, opposed to the stringent needs for system efficiency and availability, due to the RFID resource constraints, the working efficiency of some key operations in RFID systems is still severely restricted, concerning RFID identification, estimation, searching, etc. For example, currently an RFID reader has to sequentially interrogate the individual tags and identify these tags one after another. When there are massive tags in the area (e.g., during baggage classification), the operational efficiency could be excessively low. Due to the severe wireless collisions and channel contention backoffs, the operational efficiency could get even lower.

This dissertation presents novel techniques and approaches to fundamentally improve the operational efficiency of the most basic and key operations in RFID systems. Such

key operations are widely adopted across many RFID systems in practice, and most applications may benefit from the efficiency improvement.

1) We design PET and ZOE protocols to estimate the number of RFID tags in the coverage of RFID readers without identifying the tags. Such approximation protocols trade accuracy for execution time and substantially improve operation efficiency with guaranteed estimation accuracy. PET improves state-of-the-art estimation efficiency from $O(\log n)$ to $O(\log \log n)$ while ZOE further improves the efficiency to $O(1)$.

2) We further design CATS, the first tag search protocol, to enable efficient tag search in large-scale RFID systems. CATS encodes the tag set using compact approximators to reduce communication overhead, and efficiently aggregates tag responses and extracts useful information without explicit channel arbitration.

3) We develop P-MTI, the first physical layer missing tag identification scheme, to efficiently monitor RFID tags and quickly identify the missing tags. P-MTI leverages the sparsity of missing tag events and recovers physical layer collisions through compressive sensing.

To investigate the feasibility of our proposed designs and fundamentally improve our understanding of RFID systems, we build the open source platform, named Open RFID Lab, which allows the full access and control over the RFID systems. The platform unveils several fundamental problems which motivates future researches. For future work, we develop data transfer protocol to efficiently read bulk data using commodity RFID readers from RFID based sensors. Our design allows resource-constrained RFID sensors to meet the stringent response deadlines of commodity communication protocol only with software extensions.

Chapter 1

Introduction

1.1 Background

Due to its practical importance, Radio Frequency Identification (RFID) technology [27] has attracted dramatic attentions. A typical RFID system consists of RFID readers, RFID tags, and the middleware software to support proper working of the system. An RFID tag harvests energy from reader interrogation, does lightweight computation, and transmits data in response to reader queries. RFID tags can be attached to objects for explicitly labelling, and RFID readers can thus identify these objects by verifying the attached RFID tags. Due to the simple structure, small size, and low manufacturing cost, RFID tags serve as an economic and competitive method for massive object management, e.g., inventory control [40, 66], object tracking [82], activity monitoring [45], authentication [68], localization [51], and etc.

Having been widely adopted across many applications, RFID technology is fast growing as a fundamental component of Internet of Things (IoT) for enabling ubiquitous and pervasive computing and applications. For instance, in warehouses with RFID-based monitoring systems, managers may perform quick estimation of the number of products left in stock for various purposes such as the detection of employee theft. Manufacturers know the IDs of tags that are attached to the suspected products may quickly find and recall the products for further inspection. RFID tags can also be used to label weapons

in military bases for monitoring and tracking. In the case that the weapons are missing, we can quickly identify the missing items with the RFID technology.

Opposed to the stringent needs for operational efficiency and system availability, constrained by limited resources, the working efficiency of some key RFID operations is still severely restricted, concerning RFID identification, cardinality estimation, tag search, etc. For example, currently an RFID reader has to sequentially interrogate each individual tag and carry out identification one after another. When a large number of tags in the area contend for the identification opportunities (e.g., in large-scale RFID enabled distribution centers), the operational efficiency could be excessively low. Considering the severe wireless collisions and contention backoffs, the operational efficiency could get even lower.

1.2 Review of Existing Research Works

Many prior works have been proposed to improve the operation efficiency of large-scale RFID systems. We briefly summarize the most related works as follows.

1.2.1 Collision Arbitration

The problem of estimating the number of RFID tags can be directly reduced to identifying the IDs of all RFID tags and itemizing them. Since a large number of RFID tags normally share the same physical communication channel, unordered concurrent communications may result in transmission collisions among tags. To address such a problem [41, 49, 50, 52, 74], many anti-collision time-domain methods have been proposed [16, 57, 62, 93], which can generally be classified into two categories: slotted ALOHA protocols [57, 62] and tree based protocols [16, 93].

In an ALOHA based protocol, an RFID tag replies immediately to reader's interrogation. If a collision occurs, the tag replies again after a random delay. The process

continues until all tags are successfully recognized by the reader. The ALOHA based protocols mitigate the negative impact of collisions with retransmissions but cannot remove collisions. With ALOHA based protocols, a specific tag may not be identified for an excessively long time.

Figure 1.1 depicts an example of Aloha-based RFID identification process. In this example, the reader first queries all the tags (Tag 1 to Tag 5) to contend 3 time slots in each frame. From the figure, we can see that in the first and the second time slots, 2 tags select each time slot which result in collisions. When collision occurs, the tags not identified will send their IDs in the following frames until all the tags are successfully identified by the reader. Since only one tag (i.e., Tag 4) selects the third time slot, the reader can successfully recognize the tag. When a large amount of RFID tags contend to transmit their IDs, the tag-tag collisions would substantially degrade the identification throughput of the RFID systems. Besides, in large-scale RFID system, a particular tag may contend a large number of frames to be eventually identified.

In a Tree based anti-collision protocol, an RFID reader interrogates tags and detects whether or not there are any collisions. Once collisions occur, the reader splits the tag set into two subsets by tag IDs and queries the subsets with fewer tags. The reader

Figure 1.1: Aloha-based RFID identification [42]

1.2.2 Cardinality Estimation

In small-scale RFID systems, RFID identification schemes can be directly applied to estimate the exact number of tags by itemizing each tag within the reader's interrogation region. Those solutions, however, become infeasible when the RFID system scales up. The processing time rapidly grows as the number of RFID tags increases. In particular, compared with tree based anti-collision protocols for RFID identification, the cardinality estimation protocols have been designed for quickly estimating the number of distinct RFID tags.

Kodialam and Nandagopal presented Unified Simple Estimator (USE) and Unified Probabilistic Estimator (UPE) in [37]. One drawback of those schemes is that they are vulnerable to replications when one tag is read by multiple readers, and the schemes require approximate magnitude of the tag number as a prior knowledge. In [38], an Enhanced Zero-Based (EZB) estimator was proposed which provides anonymous estimation and can estimate relatively larger number of tags. This approach, however, suffers the multiple counting problem. In many large-scale RFID systems, we need to deploy several readers to cover the large volume of tags. In such scenarios, some tags may be covered several RFID readers which would result in multiple counting problems. According to [53], the multiple counting problem may substantially degrade the cardinality estimation accuracy. Recently, Shahzad et al. [58] propose the Average Run based Tag (ART) protocol to improve estimation efficiency.

Some most recent approaches advance the estimation efficiency, and achieve $\mathcal{O}(\log n)$ processing efficiency to the number of RFID tags n . Qian et al. [53] propose LoF estimating algorithm, which leverages a geometric distribution hashing process to code tags with $\mathcal{O}(\log n)$ bits and by so estimates the tags with $\mathcal{O}(\log n)$ time slots. Using the duplicate-insensitive FM sketch [25], LoF [53] is able to address the multiple reader problem as well.

In [33], Han et al. present an $\mathcal{O}(\log n)$ estimator by quickly positioning the first non-empty slot with binary search algorithm. They further provide an adaptive shrinking algorithm to adjust the upper bound of the tag number so as to speed up the estimation process. Both approaches require that RFID tags react to the reader with on-chip computations for generating some kind of randomness (uniform or geometric distribution hashing). More recently, Chen et al. [18] provide fundamental insights and achieve near-optimal performance by two phase estimations.

1.2.3 RFID Tag search

Searching a particular set of RFID tags in a large-scale RFID system is of practical importance for those applications. For example in inventory management, there is usually a need of taking stock according to a list of items. Formally, the problem of tag searching can be defined as follows: given a set of wanted RFID tags, we wish to know which among them, if any, currently exist within the interrogation zone of RFID readers. While many active efforts have been put in studying RFID systems and significant advance in recent years has been achieved, surprisingly, the problem of searching with a large number of tags is yet under-investigated by the research community.

Our work CATS [89] presents the first tag search protocol for large-scale RFID systems. We first formulate the tag searching problem in large-scale RFID systems. To solve this problem, we propose utilizing compact approximators to efficiently aggregate a large volume of RFID tag information and exchange such information with a two-phase communication protocol. By estimating the intersection of two compact approximators, the proposed tag search protocol significantly reduces the execution time compared with all alternative solutions we can directly borrow from prior studies. Many follow-up works extend CATS in various scenarios and further improve the tag search efficiency. Unlike CATS which adopts two-phase filtering, Chen et al. [20] propose an iterative tag search

protocol and achieves higher execution efficiency compared with CATS. Sudaresan et al. [64] ensure the security and privacy of tags during tag search process. Zhang et al. [84] coordinate and schedule multiple RFID readers to mitigate inter-reader interferences in tag search.

1.2.4 Missing tag identification

Many works study the problem of tag monitoring and identify the missing tags [87]. Typically, such approaches iteratively identify both present and absent tags until all the tags are identified. As each tag has to transmit at least one-bit information to announce its presence, the execution time increases as the number of tags under monitoring [43]. In [66], Tan *et al.* present a missing tag monitoring protocol which can detect the missing tag events when the number of missing tag exceeds a user-defined threshold. In [43], Li *et al.* propose a missing tag identification protocol which can detect the missing tag event with certainty and identify the missing ones. Zhang *et al.* [87] significantly reduce the missing tag identification time by more efficiently scheduling and utilizing multiple readers.

Unlike the existing approaches which focus on upper layer information, we explore to effectively leverage the aggregated responses in the physical layer to fundamentally improve the monitoring efficiency. We exploit the lower layer information to counteract the interference from the present tags, thereby revealing the absence of missing tags. Moreover, prior missing tag identification schemes cannot leverage the sparsity of missing tag events, since those schemes solely focus on instant tag responses and ignore the response dynamics due to missing tags. Leveraging the sparsity of missing tag events, we efficiently identify the missing tags via compressive sensing.

1.3 Dissertation Organization

In the previous section, we briefly summarize numerous problems and solutions. Although many advances have been made, there are still great potentials for efficiency improvement. This dissertation mainly focuses on the following three important problems – cardinality estimation (Chapter 3), tag search (Chapter 4), and missing tag identification (Chapter 5) in large-scale RFID systems. The rest of this dissertation is organized as follows.

In Chapter 2, we first introduce the system model that we are dealing with. We explicitly state the performance metrics in this chapter.

In Chapter 3, we present our cardinality estimation protocols. We first formally describe the design requirement of cardinality estimation protocols. We then present the detailed protocol design with analytical results. We evaluate our protocol and present performance comparison results compared with prior cardinality estimation protocols.

In Chapter 4, we introduce the tag search problem in large-scale RFID systems. We mathematically formulate the tag search problem and present our two-phase tag search protocol. We then present performance evaluation results and compare with all alternative solutions that we can directly borrow from prior works.

In Chapter 5, we present a physical layer missing tag identification protocol. We further exploit the sparsity of missing tag events and substantially reduce communication overhead using compressive sensing based collision recovery. We carry out extensive performance evaluation and present comparison results against prior schemes.

Finally, Chapter 6 concludes this dissertation and briefly discusses the future research directions.

Chapter 2

System Models and Performance Metrics

2.1 System Models

RFID systems typically consist of three components: a number of RFID readers, a large number of RFID tags, and a backend server. The backend server coordinates the RFID readers and has powerful computation capability. The RFID readers, connected to the backend server via high speed networks, transmit commands of the backend server and later report responses back to the server. When multiple readers are synchronized, we may logically consider them as a whole and regard the multiple readers as a single coordinated logical reader.

There are three different types of RFID tags. Active tags are equipped with batteries and capable of doing complex computations. As the active tags actively generate radio signals, they can reach a longer distance. Semi-passive tags adopt internal batteries to power the tags while send data to readers by backscattering radio signals similar to passive tags. Passive tags harvest energy from reader interrogation and thus do not require internal power supplies. Passive tags communicate with readers by backscattering readers' incoming radio signals and therefore have very small energy footprints. Thanks to their small form factors and low manufacturing cost, passive tags are most widely adopted in large-scale RFID systems.

According to the EPC global Class-1 Gen-2 standard [3], passive RFID tags are required to implement a set of mandatory commands, e.g., the inventory commands. The standard also provides flexibility for manufacturers to implement customized commands. In conformance with the standard, one may extend the functionalities of RFID systems and include some value-added features. Such a flexible design motivates us to explore and design new features for RFID tags for practical needs. The passive tags typically have the lightweight computation modules (e.g., random number generator [3, 34], lightweight hash function [80], etc) to carry out cryptographic operations as well as random access to wireless channels. We assume that the random number generators meet the randomness criteria of the EPC standard, and the lightweight hash functions satisfy the uniform distribution requirements. Considering recent development of passive tags, we envision that similar functions will be more widely adopted in lightweight passive tags.

RFID systems may operate over a wide variety of frequency bands (e.g., 13.56/433/900MHz). We exclusively focus on the RFID systems operating in the 900MHz ultra-high frequency band. The underlying RFID system is assumed to work on a frame-slotted MAC model. Each tag waits for the reader's command in each round of communication, which is also known as the Reader Talks First mode. Each tag contains a unique 96-bit ID according to the standard setting in the EPC global Class-1 Gen-2 standard [3].

For each communication frame, a reader initiates communication first by sending commands and the parameters to tags, e.g., selecting the tags to participate in the frame and configuring the number of slots in the frame size. If no tag transmits signals in the slot, the slot is called an *empty slot*. If one or more tags transmit signals in one slot, the slot is called *non-empty slot* or *busy slot*. The transmitted message can be successfully decoded if a single tag responds, and messages get corrupted when multiple tags respond in the same slot. Nevertheless, if an RFID reader only needs to determine whether one slot is chosen by any tags, we can let each tag transmit a short response in the selected slot.

The tag-to-reader ($T \Rightarrow R$) transmission rate and the reader-to-tag ($R \Rightarrow T$) transmission rate are not necessarily symmetric depending on the physical implementation and the interrogation environment [13, 54, 75]. As specified in the EPC global Class-1 Gen-2 standard, the $T \Rightarrow R$ datarate is either 40 kbps to 640kbps (FM0 encoding format) or 5kbps to 320kbps (Miller-modulated subcarrier encoding format), while $R \Rightarrow T$ datarate is normally 26.7kbps to 128kbps [3].

2.2 Performance Metrics

We mainly consider the following performance metrics in the protocol design.

2.2.1 Execution Time

Many existing protocols study the tag identification problem [39, 49, 69, 94], which aims to arbitrate collisions and identify each of a large number of RFID tags as quickly as possible. As a matter of fact, RFID identification schemes can be directly borrowed to address the problem of cardinality estimation, tag search, and missing tag identification. For instance, when the number of RFID tags is small, we are able to collect all tag IDs within the interrogation zone and compute the intersection with a given set of wanted tags. Such solutions, however, bear the common communication collision problems among RFID tags, and in particular the long data collection process renders it inappropriate for applications with stringent delay requirement.

The primary goal of this work is to reduce execution time while ensuring accuracy requirements. The overall execution time involves the communication time between tags and readers for data transmission as well as the computation time at the tags and readers. The computation overhead involved in RFID protocols is typically very small, while the throughput of the RFID system of both uplink and downlink is very low. As a result, the communication time normally contributes most to the overall execution time.

In RFID approximation protocols, the execution time and the accuracy can be traded for one another in many applications. Generally, given more execution time, an approximation protocol (e.g., cardinality estimation protocol) is allowed to collect more information from tags and thereby able to obtain a more accurate result (e.g., estimation on the number of tags). On the other hand, if the accuracy requirement is less stringent and can be relaxed in practice, an approximation protocol can terminate the information collection process right after sufficient information is collected, thereby saving execution time.

Along with the accuracy requirement, another important factor that substantially influences the execution time is the protocol design. As we will demonstrate in the rest chapters, a carefully designed protocol is able to schedule tag responses more efficiently, aggregate more tag responses with the same execution time, and extract more useful information from the collected tag responses. As a result, the execution time of different protocols targeting at the same accuracy requirement might be dramatically different. As the accuracy requirement is normally determined by a specific application, we aim at carefully designing more efficient protocols so as to reduce the execution time.

2.2.2 Memory Overhead

Current commodity passive RFID tags have different sizes of non-volatile memory which can be used to store small amount of data. The memory banks are divided into 4 different sectors (namely Reserved, EPC, TID, and User) for different application usages. A 96-bit EPC code is globally unique. Protocol related data can be stored into the user memory, which is typically only 512-bit length. As such, RFID protocols should be memory-efficient on the tag side.

The storage and computation capability in typical RFID systems is highly asymmetric, i.e., the RFID readers and the backend server are much more powerful than

the resource-constrained RFID tags. RFID readers are normally connected to backend database via high speed wired links. Thus, the RFID readers can efficiently store and retrieve data, and the tag ID can be used as an index to retrieve the tag related data from backend server.

2.2.3 Computational Overhead

The computational capability is also very limited for RFID tags. The passive tags can only carry out lightweight computations, e.g., random number generation, lightweight hashing, etc. Thus, RFID protocols should mitigate the computation overhead on passive tags. In contrast, the RFID readers and the backend server adopt powerful CPUs. Thus, the RFID readers, resorting to more powerful backend servers if necessary, can easily handle the reader side computation tasks.

Chapter 3

Cardinality Estimation

In this chapter, we study the fundamental problem of estimating the number of tags in large-scale RFID systems. Fast estimating the cardinality of RFID tags, accordingly the number of labeled items, is of primary importance to many applications [71]. For instance, estimating the number of conference attendees with RFID badges allows us to track the movement and distribution of attendees in different conference rooms. A warehouse manager may benefit from a quick estimate of products in stock. A fast cardinality estimation scheme also serves as primary inputs to various RFID protocols. For instance, Aloha-based RFID identification protocols achieve near-optimal performance if the contention frame size can be set according to the number of contending tags [16, 36, 42]. Many missing tag monitoring protocols can also build upon accurate estimation results [43, 87].

To this end, probabilistic estimation approaches have been proposed to efficiently estimate the number of RFID tags. Some recent approaches achieve $\mathcal{O}(\log n)$ estimation efficiency to the number of RFID tags n [33, 53]. Nevertheless, existing protocols require many independent estimation rounds to achieve high accuracy. For instance, PET takes several seconds to achieve an accurate estimation. As a basic component which may be frequently invoked by many applications, an estimation protocol can easily become the bottleneck which limits the overall performance of large-scale RFID systems. Further

improving the time efficiency of each estimation round will significantly benefit the entire cardinality estimation process, meet the stringent time requirement of many realtime applications, and support larger scale RFID systems.

While pursuing the estimation efficiency at the optimum, we are also aiming at reducing the computation and memory overhead of resource-constrained RFID tags. Most existing probabilistic approaches require generating large volume of random numbers or alternatively pre-storing them at RFID tags, which lead to heavy computation and storage burden for RFID tags. We aim at shifting such overhead from resource-constrained RFID tags to powerful RFID readers. Besides, most existing works assume a reliable wireless channel between the RFID reader and tags, which is contradicting with the fact that the wireless channel is mostly error-prone. This chapter introduces cardinality estimation protocols to achieve fast RFID cardinality estimation with guaranteed accuracy requirement.

In the following, we first describe the cardinality estimation problem in Section 3.1. In Section 3.2, we introduce the Probabilistic Estimating Tree (PET) based cardinality estimation protocol which achieves $\mathcal{O}(\log \log n)$ time efficiency for each estimation round. In Section 3.3, we present the Zero-One Estimator (ZOE) based cardinality estimation protocol which further improves the time efficiency to $\mathcal{O}(1)$ for each estimation round. Section 3.4 presents evaluation results in comparison with prior cardinality estimation schemes. Section 3.5 briefly summarizes this chapter.

3.1 Problem Description

A large scale RFID system consists of one or more RFID readers and a vast number of RFID tags attached on physical objects. The goal of efficient RFID cardinality estimation is to obtain the approximate number of RFID tags in the region in a fast and accurate manner. Since the number of RFID tags can be extremely large in meeting large scale

application demands, like product amount estimation in shipping cargo containers, the processing approach needs to be designed scalable with the quantity of RFID tags while meeting pre-required accuracy and confidence level.

Similar with [33, 37], the accuracy requirement of estimation is defined by two parameters: a *confidence interval* ε and an *error probability* δ . Assume that an estimating result of the RFID tag number is $\hat{n}$ while the actual number is n . We consider the estimator accurate and precise if $\hat{n}$ satisfies $Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$. For instance, if the actual number of RFID tags in a region is 50000, and the accuracy requirement is specified as $\varepsilon = 5\%$ and $\delta = 1\%$, an accurate estimation approach is expected to output the estimated number within the interval $[47500, 52500]$ with more than 99% probability.

The underlying RFID system is assumed to work on a slotted MAC model. The time period is divided into small time slots. In each slot the reader first transmits continuous waves to energize the RFID tags as well as the command for tags' response and the tags then accordingly response in the second half of the time slot, which is denoted as Reader Talks First mode and has been widely accepted and used in many RFID systems [16, 33, 57, 93]. Some works also assume that the reader sends out the command at the very beginning of a frame of slots and RFID tags then consequently response at consecutive time slots. Such an assumption, however, requires that all RFID tags need to synchronize their frame of slots and will have their own energy resource to support their proper working for the entire frame.

One main requirement for a good RFID estimation approach is *efficiency*, which requires a short processing period, i.e., a small number of time slots for reader-tag communication to achieve the desired accuracy. We want to keep it a small order to the total number of RFID tags so as to support scalable RFID systems.

We also want to make the approach *lightweight* for RFID tags. There are two types of RFID tags, active ones and passive ones [61]. Active tags are capable of doing complex computations with self energy supply but are expensive and bulky. Passive tags

are instantly energized by the reader to carry out extremely limited computations but are cheap and easy to be massively used. We want to design the estimation approach lightweight so as to support a variety of applications using passive tags.

Besides, we hope to ensure the RFID estimation process *anonymous*. In some applications the tag ID carries identity information about the object it is associated with. Revealing such information to the public might lead to leak of private information. We need to design the RFID estimation process *robust* as well, effective with dynamic RFID tag set and multi-reader environment. In such environment, the tags are attached to mobile objects which may move across the coverage areas of multiple readers. Without careful design consideration, duplicate tag counts may lead to erroneous estimation.

3.2 PET: Probabilistic Estimating Tree based Cardinality Estimation

To estimate the total number of RFID tags in the region of interest we code those tags and divide them into small subgroups. We show that a probabilistic estimation tree well organizes the coding structure. By defining an arbitrary estimating path on PET, we get an actual querying scheme to estimate tags across different subgroups and approximate the total number of tags.

3.2.1 Basic Algorithm

PET is built on top of a binary tree as shown in Figure 3.1. We call a non-leaf node as *node*, and a leaf node simply as *leaf*. Each node has two branches, labeled as the *0-branch* and *1-branch*. Each top-down path from the root to a leaf thus gives a bit string with branch labels, and such a bit string codes that leaf. Different RFID tags are mapped to different leaves according to the codes.

For the sake of simplicity, we use an example to illustrate the design of PET. We assume that there are $n = 4$ RFID tags in the system. First, PET uses a uniform

Figure 3.1: Probabilistic Estimating Tree

random hash function $\mathcal{H}(tagID) \rightarrow [0, 2^H - 1]$ to generate a random code for each RFID tag. In this example, we set $H = 4$, and assume the 4 RFID tags are assigned random codes 0001, 0110, 1011, and 1110, respectively. As Figure 3.1 depicts, they are mapped to the four black leaves. For an arbitrary node in PET, if there are no black leaves within the subtree rooted at it, we say the subtree is *white*, and label the node *white*; otherwise the subtree is *black* and the node is labeled *black*.

We define the *height* of a node as the distance of the path between the node and a leaf in its subtree. The height of PET, denoted as H , in this example is 4.

To estimate the number of tags, an RFID reader generates a random bit string r , say 0011 in this example, indicating an estimating path from the PET root down to a leaf. For each node i , there are two subtrees along its two branches. We denote the subtree of node i along the estimating path r as $\mathcal{ST}_r^i$, and the other subtree of node i that does not follow r as $\mathcal{ST}_{\sim r}^i$. If node i is black, either $\mathcal{ST}_r^i$ or $\mathcal{ST}_{\sim r}^i$ is black, and if node i is white, both $\mathcal{ST}_r^i$ and $\mathcal{ST}_{\sim r}^i$ should be white. There is a particular black node i along the path whose $\mathcal{ST}_r^i$ is white (all other black nodes have black $\mathcal{ST}_r$) and such a node is the

Table 3.1: Key notations in PET

Symbols	Descriptions
n	Number of tags
$\hat{n}$	Estimated number of tags
p	Fraction of white leaves on the estimating tree
r	A random estimating path
H	Height of the probabilistic estimating tree
h	Height of the gray node
$\mathcal{H}()$	A uniform hash function
$node_r^i$	Node along r at the height of i
$\mathcal{ST}_r^i$	Subtree of node i along r
$\mathcal{ST}_{\sim r}^i$	Subtree of node i <i>NOT</i> along r

lowest black node along the path. We define it as a *gray node* (node A in Figure 3.1). The height h of the gray node, as our later analysis in Section 4.2, implies the number of RFID tags (black leaves in PET). Intuitively, we can imagine that the bigger fraction of white leaves there are in PET, the higher the gray node is. Thus we can use h to estimate the number of RFID tags. Table 3.1 summarizes the notations used in PET.

To find out h , the RFID reader initiates prefix query along the selected estimating path r , for the example in Figure 3.1, 0011. First the reader requests those tags whose random codes match prefix 0*** to respond. As the 4 tags (black leaves in PET) are assigned 0001, 0110, 1011, and 1110, respectively, the ones with 0001 and 0110 will response to the reader at the reply slot. Though the responses result in a collision slot, the reader detects the existence of responsive signal and is aware of the existence of 0*** prefix tags. The reader then goes ahead with the estimating path and requests the respond of 00** prefix tags, and the tag with 0001 responses. The reader continues such a process till there is no response from RFID tags. In this example when the reader queries 001* prefix, as no tags are with such a prefix, no response is made and the reader detects an idle slot. The reader can thus infer that there must be some black leaves

matching prefix $000*$ and find out the only gray node A (with path prefix $00**$) on the estimating path $r = 0011$ in PET. Consequently, the height h of A is 2. We show in the following section how the height of A is used to derive the approximate number of RFID tags.

In practice, we use a relatively large H , say 32, to build a large PET that is able to accommodate billions of black leaves. Querying along the 32-bit estimating path will lead us to h and thus the number of RFID tags. One thing worth noting is that, the PET structure is neither created nor maintained at the RFID reader. It is only a *conceptual data structure* that illustrates the organization of RFID tag groups as well as the reader query process over such tag groups. In a practical protocol, the reader simply queries the tags with a randomly selected estimating path and calculates h with tag responses.

3.2.2 Algorithm Analysis

Section 3.2.1 describes the intuitive insight why the height of the gray node implies the number of RFID tags. In the following, we present theoretical analysis for the estimation accuracy of PET algorithm. We can use h to estimate the number of black leaves in PET, accordingly the number of RFID tags. We denote the fraction of white leaves in PET as p and the fraction of black leaves is $1 - p$.

We start from two extreme cases, $p = 1$ and $p = 0$, respectively. $p = 1$ corresponds to that all the leaves in PET are white. In such a case we can infer that the number of tags is 0. $p = 0$ corresponds to that all the leaves in PET are black. In such a case, we can roughly estimate that the number n of tags hashed to the leaves of PET

$$n \geq 2^h. \tag{Eq. 3.1}$$

As a matter of fact, in the case of $p = 0$ the hashing process can be modeled as the famous coupon collector problem taking the hashing collision into consideration. We do

not try to deeply investigate such a case, as we can always choose a sufficiently big H such that we can make $p = (1 - \frac{1}{2^H})^n \approx 1$ for an arbitrary number of n ($H = 32$ can accommodate $n = 40,000,000$ with $p \approx 0.99$), leading to rare hashing collisions.

For the ease of analysis, we focus on the case where both n and 2^H are sufficiently large, and $p \approx 1$.

Let the random variable h be the height of the gray node i on a randomly selected estimating path r . Then we have

$$Pr(h) = Pr\{\mathcal{ST}_r^i = \text{white}, \mathcal{ST}_{\sim r}^i = \text{black}\}. \quad (\text{Eq. 3.2})$$

As the PET random codes of tags are independently assigned for the 2^{h-1} leaves in either $\mathcal{ST}_r^i$ or $\mathcal{ST}_{\sim r}^i$ with uniformly random distributed hash function. So we have

$$Pr\{\mathcal{ST}_r^i = \text{white}\} = p^{2^{h-1}}. \quad (\text{Eq. 3.3})$$

We can also obtain

$$Pr\{\mathcal{ST}_{\sim r}^i = \text{black}\} = 1 - p^{2^{h-1}}. \quad (\text{Eq. 3.4})$$

Hence we have

$$Pr(h) = p^{2^{h-1}} \cdot (1 - p^{2^{h-1}}). \quad (\text{Eq. 3.5})$$

As a result, the expectation of h is

$$E(h) = \sum_{k=1}^H k \cdot Pr(k) = -Hp^{2^H} + \sum_{k=0}^{H-1} p^{2^k}. \quad (\text{Eq. 3.6})$$

Since $p = (1 - \frac{1}{2^H})^n \approx e^{-n2^{-H}}$, we have

$$E(h) = -He^{-n} + \sum_{k=0}^{H-1} e^{-n2^{-k-1}} \approx \sum_{k=0}^{H-1} e^{-n2^{-k-1}}. \quad (\text{Eq. 3.7})$$

We appeal to Mellin transforms to derive the asymptotic closed form of the harmonic summation [26, 36] as follows.

$$E(h) \approx H - [\log_2 n + \frac{\gamma}{\ln 2} - \frac{1}{2} + \mathcal{P}(\log_2 n) + O(\frac{1}{\sqrt{n}})], \quad (\text{Eq. 3.8})$$

where γ is Euler's constant, $\mathcal{P}(x)$ is a periodic and continuous functions of x with period 1 and amplitude bounded by 10^{-5} . We omit the term $\mathcal{P}(\log_2 n) + O(\frac{1}{\sqrt{n}})$, and let $\phi = \frac{e^\gamma}{\sqrt{2}} = 1.25941 \dots$, then

$$E(h) \approx H - \log_2(\phi n). \quad (\text{Eq. 3.9})$$

Correspondingly, the standard deviation of h is

$$\sigma(h) = \sqrt{\text{Var}(h)} = \sqrt{\sum_{k=1}^H [k - E(h)]^2 Pr(k)}. \quad (\text{Eq. 3.10})$$

Similar to the method we derive $E(h)$, We appeal to Mellin transforms to approximate the standard deviation [26].

$$\sigma(h) \approx \sqrt{\frac{\pi^2}{6(\ln 2)^2} + \frac{1}{12}} = 1.87271 \dots \quad (\text{Eq. 3.11})$$

For detailed analysis, we refer the reader to [35], which presents mathematics of probabilistic counting theory and can be used to derive $E(h)$ and $\sigma(h)$.

According to Equation (Eq. 3.9), the observation of h , the height of gray node, can be used to estimate n , the number of tags.

However there may exist variance between the observed height h of gray node and its expectation $E(h)$. According to law of large numbers [31, 70], the average of the observation results from a large number of trials should be close to the expectation of the value, and will tend to become closer as more trials are performed.

We define the random process $\bar{h} = \frac{1}{m} \sum_{i=1}^m h_i$ as the average value of m independent observations, where h_i denotes the i^{th} observation of random variable h . Since both the estimating path of the reader and the PET codes at tags are randomly generated in

each round of estimation, the trials of $h_i (1 \leq i \leq m)$ are independent random processes.

Therefore, we have

$$E(\bar{h}) = \frac{1}{m} \sum_{i=1}^m E(h_i) = E(h), \quad (\text{Eq. 3.12})$$

$$\sigma(\bar{h}) = \sqrt{\text{Var}(\bar{h})} = \sqrt{\frac{\text{Var}(\sum_{i=1}^m h_i)}{m^2}} = \frac{\sigma(h)}{\sqrt{m}}. \quad (\text{Eq. 3.13})$$

According to Equations (Eq. 3.9) and (Eq. 3.12), we can estimate the number of tags as follows

$$\hat{n} = \phi^{-1} \times 2^{H-\bar{h}} = \phi^{-1} \times 2^{H-\frac{1}{m} \sum_{i=1}^m h_i}. \quad (\text{Eq. 3.14})$$

Then, according to Equation (Eq. 3.13), we will be able to reduce the variance and improve the estimating accuracy by performing m rounds of estimation.

Next, we show that given a particular accuracy requirement, e.g., $Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$, how many rounds of estimation PET should take to output satisfying results.

We define a random variable as follows

$$X = \frac{\bar{h} - \mu}{\sigma}. \quad (\text{Eq. 3.15})$$

By the central limit theorem [31, 70], we know the random variable X is asymptotically standard normal distribution, where $\mu = E(\bar{h}) = H - \log_2(\phi n)$, and $\sigma = \sigma(\bar{h}) = \frac{\sigma(h)}{\sqrt{m}}$. So, the cumulative distribution function of variable X is

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt. \quad (\text{Eq. 3.16})$$

Given a particular error probability δ , we can always find a constant range c which satisfies

$$1 - \delta = Pr\{-c \leq X \leq c\} = \text{erf}\left(\frac{c}{\sqrt{2}}\right), \quad (\text{Eq. 3.17})$$

where erf is the Gaussian error function [9]. On the other hand, we can rewrite the accuracy requirement as follows

$$\begin{aligned}
 & Pr\{|\hat{n} - n| \leq \varepsilon n\} \\
 &= Pr\{(1 - \varepsilon)n \leq \hat{n} \leq (1 + \varepsilon)n\} \\
 &= Pr\{(1 - \varepsilon)n \leq \phi^{-1} \times 2^{H-\bar{h}} \leq (1 + \varepsilon)n\} \\
 &= Pr\{H - \log_2 \phi(1 + \varepsilon)n \leq \bar{h} \leq H - \log_2 \phi(1 - \varepsilon)n\}.
 \end{aligned} \tag{Eq. 3.18}$$

Therefore, if we have the following condition

$$\begin{aligned}
 \frac{H - \log_2 \phi(1 + \varepsilon)n - \mu}{\sigma} &\leq -c, \\
 \frac{H - \log_2 \phi(1 - \varepsilon)n - \mu}{\sigma} &\geq c
 \end{aligned} \tag{Eq. 3.19}$$

we can guarantee the accuracy requirement $Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$. Combining Equations (Eq. 3.12), (Eq. 3.13) and (Eq. 3.19), we have

$$m \geq \max\left\{\left[\frac{-c\sigma(h)}{\log_2(1 - \varepsilon)}\right]^2, \left[\frac{c\sigma(h)}{\log_2(1 + \varepsilon)}\right]^2\right\}. \tag{Eq. 3.20}$$

Therefore, with such calculated m rounds of estimation, PET can guarantee the accuracy requirement of $Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$. Note that m is solely determined by the accuracy requirement ε and δ . Given the pre-required accuracy level, we can use a constant m that does not relate to the scale of RFID tags.

3.2.3 General Protocol

As elaborated in previous sections, the number of RFID tags is estimated based on the height of gray nodes in PET, i.e., the idle slots when the reader query with the selected estimating path. In this section we formally present the general estimation protocol, regulating both the reader behaviors and RFID tag behaviors.

Algorithm 1 PET algorithm for RFID readers

```

1:  $m \leftarrow \max\{\lceil \frac{-c\sigma(h)}{\log_2(1-\varepsilon)} \rceil^2, \lceil \frac{c\sigma(h)}{\log_2(1+\varepsilon)} \rceil^2\}$ 
2: for  $i \leftarrow 1$  to  $m$  do
3:   Select a random estimating path  $r$  and a random seed  $s$ ; Broadcast  $r$  and  $s$ 
4:   for  $j \leftarrow 1$  to 32 do
5:     Set high  $j$  bits of  $mask$ 
6:     Broadcast  $mask$ ; Listen in the following slot
7:     if there is no response in the slot then
8:        $h_i \leftarrow j - 1$ 
9:       break
10:    end if
11:  end for
12: end for
13: return  $\hat{n} \leftarrow \phi^{-1} \times 2^{H - \frac{1}{m} \sum_{i=1}^m h_i}$ 

```

Algorithm 2 PET algorithm for RFID tags

```

1: Receive the estimating path  $r$  and the random seed  $s$ 
2: Compute PET random code  $prc \leftarrow \mathcal{H}(s, tagID)$ 
3: while TRUE do
4:   Receive  $mask$ 
5:   if  $prc \wedge mask = r \wedge mask$  then
6:     \* Check whether high  $mask$  bits of  $prc$  is equal to that of  $r$  *
7:     Response immediately
8:   else
9:     Keep silent
10:  end if
11: end while

```

Algorithm 1 defines the behaviors of the RFID reader during each round of estimation. The RFID reader uses Reader Talk First mode to communicate with tags. At the first, the reader computes the number estimation rounds according to accuracy requirement (line 1). In each round the reader selects a random estimating path r and a random seed s , and broadcast them to the tags (line 3). The reader queries the tags with the additively increased path prefix in the following 32 time slots (line 4-11). In particular, at the j -th time slot the reader queries with the j -prefix of the selected estimating path r (line 5). At each time slot the reader broadcasts the prefix $mask$ for each tag's comparison (line 6).

Figure 3.2: Gray node on the estimating path

The reader listens to the channel and obtains the idle slot j when no response is received and stores the value of $j - 1$ (line 7-10). Finally the reader derives the approximate number of RFID tags according to Formula (Eq. 3.14) described in previous section (line 13).

Algorithm 2 defines the behaviors of RFID tags during each round of estimation. Compared with reader behaviors, the task of each tag is simpler. The tag receives the estimating path r as well as the random seed s , and generates the PET random code (line 1-2). The tag keeps receiving the additively increased *mask* for the path prefix and compares the path prefix with the prefix of its own generated random code. If they are the same the tag responds to the reader and otherwise the tag simply keeps silent (line 3-11). With such behaviors fewer RFID tags response as the querying process goes on and finally all tags will keep silent.

Similar with previous estimation approaches like FNEB [33] and LoF [53], the random code of each tag is generated with a random seed sent from the reader at the beginning of each estimation round. In such a case, we need to use active tags such that each tag is capable of executing the random hashing functions to generate the random code.

3.2.4 $\mathcal{O}(\log \log n)$ Algorithm

Given an estimating path r , we define the node along r with the height of i in PET as $node_i^r$. In the basic algorithm the reader queries the path prefix additively, which can

Table 3.2: Node classification

Node i	Remarks
<i>black node</i>	$\mathcal{ST}_r^i = \text{black}$
<i>white node</i>	$\mathcal{ST}_r^i = \text{white}$ AND $\mathcal{ST}_{\sim r}^i = \text{white}$
<i>gray node</i>	$\mathcal{ST}_r^i = \text{white}$ AND $\mathcal{ST}_{\sim r}^i = \text{black}$

be mapped to searching the gray node from the root down to the leaf on the estimating path r in PET.

Figure 3.2 gives an example to illustrate such a process. In Figure 3.2, the estimation process first probes $node_5^r$, and searches along r with additive path prefix query until the gray node $node_3^r$ is found. We need $\mathcal{O}(H)$ time slots for such a sequential path prefix query in searching the height h of the gray node in PET. When the number of RFID tags is large, the height of PET $H \approx \log_2 n$, and thus the basic estimation protocol has $\mathcal{O}(\log n)$ efficiency, which is comparable performance with the state-of-the-art approaches, such as FNEB [33] and LoF [53].

In this section, we improve the estimating efficiency by speeding up the process of path prefix query. As a matter of fact, the nodes along the estimating path r can be classified as shown in Table 3.2. For arbitrary $i > j$, we have

$$(\mathcal{ST}_r^j \cup \mathcal{ST}_{\sim r}^j) \subseteq \mathcal{ST}_r^i. \quad (\text{Eq. 3.21})$$

Thus we have the following relations between $node_i^r$ and $node_j^r$. For $i > j$,

- if $node_i^r$ is white or gray node, then $node_j^r$ is white node.
- if $node_j^r$ is black or gray node, then $node_i^r$ is black node.
- In either case, only one gray node exists in an estimating path.

Such an observation directly reveals the monotonic feature of the node colors along the estimating path, i.e., the black and white nodes are consecutively aligned and concatenated by the only gray node. In the example shown in Figure 3.2, $node_{0,1,2}^r$ are white

	0	1	2	3	4	5	6
Reader (000011)	0*****	00****	000***	0000**	00001*	STOP	
1 Tag (000001)							
1 Tag (000110)							
1 Tag (001011)							
1 Tag (001110)							
4 Tags (01****)							
8 Tags (1*****)							

3.3.a: Basic algorithm

	0	1	2	3
Reader (000011)	0000**	00001*	STOP	
1 Tag (000001)				
1 Tag (000110)				
1 Tag (001011)				
1 Tag (001110)				
4 Tags (01****)				
8 Tags (1*****)				

3.3.b: Binary search algorithm

Figure 3.3: Protocol execution process

nodes, $node_{4,5}^r$ are black nodes, and $node_3^r$ is the only gray node in between. Utilizing the monotonic feature of the estimating path, we can use a binary search algorithm to rapidly find the gray node in PET.

Mapped back to the estimation protocol, the reader no longer conducts a sequential path prefix query. Instead, the reader queries the path prefix with binary search. Algorithm 3 gives the improved protocol for the reader. The binary search algorithm is applied to query the path prefix (line 5-15). Instead of querying the additively increased path prefix, the $mid = \lceil (low + high)/2 \rceil$ bit prefix is chosen at each time slot (line 6).

Algorithm 3 PET algorithm for RFID readers with binary search

```

1:  $m \leftarrow \max\{\lceil \frac{-c\sigma(h)}{\log_2(1-\varepsilon)} \rceil^2, \lceil \frac{c\sigma(h)}{\log_2(1+\varepsilon)} \rceil^2\}$ 
2: for  $i \leftarrow 1$  to  $m$  do
3:    $low \leftarrow 1, high \leftarrow 32$ 
4:   Select a random estimating path  $r$ ; Broadcast  $r$ 
5:   while  $low < high$  do
6:      $mid \leftarrow \lceil (low + high)/2 \rceil$ 
7:     Set high  $mid$  bits of  $mask$ 
8:     Broadcast  $mask$ ; Listen in the following slot
9:     if there is no response in the slot then
10:       $high \leftarrow mid - 1$ 
11:     else
12:       $low \leftarrow mid$ 
13:     end if
14:   end while
15:    $h_i \leftarrow low$ 
16: end for
17: return  $\hat{n} \leftarrow \phi^{-1} \times 2^{H - \frac{1}{m} \sum_{i=1}^m h_i}$ 
    
```

The *high* end and *low* end of the prefix range are adjusted according to whether or not the response is received from the tags (line 9-13). Finally the *high* end and *low* end converge to the height of the gray node on the estimating path, and it is used to derive the approximate number of RFID tags (line 17).

With the above new protocol, the estimation efficiency is further improved. Only $\mathcal{O}(\log H)$ time slots are used to find the gray nodes in PET, which finally gives us $\mathcal{O}(\log \log n)$ estimation efficiency.

Figure 3.3 uses an example to demonstrate the performance gain of the improved protocol compared with the basic one. The example contains one RFID reader and 16 RFID tags. The height H of PET is chosen to be 6. Each tag generates a random code with 6 bits and the reader selects a 6-bit random estimating path $r = 000011$.

Figure 3.3.a depicts the estimation process with the basic protocol. At time slot 0, the reader queries the path prefix 0*****. As a result, the first 4 tags and the 4 tags with prefix 01**** response. The reader is aware of that and keeps proceeding at time

Algorithm 4 PET algorithm for RFID tags with binary search

```

1: PET random code  $prc$  = preloaded random number
2: Receive the estimating path  $r$ 
3: while TRUE do
4:   Receive  $mask$ 
5:   if  $prc \wedge mask = r \wedge mask$  then
6:     \* Check whether high  $mask$  bits of  $prc$  is equal to that of  $r$  *
7:     Response immediately
8:   else
9:     Keep silent
10:  end if
11: end while

```

slot 1 with a path prefix 00****. The query process continues until time slot 4, during which the reader queries the path prefix 00001* and identifies an idle slot. The entire process contains 5 time slots.

Figure 3.3.b depicts the estimation process with the improved protocol. At time slot 0, the reader directly queries the mid ($mid = \lceil (low + high)/2 \rceil = \lceil (1 + 6)/2 \rceil = 4$) path prefix 0000** and one tag responses. Receiving the response, the reader then raises the low end of the query range and at time slot 1 queries path prefix 00001*. At this time, there are no tag responses. The reader then lowers the high end of the query range and the estimation converges. The entire process contains only 2 time slots.

3.2.5 Discussion

Shifting the Computation Burden from RFID Tags. With the basic protocol each tag will generate a random PET number for mapping into the PET structure at each round of estimation. Generating the random code at each tag requires a fair amount of computation, which is infeasible for passive tags. An alternative is to preload a number of such random codes on the chip of each RFID tag during manufacturing. At each round of estimation the tag uses one of the preloaded random codes. As a trade-off, however, extra memory cost is required to store those random codes, which is proportional to the

number of estimation rounds m . As there will be generally many rounds of estimations for accurate results, the memory cost for preloaded PET random codes will be high.

With the above concern, we propose to shift such computation burden from RFID tags to the reader. We rely on the random estimating paths generated on the reader rather than refreshing the random codes at tags. Instead of using new random codes at different estimating rounds, a 32-bit PET random code is preloaded on each tag during manufacturing and used across all rounds of estimation. A group of off-the-shelf uniformly distributed hash functions can be used to generate the PET numbers, including Message-Digest algorithm 5 (MD5) and Secure Hash Algorithm (SHA-1). Note that MD5 generates a 128-bit hash value, but we can trivially convert them to shorter length, e.g., a 32-bit hash value, at will.

The reader generates a uniformly distributed random number as an estimating path at each round of estimation. By solely changing the 32-bit estimating path in the 2^{32} combinatorial space at each round of estimation, even the PET codes of tags keep unchanged, we are still expecting near independent estimating instances [9, 32]. Algorithm 4 defines the new behaviors of RFID tags. The tags use the preloaded random codes through all rounds of estimation (line 1), while in Algorithm 2, tags generate PET random codes at each round. In such a way, a tag only performs bitwise comparison on the PET code and path prefix during each round of estimation (line 3-11).

Processing Efficiency and Computational Overhead. According to our analysis, PET achieves the estimation efficiency of $\mathcal{O}(\log\log n)$ which significantly improves the state-of-the-art performance. In PET, RFID tags are only required to perform bitwise operations as well as to store a 32-bit PET random code. The random codes can be preloaded to the tags by RFID manufacturers. Such cost of implementing PET on RFID tags is negligible in terms of both computation and memory requirement in comparison with other probabilistic counting schemes, such as FNEB [33] and LoF [53]. In both

approaches, RFID tags need to generate random numbers at each round of estimation, or the random numbers are preloaded and stored at the RFID tags during manufacturing. In order to pursue a high estimation accuracy, both FNEB and LoF will conduct hundreds or even thousands of rounds of estimation. In such a case, the computation or memory cost would be significant and render it infeasible to work with passive tags.

Multiple Readers and Mobile Tags. Since the propagation range of both RFID readers and RFID tags, especially the passive tags, are limited, we usually deploy multiple readers to enhance the coverage for a large number of tags in the region of interest. In such a scenario, a back end controller coordinates multiple RFID readers. The controller generates an estimating path r , computes the *mask* for each round of estimation, and sends the estimating path r and *mask* to the readers. The readers accordingly query RFID tags with r and *mask*, and wait for their replies. The controller aggregates the reports from all readers and takes a slot as idle only when no tag response is reported from any readers. The controller repeats the estimating process until the height of gray node is determined. In the multiple reader scenario, the controller functions as if there were a single RFID reader. Even if a tag is located in the overlapped region and responses to multiple readers, its impact on the controller side is the same as a single response. Thus the PET protocol well handles the multiple reader scenario due to the duplicate-insensitive nature in tag responses. When RFID tags are attached to mobile objects and moving across interrogation regions of different readers, the responses of the same tag to different readers will converge at the controller as well. Due to the duplicate-insensitive nature, such a scenario is equivalent to that of the multiple readers and can be correctly handled by PET.

Anonymity. In some applications the tag ID carries private information about the associated object. Revealing such information to the public might lead to the leak of personal privacy. PET fully resists such privacy threats. In PET, during the estimating

Table 3.3: Key notations in ZOE

Symbols	Descriptions
n	Actual number of tags
$\hat{n}$	Estimated number of tags
m	Estimation rounds
$\mathcal{H}(\cdot)$	A uniform hash function
$\mathcal{H}_B(\cdot)$	Binary representation of $\mathcal{H}(\cdot)$
$\mathcal{R}(\cdot)$	Position of right-most zero in $\mathcal{H}_B(\cdot)$
θ	A threshold affecting the behavior of tags
λ	Load factor $n/2^\theta$
q	Channel error rate

process, each RFID tag does not participate with its own ID. Instead each tag responses to the reader's query according to a random code which is not directly bound to the tag ID. Even that, the tag does not reveal the random code directly to the public. At each time slot, a number of tags response to the reader, and their responses cumulate. The readers or any overhearing devices cannot distinguish the exact set of tags which response at a collision slot. As a result, the random PET numbers of RFID tags are not revealed to the readers as well.

3.3 ZOE: Zero-One Estimator based Cardinality Estimation

In this section, we propose a new design of our cardinality estimation protocol, which provides $\mathcal{O}(1)$ estimation efficiency for each estimation round. Table 3.3 summarizes the key notation used in ZOE.

3.3.1 Basic Algorithm

Suppose there exist n RFID tags, and each tag generates a random 0/1 bit with equal probability at each step. The tags stop generating random numbers until each of them

obtains a ‘0’(or ‘1’). Intuitively, the more tags there are, the longer the random number generating steps will last, and vice versa.

1) Tag: When probed by a reader in the cardinality estimation process, each tag independently computes a random number with a hash function $\mathcal{H}(\text{id}, s)$, where s is a random seed. For simplicity, we omit the notation of s for hash functions. We denote by $\mathcal{H}_B(\text{id})$ the binary representation of $\mathcal{H}(\text{id})$. We also denote by $\mathcal{R}(\text{id})$ the index of the right-most zero bit in $\mathcal{H}_B(\text{id})$ as follows

$$\mathcal{R}(\text{id}) = \min\{i | \mathcal{H}_B(\text{id})[i] = 0\} \quad (\text{Eq. 3.22})$$

To reduce the burden of generating random numbers in multiple estimation rounds, similar to the method in Section 3.2.5, the backend server can generate a uniformly distributed random number (denoted as R_B) in each round and broadcast it to all RFID tags. Receiving R_B , each tag computes $\mathcal{R}(\text{id}) = \min\{i | [\mathcal{H}_B(\text{id}) \oplus R_B][i] = 0\}$, where $\oplus$ denotes the bitwise XOR operation, and participates in the rest estimation rounds with the XORed random number.

If $\mathcal{R}(\text{id}_i) \geq \theta$, then the tag responds to the reader where θ is a threshold received from the reader; otherwise the tag keeps silent in the rest cardinality estimation process.

Let the random variable of $\mathcal{R}(\text{id}_i)$ be R_i , $1 \leq i \leq n$, then we have the probability

$$\Pr(R_i) = p^{R_i}(1 - p)$$

where p denotes the probability that a bit of $\mathcal{H}_B(\text{id})$ turns out to be ‘1’. Typically, we assume $p = 0.5$, i.e., the hash function is a uniform distribution hash function. Then we have $p = 1 - p = 0.5$, and

$$\Pr(R_i) = p^{R_i+1} = \frac{1}{2^{R_i+1}}$$

The probability that each tag keeps silent given a threshold θ is

$$\Pr(R_i < \theta) = \sum_{k=0}^{\theta-1} \Pr(k) = \sum_{k=0}^{\theta-1} \frac{1}{2^{k+1}} = 1 - \frac{1}{2^\theta}$$

On the other hand, each tag will respond to the reader with a probability of $1 - \Pr(R_i < \theta) = \frac{1}{2^\theta}$.

2) Reader: In the beginning, a reader transmits a random seed and the threshold θ to the tags, and waits for the reply from the tags. In the case that $\mathcal{R}(\text{id}_i) < \theta$, $i \in \{1, 2, \dots, n\}$, the reader observes no reply from the tags. Therefore, with the assumption of independent, identical distribution (i.i.d) for R_i , the probability that there is no reply from the tags (i.e., the channel is idle) is as follows.

$$\Pr(\text{idle}) = [\Pr(R_i < \theta)]^n = \left(1 - \frac{1}{2^\theta}\right)^n \approx e^{-n/2^\theta} = e^{-\lambda}$$

where the load factor $\lambda = \frac{n}{2^\theta}$, and $\lambda > 0$.

The probability that there is a reply (no matter a singleton reply from one tag or replies from multiple tags) is

$$\Pr(\text{reply}) = 1 - \Pr(\text{idle}) \approx 1 - e^{-n/2^\theta} = 1 - e^{-\lambda}$$

We define a random variable X which takes value 1 with probability $\Pr(\text{idle}) \approx e^{-n/2^\theta} = e^{-\lambda}$ and value 0 with probability $\Pr(\text{reply}) \approx 1 - e^{-n/2^\theta} = 1 - e^{-\lambda}$. Then we have

$$\Pr(X = 1) \approx e^{-\lambda}, \Pr(X = 0) \approx 1 - e^{-\lambda}$$

Obviously, the random variable X follows the Bernoulli distribution. Therefore, the expectation and the standard deviation of X are as follows.

$$E(X) = e^{-\lambda}, \sigma(X) = \sqrt{\text{Var}(X)} = \sqrt{e^{-\lambda}(1 - e^{-\lambda})}$$

We can calculate the maximum standard deviation of X is

$$\sigma(X)_{\max} = 0.5, \text{ when } e^{-\lambda} = 0.5$$

We define the random process $\bar{X} = \frac{1}{m} \sum_{i=1}^m X_i$ as the average of m independent observations, where X_i denotes the i^{th} observation of random variable X . We assume the trials of $X_i (1 \leq i \leq m)$ are i.i.d, then we have $E(\bar{X}) = E(X)$ and $\sigma(\bar{X}) = \frac{\sigma(X)}{\sqrt{m}}$.

According to the law of large numbers [31], we have

$$\bar{X} = E(\bar{X}) = E(X) = e^{-n/2^\theta} = e^{-\lambda}, m \rightarrow \infty \quad (\text{Eq. 3.23})$$

According to (Eq. 3.23), we can estimate the load factor as follows

$$\hat{\lambda} = -\ln \bar{X}$$

where $\hat{\lambda}$ denotes the estimated value of λ .

The observation of $\bar{X}$ can thus be used to estimate the tag cardinality $\hat{n}$ as follows

$$\hat{n} = -2^\theta \ln \bar{X} \quad (\text{Eq. 3.24})$$

Since the result may vary slightly because of the estimation variance, we seek a guaranteed cardinality estimation result, e.g., $\Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$. The estimation accuracy requirement can be represented as follows

$$\Pr\{|\hat{n} - n| \leq \varepsilon n\} = \Pr\{e^{-\lambda(1+\varepsilon)} \leq \bar{X} \leq e^{-\lambda(1-\varepsilon)}\}$$

We define a random variable $Y = \frac{\bar{X} - \mu}{\sigma}$, where $\mu = E(\bar{X}) = e^{-\lambda}$, and $\sigma = \sigma(\bar{X}) = \frac{\sigma(X)}{\sqrt{m}}$. By the central limit theorem [31], we know Y is asymptotically standard normal distribution.

Given a particular error probability δ , we can find a constant c that satisfies

$$1 - \delta = \Pr\{-c \leq Y \leq c\} = \text{erf}\left(\frac{c}{\sqrt{2}}\right)$$

where erf is the Gaussian error function [31]. Therefore, we can guarantee the accuracy requirement $\Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$ if we have the following conditions

$$\frac{e^{-\lambda(1+\varepsilon)} - e^{-\lambda}}{\sigma} \leq -c \text{ and } \frac{e^{-\lambda(1-\varepsilon)} - e^{-\lambda}}{\sigma} \geq c \quad (\text{Eq. 3.25})$$

Algorithm 5 ZOE algorithm for RFID readers

```

1:  $m \leftarrow \lceil \frac{c\sigma(X)_{\max}}{e^{-\lambda}(1-e^{-\varepsilon\lambda})} \rceil^2$ 
2: Initiate the estimation, broadcast  $\theta$ 
3: for  $i \leftarrow 1$  to  $m$  do
4:   Generate a random seed  $s$  and broadcast it
5:   if there is no response in the slot then
6:      $X_i \leftarrow 1$ 
7:   else
8:      $X_i \leftarrow 0$ 
9:   end if
10: end for
11:  $\bar{X} \leftarrow \frac{1}{m} \sum_{i=1}^m X_i$ 
12: return  $\hat{n} \leftarrow -2^\theta \ln \bar{X}$ 

```

Algorithm 6 ZOE algorithm for each RFID tag

```

1: Receive the threshold  $\theta$ 
2: while TRUE do
3:   Receive the random seed  $s$ ; Compute  $\mathcal{R}(\text{id})$ 
4:   if  $\mathcal{R}(\text{id}) \geq \theta$  then
5:     Respond immediately
6:   else
7:     Keep silent
8:   end if
9: end while

```

According to (Eq. 3.25), we have

$$\begin{aligned}
m &\geq \max\left\{\left[\frac{c\sigma(X)_{\max}}{e^{-\lambda}(1-e^{-\varepsilon\lambda})}\right]^2, \left[\frac{c\sigma(X)_{\max}}{e^{-\lambda}(e^{\varepsilon\lambda}-1)}\right]^2\right\} \\
&\geq \left[\frac{c\sigma(X)_{\max}}{e^{-\lambda}(1-e^{-\varepsilon\lambda})}\right]^2
\end{aligned} \tag{Eq. 3.26}$$

Therefore, with such m trials to estimate the tag cardinality, ZOE can guarantee the accuracy requirement of $Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$.

3.3.2 General Protocol

We estimate the tag cardinality according to the m rounds of independent experiments in which each tag transmits a 1-bit short messages or keeps silent. We formally present the

general estimation protocol by regulating the behaviors of readers and tags in Algorithm 5 and 6, respectively.

Algorithm 5 regulates the behavior of the RFID reader. The reader calculates the estimation rounds m according to (Eq. 3.26) given an accuracy requirement (line 1). The reader initiates the estimation process by energizing and synchronizing the tags, and broadcasts the threshold θ (line 2). In the following m rounds, the reader generates random seeds and broadcasts them (line 3-4), and then the reader records the tags' responses (line 5-9) at each round. The average $\bar{X}$ is thus calculated based on the m estimation rounds (line 11). Finally, the estimated tag cardinality is computed according to (Eq. 3.24) (line 12).

Energized by the RFID reader, each tag performs simple tasks as regulated in Algorithm 6. Each tag will receive a threshold θ (line 1). In each of the following m estimation rounds, when receiving a random seed s , the tag computes the random number $\mathcal{R}(\text{id})$ according to (Eq. 3.22). Then the tag responds to the reader according to $\mathcal{R}(\text{id})$ and the threshold θ . If $\mathcal{R}(\text{id}) \geq \theta$ the tag sends a response, and keeps silent otherwise (line 2-9).

Figure 3.4 further illustrates the estimation process. The subfigures depict the three cases where the total numbers of RFID tags are 512, 1024, and 2048, respectively. The left three subfigures depict distribution of the random numbers $\mathcal{R}(\text{id}_i)$, $1 \leq i \leq n$, in 100 independent estimation rounds. On the right side, we depict the random numbers with better granularity. The x-axis represents 100 independent estimation rounds and the y-axis represents the random numbers. Each grid (x, y) indicates a random number y generated at x^{th} estimation round, and the gray level intensity indicates the amount of tags which generate y at that estimation round. The skyline indicates the maximum value $\mathcal{R}(\text{id}_i)_{\max}$ among all tags along with the 100 estimation rounds. In ZOE each estimation round contains only one slot. A threshold θ is selected and applied along with

Figure 3.4: The distribution of random numbers. The left three subfigures depict the random number distribution in 100 estimation rounds. The right three subfigures depict the random number distribution in each estimation round. The skyline depicts $\mathcal{R}(\text{id}_i)_{\max}$ for 100 estimation rounds.

all 100 estimation slots. At each slot if the skyline is higher than the threshold θ the reader will receive response from some tags (we can take it as ‘0’), and if the skyline is lower than the threshold θ the reader will observe an idle slot (we can take it as ‘1’). The average of the 0/1 values is thus utilized to estimate the total number of tags (line 11-12, Algorithm 5).

3.3.3 Parameter Setting

Before we perform the estimation process, we need to set the threshold θ which directly influences the behaviors of the tags and the estimation efficiency. If θ is too big, the reader will consistently observe idle slots, i.e., $\bar{X} = 1$; if θ is too small, the reader will receive responsive messages from tags in almost every time slots, i.e., $\bar{X} = 0$, with high probability. In either situation, it consumes extra processing time to meet an accuracy requirement. As a matter of fact, if we look at the lower bound of the estimation round m measured in (Eq. 3.26), since $\lambda = n/2^\theta$, the lower bound depends on the tag cardinality which is not known in advance. For simplicity, we denote by $f(\lambda) = e^{-\lambda}(1 - e^{-\varepsilon\lambda}) \approx e^{-\lambda}\varepsilon\lambda$, the

Algorithm 7 Threshold setting algorithm

```

1:  $low \leftarrow 0, high \leftarrow 32$ 
2: while  $low < high$  do
3:    $mid \leftarrow (low + high)/2$ 
4:    $\theta \leftarrow mid, m \leftarrow 32$ , Compute  $\bar{X}$  with Algorithm 5
5:   if  $\bar{X} \geq \frac{e^{-2}+e^{-1}}{2}$  and  $\bar{X} \leq \frac{e^{-0.5}+e^{-1}}{2}$  then
6:      $\theta \leftarrow mid$ ; break;
7:   end if
8:   if  $\bar{X} > \frac{e^{-0.5}+e^{-1}}{2}$  then
9:      $high \leftarrow mid$ 
10:  else
11:     $low \leftarrow mid$ 
12:  end if
13: end while
14: return  $\theta$ 
    
```

denominator of $\frac{c\sigma(X)_{\max}}{e^{-\lambda}(1-e^{-\varepsilon\lambda})}$ in (Eq. 3.26). To reduce the number of estimation rounds, we maximize the denominator $f(\lambda)$ since the numerator $c\sigma(X)_{\max}$ can be regarded as a constant given the accuracy requirement. Figure 3.5 plots $f(\lambda)$ against λ for different ε .

We observe that $f(\lambda)$ reaches the maximum value at $\lambda \approx 1$. We compute the first order derivative of $f(\lambda) \approx e^{-\lambda}\varepsilon\lambda$ as follows,

$$\frac{df(\lambda)}{d\lambda} = \varepsilon e^{-\lambda}(1 - \lambda) \quad (\text{Eq. 3.27})$$

According to (Eq. 3.27), we find the first order derivative vanishes at $\lambda \approx 1$, and we have $\varepsilon e^{-\lambda} > 0$. Therefore, the minimum lower bound $m_{\min}$ is achieved at $\lambda \approx 1$. When $\lambda \approx 1$, we have $\theta \approx \log_2 n$ and $\Pr(\text{idle}) \approx e^{-1}$.

This observation motivates us to adaptively optimize the threshold θ according to the observation of a short sequence of the tags' responses such that $\bar{X}$ tends to e^{-1} . Intuitively, when the reader observes too many idle slots, i.e., $\bar{X} \gg e^{-1}$, it decrements the threshold to increase the probability that tags would send responsive messages; when the reader observes almost all the responsive slots, i.e., $\bar{X} \ll e^{-1}$, it increments the threshold to decrease the probability that tags would send responsive messages. In summary, we adaptively tune the threshold θ in the manner of negative feedback mechanism.

Figure 3.5: The denominator $f(\lambda)$ achieves maximum value at $\lambda \approx 1$.

Besides, the expected value of $\bar{X}$ is monotonically non-decreasing function against the threshold. Exploiting such monotonic feature allows us a fast convergence to an optimal threshold, e.g., we can reach a suitable θ that provides us $\bar{X} \approx e^{-1}$ with bisection search. Since we know the target average of $\bar{X}$, i.e., $e^{-1} \approx 0.37$, we can terminate the bisection search when the intermediate value of $\bar{X}$ during the process already becomes very close to $e^{-1} \approx 0.37$. In particular, we adaptively tune the threshold θ and terminate the bisection process when the intermediate value $\bar{X}(\theta_{\text{int}})$ reaches the interval $[(e^{-2} + e^{-1})/2, (e^{-0.5} + e^{-1})/2]$ and use θ_{int} as the threshold.

Algorithm 7 presents the threshold setting process using bisection search method. The threshold is set to be the average of *low* and *high*. The *low* end and *high* end are adjusted according to $\bar{X}$ (line 8-12). $\bar{X}$ is computed according using Algorithm 5 with the parameters $\theta = \text{mid}$, and $m = 32$ (line 4). Finally, the two ends converge, and the average is used as the threshold θ (line 2). Besides, when the intermediate value $\bar{X}$ becomes very close to the target average e^{-1} , the parameter setting process terminates and $\theta = \text{mid}$ is used as the optimal threshold (line 5-7).

Figure 3.6.a depicts the threshold setting process where we use the bisection search method to increase the convergence efficiency. In the experiment, the actual tag cardi-

nality is 1024. We repeat a small number of trials in each bisection step to derive $\bar{X}$. The experiment consists of 4 steps, and the number of trials is set to be 32. At the first step, we start with the threshold $\theta=16$. The reader observes 32 consecutive idle slots denoted by ‘1’s in Figure 3.6.a. At the second step, we adjust the parameter by decreasing θ in the bisection manner, and we repeat again 32 trials with the threshold $\theta=8$. The reader observes 32 straight busy slots denoted by ‘0’s in Figure 3.6.a. At the third step, the threshold is tuned to be $(8+16)/2 = 12$. In this case, the reader observes both ‘1’s and ‘0’s ($\bar{X} = 24/32 = 0.75 > e^{-1}$). At the final step, we run the estimation with $\theta = (8+12)/2 = 10$, in which case the reader also observes mixed ‘1’s and ‘0’s ($\bar{X} = 11/32$). Since $\bar{X} = 11/32 \approx 0.34$ at the final step is quite close to $e^{-1} \approx 0.37$, we set the threshold θ to be 10.

Figure 3.6.b plots the variance of X against $e^{-\lambda}$, which is the expectation of X , i.e., $E(X) = e^{-\lambda}$. We find that when $e^{-\lambda} \rightarrow 0$ or $e^{-\lambda} \rightarrow 1$, the variance $Var(X) \rightarrow 0$ indicating that while inaccurate to estimate the cardinality, $\bar{X}$ should be relatively stable around $E(X)$ to tell the scale of n . Therefore, it is safe to roughly and rapidly estimate the scale of tag cardinality and set the threshold accordingly with a small number of runs. This is the reason why we use a small sequence of 32 slots in calculating the optimal θ .

The parameter setting process involves at the worst case $\log_2 32 = 5$ bisection steps (32 rounds per step) to determine a threshold. This small amount of overhead after further reduction by early termination becomes almost negligible ($\sim 3\%$ of the total estimation overhead). Therefore, we can first tune the threshold θ and converges to an optimal parameter setting at a very small cost, and then using such an optimized θ in Algorithm 5 we can estimate the accurate cardinality with minimized number of estimation round m .

3.6.a:

3.6.b:

Figure 3.6: Parameter setting process: (a) Fast convergence to the optimal threshold value with bisection search method; (b) When $e^{-\lambda} \rightarrow 0$ or $e^{-\lambda} \rightarrow 1$, the variance is very small.

3.3.4 Reliable Estimation with Unreliable Channel

Most existing protocols study the cardinality estimation with reliable communication channel, while the realistic communication channel condition between tags and readers is much more complex and challenging. Wireless communication is error-prone depending on various channel conditions, including the transmission power, communication distance, antenna gain, radio frequency environment, software and hardware implementation imperfection, etc. We abstract such errors by communication errors regardless of the essential causes.

The cardinality estimation results of existing protocols fail to capture the actual value under unreliable channels. For instance, the false detection of responsive signal might tamper the monotonic feature of responsive signal along the estimation path in PET protocol [88], and substantially degrades estimation results. Similarly, the estimation results of LoF significantly rely on the channel condition, and the estimation accuracy decreases

dramatically with even small error rate. Relying on the structure of tag responses, it becomes very challenging and complicated for existing protocols to adjust the estimation and compute an accurate results even the communication error rate is available.

Before the estimation process, we can measure the communication error rate between tags and readers. For example, the reader requests the tags to transmit zero-one alternating messages and compares them with the received messages to compute the error rate, assuming that the error rate is time-invariant during the estimation process.

For simplicity, we assume the false negative rate and false positive rate are both q , i.e., it is equally likely to miss a tag response or to trigger a false detection with an idle slot. One may trivially extend it to the cases of asymmetric error rates. We propose Error Estimation and Adjustment (EEA) algorithm to adjust the estimation result according to communication error rates.

We denote by $\bar{X}_{\text{Error}}$ as the average value of m independent observations with the error rate q . Then we have

$$E(\bar{X}_{\text{Error}}) = E(\bar{X} - q(2\bar{X} - 1)) \quad (\text{Eq. 3.28})$$

According to (Eq. 3.28), we can compute $E(\bar{X})$ with the error rate q as follows,

$$E(\bar{X}) = \frac{E(\bar{X}_{\text{Error}}) - q}{1 - 2q}$$

We extend (Eq. 3.24) and estimate the tag cardinality as follows,

$$\hat{n}_{\text{Error}} = -2^\theta \ln\left(\frac{\bar{X}_{\text{Error}} - q}{1 - 2q}\right) \quad (\text{Eq. 3.29})$$

From (Eq. 3.29), we find that the ideal communication condition is a special case where $q = 0$. When $q = 0.5$, the communication channel becomes totally random, i.e., the outcome of the measurement totally depends on the channel noise. In such an extreme

case, theoretically none probabilistic estimation protocol can provide any useful information about the tag cardinality. We can successfully compensate the communication error given $0 < q < 0.5$.

Since the ZOE protocol relies solely on 0/1 responses from tags and does not assume any patterns of the tag responses at each estimation round, it is inherently simple yet more robust to unreliable communication channels.

3.4 Evaluation

We conduct comprehensive simulations under various situations to study the performance of PET and ZOE.

3.4.1 Simulation Setting

In the simulations, we first focus on the ideal communication channel, i.e., there is no transmission error between RFID tags and RFID readers, and the reader is capable of correctly detecting the responses from tags. After that, we evaluate the robustness and reliability of the estimation protocols with unreliable channel conditions. For all simulation instances, we repeat 300 runs and report the average if not explicitly specified otherwise.

The estimation accuracy is one of the most important metrics for an estimator. Consistent with existing works, we use the same accuracy metric as defined in LoF and PET.

$$\text{Accuracy} = E(\hat{n}/n)$$

where $\hat{n}$ denotes the estimation result and n denotes the actual number of tags. This parameter evaluates the estimation accuracy and bias. An ideal estimator is expected to return an estimation result close to the actual value. The closer it is to 1, the higher the estimation accuracy is.

Figure 3.7: Estimation Accuracy.

We use the standard deviation to measure the estimation precision.

$$\sigma = \sqrt{E[(\hat{n} - n)^2]}$$

where the operator $E[.]$ denotes the average of all runs. A high standard deviation means the estimation results spread out, whereas a low standard deviation means the estimation results concentrate. Therefore, we expect that an ideal estimator has the accuracy close to 1 with a low standard deviation towards 0.

Given an estimation accuracy requirement, we consider the estimating time that it takes to meet the requirement. Since the transmission rate varies depending on various factors (e.g., underlying physical implementations, protocol specifications, and channel conditions), same as the previous works we abstract the estimation time with the number of time slots that each protocol consumes.

Finally, another metric we consider is the computation and memory overhead at RFID tags. We measure the overhead by comparing the quantity of random numbers generated or stored at RFID tag side.

Figure 3.8: Standard deviation.

Figure 3.9: Normalized standard deviation.

3.4.2 PET Investigation

First, we demonstrate that PET provides tunable estimation accuracy at the cost of estimating time. As illustrated in Figure 3.7, one can improve the estimation accuracy of PET by performing more rounds of estimation. With 32 to 64 rounds of estimation, PET already maintains the accuracy very close to 1. Such a characteristic of PET enables modulating the estimating accuracy and efficiency according to the distinctive application needs. Figure 3.7 also suggests that the change of the number of tags has no significant impact on the estimation accuracy of PET, i.e., we can accurately estimate a wide range

Table 3.4: Total time slots needed for PET

Rounds	1	4	16	64	256	1024	4096
Time slots	5	20	80	320	1280	5120	20480

of RFID quantities without a priori of the tag number.

We are also interested in the standard deviation of the estimation results. The standard deviation and normalized standard deviation of PET estimation results are depicted in Figure 3.8 and 3.9, respectively. The figures suggest that by performing more rounds of estimation, the standard deviation of the estimation results can be reduced. Figure 3.9 further suggests that if we take a look at the normalized standard deviation, the number of tags will affect little. 64 rounds of estimation lead to nearly 0.2 normalized standard deviation. According to the simulation results, repeating a constant number of estimation rounds suffices to meet the requirement of estimation accuracy regardless of the number of tags.

The processing time of PET is also examined. In the simulation we use a fixed length of PET number $H = 32$. In such a setting, PET only takes 5 time slots to complete each round of estimation. Therefore the total number of time slots needed in m rounds of estimation can be listed in Table 3.4.

3.4.3 ZOE Investigation

We first evaluate the performance of the ZOE protocol with the assumption that there is no transmission error between tags and readers. We demonstrate that the ZOE protocol provides tunable estimation accuracy at the cost of processing time. Figure 3.10.a depicts different estimation accuracy while different number of estimation rounds are applied. The threshold θ is set at the optimal value for all cases. The figure suggests that one can improve the estimation accuracy by running additional rounds of estimation. By repeating 64 rounds of estimation, ZOE already achieves the accuracy very close to 1

3.10.a: Estimation accuracy

3.10.b: Standard deviation

Figure 3.10: Evaluation of ZOE protocol with different numbers of estimating rounds.

regardless of the actual tag cardinality, which suggests that the variation of tag cardinality under investigation has little impact on the estimation accuracy.

The standard deviation, indicating the precision of the estimator, is another important metric that we take into consideration. Figure 3.10.b illustrates the standard deviation. The figure suggests that one can reduce the standard deviation and thus improve the estimation accuracy by performing extra estimation rounds. With 64 estimation rounds, ZOE achieves standard deviation less than 20% of total RFID tag number, i.e., it achieves less than 0.2 of normalized standard deviation.

We also investigate the performance of ZOE with unreliable communication channels. We investigate the estimation accuracy for different communication error rates varying from 5% to 30%. The estimation round m is fixed at 64 in all experiments. In the cases that $m \neq 64$, the simulation results suggest similar results.

Figure 3.11 plots the estimation accuracy with and without Error Estimation and Adjustment (EEA) algorithm, respectively. As shown in Figure 3.11, the estimation

3.11.a: $q = 5\%$ 3.11.b: $q = 10\%$ 3.11.c: $q = 20\%$ 3.11.d: $q = 30\%$

Figure 3.11: Evaluation of estimation accuracy under different error rates.

accuracy of the basic ZOE protocol degrades substantially against the increase of the communication error, whereas the estimation accuracy with EEA remains reliable with various error rates. That is because EEA takes into the consideration the communication error and integrates such information into the estimation exploiting the simplicity of the ZOE protocol.

Table 3.5: Total time slots needed to meet the estimation accuracy requirement with different ε ($\delta = 1\%$).

ε	FNEB	LoF	PET	ZOE
20%	5366	3857	1681	533
15%	8524	6566	2862	792
10%	17067	14120	6154	1514
5%	60483	53885	23484	5305

Table 3.6: Total time slots needed to meet the estimation accuracy requirement with different δ ($\varepsilon = 5\%$).

δ	FNEB	LoF	PET	ZOE
15%	18759	16734	7406	1789
10%	24699	22004	9590	2270
5%	34465	30705	13382	3104
1%	60483	53885	23484	5305

3.4.4 Performance Comparison

We compare the performance of the proposed protocol with most recent estimation protocols. Since such protocols do not provide any mechanism to tolerate the communication errors, they fail to compute accurate estimate results with the practical channel condition. Therefore, we first focus on the performance comparison with the ideal channel, i.e., $q = 0$. We will evaluate the performance with unreliable channel later in the end.

We compare the estimating time slots given the same estimation accuracy requirement in $Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$. The actual tag cardinality is 50000. For the proposed ZOE protocol, the entire estimation process consists of $\log_2 32 = 5$ bisection steps (32 rounds for each step) to set a suitable threshold, and m time slots to improve the accuracy. We add the time slots for the two stages and present the sum for the comparison with other protocols.

The recent protocols shall perform multiple estimation rounds to meet the certain

3.12.a:

3.12.b:

Figure 3.12: Performance comparison: (a) Protocol performance with different confidence interval ε , and the same error probability $\delta = 1\%$; (b) Protocol performance with different error probability δ and the same confidence interval $\varepsilon = 5\%$.

estimation accuracy. We first keep the error probability $\delta = 1\%$ fixed and vary the confidence interval from 5% to 20%. Table 3.5 summarizes the total time slots needed by each protocol. Figure 3.12.a gives a more detailed comparison with a better granularity on ε . Then we keep the confidence interval $\varepsilon = 5\%$ fixed and vary the error probability δ from 1% to 15%, and the simulation results are presented in Table 3.6 and Figure 3.12.b. According to the simulation results, ZOE only consumes about 31% processing time of PET to provide the same estimation accuracy, which translates to more than 3x performance improvement in terms of time efficiency and even more compared with LoF and FNEB. The simulation results also imply that provided the same amount of processing time, the estimation accuracy of ZOE should be more accurate.

Figure 3.13.a presents the simulated performance against the analytical performance of the proposed ZOE protocol given the accuracy requirement of $\varepsilon = 5\%$ and $\delta = 1\%$ with the actual tag cardinality of 50000. In Figure 3.13.a, we observe that the simulation

3.13.a: Theoretical performance vs simulation results of ZOE

3.13.b: Cumulative distribution comparison of different protocols

Figure 3.13: Cumulative distribution of estimation results.

results match the analytical performance. Almost all the estimated values fall into the 5% confidence interval $[47500, 52500]$ with small tails very close to the confidence range.

We provide PET, FNEB and LoF the same amount of time slots to estimate the actual tag cardinality of 50000, and present the cumulative distributions in Figure 3.13.b. According to the simulation results, we find that the estimated results of ZOE are much more concentrated about the actual cardinality. Besides, the number of outliers is much smaller than those of PET, FNEB and LoF. In particular, with the same processing time

Figure 3.14: Memory consumption in storing the random numbers: (a) with different confidence interval ε , and the same error probability $\delta = 1\%$; (b) with different error probability δ , and the same confidence interval $\varepsilon = 5\%$.

that about 99% estimated values fall into the confidence interval $[47500, 52500]$ in ZOE, other protocols only guarantees less than 80% results within such an interval.

We also compare the computation and memory overhead at RFID tags. Most other protocols need to generate huge amount of random numbers. Constrained by the computation capacity, those protocols propose to store such random numbers at tags. We examine the memory overhead and compare ZOE with recent protocols in Figure 3.14. We fix the error probability $\delta = 1\%$ and vary the confidence interval ε from 5% to 20% in Figure 3.14.a. We vary the error probability δ from 1% to 15% with fixed confidence interval $\varepsilon = 5\%$ in Figure 3.14.b. According to the statistics, we observe that ZOE and PET consume constant small storage, and outperforms FNEB and LoF which require much larger memory cost.

Till now we focus on the performance comparison with ideal channel. When the bit error rate $q \neq 0$, the bit errors alter the pattern of tag responses, and it becomes

Figure 3.15: Accuracy comparison of different protocols under varying error rates.

very complicated and challenging for FNEB, LoF, and PET to make accurate estimation even if the communication error rate is known. For example, FNEB uses the bisection search method to locate the first non-empty slot. But when the monotonic feature of tag responses is altered, the estimation would dramatically influence the estimation results. PET also suffers a similar problem for the same reason. Therefore, we compare the core algorithms of FNEB, LoF, and PET with the ZOE protocol with the erroneous channel.

We examine the estimation accuracy of ZOE compared with that of FNEB, LoF, and PET in Figure 3.15 with different error rates. We vary the error rate from 5% to 30%, and the actual tag cardinality is 50000 in this experiment. According to Figure 3.15, we find that the estimation accuracies of LoF and PET are significantly biased from the actual value, i.e., accuracy is much smaller than 1. Though FNEB is more robust than LoF and PET, it still fails to provide an unbiased and accurate estimation. In contrast, ZOE can adjust the estimation results with the Error Estimation and Adjustment (EEA) algorithm described in Section 3.3.4. As a result, we see that ZOE with EEA can resist the various error rates and provide accurate estimation results even when the error rate reaches 30%.

3.5 Summary

In this chapter, we first propose the Probabilistic Estimating Tree (PET) based cardinality estimation scheme which outperforms prior scheme, such as LoF and FNEB. We further propose a cardinality estimation protocol based on Zero-One Estimator (ZOE) which improves the estimation time efficiency in meeting arbitrary accuracy requirement. ZOE only requires one-bit response from each of the RFID tags while prior works require several time slots per estimation round, which significantly saves the estimation time while excluding resource constrained tags from heavy computation. The proposed ZOE protocol also considers noisy communication channel and improves the robustness of the cardinality estimation with more practical communication channels with transmission errors. The intensive simulation results demonstrate that ZOE outperforms the most recent cardinality estimation protocols with both reliable and unreliable communication channels.

Chapter 4

RFID Tag Search

Searching a particular set of RFID tags in a large-scale RFID system is of practical importance for those applications. For example in inventory management, there is usually a need of taking stock according to a list of items. A manufacturer knows the IDs of tags that are attached to the suspected products wants to recall them for further inspection. In such practical scenarios, users may want to quickly find out which tags are in the interrogation area. Formally, the problem of tag searching can be defined as follows: given a set of wanted RFID tags, we wish to know which among them, if any, currently exist within the interrogation zone of RFID readers (as Figure 4.1 depicts, we want to know which tags within set $\mathbf{X}$ exist in tag set $\mathbf{Y}$ in the zone). While many active efforts have been put in studying RFID systems and significant advance in recent years has been achieved, surprisingly, the problem of searching with a large number of tags is yet under-investigated by the research community.

Many existing works concentrate on the RFID tag identification problem [39, 69, 94], which aims to identify each of a large number of RFID tags as quickly as possible. The major research issue in tag identification problem is to design efficient algorithms that resolve the tag collision problem since multiple RFID tags may contend the spatially and temporally shared communication channel. As a matter of fact, RFID identification schemes can be directly borrowed to address the tag search problem in small-scale

Figure 4.1: RFID Tag Searching

RFID systems, i.e., when the number of RFID tags is small, we are able to collect all tag IDs within the interrogation zone and compute the intersection with a given set of wanted tags. Such solutions, however, bear the common communication collision problems among RFID tags, and in particular the long data collection process renders it inappropriate for applications with stringent delay requirement. Due to tag-tag collisions, in slotted Aloha-based identification protocols, each tag attempts 2.72 times on average to successfully deliver its ID to the reader even with ideal frame sizing [39]. As a result the efficiency of identification protocol is very low. According to the RFID standard ISO-18000, the average identification throughput is about 100 tags per second [78]. In large-scale RFID systems, the number of RFID tags could be huge, e.g., there may store millions of products in a supermarket inventory. Collecting such a vast volume of RFID tag IDs often fails to meet a stringent delay requirement.

By painstakingly collecting high volume of RFID tag IDs, such identification based approaches are far from fast and efficient tag searching. The current inadequacy motivates our work to design an efficient and elegant tag searching protocol without the expensive tag ID identification phase so as to meet real-time application demands. We observe that many real applications can tolerate and are robust to certain error rate within a

sufficiently small range. For example, the inventory manager may want to search for a subset of products with manufacturing flaws and such a practical application can tolerate small false positive errors, e.g., provided that all flawed products (say 100) can be found, it is acceptable even if a small number of good products (say 1 or 2) are wrongly identified as flawed. It is thus desirable to leverage such characteristics and seek dramatic efficiency improvement as long as the error rate can be preserved within the tolerable range.

In order to reduce the overhead of tag searching in large-scale RFID systems, we propose utilizing compact approximators to efficiently aggregate a large volume of RFID tag information, and transmit the compact approximators instead of directly broadcasting or collecting tag IDs. By transmitting the succinctly encoded sets instead of raw data sets between RFID tags and readers, the communication cost involved in the tag searching protocol is significantly reduced. We choose the Bloom filter as a vehicle to illustrate the principle of our idea. Many variants of the Bloom filter can be used as well for the specific design purpose. Nevertheless due to the large number of RFID tags and the heavy channel competition how to optimize approximator utility and reduce the communication cost remains non-trivial. Constrained by computation and storage resource of RFID tags, achieving the optimal protocol performance brings challenging problems as well.

In this chapter, we propose a time-efficient protocol for tag searching in large-scale RFID systems. The objective of the protocol is to 1) examine whether any wanted tags exist in the reader interrogation region, and 2) if there are any, report the IDs of those tags. We also develop a light-weight tag cardinality range estimation algorithm for input of the tag searching protocol. We seek to reduce the communication cost and time delay involved in the tag searching process with a two-phase compact approximator exchange approach. The wanted tag set is first aggregated and broadcasted to the RFID tags and the tag feedbacks are once again aggregated and revealed to the readers. With

optimal parameter settings for such a two-phase approximator exchange process, we are able to significantly reduce the transmission overhead according to particular accuracy requirement.

The contributions of this chapter can be summarized as follows. We formally introduce the tag searching problem in large-scale RFID systems. To the best of our knowledge, this is the first attempt aiming to address such a practically important and yet under-investigated problem. We propose a baseline protocol, and then on top of it propose a much more efficient two-phase compact approximator based tag searching protocol which significantly improves the communication and time efficiency in tag searching with guaranteed error rate. We note that the proposed two-phase scheme is particularly designed for the scenarios of $|\mathbf{X}| < |\mathbf{Y}|$, which can be generalized to $|\mathbf{X}| > |\mathbf{Y}|$ by iteratively adapting the frame lengths as in [20]. The extensive simulation results show that compared with the baseline protocol, the proposed tag searching protocol reduces the communication time by around 79%, and reduces even more compared with the tag identification based protocols.

In the following, we first describe the tag search problem in Section 4.1. Section 4.2 presents the detailed design of the Compact Approximator based Tag Search (CATS) protocol. Section 4.3 presents evaluation results in comparison with alternative approaches to search tags. Section 4.4 briefly summarizes this chapter.

4.1 Problem Description

As Figure 4.1 depicts, we consider a large-scale RFID system with $\mathbf{Y} = \{y_1, y_2, \dots\}$ representing all the RFID tags in the interrogation zone covered by readers, and $\mathbf{X} = \{x_1, x_2, \dots\}$ representing the wanted RFID tags we are interested in. **The problem of searching for RFID tags is to find the intersection $\mathbf{X} \cap \mathbf{Y}$.** Each tag $x \in \mathbf{X}$ is called a *wanted* tag. The wanted tags $\mathbf{X}$ are not necessarily in the interrogation area,

i.e., the intersection can be an empty set. In other instances, all the wanted tags may be covered by readers, i.e., $\mathbf{X} \subseteq \mathbf{Y}$. We do not restrict the spatial distribution of $\mathbf{Y}$. We denote by $|\cdot|$ the cardinality of the set. For example, as depicted in Figure 4.1, many tags are covered by 4 readers in the interrogation zone $\mathbf{Y}$. There are $|\mathbf{X}| = 24$ wanted tags, among which 16 tags (black dots) are indeed in the interrogation zone and the other 8 tags (white dots) are not. For a tag $y \in \mathbf{Y}$, with a priori knowledge during the searching process, if $\Pr\{y \in \mathbf{X}\} \neq 0$, the tag is considered as a *candidate* tag, and if $\Pr\{y \in \mathbf{X}\} = 0$, the tag becomes a *non-candidate* tag. The candidate tags form a subset $\mathbf{Y}_C$, and the non-candidate tags form the other subset $\mathbf{Y}_{\sim C}$. Obviously, for any time instance the subset of candidate tags and the subset of non-candidate tags are by definition mutually exclusive and complementary, i.e., $\mathbf{Y}_C \cap \mathbf{Y}_{\sim C} = \emptyset$ and $\mathbf{Y}_C \cup \mathbf{Y}_{\sim C} = \mathbf{Y}$. As we accumulate more knowledge during the tag searching process, some candidate tags may eventually become non-candidate tags.

In practical situations, many applications tolerate a small false rate, as long as the false rate is sufficiently small. For most applications, in order to make the searching protocol scalable, it seems natural to trade the accuracy within a tolerable range for significant efficiency improvement. Given a false rate requirement, an ideal tag searching protocol is expected to compute the intersection $\mathbf{X} \cap \mathbf{Y}$ at minimum time and communication cost with a false rate smaller than the requirement. As a matter of fact, since the low-cost light-weight RFID tags are inherently error prone, the impractical pursuit of perfect $\mathbf{X} \cap \mathbf{Y}$ calculation may lead to excessively high overhead in realistic implementations. In spite of the search accuracy and efficiency, in many applications the anonymity of the RFID tags should be protected for privacy issues. Revealing identity information to the public might raise security and privacy concerns.

To meet above constraints and requirements, the goal of this work is to propose a fast RFID tag searching protocol that is able to efficiently calculate the intersection

$\mathbf{X} \cap \mathbf{Y}$ within a guaranteed false rate. To this end, the protocol should avoid the explicit identification of tag IDs, prevent heavy communication collisions between massive RFID tags, and reduce the transmitted bits of information.

4.2 CATS: Compact Approximator based Tag Search Protocol

In this section, we introduce several approaches to develop efficient tag searching protocols for large-scale RFID systems. We first give a baseline protocol which prevents collecting all tag IDs from the interrogation zone. We further propose a two-phase compact approximator based tag searching protocol to significantly reduce the transmission cost and achieve extra high efficiency. In addition, we propose a light-weight cardinality range estimation approach for providing rough cardinality as input to the tag searching protocol.

4.2.1 Baseline protocol

According to the aforementioned tag identification algorithms, we know that directly collecting tag IDs from all tags in set $\mathbf{Y}$ is highly inefficient because the transmission amount is high and the tag-tag collisions are heavy. The quantity of the tags in the interrogation zone can scale up to millions in some large-scale systems and identifying such amount of tags is impractical. Since we only concern the set of wanted tags $\mathbf{X}$ rather than all the tags in set $\mathbf{Y}$, an obvious optimization is that, instead of identifying the tag set $\mathbf{Y}$ in interrogation zone we let the RFID reader broadcast the IDs of tags in set $\mathbf{X}$ one by one and wait for the responses from tags in set $\mathbf{Y}$. Upon receiving the broadcasted IDs, each tag compares with its own ID, and reply immediately by sending a short response if the broadcasted ID matches its own ID or keep silent otherwise. For each ID, we can reserve a one-bit slot for identifying tags' binary response, i.e., '1' when tag

response is received or ‘0’ otherwise. Instead of ‘polling’ IDs from set $\mathbf{Y}$, by ‘pushing’ the tag IDs from set $\mathbf{X}$, the baseline protocol avoids collecting a large amount of tag IDs as well as the heavy tag-tag collisions during the process. Since tag IDs are 96 bits long and we need a binary response from each tag, the expected execution time of the baseline protocol is approximately

$$T_{Base} = |\mathbf{X}| \times (96 \times \alpha \times T_b + \beta \times T_b), \quad (\text{Eq. 4.1})$$

where $\alpha \times T_b$ denotes the per bit transmission time from readers to tags, and $\beta \times T_b$ denotes the per bit transmission time from tags to readers.

In practical large-scale RFID systems, the number of tags can scale to millions, while the number of wanted tags is usually much smaller, i.e., $|\mathbf{X}| \ll |\mathbf{Y}|$. Therefore, the baseline protocol significantly reduces the searching time. Even if $|\mathbf{X}|$ approaches $|\mathbf{Y}|$, since the baseline protocol inherently avoids tag-tag collisions, it still significantly outperforms the identification protocols in large-scale tag searching applications.

Although the baseline protocol demonstrates a promising performance improvement, it suffers several limitations which motivate our study for a more efficient and secure tag searching protocol. First, though $|\mathbf{X}|$ is probably much smaller than $|\mathbf{Y}|$, $|\mathbf{X}|$ can still be a large number for large-scale RFID systems where there are many wanted items. As the searching time increases proportionally with $|\mathbf{X}|$, the baseline protocol may still fail to meet stringent delay requirement. It is yet significant to further improve the searching efficiency. Second, the baseline protocol requires that all tags participate during the entire tag searching process, which results in unnecessary power consumptions on both reader and tag ends. Third, in the baseline protocol unique tag IDs are explicitly transmitted and acknowledged on the air which leads to potential privacy leaks.

4.2.2 Two-phase Tag Search Protocol

Based on the baseline protocol, we propose a two-phase compact approximator based tag searching protocol to further reduce transmission overhead and searching time. In particular, we transmit the compact approximators instead of explicit tag IDs to reduce transmission amount otherwise involved in broadcasting or collecting those IDs. One well known compact approximator, Bloom filter, is capable of encoding itemized information in a hashed Boolean vector. We use the Bloom filter as a representative approximator to carry aggregated tag ID information. One can choose a variety of compact approximators to aggregate the tag sets. [10] surveys various existing techniques using such approximators.

The compact approximator based tag searching protocol consists of two phases. In the first phase, we significantly reduce the number of candidate tags in $\mathbf{Y}_C$ by filtering the candidate tags using a Bloom filter produced with the wanted tags in $\mathbf{X}$. In the second phase, we estimate the intersection $\mathbf{X} \cap \mathbf{Y}$ by filtering the wanted tags with a virtual Bloom filter produced with the responses from the candidate tags.

4.2.2.1 Preliminary

Compact approximators are capable of succinctly representing a large volume of information. By transmitting the concisely encoded sets instead of the raw data sets, the communication cost involved can be significantly reduced. We choose the Bloom filter as a vehicle to demonstrate how the compact approximator can be used to develop efficient tag searching protocol.

The Bloom filter representing a set $\mathbf{A} = \{a_1, a_2, \dots, a_M\}$ of M elements comprises of a Boolean vector of L bits and K independent hash functions $h_i(\cdot), 1 \leq i \leq K$. Each hash function $h_i(\cdot)$ maps an element $a \in \mathbf{A} \subseteq \Omega$ to a bit $h_i(a) \in \{1, 2, \dots, L\}$, where Ω represents the universal set. Initially, the L -bit array is set to '0'. For each element $a \in \mathbf{A}$,

the bits $h_i(a)$ are set to ‘1’, $1 \leq i \leq K$. In order to determine whether a given element $b \in \Omega$ belongs to $\mathbf{A}$, we compute K hash functions $h_i(b)$, $1 \leq i \leq K$. If all $h_i(b)$ bits on the vector have been set to ‘1’, we assert that $b \in \mathbf{A}$, and otherwise $b \notin \mathbf{A}$. Generally, membership testing using Bloom filter has no false negatives [10]. Nevertheless, it may produce false positives, i.e., an element might be misclassified to be within the set while it is not. Many practical applications tolerate such false positives, as long as the rate is sufficiently small.

Given the assumption that the K hash functions are independently and identically distributed (i.i.d.), and can uniformly map M elements into the range $\{1, 2, \dots, L\}$, the probability of a false positive can be calculated in a straightforward way. Let P denote the probability that a particular bit remains ‘0’. Then $P = (1 - \frac{1}{L})^{M \times K} \approx e^{-M \times K/L}$ as $M \times K$ bits are independently selected by hash functions with probability $\frac{1}{L}$ for each bit. Therefore, the probability that a specific bit is set to ‘1’ is $1 - P$. A false positive occurs when for an element $b \notin \mathbf{A}$, all $h_i(b)$ bits ($1 \leq i \leq K$) are set to ‘1’ due to the hash results of other elements. We denote the false positive rate as P_{FP} , and we have

$$P_{FP} = (1 - P)^K \approx (1 - e^{-M \times K/L})^K \quad (\text{Eq. 4.2})$$

It is easy to see that the minimum value of $P_{FP} = 0.6185^{\frac{L}{M}}$ when the number of hash functions is $K = \frac{L}{M} \times \ln 2$ given the number of wanted tags M and the size of Bloom filter L .

In practical applications, the number of wanted tags M is usually known a priori. On the other hand, the size of Bloom filter vector L should be carefully selected. Since the false positive rate P_{FP} monotonically decreases with the increase of L , a larger size of Bloom filter vector guarantees a lower false positive rate. A larger L , however, in our tag searching problem may result in a larger volume of transmission in broadcasting the Bloom filter vector from the readers to RFID tags. The number of hash functions, K

determines the computation intensity on RFID tags since each tag should perform K hash functions with its constrained computation resources and storage capacities. As $K = \lfloor \frac{L}{M} \times \ln 2 \rfloor$ is proportional to the vector size L , choosing appropriate Bloom filter vector size L is critical to the protocol performance, helping to achieve better accuracy and efficiency trade-offs. In later analysis, we assume all parameters including L and K , are real numbers. In practical implementation, one can first compute the parameters, and then round them into integers.

There are many lightweight hash functions instantly available in the literatures. To simplify the circuit design of passive RFID tags, one may preload random numbers on the memory chip at each tag during manufacturing. One may adopt the efficient design that organizes the random bits into a logical ring, described in [43]. Since each tag only needs a small number of random numbers, the memory overhead is very small and within the current storage capacity of (even passive) RFID tags [3].

4.2.2.2 Two-phase Compact Approximator based Tag Search Protocol

We propose a Compact Approximator based Tag Searching (CATS) protocol. CATS consists of two phases. Referring to Figure 4.2, we introduce the CATS protocol in this subsection.

In the first phase, we reduce the candidate tags using a Bloom filter. In particular, the backend server constructs a Bloom filter vector by mapping the wanted tags in set $\mathbf{X}$ into an L_1 -bit array using K_1 hash functions with random seed S_1 . The RFID readers broadcast the L_1 -bit Bloom filter vector, K_1 , and S_1 to all RFID tags in interrogation zone. Here we denote the Bloom filter vector as $BF_1(\mathbf{X})$. When receiving $BF_1(\mathbf{X})$, K_1 , and S_1 , each tag $y \in \mathbf{Y}$ locally performs K_1 hash functions with random seed S_1 which are identical to those used to build $BF_1(\mathbf{X})$, and checks whether the bits $h_i(x)$ of $BF_1(\mathbf{X})$ are ‘1’s for $1 \leq i \leq K$. If all the K_1 bits in $BF_1(\mathbf{X})$ are ‘1’s, then we say the tag y passes

Figure 4.2: Two-phase filtering: (1) Transmission of the compact approximator $BF_1(\mathbf{X})$ from backend server to the interrogation zone. The readers broadcast $BF_1(\mathbf{X})$ to the tags. The tags that pass $BF_1(\mathbf{X})$ remain in candidate tag set $\mathbf{Y}_C = \mathbf{Y} \cap BF_1(\mathbf{X})$; (2) The candidate tags $\mathbf{Y} \cap BF_1(\mathbf{X})$ respond in the second-phase communication. The responses forms $BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))$ for server examination.

$BF_1(\mathbf{X})$. We denote by $\mathbf{Y} \cap BF_1(\mathbf{X})$ the set of tags that pass the test of $BF_1(\mathbf{X})$ which is the current candidate set $\mathbf{Y}_C$. Since the Bloom filter has no false negatives, the tags that cannot pass the test of $BF_1(\mathbf{X})$, denoted as $\mathbf{Y} - \mathbf{Y} \cap BF_1(\mathbf{X})$, may directly classify themselves into non-candidate set $\mathbf{Y}_{\sim C}$. The non-candidate tags will keep silent and do not participate in later tag searching operations. On the other hand, the tags in $\mathbf{Y}_C$ will stay alert and participate in the following phases of CATS. We emphasize that because of the false positive problem of the Bloom filter, there are a few tags $y \notin \mathbf{X} \cap \mathbf{Y}$ that may pass the test of $BF_1(\mathbf{X})$ in the first phase with a probability of P_{FP1} .

The expected number of the candidate tags after the first phase $\mathbf{Y} \cap BF_1(\mathbf{X})$ is

$$\begin{aligned} E(|\mathbf{Y} \cap BF_1(\mathbf{X})|) &= |\mathbf{Y} - \mathbf{X} \cap \mathbf{Y}| \times P_{FP1} + |\mathbf{X} \cap \mathbf{Y}| \\ &\leq |\mathbf{Y}| \times P_{FP1} + |\mathbf{X} \cap \mathbf{Y}| \end{aligned}$$

The false positive rate is

$$P_{FP1min} = 0.6185^{\frac{L_1}{|\mathbf{X}|}} = \phi^{\frac{L_1}{|\mathbf{X}|}} \quad (\text{Eq. 4.3})$$

In later description, we define the constant $\phi = 0.6185$. Since P_{FP1} exponentially decreases with the increase of L_1 , we can reduce the false positive rate P_{FP1} at the cost of

additional transmission of a longer $BF_1(\mathbf{X})$. The transmission time of $BF_1(\mathbf{X})$ is

$$T_1 = L_1 \times \alpha \times T_b \quad (\text{Eq. 4.4})$$

For the purpose of clarity, we ignore the transmission cost of the configuration parameters including K_1 , S_1 , and L_1 , which normally take several bytes to encode.

After filtering $\mathbf{Y}$ with $BF_1(\mathbf{X})$, the cardinality of candidate tags $|\mathbf{Y} \cap BF_1(\mathbf{X})|$ will become much smaller than the original cardinality $|\mathbf{Y}|$, in the cases that $|\mathbf{X} \cap \mathbf{Y}| < |\mathbf{Y}|$ and P_{FP1min} is small. At current stage, each of the RFID tag IDs in the candidate set $\mathbf{Y}_C$ is preserved locally on the tag chip and $\mathbf{X} \cap \mathbf{Y}$ still remains unknown to the reader. Such a set is not adequately accurate, yet explicitly letting tags within the set send their IDs back to the readers may result in heavy collisions in a large scale RFID system.

In the second phase, the RFID readers broadcast the parameters K_2 , L_2 , and a new random seed S_2 to the RFID tags and initiate another round of filtering. Upon receiving the configuration parameters, each candidate tag $y \in \mathbf{Y}_C = \mathbf{Y} \cap BF_1(\mathbf{X})$ calculates and selects K_2 slots at the indexes $h_i(y), (1 \leq i \leq K_2)$ in a frame of L_2 length. In later process, each candidate tag transmits a short response at each of the K_2 corresponding slots. Such a process is similar with the first phase of CATS. In the frame, there are two different types of slots: empty slots and non-empty slots. In particular, according to the responses from the candidate tags, the reader encodes a L_2 -bit vector as follows: if the i^{th} slot is an empty slot the reader set the i^{th} bit of the vector to be ‘0’, otherwise ‘1’. A virtual Bloom filter is thus constructed based on the responses from each of the remaining candidate tags in $\mathbf{Y} \cap BF_1(\mathbf{X})$. We denote the L_2 -bit long vector as $BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))$. Note that $BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))$ is not managed by any RFID tags but the backend server beyond the readers.

With the knowledge of $BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))$, the backend server performs membership testing using tag IDs from the wanted tag set $\mathbf{X}$ to determine the intersection $\mathbf{X} \cap \mathbf{Y}$.

The tag IDs which pass $BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))$ test are considered to be within the set $\mathbf{X} \cap BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))$.

According to the characteristics of the Bloom filter, for arbitrary sets $\mathbf{A}$ and $\mathbf{B}$, we have

$$\begin{aligned} \mathbf{A} \cap \mathbf{B} &\subseteq \mathbf{A} \cap BF(\mathbf{B}), \\ |\mathbf{A} \cap \mathbf{B}| &\leq |\mathbf{A} \cap BF(\mathbf{B})| \end{aligned} \quad (\text{Eq. 4.5})$$

We can infer from (Eq. 4.5)

$$\begin{aligned} \mathbf{X} \cap \mathbf{Y} &\subseteq \mathbf{X} \cap BF_2(\mathbf{Y} \cap BF_1(\mathbf{X})), \\ |\mathbf{X} \cap \mathbf{Y}| &\leq |\mathbf{X} \cap BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))| \end{aligned} \quad (\text{Eq. 4.6})$$

In the CATS protocol, we use $\mathbf{X} \cap BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))$ to approximate the intersection $\mathbf{X} \cap \mathbf{Y}$.

4.2.3 Joint Optimization

Although CATS guarantees that any wanted tag $x \in \mathbf{X} \cap \mathbf{Y}$ can be correctly classified, due to false positive property of the Bloom filter there might be unwanted tags misclassified as in $\mathbf{X} \cap \mathbf{Y}$.

The false positive probability after the second phase filtering is

$$P_{FP2} \approx (1 - e^{-|\mathbf{Y} \cap BF_1(\mathbf{X})| \times K_2 / L_2})^{K_2}. \quad (\text{Eq. 4.7})$$

With the optimal setting $K_2 = \frac{L_2}{|\mathbf{Y} \cap BF_1(\mathbf{X})|}$, the minimum false probability is

$$P_{FP2 \min} = \phi^{L_2 / |\mathbf{Y} \cap BF_1(\mathbf{X})|} \quad (\text{Eq. 4.8})$$

Therefore, the expected number of tags which are misclassified is $|\mathbf{X} - \mathbf{X} \cap \mathbf{Y}| \times P_{FP2}$. Given a tolerable false positive rate P_{REQ} , the CATS is able to guarantee $P_{FP2} \leq P_{REQ}$ with joint optimization on L_1 and L_2 .

The transmission time of the frame with L_2 response slots in the second phase is

$$T_2 = L_2 \times \beta \times T_b \quad (\text{Eq. 4.9})$$

Similar to the first phase, we ignore the transmission cost of configuration parameters including K_2 , S_2 , and L_2 .

According to (Eq. 4.4) and (Eq. 4.9), the total transmission time in CATS is

$$T_t = T_1 + T_2 = T_b \times (\alpha L_1 + \beta L_2) = T_b \times L_t \quad (\text{Eq. 4.10})$$

where $L_t = \alpha L_1 + \beta L_2$ abstracts the total bits transmitted in CATS.

According to (Eq. 4.5), (Eq. 4.7), and (Eq. 4.8), we have

$$\begin{aligned} P_{FP2} &= \phi^{L_2/|\mathbf{Y} \cap BF_1(\mathbf{X})|} \leq \phi^{L_2/(|\mathbf{Y}|P_{FP1} + |\mathbf{X} \cap \mathbf{Y}|)}, \\ &\leq \phi^{L_2/(|\mathbf{Y}|\phi^{L_1/|\mathbf{X}|} + |\mathbf{X} \cap \mathbf{Y}|)} \leq \phi^{L_2/(|\mathbf{Y}|\phi^{L_1/|\mathbf{X}|} + |\mathbf{X}|)} \end{aligned} \quad (\text{Eq. 4.11})$$

To maximize the protocol efficiency, we expect to meet the false rate requirement of P_{REQ} with a minimum total transmission time. Therefore, the problem can be modeled as the following optimization problem:

$$\begin{aligned} \text{Minimize : } & L_t = \alpha L_1 + \beta L_2 \\ \text{Subject to : } & \phi^{L_2/(|\mathbf{Y}|\phi^{L_1/|\mathbf{X}|} + |\mathbf{X}|)} \leq P_{REQ} \end{aligned}$$

We solve such a problem with Lagrange multiplier technique. We define

$$\Lambda(L_1, L_2, \lambda) = \alpha L_1 + \beta L_2 + \lambda(\phi^{L_2/(|\mathbf{Y}|\phi^{L_1/|\mathbf{X}|} + |\mathbf{X}|)} - P_{REQ})$$

and solve $\nabla_{L_1, L_2, \lambda} \Lambda(L_1, L_2, \lambda) = 0$.

We obtain the optimal settings for L_1 and L_2 for the optimization problem as follows:

$$\begin{aligned} L_1 &= |\mathbf{X}| \log_{\phi} \left(-\frac{\alpha |\mathbf{X}|}{\beta |\mathbf{Y}| \ln P_{REQ}} \right), \\ L_2 &= \frac{|\mathbf{X}|}{\ln \phi} \left(\ln P_{REQ} - \frac{\alpha}{\beta} \right) \end{aligned} \quad (\text{Eq. 4.12})$$

 Figure 4.3: Sensitivity analysis: L_1 against $|\mathbf{Y}|$

This way, we can guarantee the false positive probability of P_{REQ} with the total transmission of L_1 -bit $R \Rightarrow T$ and L_2 -bit $T \Rightarrow R$ transmissions.

According to (Eq. 4.12), one may notice that CATS requires the cardinality $|\mathbf{X}|$ of the wanted tags as well as the cardinality $|\mathbf{Y}|$ of the tags in the interrogation zone as inputs to set the optimal frame size L_1 and L_2 . While in most cases $|\mathbf{X}|$ is already known in advance to the backend server, in practice, we may have only rough estimation on the cardinality $|\mathbf{Y}|$ of the RFID tags in interrogation zone.

Therefore, we investigate the sensitivity and robustness of the protocol optimality to the variance of $|\mathbf{Y}|$. If the optimal frame size setting is very sensitive to the variance on $|\mathbf{Y}|$, tuning the frame sizes with an inaccurate set cardinality might result in huge performance degradation.

From (Eq. 4.12), we find that $|\mathbf{Y}|$ directly influences the setting of frame size L_1 in the first-phase of CATS. We compute the first order derivative as follows.

$$\frac{dL_1}{d|\mathbf{Y}|} = -\frac{|\mathbf{X}|}{|\mathbf{Y}| \ln \phi} \approx \frac{2.08 |\mathbf{X}|}{|\mathbf{Y}|} \quad (\text{Eq. 4.13})$$

From (Eq. 4.13) we notice that the fraction $\frac{|\mathbf{X}|}{|\mathbf{Y}|}$ determines the first order derivative. Since $|\mathbf{Y}|$ tends to be much bigger than $|\mathbf{X}|$ in most scenarios, the first order derivative can be very small, meaning that L_1 is not sensitive to the variance of $|\mathbf{Y}|$.

Figure 4.3 plots the optimal setting of frame size L_1 given $|\mathbf{X}| = 5 \times 10^3$, 5×10^4 , and 5×10^5 , and $|\mathbf{Y}|$ varying from 1×10^6 to 5×10^6 . When $\frac{|\mathbf{X}|}{|\mathbf{Y}|}$ is small, the derivative of L_1 is very small. On the other hand, Figure 4.3 suggests us that by slightly increasing L_1 , we are able to accommodate a much larger tag set $\mathbf{Y}$.

In the next subsection, we will later show that it is communication economic to use a slightly increased Bloom filter length L_1 so as to guarantee P_{REQ} with high probability.

4.2.4 Cardinality Range Estimation

In some application scenarios, even a rough estimation of $|\mathbf{Y}|$ is not available, and thus we have to estimate the cardinality of the tag population. There are many existing tag cardinality estimation algorithms. Those algorithms, however, need relatively long processing time to derive accurate estimation result [53, 88]. The marginal accuracy improvement decreases dramatically when one pursues higher accuracy estimation results. This observation motivates a rough estimation of $|\mathbf{Y}|$ range rather than an accurate estimation with excessive communication overhead.

Instead of pursuing an accurate cardinality estimation result at excessive communication cost, since the parameters are not sensitive to the estimation error of the tag cardinality, we propose a light-weight cardinality range estimation component to provide cardinality input for CATS. Aiming at a rough cardinality range estimation, the proposed approach tries to maximize the marginal gain of estimation accuracy. During the process, there are several estimation rounds. In each round, the reader collects the 1-bit slot containing empty or non-empty responses from tags in order to roughly estimate the size of tag set.

At the beginning, the reader broadcasts a threshold u and the number of estimation rounds n to the tags, and monitors the communication channel for the responses from tags in the following n slots. When receiving the threshold u and the number of estimation

rounds n , each tag independently computes a binary random vector with a uniform distribution hash function $h_B(ID)$. We denote by $R(ID)$ the position of right-most zero of $h_B(ID)$. For example, assuming that

$$h_B(ID_1) = 0100 \underbrace{11}_2, h_B(ID_2) = 100 \underbrace{111}_3, \quad (\text{Eq. 4.14})$$

$R(ID_1) = 2$ and $R(ID_2) = 3$, respectively. Obviously, the random number $R(ID)$ follows geometric distribution with probability $\Pr\{R(ID) = k\} = \frac{1}{2^{k+1}}$.

We denote by R_{ij} , ($1 \leq i \leq |\mathbf{Y}|, 1 \leq j \leq n$), the random number of tag i in the j^{th} estimation round. In the estimation round j , if $R_{ij} > u$, then tag i responds to the reader by sending a 1-bit short response; otherwise the tag keeps silent. As a result, n consecutive empty or non-empty slots will be sensed by the reader.

The probability that each tag keeps silent in the j^{th} estimation round is

$$\Pr(R_{ij} < u) = \sum_{k=0}^{u-1} \Pr(k) = \sum_{k=0}^{u-1} \frac{1}{2^{k+1}} = 1 - \frac{1}{2^u} \quad (\text{Eq. 4.15})$$

In the case that all $R_{ij} < u$, $i \in \{1, 2, \dots, |\mathbf{Y}|\}$, the reader observes no single from the tags, i.e., the channel is idle. Therefore, with the assumption of i.i.d. for R_{ij} , the probability that the reader observes an idle channel in the j^{th} estimation round is as follows.

$$\begin{aligned} \Pr(idle) &= [\Pr(R_i < u)]^{|\mathbf{Y}|} = \left(1 - \frac{1}{2^u}\right)^{|\mathbf{Y}|} \\ &\approx e^{-|\mathbf{Y}|/2^u} = e^{-\rho}, \end{aligned}$$

where we define $\rho = \frac{|\mathbf{Y}|}{2^u}$.

We define a Bernoulli random variable X which takes value 1 with probability $\Pr(idle) \approx e^{-\rho}$ and value 0 with probability $1 - \Pr(idle) \approx 1 - e^{-\rho}$, so we have

$$\Pr(X = 1) \approx e^{-\rho}, \Pr(X = 0) \approx 1 - e^{-\rho} \quad (\text{Eq. 4.16})$$

Therefore, according to (Eq. 4.16), the expected value and the variance of X are

$$E(X) = e^{-\rho}, \sigma^2(X) = e^{-\rho}(1 - e^{-\rho}) \quad (\text{Eq. 4.17})$$

The maximum variance of X is $\sigma^2(X)_{MAX} = 0.25$, when $e^{-\rho} = 0.5$.

We define the average value of n measurements as $\bar{X} = \frac{1}{n} \sum_{j=1}^n X_j$. Then the expectation and the variance of $\bar{X}$ are

$$E(\bar{X}) = E(X) = e^{-\rho}, \sigma^2(\bar{X}) = \frac{\sigma^2(X)}{n} \leq \frac{0.25}{n} \quad (\text{Eq. 4.18})$$

According to (Eq. 4.18), the observation of $\bar{X}$ can be used to estimate the set cardinality $|\hat{\mathbf{Y}}|$ as follows

$$|\hat{\mathbf{Y}}| = -2^u \ln \bar{X} \quad (\text{Eq. 4.19})$$

The challenge in such a cardinality estimation approach arises however, when either $\Pr(\text{idle}) \approx e^{-\rho} \rightarrow 0$ or 1. Without observing adequate number of distinct channel states, the estimation accuracy on $|\hat{\mathbf{Y}}|$ would be poor [17]. This motivates us to adjust the threshold u so as to quickly adapt to the cardinality range. The RFID reader adaptively calibrates u according to the tags' responses and progressively narrows down the estimation range, e.g., if we observe very few idle states (i.e., $\bar{X} \rightarrow 0$), we infer that $|\mathbf{Y}| \gg 2^u$ and increment u ; if we observe idle channel in almost all rounds (i.e., $\bar{X} \rightarrow 1$), we infer that $|\mathbf{Y}| \ll 2^u$ and decrement u . The expected value of $\bar{X}$ is nondecreasing with the increase of u . With the monotonic feature, we can speed up the convergence of u and narrow down the estimating range with bisection search.

Since the result may still vary slightly because of the estimation variance, we seek a guaranteed cardinality estimation confidence range, i.e., $\Pr\{|\mathbf{Y}|(1 - \varepsilon) \leq |\hat{\mathbf{Y}}| \leq |\mathbf{Y}|(1 + \varepsilon)\} \geq 1 - \frac{1}{k^2}$. We can rewrite the estimation range requirement as follows

$$\begin{aligned} & \Pr\{|\mathbf{Y}|(1 - \varepsilon) \leq |\hat{\mathbf{Y}}| \leq |\mathbf{Y}|(1 + \varepsilon)\} \\ &= \Pr\left\{e^{-\frac{|\mathbf{Y}|(1+\varepsilon)}{2^u}} \leq \bar{X} \leq e^{-\frac{|\mathbf{Y}|(1-\varepsilon)}{2^u}}\right\} \end{aligned} \quad (\text{Eq. 4.20})$$

According to Chebyshev's inequality, we have

$$\Pr\{e^{-\frac{|\mathbf{Y}|}{2u}} - \frac{k}{2\sqrt{n}} \leq \bar{X} \leq e^{-\frac{|\mathbf{Y}|}{2u}} + \frac{k}{2\sqrt{n}}\} \geq 1 - \frac{1}{k^2} \quad (\text{Eq. 4.21})$$

Combining (Eq. 4.20) and (Eq. 4.21), we compute the minimum estimation round n^* which can guarantee the estimation requirement $\Pr\{|\mathbf{Y}|(1 - \varepsilon) \leq |\hat{\mathbf{Y}}| \leq |\mathbf{Y}|(1 + \varepsilon)\} \geq 1 - \frac{1}{k^2}$.

The estimation result influences the first-phase parameter setting in the CATS protocol, such as L_1 and K_1 . Since the CATS protocol can accommodate a larger number of tags by slightly increasing L_1 , it is communication economic to enhance the robustness at a small communication cost. Assume that the estimation value is $|\hat{\mathbf{Y}}|$ and the length of the Bloom filter in the first phase is $L_1(|\hat{\mathbf{Y}}|)$, then to accommodate $|\hat{\mathbf{Y}}|(1 + \varepsilon)$ tags, according to (Eq. 4.12), we can slightly increase L_1 by $|\mathbf{X}| \log_\phi \frac{1}{(1+\varepsilon)}$. In later evaluation part, we show that, $\varepsilon = 10\%$ already gives us quite good performance.

4.2.5 Discussion

4.2.5.1 Estimate Candidate Cardinality before the Second-phase Filtering

We jointly optimize the Bloom filter sizes L_1 and L_2 and pre-set the frame lengths. As a matter of fact, after the filtering, we have another chance to estimate the cardinality of shortlisted candidate tags $|\mathbf{Y}_C| = |\mathbf{Y} \cap BF_1(\mathbf{X})|$, and further tune the second-phase parameters L_2 and K_2 accordingly,

$$\begin{aligned} L_2 &= |\mathbf{Y} \cap BF_1(\mathbf{X})| \times \log_\phi P_{REQ}, \\ K_2 &= \frac{L_2}{|\mathbf{Y} \cap BF_1(\mathbf{X})|} \times \ln 2 \end{aligned} \quad (\text{Eq. 4.22})$$

This provides one more chance to optimize the frame size L_2 in order to achieve the false positive requirement with even lower communication cost. In the scenario that $|\mathbf{Y} \cap BF_1(\mathbf{X})|$ is small and execution time involved in collecting the IDs is within the delay requirement, the tag searching protocol may even directly collect the IDs from those tags instead of performing the second filtering phase.

4.2.5.2 Multiple Readers and Mobile Tags

Since the propagation range of both RFID readers and RFID tags are limited, many large-scale applications deploy multiple readers to enhance the coverage for a large number of tags in the interrogation region. In such scenarios, duplicate readings of the same object are very common. In CATS, however, the backend server aggregates the 1-bit responses from all readers. Even if a tag is located in the overlapped interrogation region and its response is overheard by multiple readers, its impact on the backend aggregation is equivalent to a single response. Therefore, the CATS protocol well handles the multiple reader scenario with the duplicate-insensitive nature in tag responses. When RFID tags are attached to mobile objects and move within the interrogation region of multiple readers, the response from the same tag will finally converge at the backend server even it might go through multiple readers. Such a scenario is equivalent to that of the multiple readers and thus can be correctly handled by the CATS protocol.

4.2.5.3 Anonymity

In some applications where the tag ID carries private information about the associated item, explicitly transmitting such information might lead to privacy leakage. CATS resists such privacy threats because each RFID tag does not explicitly broadcast its ID. Instead each tag responds to the reader's query according to implicit hash results. In addition, the tag does not reveal the hash result directly to publics. At each response slot, a number of tags reply to the reader, and their responses cumulate. Neither the readers nor any overhearing entities can distinguish the exact set of tags which respond at a collision slot. Consequently, the hash values of RFID tags are well preserved from any eavesdroppers.

4.2.5.4 Enhance robustness to noise

The channel errors would lead to confused responses in the second phase for each tag. To enhance the robustness to noise, the reader may divide the Bloom filter into several blocks (e.g., 96-bit block) and add a checksum for each block. Receiving each block, the tags can use the checksum to detect corrupted data in the tag side. We note that it remains as an open problem to ensure the reliable data collection from a large number of tags in practice. One promising research direction might be to enhance the reliability with rateless codes [72]. We would like to further explore along this direction in the future.

4.3 Evaluation

4.3.1 Simulation Setting

In the simulations, we assume that there is no transmission loss between RFID tags and the reader. In each frame the reader initiates the communication by sending commands to the tags and waits for tag's response. The RFID reader is capable of detecting and distinguishing empty slots from non-empty slots. All presented results are obtained by averaging over 150 runs.

We mainly consider the searching efficiency given a tolerable false positive. Since both $T \Rightarrow R$ and $R \Rightarrow T$ transmission rates vary depending on various factors, we assume both $T \Rightarrow R$ and $R \Rightarrow T$ data rates to be 40kbps, i.e., the transmission time for each bit equals to $25\mu s$ [3]. The total transmission time reflects the protocol efficiency. The protocol with short transmission time will be able to scale up with more RFID tags.

We also concern whether the tag searching algorithm can guarantee the tolerable false positive specified by users. We use the false positive fraction as the accuracy indicator, denoted as

$$fraction_{FP} = \frac{|\{x | x \in \mathbf{X} - \mathbf{X} \cap \mathbf{Y}, x \in \mathbf{X} \cap BF_2\}|}{|\mathbf{X} - \mathbf{X} \cap \mathbf{Y}|},$$

4.4.a:

4.4.b:

Figure 4.4: Cardinality range estimation with $|\mathbf{Y}| = 10000$: (a) Cumulative distribution of estimation results; (b) Estimation results with estimation rounds $n = 512$, 1024 , and 2048 .

where $X \cap BF_2$ denotes the set of the tags that pass the two phase membership test. Given a false positive requirement P_{REQ} , the protocol is expected to guarantee $fraction_{FP} \leq P_{REQ}$.

4.3.2 CATS Investigation

4.3.2.1 Cardinality Range Estimation

We first investigate the cardinality range estimation algorithm and demonstrate its effectiveness. In the simulation, we set the tag cardinality in the interrogation region to be

4.5.a:

4.5.b:

Figure 4.5: (a) $\frac{\alpha}{\beta}$ varies from $\frac{1}{16}$ to 16, and the same false positive rate $P_{REQ} = 5\%$; (b) P_{REQ} varies from 1% to 10%, and the symmetric data rates $\frac{\alpha}{\beta} = 1$.

10000. Figure 4.4.a illustrates the cumulative distribution of the estimated values for the tag cardinality with different rounds of estimation. Figure 4.4.b plots 1000 estimation results for each of the 512, 1024, and 2048 estimation rounds, respectively. We observe that the cardinality estimation protocol provides tunable estimation accuracy in terms of estimation time, i.e. the more estimation rounds we run, the more accurate estimation results will be. The marginal accuracy improvement, however, decreases when we reach 1024 rounds. Besides, since the CATS algorithm is not sensitive to the tag cardinality in the interrogation region, we prefer rough estimation rather than expensive high accuracy estimation. The empirical results show that 1024 estimation rounds suffice to get accurate cardinality range input for CATS. As depicted in Figure 4.4.b, with 1024 estimation rounds we observe that 1) most estimation results, 99.3%, are within the confidence range $[9000, 11000]$, and 2) for the small portion, 0.7%, of the estimation results off the confidence range are still very close to the confidence range. Therefore, in later experiments we estimate the tag cardinality range with 1024 rounds of estimation, and set tolerable cardinality range estimation error to be $\varepsilon = 10\%$.

Table 4.1: Parameter setting ($\frac{\alpha}{\beta} = 1$, $P_{REQ} = 0.05$, and $|\mathbf{Y}| = 5000000$)

$ \mathbf{X} (10^5)$	$K_{(1,2)}$	L_1	L_2	L_{est}	L_{sum}
0.5	(8,4)	593429	415825	11099	1020353
1	(7,4)	1042590	831650	21014	1895254
2	(6,4)	1796645	1663301	40844	3500790
4	(5,4)	3016219	3326602	80504	6423325
8	(4,4)	4878294	6653204	159824	11691322
16	(3,4)	7448301	13306408	318464	21073173
32	(2,4)	10280027	26612817	635744	37528588

4.3.2.2 CATS Investigation

In this subsection, we investigate the CATS protocol performance. The cardinality of tag set $\mathbf{Y}$ is either known in advance or estimated with the previous cardinality range estimation approach.

Figure 4.5 plots the optimal frame sizes L_1 and L_2 under different settings of $R \Rightarrow T$ and $T \Rightarrow R$ data rates, as well as different false positive requirements. In Figure 4.5.a, we fix $P_{REQ} = 5\%$ and vary $\frac{\alpha}{\beta}$ from 0.0625 to 16. When $\frac{\alpha}{\beta}$ is small, meaning that $R \Rightarrow T$ per bit transmission time is smaller than that of $T \Rightarrow R$, CATS prefers a longer L_1 and a shorter L_2 . When $\frac{\alpha}{\beta}$ is big, CATS prefers a shorter L_1 and a longer L_2 so as to shift more communications to the second phase. In Figure 4.5.b, we fix $\frac{\alpha}{\beta} = 1$, and vary P_{REQ} from 1% to 10%. A larger tolerable false positive rate trades for shorter frame sizes and thus shorter transmission time.

For simplicity, in the following experiments we assume a symmetric $T \Rightarrow R$ and $R \Rightarrow T$ data rates (i.e., $\alpha = \beta$), and a required false positive rate $P_{REQ} = 5\%$. In Table 4.1, we show the optimal parameter settings given $|\mathbf{X}|$ varying from 50000 to 3200000 and $|\mathbf{Y}| = 5000000$. $K_{(1,2)}$ represents the hash functions that each tag will take in the first and second phase Bloom filter membership testing and inserting. L_1 and L_2 are the optimized

Bloom filter sizes in the two phases. L_{est} , representing the cardinality range estimation overhead, consists of $\log_2 32 = 5$ bisection steps (32 rounds for each step) to converge to an estimation range, and 1024 rounds to sharpen the accuracy (with $\varepsilon = 10\%$). We count the extra $|\mathbf{X}| \log_{\phi} \frac{1}{(1+\varepsilon)}$ in L_{est} as well. L_{sum} denotes total bit transmission involved in the entire CATS protocol, i.e., $L_{sum} = L_1 + L_2 + L_{est}$. According to Table 4.1, we find L_{est} counts for less than 2% of the total bit transmission L_{sum} . L_{sum} has already taken the cardinality range estimation overhead into consideration, and is used to abstract the overall transmission overhead of CATS.

Figure 4.6.a compares the total transmission time of tag searching with CATS, the baseline protocol, and an ideal ALOHA-based identification protocol with the optimal efficiency of 36.8% [39]. The expected transmission time of the baseline protocol is almost 4.81 to 8.41 times that of CATS. According to the results, CATS significantly outperforms the optimal performance state-of-the-art identification based protocols can achieve. Besides the search efficiency, we are also interested in whether the false positive rate of CATS is within the requirement bound. Figure 4.6.b plots the false positive fraction of CATS with different scenarios, where we fix $|\mathbf{Y}| = 5000000$ and vary $|\mathbf{X}|$ from 50000 to 3200000. We observe that for each setting the false positive rate increases as $\text{Ratio} = \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X}|}$ increases, and that the false positive rate also increases with $|\mathbf{Y}|$. The false positive rates of all tests remain within the tolerable bound $P_{REQ} = 5\%$.

One may notice that in the case of $|\mathbf{X}| = 50000$, the false positive rate is below 2% indicating that CATS performs beyond the expected false positive rate. It is mainly because we consider the worst case in tuning CATS parameters (L_1 , L_2 , and the like) without the knowledge of $|\mathbf{X} \cap \mathbf{Y}|$, i.e., we bound $|\mathbf{Y} \cap BF_1(\mathbf{X})|$ with $|\mathbf{Y}|P_{FP1} + |\mathbf{X}|$. When $|\mathbf{Y}|$ is small and $\frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X}|}$ is big, $|\mathbf{Y} \cap BF_1(\mathbf{X})|$ is much smaller than $|\mathbf{Y}|P_{FP1} + |\mathbf{X}|$. On the other hand, when $|\mathbf{Y}| \gg |\mathbf{X}|$, $|\mathbf{Y}|P_{FP1} + |\mathbf{X}|$ becomes a good estimation of $|\mathbf{Y} \cap BF_1(\mathbf{X})|$.

In Figure 4.7, we examine the same metrics by varying $|\mathbf{Y}|$ from 50000 to 3200000 with the fixed $|\mathbf{X}| = 50000$. Similarly, Figure 4.7.a shows that CATS significantly reduces

4.6.a:

4.6.b:

Figure 4.6: CATS investigation with varied $|X|$ from 0.5×10^5 to 32×10^5 and the fixed $|Y| = 5000000$: (a) Total transmission time (in *log scale*); (b) False positive fraction.

4.7.a:

4.7.b:

Figure 4.7: CATS investigation with varied $|Y|$ from 0.5×10^5 to 32×10^5 and the fixed $|X| = 50000$: (a) Total transmission time (in *log scale*); (b) False positive fraction.

transmission time compared with the other two approaches. Figure 4.7.b shows that

CATS secures the false positive rate $P_{REQ} = 5\%$ for all tests.

4.4 Summary

In this chapter, we study the tag searching problem in large-scale RFID systems. The solution to such problem is of significant importance for many RFID management applications. To meet the stringent delay requirement, we propose CATS, a compact approximator based tag searching protocol, which significantly improves the searching efficiency in comparison with the state-of-the-art approaches, while being able to secure an arbitrary required false rate. We propose a light-weight cardinality range estimation algorithm for providing cardinality input to CATS. We do extensive simulations to evaluate the performance of CATS and the results demonstrate that CATS outperforms other possible solutions originated from existing approaches.

Chapter 5

Missing Tag Identification

The problem of missing tag identification has attracted wide attention due to its practical importance [43, 87]. For example, RFID tags can be attached to items as labels in a warehouse, and warehouse managers can monitor them for anti-theft purpose. Such a problem of item-level monitoring also appears in applications of healthcare, logistics, military, etc. One straightforward solution might be to identify all the tags and check whether anyone is missing using traditional RFID tag identification schemes. When the number of tags is small, such a method can frequently identify and monitor each tag. When the number of tags scales up, tag-tag collisions become increasingly severe and incur substantial collision arbitration overhead as well as the transmission overhead of tag IDs.

Recently, many probabilistic missing tag identification schemes [43, 87] try to identify missing tags without explicitly sending tag IDs. Due to the inherent nature of sequential look-up, those approaches consume a substantial number of time slots that is linearly proportional to the total number N of tags. Despite of many advances, such approaches largely overlook the physical layer information and solely rely on upper layer information, rendering them less efficient for realtime monitoring.

The compelling practical demands and the inadequacy of the status quo motivate the design of more efficient missing tag identification schemes. In this chapter, we present

Physical-layer Missing Tag Identification (P-MTI) which efficiently utilizes the physical layer information of tag responses and thereby substantially improves the monitoring efficiency. Unlike conventional approaches focusing on individual tag responses, we look into the aggregated signals from concurrent tag responses which provide us much richer information.

Suppose that we wish to identify K missing tags out of N , where $N \gg K \geq 0$. We let each tag i transmit a sequence of M random bits A_i concurrently with other tags. Each tag i transmits one bit of A_i at each time slot. The tags modulate the bits into physical layer symbols using on-off keying. In physical layer, the transmitted symbols from multiple tags will mix in the air and arrive at the reader as PHY symbol superpositions. In the j th time slot the reader receives $y_j = \sum_{i=1}^N A_i(j)$, and thus after M time slots the received symbols at the reader $\mathbf{y}$ can be concisely represented as $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $M \times N$ matrix $\mathbf{A} = [A_1, A_2, \dots, A_N]$. $\mathbf{y}$ denotes an $M \times 1$ vector where each entry represents one of the M PHY symbol measurements. $\mathbf{x}$ denotes an $N \times 1$ binary vector where the non-zero entries indicate presence of tags while the zero entries imply the missing tags. For ease of presentation, here we omit channel coefficients. To compute $\mathbf{x}$ and figure out the K zero entries, generally we need $M = N$ measurements. As a matter of fact, it suffices for identifying the missing tags to know the differential of two consecutive instances, $\mathbf{x}_\Delta = \mathbf{x}_t - \mathbf{x}_{t-\Delta}$. A non-zero entry in $\mathbf{x}_\Delta$ indicates that the corresponding tag is present in the former measurement while absent in the latter measurement, which implies a missing tag event. Thus, we only need to reconstruct $\mathbf{x}_\Delta$ to identify the missing tags that are missed during the period $[t - \Delta, t]$. Moreover, we know that the missing tag events are inherently sparse, i.e., $N \gg K \geq 0$. Therefore, $\mathbf{y}_\Delta = \mathbf{A}\mathbf{x}_\Delta$ can be formulated as a standard compressive sensing problem, which allows us to recover $\mathbf{x}_\Delta$ with only a substantially smaller number of measurements. The tags that are missed during the time $[0, t - \Delta]$ can be identified by examining $\mathbf{x}_{t-\Delta} - \mathbf{x}_0$ in a similar manner.

We implement a prototype system and validate P-MTI based on the Universal Software Radio Peripheral (USRP) software defined radio with the Intel Wireless Identification and Sensing Platform (WISP) computational RFID tags. We also investigate our approach in large-scale settings with extensive simulations compared with existing approaches [43, 87]. The experiment results show that the proposed P-MTI scheme significantly improves the time efficiency. In particular, P-MTI can effectively reduce the communication time by approximately 64% over state-of-the-art approaches. The improvement stems from both the efficient use of lower layer information and the compressive sensing based reconstruction.

The contributions of this chapter are briefly summarized as follows: For the first time, (1) we present the physical layer missing tag identification scheme, which makes extensive use of lower layer information in large-scale RFID systems; (2) exploiting the sparsity of missing tag events, the proposed scheme further improves missing tag identification efficiency with compressive sensing technique; (3) we consolidate our design with a prototype system using the software-defined RFID reader and the computational RFID tags.

In the following, we first describe the problem of missing tag identification in Section 5.1. Section 5.2 presents the detailed design of the Physical-layer Missing Tag Identification (P-MTI) protocol. Section 5.3 presents the implementation of P-MTI on software defined platforms. Section 5.4 presents evaluation results in comparison with state-of-the-art approaches. Finally, Section 5.5 briefly summarizes this chapter.

5.1 Problem Description

Consider an RFID system $\mathbf{N} = \{n_1, n_2, \dots, n_N\}$ representing all the N tags covered by a reader, and $\mathbf{K} \subseteq \mathbf{N}$ representing the set of missing tags. The reader has the access to the IDs of the tags, among which K tags are missing. **The problem of missing tag**

identification is to identify the K missing tags. The missing tag set may be empty meaning that all the tags in $\mathbf{N}$ are present. We denote by $|\cdot|$ the cardinality of a set, and thus $K = |\mathbf{K}|$ and $N = |\mathbf{N}|$. We particularly focus on the cases where $N \gg K \geq 0$. Although the reader has the knowledge of $\mathbf{N}$, the number of missing tags K is not known in advance. If most tags are missing and only a small number of tags are present, one may directly identify the present tags with collision arbitration schemes [52, 79]. Many recent cardinality estimation schemes [33, 53, 90] can be used to estimate the number of tags.

To ensure realtime monitoring, we need to reduce the processing time of missing tag identification and design a scalable scheme to support thousands or even more tags. As each tag has to transmit at least one-bit information to announce its presence, prior work believes that at least N time slots are needed to monitor N tags [43]. Generally, such schemes sequentially look up individual tag responses and only differentiate the empty/singleton/collision slots, meaning that only $\log_2 3$ bits of information can be extracted per slot. Moreover, each tag transmits multiple physical layer symbols to allow the reader to distinguish different types of slots.

To improve monitoring efficiency, we wish to make extensive use of physical layer symbols by extracting more amount of information. On the other hand, as the tags are generally resource-constrained, we wish to make minimum modifications to the tags without introducing extra computation or communication overhead.

5.2 P-MTI: Physical-layer Missing Tag Identification Protocol

We first describe the rationale of P-MTI design in Section 5.2.1. We present a basic solution which leverages the physical layer information for missing tag identification in Section 5.2.2. We then exploit the sparsity of missing tag events and use compressive

Figure 5.1: Frame-slotted Aloha: 7 tags contend for 11 time slots.

sensing technique to enhance the missing tag identification efficiency in Section 5.2.3. We enhance decoding robustness against noisy measurements in Section 5.2.4. We present an operation overview of our approach in Section 5.2.5. We finally discuss additional design considerations in Section 5.2.6 and Section 5.2.7.

5.2.1 P-MTI Rationale

The conventional schemes generally exploit the frame-slotted Aloha protocol to identify the missing tags [43, 87]. Figure 5.1 illustrates an instance where 7 tags contend for 11 time slots. Say that one tag should have responded in a singleton slot (e.g., the 7th time slot) if present, but no response is received from the time slot. Then the reader may infer the tag is missing. In such schemes, less information can be extracted from empty and collision slots. As each tag has to transmit at least one-bit information to announce its presence, prior work believes that at least N time slots are needed to monitor N tags [43]. Although many advances have been achieved in missing tag identification [43, 87], existing approaches largely overlook the PHY information and extract limited amount of information at upper layer.

To fundamentally improve the protocol efficiency, we conduct initial experiments to explore the physical layer of RFID system using the USRP software defined radio and the WISP programmable tags. Section 5.3 presents the implementation and experiment settings in detail. A PHY symbol can be represented as a complex number (amplitude

Figure 5.2: Received signals at RFID reader: (a) Signals from tag 1; (b) Signals from tag 2; (c) Aggregated signals from the two tags.

and phase components). As RFID backscatter communication uses on-off keying (a simple amplitude-shift keying scheme), at physical layer an RFID reader decodes tag responses by measuring only amplitude and ignoring phase component [3]. We note that such on-off keying is compatible with existing RFID modulation schemes as specified in the C1G2 standard. Figure 5.2(a) shows the magnitude of backscatter signals received at the USRP reader when one WISP tag (tag 1) transmits a sequence of pseudo random bits using on-off keying. Similarly, Figure 5.2(b) depicts the signals from tag 2. The magnitude of received signals depends on the wireless channel attenuation. The reader can decode tag 1 and tag 2 individually when no collision occurs. When both tags transmit at the same time (Figure 5.2(c)), we find that the aggregated PHY symbols exhibit as the superposition of the symbols from both tags. When more tags respond concurrently, the responsive signals from multiple tags similarly exhibit as a sum when they arrive at the reader. We propose to efficiently utilize the aggregated signals from concurrent tag responses (which were previously treated as collision slots), and present P-MTI which makes extensive use of such physical layer information.

Figure 5.3: Response signals from 7 tags mix in the air and aggregate at the reader. If tag i is missing, then its contribution to the aggregated physical layer symbols would be missing.

The rationale behind P-MTI is that the missing tags reveal themselves by the absences of responses. Say that there are 7 tags (i.e., tag 1 – 7), and we let them send random bits concurrently as in Figure 5.3. The responses from tags mix in the air and aggregate at the reader. The reader will receive physical layer symbols from all the 7 tags if none is missing. If any tags are missing, the received symbols may differ from the expected ones. We denote the symbols of tag i as A_i , and denote the channel coefficient as h_i . The aggregated physical layer responses received by the reader can be represented as $\sum_{i=1}^7 A_i h_i$. If tag k is missing, the aggregated responses become $\sum_{i=1}^7 A_i h_i - A_k h_k$. We can thus leverage the differences of physical layer symbols to detect and identify the missing tags. Departing from conventional schemes which look at individual tag response at each slot, P-MTI allows multiple tags to concurrently respond in each time slot which fundamentally improves the protocol efficiency.

5.2.2 Physical-layer Missing Tag Identification: A Basic Solution

In P-MTI, an RFID reader initiates the missing tag monitoring by broadcasting an operation code. When receiving the command, each tag generates random bits using a pseudo random number generator with its ID as the seed. The pseudo random bits of tag i is denoted as $\mathbf{A}_i$, an $M \times 1$ bit vector. At each time slot, each tag backscatters

radio frequency signals if the random bit turns out to be 1, or keep silent otherwise. The physical layer symbols from N tags mix in the air and aggregate at the reader. We model the RFID communication channel with a complex matrix $\mathbf{H}_{N \times N}$. Then the aggregated symbols at the reader can be represented as follows.

$$\mathbf{y} = \mathbf{A}\mathbf{H}\mathbf{x}, \quad (\text{Eq. 5.1})$$

where $\mathbf{y}$ is an $M \times 1$ complex vector representing the aggregated symbols, $\mathbf{A}$ is an $M \times N$ binary matrix, and $\mathbf{x}$ is an N -entry binary vector with non-zero (zero) entries representing presence (absence) of tags. As the backscatter communication is generally within a narrow wireless band [72], the wireless channel between each tag and the reader can be mathematically modeled with a complex number, incorporating both signal attenuation and phase shift of wireless channel [72]. Therefore, we assume $\mathbf{H} = \text{diag}(h_1, h_2, \dots, h_N)$, where h_i denotes the channel coefficient from tag i to the reader. Thus $\mathbf{H}\mathbf{x}$ can be represented with an $N \times 1$ complex vector $\mathbf{z}$, where the i th entry $z_i = h_i x_i$ [72]. As $\mathbf{x}$ is binary vector, we have $z_i = h_i$ if $x_i = 1$, and $z_i = 0$ otherwise. Hence, Eq.(Eq. 5.1) can be rewritten as follows,

$$\mathbf{y} = \mathbf{A}\mathbf{z}. \quad (\text{Eq. 5.2})$$

Since the reader has access to the tag IDs through database lookups, the reader can generate the pseudo random binary matrix $\mathbf{A}$ using the same random number generator with the tags IDs as seeds. Therefore, we can compute $\mathbf{z}$ by solving the linear equation Eq.(Eq. 5.2) with the knowledge of $\mathbf{y}$ and $\mathbf{A}$. The non-zero entry $z_i = h_i$ is the channel coefficient between tag i and the reader. A zero entry in $\mathbf{z}$ indicates that the corresponding tag is missing. In such a way, P-MTI not only identifies missing tags but also measures the channel coefficients $\mathbf{H}$ between present tags and the reader.

To solve the linear equation Eq.(Eq. 5.2), we need N measurements of physical layer symbols, since the unknown $\mathbf{z}$ is an $N \times 1$ vector. As the reader does not need to distinguish empty/singleton/collision slots, the tags do not need to transmit multiple physical layer symbols in each slot, which further reduces the transmission time.

5.2.3 Compressive Sensing based Recovery

We notice that P-MTI needs N measurements per monitoring operation. In practical scenarios, one may need to frequently run the missing tag identification protocol to ensure a timely report of missing tag events.

As a matter of fact, it suffices to compute the differential of aggregated responses between two consecutive instances to achieve continuous monitoring. The rationale is straightforward: if a response from a tag is detected while no more response is detected later from the tag, then the tag is probably missing. We therefore compute the differential of the aggregated responses between two consecutive instances at time t and $t - \Delta$, $\mathbf{y}_\Delta = \mathbf{y}_t - \mathbf{y}_{t-\Delta}$, and infer the dynamics of the tags. From Eq.(Eq. 5.2), we have

$$\mathbf{y}_\Delta = \mathbf{y}_t - \mathbf{y}_{t-\Delta} = \mathbf{A}\mathbf{z}_t - \mathbf{A}\mathbf{z}_{t-\Delta}. \quad (\text{Eq. 5.3})$$

We refer the differential of $\mathbf{z}$ similarly by $\mathbf{z}_\Delta = \mathbf{z}_t - \mathbf{z}_{t-\Delta}$. Then, from Eq.(Eq. 5.3), we have

$$\mathbf{y}_\Delta = \mathbf{A}\mathbf{z}_\Delta. \quad (\text{Eq. 5.4})$$

Ideally, the zero entries in $\mathbf{z}_\Delta$ imply that corresponding tags are present, while the non-zero entries indicate that the tags are missing during the period $[t - \Delta, t]$. The tags that are missed during the time $[0, t - \Delta]$ can be identified by examining $\mathbf{z}_{t-\Delta} - \mathbf{z}_0$ in a similar manner. In practical RFID systems, the number of missing tags K is typically much smaller than the number of tags under monitoring N during a short monitoring interval, i.e., $N \gg K \geq 0$ [43, 87]. In other words, $\mathbf{z}_\Delta$ is K -sparse, meaning that there are

at most K non-zero entries in $\mathbf{z}_\Delta$ [23]. According to the theory of compressive sensing, the K -sparse vector $\mathbf{z}_\Delta$ can be accurately recovered with only $M = O(K \log(N/K))$ measurements, by solving the following convex optimization problem [23].

$$\begin{aligned} &\text{Minimize: } \|\mathbf{z}_\Delta\|_{\ell_1} \\ &\text{Subject to: } \mathbf{A}\mathbf{z}_\Delta = \mathbf{y}_\Delta, \end{aligned} \tag{Eq. 5.5}$$

where $\|\cdot\|_{\ell_p}$ denotes ℓ_p -norm, i.e., $\|\mathbf{x}\|_{\ell_p} \triangleq (\sum_{i=1}^N |x_i|^p)^{1/p}$. Intuitively, the objective function of minimizing $\|\mathbf{z}_\Delta\|_{\ell_1}$ incorporates the fact that $\mathbf{z}_\Delta$ is sparse, while the constraint function is self-explanatory. Therefore, we can refer to convex optimization solvers (e.g., CVX [2], ℓ_1 -Magic [6]) to compute $\mathbf{z}_\Delta$. In our implementation, we use the CVX solver [2] based on the interior-point algorithm [11]. It has been reported that $M = 3K \sim 4K \ll N$ measurements suffice to recover the K -sparse vector [47]. The number of missing tags K however is not known in advance. We set the number of measurements as $M = CK_{\max} \log(N)$ in practice, where $K_{\max}$ represents the estimated maximum number of missing tags and C is a constant. Our experiment results show that M measurements of physical layer symbols are sufficient to reconstruct the K -sparse vector of $\mathbf{z}_\Delta$. When the number of missing tags exceeds the estimated maximum number $K_{\max}$, our approach can identify at least $K_{\max}$ missing tags, which allows P-MTI to adjust $K_{\max}$ (Section 5.4).

Leveraging the differential of aggregated responses, we improve the performance of P-MTI from $O(N)$ to $O(K \log(N/K))$. The compressive sensing based information reconstruction allows us to extract the missing tag events directly from the differential of aggregated physical layer symbols, thereby significantly reducing the required total time slots compared with the existing schemes. Besides, unlike existing schemes which need multiple physical symbols per slot, P-MTI needs only one symbol per time slot. Therefore, the performance gain of such joint optimization is promising.

5.2.4 Enhancement against Noisy Measurement

The above analysis ignores the noise in measurements. Wireless channel is mostly error-prone, subjected to various factors, such as interference, quantization, etc [13]. Channel dynamics may also introduce noise to the measurements. Without robustness enhancement against noise, a detection system may result in unfavorable false alarms over noisy channels [30]. In the following, we enhance P-MTI's robustness against noise based on the theory of stable recovery [14]. Incorporating the noise, Eq.(Eq. 5.4) can be rewritten as follows.

$$\mathbf{y}_\Delta = \mathbf{A}\mathbf{z}_\Delta + e, \quad (\text{Eq. 5.6})$$

where e denotes the error due to noise. Then the optimization problem with relaxed constraints for recovery of $\mathbf{z}$ can be written as follows.

$$\begin{aligned} &\text{Minimize: } \|\mathbf{z}_\Delta\|_{\ell_1} \\ &\text{Subject to: } \|\mathbf{A}\mathbf{z}_\Delta - \mathbf{y}_\Delta\|_{\ell_2} \leq \epsilon, \end{aligned} \quad (\text{Eq. 5.7})$$

where the magnitude e is bounded by ϵ , i.e., $\|e\|_{\ell_2} \leq \epsilon$. The theory of stable recovery [14] tells us that the solution $\hat{\mathbf{z}}_\Delta$ to the convex optimization problem (Eq. 5.7) is a good approximation of $\mathbf{z}_\Delta$, and $\|\mathbf{z}_\Delta - \hat{\mathbf{z}}_\Delta\|_2 \leq c\epsilon$ where c denotes a small constant. In other words, a small error in $\mathbf{y}_\Delta$ only slightly influences reconstruction of $\mathbf{z}_\Delta$. In order to identify missing tags, P-MTI only needs to distinguish the zero and non-zero entries in $\mathbf{z}_\Delta$. The noise may disturb $\mathbf{z}_\Delta$ and render the zero entries in $\mathbf{z}_\Delta$ non-zero (yet remaining small), affecting the detection accuracy. As the error is well bounded by the noise in practice, if the noise is small we can use a threshold θ to accurately classify the contaminated signals with high probabilities. In particular, we define the detection function $f(z_{\Delta i})$ for tag i as follows.

$$f(z_{\Delta i}) = \begin{cases} \text{Present,} & \text{if } \|z_{\Delta i}\|_{\ell_2} < \theta \\ \text{Absent,} & \text{otherwise.} \end{cases} \quad (\text{Eq. 5.8})$$

If the magnitude of $z_{\Delta i}$ is smaller than θ , then tag i is present; otherwise, we say tag i is absent. When the channel condition dramatically deteriorates, the RFID reader cannot always accurately identify the missing tags. In such scenarios, the reader may use the basic P-MTI to identify the missing tags, and monitor the channel $\mathbf{H}$. If the channel becomes reasonably good and stable, P-MTI can switch back to monitor the differential and identify the missing tags. The reader may also increase transmission power to increase the signal strength of backscatter responses, reduce data rates to enhance robustness against noise, and notify coexisting wireless devices to mitigate interferences. Although our approach cannot magically fight against strong channel variation (due to intentional inference, tag mobility, dramatic environment changes), P-MTI is experimentally proven to be robust in reasonably good channel conditions (Section 5.4) and stationary environments (Section 5.3).

5.2.5 Operation Overview

Tag overview. When receiving P-MTI command, each tag generates random bits with its ID as the seed, and backscatters signals if the random bit is 1 or keeps silent otherwise. Each tag repeats the transmission until the reader terminates the interrogation.

Reader overview. The reader initiates one monitoring operation by broadcasting P-MTI command to tags. The reader then measures $M = O(K \log(N/K))$ PHY symbols before terminating the interrogation. The reader decodes the PHY measurements by solving the optimization problem Eq.(Eq. 5.7) and detects missing tags with $f(z_{\Delta i})$ as in Eq.(Eq. 5.8). We discuss parameter settings in Section 5.4.

5.2.6 Multiple Readers

Multiple readers are normally deployed to ensure a full coverage of a large monitoring area [87] due to power constraints, interrogation environment, etc. For instance, current RFID readers can only power up the passive RFID tags within approximately 10m [1, 13, 73]. With more RFID readers, the number of tags in each monitoring area can be reduced, which mitigates the tag-tag contention and collision in the area. Besides, the RFID readers with non-overlapping coverage can interrogate tags in parallel without the reader-reader interference. Here, we explicitly assume that the tags under the coverage of each reader are known to the reader, and multiple readers monitor the tags and identify missing tags in parallel. Thus, an RFID reader only needs to coordinate with its immediate neighbor readers and identify the missing tags with the compressive sensing technique.

Recent works propose to efficiently schedule multiple RFID readers to improve the performance [87]. P-MTI is able to adopt similar coordination strategies to achieve the parallel interrogation of multiple readers. We note that a tag can be in the overlapped coverage of multiple RFID readers. As the readers with overlapped coverage can easily be scheduled into different time slots by the server, the tags will only talk to one reader at a time. Therefore, each RFID reader with non-overlapping coverage area can interrogate the tags in its coverage. Each RFID reader reports all the missing tags as well as the present tags in its coverage to the backend server. The server claims a missing tag event only when a tag is absent in all the monitoring area of the RFID readers. Such a simple data processing task can be easily performed by powerful backend servers.

5.2.7 Discussion

5.2.7.1 Communication Cost

The RFID reader needs to initiate the missing tag identification process by sending a command as well as communication parameters (e.g., data rate, encoding scheme, etc).

Such an initialization is typically in orders of *ms*. As the RFID readers are normally connected with backend server via high speed links, the communication cost between readers and backend server is also small compared with that of the backscatter communication. The transmission time from tags to reader involves $O(K \log(N/K))$ physical layer symbols.

5.2.7.2 Computational Complexity

The computation time of compressive sensing based decoding performed at the backend server turns out to be the major contributor to the overall computation overhead. In our implementation, we use the off-the-shelf CVX solver [2] to decode the aggregated responses and identify the missing tags, which involves computation time of $O(N^3)$. As our scheme needs the access to physical layer information, we implement our scheme using software radios. Physical layer symbols are collected using the software radios and transferred to PC via Ethernet links. Our implementation using commodity PCs can easily cope with the computation tasks in sub-second time with thousands of tags. In P-MTI, both RFID reader and RFID tag only perform lightweight routine computation tasks and the computational overhead is negligible. We note that our scheme needs the slight extensions to EPCglobal Gen-2 protocol, e.g., a new command for missing tag identification and the convex optimization based decoder at the reader, etc.

5.2.7.3 Divide and Conquer

Due to the constraints of antenna sensitivity, the measurement error of RFID reader would increase when large number of tags concurrently respond. Besides, the computation time increases with the number of tags. As the RFID reader knows the number of tags, the RFID reader can adaptively divide the tags into several smaller subsets when necessary. For instance, by querying the tags with a particular p -bit prefix, the tags can be divided into 2^p subsets with fewer tags in each subset using the **Select** primitive

operation of the EPCglobal Gen-2 standard to reduce time complexity as well as to mitigate the interferences. The RFID reader may monitor the mutually exclusive subsets iteratively and combine the results.

5.2.7.4 Channel Dynamics

In practice, wireless channels change over time. P-MTI naturally embraces the channel dynamics. First, the compressive sensing based signal reconstruction is robust to channel noise. P-MTI takes measurement noise (due to channel dynamics, interference, quantization, etc.) into consideration and enhance its robustness based on the theory of stable recovery (Section 5.2.4). Second, the detection function $f(z_{\Delta_i})$ obviates the need of accurate channel measurements. In stationary environment settings (e.g., medicine store, military basis, etc), the channel variation is small. If channel condition changes dramatically during a short period, P-MTI may draw a false detection result. One possible solution is to do extra measurements to ensure the detection results. For instance, the RFID reader may query the potential missing tags to respond immediately. If no response is sent back from the tag, then the reader can confidently conclude its absence. Tag mobility also introduces channel dynamics. While P-MTI primarily focus on monitoring static goods (e.g., in inventory management), our scheme inherently tolerates low mobility scenarios where the channel dynamics are within the stability range as discussed in Section 5.2. Following the existing approaches [43, 72], we focus on the wireless channels without the intentional interference from adversaries. P-MTI may benefit from a wise channel selection (e.g., BLINK [86]) to ensure channel quality.

5.3 Implementation

Although the EPCglobal Gen-2 standard specifies many operations of tags and readers, the practical implementation is left to the manufacturers' choices. Production RFID

Figure 5.4: Testbed: 2 circular antennas are mounted to USRP N210. The USRP N210 is connected via GigE to a laptop which acts as an RFID reader.

Figure 5.5: A WISP tag.

readers (e.g., Alien ALR 9900+ RFID reader [1]) only provide limited interfaces and do not expose physical layer information to users. To explore the lower layer information, we build a prototype missing tag identification system based on the USRP software defined radio and the programmable WISP tags. Figure 5.4 shows the testbed. We implement a prototype software defined RFID reader using USRP N210 based on the GNURadio toolkit and Gen2 RFID project [5]. In particular, one USRP RFX900 daughterboard operating in the 900MHz UHF band [4] is connected to the USRP N210 as the front-end. We connect the daughterboard to Alien ALR-8696-C circular polarized antennas with 8.5dBic antenna gain [1]. The antennas provide sufficient fidelity to detect minute variations of backscattered signals. The USRP N210 is connected via Gigabit Ethernet to a laptop equipped with one qual-core 2.67GHz processor and 2.9GB memory running Ubuntu 10.10. The source code for the USRP N210 RFID reader is available online [8].

P-MTI requires each RFID tag to generate random bits with its ID as the seed. As current commodity passive tags (e.g., Alien ALN tags) do not expose flexible interface

Figure 5.6: MTI process.

Figure 5.7: Received signal strength.

for programming, we prototype with the WISP programmable tags (Figure 5.5). Each WISP tag mainly consists of an antenna circuitry and an ultra-low power 16-bit MSP430 microcontroller. The circuitry is used to harvest energy from the radio frequency signals and backscatter the signals for communication. A capacitor stores the harvested transient energy and supplies power for computation and communication. The EPCglobal Gen-2 protocol [3] has been partially implemented in the WISP firmware, which has most of the necessary components to implement P-MTI. We extend the firmware to let the tags work in concert with the software defined reader to achieve the functionality of missing tag identification. The programming overhead is small and the extension only requires slight

Figure 5.8: Initial offsets among tags.

updates to the EPCglobal Gen-2 standard. Constrained by transmission power of USRP with 200mW and limited number of available WISP tags, currently we can only perform experiments in small-scale static settings. Yet we believe P-MTI can be implemented in large-scale production RFID systems.

The EPCglobal Gen-2 standard specifies a set of routine operations (e.g., **Query**, **Write**, **Select**, **ACK**, etc). We extend the EPCglobal Gen-2 following the conventional reader-initiated approach by adding the missing tag identification routine named **MTI**. **MTI** initiates the missing tag identification process. Figure 5.6 plots the magnitude of received signals at RFID reader during the **MTI** process. The RFID reader initiates the monitoring by transmitting continuous waves to power up the tags. The reader then broadcasts the operation code of **MTI**. When receiving the **MTI** command, each tag sends the pseudo random bits using on-off keying as specified in P-MTI. The aggregated responses from tags are received and decoded at the reader. The reader terminates the monitoring by simply stopping the radio frequency waves.

5.3.1 Backscatter Signals

The differential of aggregated responses based approach requires that the wireless channel remains relatively stable during consecutive measurements. We let 3 WISP tags transmit

Figure 5.9: Received signals at RFID reader: (a) Aggregated signals from 5 tags; (b) Aggregated signals from 4 tags; (c) Differential of the two measurements.

Figure 5.10: Ratio of successfully identified missing tags across different channel conditions.

random bits and measure the received signal strength at the USRP reader in the office environment. Each tag transmits a packet of 32 symbols per second. The RFID reader measures the average signal strength of the 32 symbols. We conduct 40 measurements for each tag, and show the strength of backscatter signals from 3 tags in Figure 5.7. We find that the received backscatter signals remain quite stable during the measurements of 40 seconds. As shown in Figure 5.2, the signal strength remains even more stable within the transmission of one packet.

Figure 5.11: Ratio of successfully identified missing tags with different number of missing tags.

Figure 5.12: Identified and missed tags with different number of missing tags, SNR=17dB, and $K_{\text{Max}}=20$.

5.3.2 Backscatter Synchronization

Passive RFID tags work on harvested energy from radio frequency signals of reader. As the interrogation environment varies, the energy harvest rate differs from tags to tags. Therefore, some tags may wake up earlier than other tags, which introduces response offsets. We set a conservative period of transmitting continuous waves to power the tags before broadcasting the MTI command. The tags transmit signals concurrently after receiving the MTI command. Ideally, the signals from multiple tags would arrive at the reader at the same time. Figure 5.8 shows the CDF of offsets among 7 WISP tags.

According to our experiments on WISP RFID tags, the offsets are within $1.4\mu s$ with the 90th percentile of $0.75\mu s$. When the data rate is 40kbps, a $1.4\mu s$ offset counts for 5.6% of bit width ($25\mu s$) which is sufficiently small for decoding. An independent work [72] reports similar synchronization accuracy on Moo RFID tags [7].

5.3.3 Identifying Missing WISP Tag

We prototype a missing tag identification system where a software defined RFID reader monitors 5 WISP tags. Figure 5.9(a) depicts the received signals at the reader when all the 5 WISP tags are present in the first measurement instance. We intentionally take away one tag to emulate a missing tag event. Figure 5.9(b) plots the received signals from the 4 remaining tags. Figure 5.9(c) plots the differential of the two measurements, and the signals when only the missing tag responds. We note that the magnitude of received signal strength is influenced by the wireless channel as well as the automatic gain control of RFID reader. We consider the influence of automatic gain control as a part of the wireless channel. Although the magnitude of received signal differs due to channel attenuation and gain control, we see that the differential exhibits similar patterns with the signals when only the missing tag responds. With the knowledge of the differential signals, we can accurately identify the missing tag by solving the convex optimization problem (Eq. 5.7). For clarity, here we only present the instance when only one tag is missing. In the experiments when more than one tag are missing, we find that P-MTI can accurately identify them as well.

5.4 Evaluation

In the following, we turn to large-scale simulations to evaluate our missing tag identification scheme. We first analyze the proposed P-MTI. We compare P-MTI with state-of-the-art protocols IIP [43] and Protocol-3 [87].

5.4.1 Simulation Setup

We build a custom simulator based on CVX solver [2] and run the simulations on MATLAB. For P-MTI, we study the various scenarios with different number of tags and readers and wireless channel conditions. For fair comparison, we adopt the same system parameters used in benchmark protocols [43, 87]. In particular, the transmission time of a short response is specified as $t_s = 0.8\text{ms}$ and the transmission time of a 96-bit tag ID is specified as $t_{tag} = 2.4\text{ms}$, which gives an approximate bitrate of 40kbps. We measure the communication time between RFID readers and tags, the initialization time, as well as the protocol execution time in the evaluation. All results are obtained by averaging over 100 runs if not specified otherwise.

5.4.2 P-MTI Investigation

Wireless communication is error-prone in practice, and thus whether P-MTI can accurately identify missing tags in the presence of channel noise is worth investigating. We simulate 1000 tags uniformly distributed in the coverage of each RFID reader, among which 20 tags are missing. We specify the number of measurements $M = CK_{\text{Max}} \log(N)$, where $N = 1000$ and $K_{\text{Max}} = 20$ with different C . We randomly select the missing tags for simulation. Figure 5.10 plots the ratio of successfully identified missing tags to the total number of missing tags across different signal to noise ratio (SNR). As expected, when SNR is very low, P-MTI cannot identify all the missing tags. With the $\text{SNR} \geq 17\text{dB}$ and $C \geq 3$, P-MTI correctly identifies more than 98% of missing tags. P-MTI can further improve the identification accuracy with extra measurements (i.e., with larger C). In Figure 5.10, the marginal accuracy improvement diminishes with the increased number of measurements. For instance, the accuracy with $C = 4$ is only slightly higher than that with $C = 3$. In practice, we set $C = 3$ to balance the accuracy and measurement overhead.

Figure 5.13: Execution time comparison for the single reader case.

Figure 5.14: Execution time comparison for the multiple reader case.

Figure 5.15: Execution time comparison with the different missing tag ratio.

Table 5.1: The error rates of classifying present tags as missing.

SNR	5dB	8dB	11dB	14dB	17dB	20dB
Error	0.42%	0.184%	0.034%	0.005%	0.0%	0.0%

Another parameter that P-MTI needs to specify is the estimated maximum number of missing tags. The number of missing tags can sometimes exceed the estimated maximum number. We evaluate P-MTI with different number of missing tags with $K_{\text{Max}}=20$, under noisy channel conditions (i.e., SNR=14, 17, and 20dB). Figure 5.11 plots the ratio of successfully identified missing tags to the total number of missing tags when the number of missing tags varies from 5 to 40. According to the results, when the number of missing tags is smaller than K_{Max} (i.e., <20), P-MTI can accurately identify the missing tags. When the number of missing tags exceeds K_{Max} , P-MTI cannot identify all the missing tags. Therefore, the RFID reader needs to adjust K_{Max} to ensure that no more than K_{Max} tags would be missing. Figure 5.12 plots in detail the missing tag identification results with different number of missing tags, SNR=17dB, and $K_{\text{Max}}=20$. Although P-MTI may fail to identify some tags when the number of missing tags exceeds K_{Max} , the number of successfully identified missing tags would become very close to or even exceed K_{Max} . When the number of missing tags exceeds 20, however, the number of successfully identified ones becomes larger than 20. In such cases, the reader can adaptively increase K_{Max} to ensure that K_{Max} is indeed larger than the number of missing ones. A larger K_{Max} can correct the error due to underestimation of missing tags at the cost of the increased communication overhead. We note that it remains challenging to set the optimal value of K_{Max} . One may use the tag cardinality estimation schemes [33, 53, 90] to adapt K_{Max} when the number of missing tags exceed K_{Max} . We show in the following that even in the most conservative setting (i.e., $K_{\text{Max}} = N$), P-MTI still achieves much higher processing efficiency compared with prior schemes.

We evaluate whether P-MTI would wrongly classify present tags as missing tags. In Table 5.1, we measure the error rates of classifying present tags as missing tags across different SNRs. According to the experiment results, on average only 1.8 out of 980 (i.e., 0.184%) present tags are wrongly considered as missing tags when SNR=8dB. Generally, the error rates decrease as the SNR increases. In particular, no present tags are wrongly considered missing when SNR>14dB.

5.4.3 Performance Comparison

We compare P-MTI with the most recent protocols IIP [43] and Protocol-3 [87]. We adopt the same system parameters used in [43, 87]. As IIP and Protocol-3 do not target at noisy channels, we do not simulate channel errors in the comparison study. For completeness, P-MTI however uses conservative parameter settings (e.g., $M = 3K_{\max} \log(N)$) which would favor the benchmark protocols. We investigate the transmission time of backscatter responses from tags to readers, the initialization overhead, as well as the protocol execution time in the evaluation. We note that the overall execution time is largely determined by the transmission time of backscatter responses from tags to readers, while the other communication overhead is very small (e.g., initialization time, reader to server transmission, etc).

Single reader. We note that Protocol-3 and P-MTI schedule multiple readers and achieve interrogation among non-overlapping readers in parallel, while IIP views multiple readers as a single logical reader and does not benefit from parallel interrogation. Therefore, we first study the single reader case and compare IIP, Protocol-3, and P-MTI. Figure 5.13 compares the overall execution time of IIP, Protocol-3, P-MTI (with $K_{\max} = 0.2N$), and P-MTI (with $K_{\max} = N$) with varied number of tags. According to the results, we find that P-MTI with conservative parameter settings of $K_{\max} = N$ can reduce the execution time by approximately 64% compared with IIP and Protocol-3. This

performance gain stems from the efficient use of physical layer information. With more realistic parameter settings of $K_{\text{max}} = 0.2N$, P-MTI achieves even higher efficiency. The performance gain stems from the compressive sensing based information reconstruction.

Multiple readers. We compare the performance of P-MTI with Protocol-3 in the scenarios of multiple RFID readers. To study the scalability of different schemes, we simulate 50,000 tags with varied number of readers. We use the same RFID network topology for P-MTI and Protocol-3 and investigate the overall execution time. As shown in Figure 5.14, with more readers both P-MTI and Protocol-3 can effectively reduce the execution time by dividing the tags into smaller subsets and interrogating them in parallel. Nevertheless, the increased number of readers requires extra deployment costs. Thus it is yet significant to further improve the communication efficiency. According to our experiment results, P-MTI can significantly reduce the execution time compared with Protocol-3, with/without the knowledge of missing tag ratio. In particular, P-MTI with conservative parameter setting only takes about 33% of the execution time of Protocol-3.

Number of missing tags. We compare the overall execution time of P-MTI with Protocol-3 with different ratio of missing tags. We simulate 50,000 tags in the coverage of 50 RFID readers. Figure 5.15 plots the execution time with different missing tag ratios varied from 0% to 50%, which covers the typical missing tag events in practical applications. The adaptive P-MTI adaptively increases K_{max} when the number of missing tags becomes close to or exceeds K_{max} . The conservative P-MTI conservatively sets $K_{\text{max}} = 0.5N$. From Figure 5.15, we find that the execution time of adaptive P-MTI increases linearly with the missing tag ratio, while the execution time of conservative P-MTI remains almost constant. We also find that the execution time of both adaptive and conservative P-MTI is much smaller than that of Protocol-3 across different missing tag ratios.

5.5 Summary

In this chapter, we study the missing tag identification problem in large-scale RFID systems. We propose P-MTI to leverage physical layer information and substantially improve monitoring efficiency. We further present several optimization techniques to improve the performance. P-MTI leverages the sparsity of missing tag events and reconstructs tag responses through compressive sensing. To validate its efficacy, we implement a prototype system and extend the EPCglobal Gen-2 standard based on the GNUMRadio/USRP and WISP platform. We do extensive evaluation of P-MTI with large-scale simulations. The results demonstrate that P-MTI substantially outperforms the state-of-the-art schemes.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this dissertation, we propose several novel RFID protocols to fundamentally improve the operational efficiency of large-scale RFID systems. Specifically, we mainly study three fundamental problems – cardinality estimation, tag search, and missing tag identification.

For cardinality estimation, we first propose the probabilistic estimating tree (PET) based approach which substantially reduces execution time to meet a certain estimation accuracy requirement compared with prior schemes. PET achieves the higher performance over other schemes mainly because of its bisection search speedup, during the search of some key nodes along a search path. We present detailed mathematical analysis on PET protocol and its performance. To further improve cardinality estimation performance, we propose the zero-one estimator (ZOE) which makes best use of each frame slot, and extract most information from each slot during two-phase estimation process. The first phase of ZOE adjusts the participation probability of each tag so that we can aggregate maximum information from collective participation of tags. ZOE significantly reduces the execution time compared with state-of-the-art schemes and increases operation efficiency of RFID systems in cardinality estimation.

For RFID tag search, we propose several algorithms to meet the stringent delay requirement. We formally formulate the tag search problem and propose the first tag search

protocol in large-scale RFID systems. In particular, we propose to utilize compact approximators to efficiently aggregate a large volume of RFID tag information and exchange such information through two-phase communication. By estimating the intersection of two compact approximators, the proposed two-phase compact approximator based tag search (CATS) protocol significantly reduces the execution time compared with all alternative solutions. We conduct comprehensive simulations to validate our design, and the results demonstrate that the proposed tag searching protocol is highly efficient in terms of both communication and computation efficiency.

For missing tag identification, we propose the first physical-layer missing tag identification (P-MTI) protocol which recovers physical layer collisions and substantially improves monitoring efficiency. P-MTI further exploits the sparsity of missing tag events and reconstructs tag responses through compressive sensing based collision recovery. To validate its efficacy, we implement a prototype system and extend the EPCglobal Gen-2 standard based on the GNURadio/USRP and WISP platform. We evaluate P-MTI with large-scale simulations which demonstrates P-MTI substantially outperforms the state-of-the-art schemes.

6.2 Future Work

To fundamentally improve the protocol efficiency, we conducted initial experiments using the USRP software defined radio and the WISP programmable tags. We see that current RFID physical layer design is generally oversimplified and rich information is simply discarded. As a result, the RFID upper layer only receives limited information passed up from the lower layer.

Motivated by this observation, we proposed the first physical-layer missing tag identification protocol, P-MTI, which outperforms prior schemes and surprisingly exceeds their

theoretical bounds, because our physical-layer approach is able to extract more information from the lower layer while prior schemes are inherently restricted by the limited information available at the upper layer.

We believe that careful physical-layer designs for RFID communication would bring more benefits and further improve its operation efficiency. For instance, prior schemes (including the proposed PET and ZOE) only detect idle and busy slots, and do not differentiate the exact number of tags colliding in a busy slot. When we examine the collision signals at physical layer, although we cannot decode the collisions we still have fair chances to accurately estimate the number of tags in a collision slot. In the future, we plan to explore such an opportunity and improve cardinality estimation efficiency by analyzing physical-layer collisions.

Tag search protocols can also benefit from more efficient physical-layer design. For instance, if we can accurately tell the number of collision tags in each collision slot, the bit vectors used to encode tag responses would become the integer vectors, which expose more information per slot measurement. As a result, we may reduce the length of the vectors and cut transmission overhead.

Beyond that, how should we optimize RFID network stacks to efficiently support a variety of applications? Current RFID network protocols enhance the communication robustness at the cost of lower communication efficiency. As the RFID based sensors are widely deployed in the future, the throughput of RFID communication will eventually become a bottleneck which calls for a new network paradigm to strike a balance between robustness and efficiency for RFID communication. We plan to design the next-generation PHY/MAC for RFID networks to increase the RFID throughput and meanwhile ensure communication robustness.

Appendix A

Author's Publications related to this Dissertation

- (i) [88] Yuanqing Zheng, Mo Li, “PET: Probabilistic Estimating Tree for Large-Scale RFID Estimation”, *IEEE Transactions on Mobile Computing*, vol. 11, issue 11, pages 1763–1774, November 2012.

This extends the conference version published in *IEEE ICDCS*, 2011.

- (ii) [89] Yuanqing Zheng, Mo Li, “Fast Tag Searching Protocol for Large-Scale RFID Systems”, *IEEE/ACM Transactions on Networking*, vol. 21, issue 3, pages 924–934, June 2013.

This extends the conference version published in *IEEE ICNP*, 2011.

- (iii) [90] Yuanqing Zheng, Mo Li, “Towards More Efficient Cardinality Estimation for Large-Scale RFID Systems”, *IEEE/ACM Transactions on Networking*, in press, 2013.

This extends the conference version published in *IEEE INFOCOM*, 2013.

- (iv) [91] Yuanqing Zheng, Mo Li, “P-MTI: Physical-layer Missing Tag Identification via Compressive Sensing”, *IEEE/ACM Transactions on Networking*, in press, 2014.

This extends the conference version published in *IEEE INFOCOM*, 2013.

- (v) [92] Yuanqing Zheng, Mo Li, “Read Bulk Data from Computational RFIDs”, in *IEEE INFOCOM*, 2014.

References

- [1] “Alien Technology”, <http://www.alientechnology.com>
- [2] “CVX Research”, <http://cvxr.com/cvx/>
- [3] “EPC Radio-Frequency Identity Protocols Class-1 Gen-2 UHF RFID Protocol for Communications at 860MHz-960MHz, EPCglobal”, <http://www.epcglobalinc.org/standards/uhfc1g2>, Apr. 2011.
- [4] “Ettus Research”, <http://www.ettus.com>
- [5] “Gen 2 RFID Tools”, <https://www.cgran.org/wiki/Gen2>
- [6] “ ℓ_1 -Magic”, <http://users.ece.gatech.edu/~justin/l1magic/>
- [7] H. Zhang, J. Gummesson, B. Ransford, and K. Fu, “Moo: A batteryless computational RFID and sensing platform”, Tech Report UMASS, 2011. <http://spqr.cs.umass.edu/moo/>
- [8] “Open RFID Lab”, <http://pdcc.ntu.edu.sg/wands/ORL/>
- [9] M. Abramowitz and I. A. Stegun, “Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables”, *Dover Publications*, 1972.
- [10] A. Broder and M. Mitzenmacher, “Network Applications of Bloom Filters: A Survey”, *Internet Math.*, vol. 1, issue. 4, pp. 485-509, 2003.
- [11] S. Boyd, “Convex Optimization”, Cambridge University Press, 2004.
- [12] K. Bu, B. Xiao, Q. Xiao, and S. Chen, “Efficient Pinpointing of Misplaced Tags in Large RFID Systems”, in *IEEE SECON*, 2011.

- [13] M. Buettner and D. Wetherall, “An Empirical Study of UHF RFID Performance”, in *Proc. of ACM MobiCom*, 2008.
- [14] E. Candès, J. Romberg, T. Tao, “Stable Signal Recovery from Incomplete and Inaccurate Measurements”, *Communications on Pure and Applied Mathematics*, vol. 59, issue 9. pp. 1207-1223, 2006.
- [15] Z. Cao, C. Sutton, Y. Diao, and P. Shenoy, “Distributed Inference and Query Processing for RFID Tracking and Monitoring”, in *Proc. of VLDB*, 2011.
- [16] J. I. Capetanakis, “Tree algorithms for packet broadcast channels”, *IEEE Trans. on Information Theory*, vol. 25, issue 5, pp. 505-515, 1979.
- [17] B. Chen, Z. Zhou, Y. Zhao, and H. Yu, “Efficient Error Estimating Coding: Feasibility and Applications”, in *Proc. of ACM SIGCOMM*, 2010.
- [18] B. Chen, Z. Zhou, and H. Yu, “Understanding RFID Counting Protocols”, in *Proc. of ACM MobiCom*, 2013.
- [19] H. Chen, W.-S. Ku, H. Wang, M.-T. Sun, “Leveraging Spatio-Temporal Redundancy for RFID Data Cleansing”, in *Proc. of ACM SIGMOD*, 2010.
- [20] M. Chen, W. Luo, Z. Mo, S. Chen, and Y. Fang, “An Efficient Tag Search Protocol in Large-Scale RFID Systems”, in *Proc. of IEEE INFOCOM*, 2013.
- [21] S. Chen, M. Zhang, and B. Xiao, “Efficient Information Collection Protocols for Sensor-augmented RFID Networks”, in *Proc. of IEEE INFOCOM*, 2011.
- [22] J. Considine, F. Li, G. Kollios, and J. Byers, “Approximate Aggregation Techniques for Sensor Databases”, in *Proc. of IEEE ICDE*, 2004.
- [23] D. Donoho, “Compressed sensing”, *IEEE Transactions on Information Theory*, vol. 52, issue 4, pp. 1289-1306, April 2006.
- [24] M. Durand and P. Flajolet, “Loglog counting of large cardinalities (extended abstract)”, in *Proc. of ESA*, 2003.
- [25] P. Flajolet and G. N. Martin, “Probabilistic counting algorithms for data base

- applications”, *Journal of Computer and System Science*, vol. 31, issue 2, pp. 182-209, 1985.
- [26] P. Flajolet, X. Gourdon, and P. Dumas, “Mellin transforms and asymptotics: Harmonic sums”, *Theoretical Computer Science*, vol.144, issues 1-2, pp. 3-58, 1995.
- [27] K. Finkenzeller, “RFID Handbook: Radio-Frequency Identification Fundamentals and Applications”, *John Wiley & Sons*, 2000.
- [28] S. Ganguly, M. Garofalakis, and R. Rastogi, “Processing set expressions over continuous update streams”, in *Proc. of ACM SIGMOD*, 2003.
- [29] S. Garfinkel, A. Juels, and R. Pappu, “RFID Privacy: An Overview of Problems and Proposed Solutions”, *IEEE Security and Privacy*, vol. 3, issue 3, pp. 34-43, 2005.
- [30] D. Guo, Y. Liu, X. Li, and P. Yang, “False Negative Problem of Counting Bloom Filter”, *IEEE Trans. on Knowledge and Data Engineering*, vol. 22, issue 5, pp. 651-664, 2010.
- [31] G. R. Grimmett, D. R. Stirzaker, “Probability and Random Processes, 3rd edition”, *Oxford University Press*, 2001.
- [32] P. J. Haas, J. F. Naughton, S. Seshadri, and L. Stokes, “Sampling-Based Estimation of the Number of Distinct Values of an Attribute”, in *Proc. of VLDB*, 1995.
- [33] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, “Counting RFID Tags Efficiently and Anonymously”, in *Proc. of IEEE INFOCOM*, 2010.
- [34] D. Holcomb, W. Burleson, and K. Fu, “Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers”, *IEEE Trans. on Computers*, vol. 58, Issue 9, pp. 1198-1210, 2009
- A. Juels, “RFID Security and Privacy: A Research Survey”, *IEEE Journal on*

- Seleted Areas in Communications*, vol. 24, issue 2, pp. 381-394, 2006.
- [35] P. Kirschenhofer and H. Prodinger, “On the analysis of probabilistic counting”, *Lecture Notes in Mathematics*, vol. 1452, pp. 117-120, 1990.
- [36] D. E. Knuth, “The Art of Computer Programming, 2nd Edition, vol. 3: Sorting and Searching”, *Addison-Wesley*, 1998.
- [37] M. Kodialam and T. Nandagopal, “Fast and reliable estimation schemes in RFID systems”, in *Proc. of ACM MobiCom*, 2006.
- [38] M. Kodialam, T. Nandagopal, and W. C. Lau, “Anonymous Tracking using RFID tags”, in *Proc. of IEEE INFOCOM*, 2007.
- [39] T. F. La Porta, G. Maselli, and C. Petrioli, “Anticollision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization”, *IEEE Trans. on Mobile Computing*, vol. 10, issue 2, pp. 267-279, 2011.
- [40] C.-H. Lee and C.-W. Chung, “Efficient storage scheme and query processing for supply chain management using RFID”, in *Proc. of ACM SIGMOD*, 2008.
- [41] J. Lee, T. Kwon, Y. Choi, S. K. Das, and K.-a. Kim, “Analysis of RFID anti-collision algorithms using smart antennas”, in *Proc. of ACM SenSys*, 2004.
- [42] S.-R. Lee, S.-D. Joo, and C.-W. Lee, “An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification”, in *Proc. of IEEE MobiQuitous*, 2005.
- [43] T. Li, S. Chen, and Y. Ling, “Identifying the Missing Tags in a Large RFID System”, in *Proc. of ACM MobiHoc*, 2010.
- [44] T. Li, S. Wu, S. Chen, and M. Yang, “Energy Efficient Algorithms for the RFID Estimation Problem”, in *Proc. of IEEE INFOCOM*, 2010.
- [45] Y. Liu, L. Chen, J. Pei, Q. Chen, and Y. Zhao, “Mining Frequent Trajectory Patterns for Activity Monitoring Using Radio Frequency Tag Arrays”, in *Proc. of IEEE PerCom*, 2007.

- [46] L. Lu, Y. Liu, and X. Li, “Refresh: Weak Privacy Model for RFID Systems”, in *Proc. of IEEE INFOCOM*, 2010.
- [47] C. Luo, F. Wu, J. Sun, C. W. Chen, “Compressive Data Gathering for Large-Scale Wireless Sensor Networks”, in *ACM MobiCom*, 2009.
- [48] K. Michael and L. McCathie, “The pros and cons of RFID in supply chain management”, in *Proc. of International Conference on Mobile Business*, pp. 623-629, 2005.
- [49] J. Myung and W. Lee, “Adaptive Splitting Protocols for RFID Tag Collision Arbitration”, in *Proc. of ACM MobiHoc*, 2006.
- [50] V. Namboodiri and L. Gao, “Energy-Aware Tag Anti-Collision Protocols for RFID systems”, in *Proc. of IEEE PerCom*, 2007.
- [51] L. M. Ni, Y. Liu, Y. C. Lau, and A. Patil, “LANDMARC: Indoor Location Sensing Using Active RFID”, in *Proc. of IEEE PerCom*, 2003.
- [52] C. Qian, Y. Liu, H. Ngan, and L. M. Ni, “ASAP: Scalable Identification and Counting for Contactless RFID Systems”, in *Proc. of IEEE ICDCS*, 2010.
- [53] C. Qian, H. Ngan, and Y. Liu, “Cardinality Estimation for Large-scale RFID Systems”, in *Proc. of IEEE PerCom*, 2008.
- [54] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, “A General Model of Wireless Interference”, in *Proc. of ACM MobiCom*, 2007.
- [55] Y. Qiao, S. Chen, T. Li, S. Chen, “Energy-efficient Polling Protocols in RFID Systems”, in *Proc. of ACM MobiHoc*, 2011.
- [56] Z. Ren, C. C. Tan, D. Wang, and Q. Li, “Experimental Study on Mobile RFID Performance”, in *Proc. of WASA*, 2009.
- [57] L. G. Roberts, “Aloha Packet System with and without Slots and Capture ”, *ACM SIGCOMM Computer Communication Review*, vol. 5, issue 2, pp. 28-42, 1975.

- [58] M. Shahzad, A. X. Liu, “Every bit counts: Fast and scalable RFID estimation”, in *Proc. of ACM MobiCom*, 2012.
- [59] M. Shahzad, A. X. Liu, “Probabilistic Optimal Tree Hopping for RFID Identification”, in *Proc. of ACM SIGMETRICS*, 2013.
- [60] S. Sarma, D. Brock, and D. Engels, “Radio Frequency Identification and the Electronic Product Code ”, *IEEE Micro*, vol. 21, issue 6, pp. 50-54, 2001.
- [61] T. A. Scharfeld, “An analysis of the fundamental constraints on low cost passive radio-frequency identification system design”, *M.S. thesis, Massachusetts Institute of Technology*.
- [62] B. Sheng, Q. Li, W. Mao, “Efficient Continuous Scanning in RFID Systems”, in *Proc. of IEEE INFOCOM*, 2010.
- [63] B. Sheng, C. C. Tan, Q. Li, and W. Mao, “Finding Popular Categories for RFID Tags”, in *Proc. of ACM MobiHoc*, pp. 159-168, May, 2010.
- [64] S. Sundaresan, R. Doss, S. Piramuthu, and W. Zhou, “Secure Tag Search in RFID Systems Using Mobile Readers”, *IEEE Transactions on Dependable and Secure Computing*, in press, 2014.
- [65] W. K. Sze, W. C. Lau, and O. C. Yue, “Fast RFID Counting under Unreliable Radio Channels”, in *Proc. of IEEE ICC*, 2009.
- [66] C. C. Tan, B. Sheng, and Q. Li, “How to Monitor for Missing RFID Tags”, in *IEEE ICDCS*, 2008.
- [67] C. C. Tan, L. Xie, and Q. Li, “Privacy Protection for RFID-based Tracking Systems”, in *Proc. of IEEE RFID*, pp. 53-6-, April, 2010.
- [68] C. C. Tan, B. Sheng, and Q. Li, “Serverless Search and Authentication Protocols for RFID”, in *Proc. of IEEE PerCom*, 2007.
- [69] S. Tang, J. Yuan, X.-Y. Li, G. Chen, Y. Liu, and J. Zhao, “RASPberry: A Stable Reader Activation Scheduling Protocol in Multi-Reader RFID Systems”,

- in *Proc. of IEEE ICNP*, 2009.
- [70] H. Tijms, “Understanding Probability: Chance Rules in Every Life”, *Cambridge University Press*, 2004.
- [71] R. Want, “An Introduction to RFID Technology”, *IEEE Pervasive Computing*, vol. 5, issue 1, pp. 25- 33, 2005.
- [72] J. Wang, H. Hassanieh, D. Katabi, P. Indyk, “Efficient and Reliable Low-Power Backscatter Networks”, in *ACM SIGCOMM*, 2012.
- [73] J. Wang, D. Katabi, “Dude, Wheres My Card? RFID Positioning That Works with Multipath and Non-Line of Sight”, in *ACM SIGCOMM*, 2013.
- [74] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li, and D. Chen, “Efficient Tag Identification in Mobile RFID Systems”, in *Proc. of INFOCOM*, 2010.
- [75] X. Xu, L. Gu, J. Wang, and G. Xing, “Negotiate Power and Performance in the Reality of RFID Systems”, in *Proc. of IEEE PerCom*, 2010.
- [76] L. Yang, J. Han, Y. Qi, C. Wang, Y. Liu, Y. Chen, and X. Zhong, “Revisiting Tag Collision Problem in RFID Systems”, in *Proc. of ICPP*, 2010.
- [77] L. Yang, J. Han, Y. Qi, C. Wang, Z. Liu, Q. Yao, Y. Chen, and X. Zhong, “Utilizing RF Interference to Enable Private Estimation in RFID Systems”, in *Proc. of IEEE ICPADS*, 2010.
- [78] L. Yang, J. Han, Y. Qi, and Y. Liu, “Identification-Free Batch Authentication for RFID Tags”, in *Proc. of IEEE ICNP*, 2010.
- [79] L. Yang, J. Han, Y. Qi, C. Wang, T. Gu, and Y. Liu, “Season: Shelving Interference and Joint Identification in Large-scale RFID Systems”, in *Proc. of IEEE INFOCOM*, 2011.
- [80] D. Yeager, A. Sample, and J. Smith, “WISP: A Passively Powered UHF RFID Tag with Sensing and Computation”, *CRC Press*, March 2008.
- [81] M. Yang, H. Wang, H. Chen, and W.-S. Ku, “Querying Uncertain Data with

- Aggregate Constraints”, in *Proc. of ACM SIGMOD*, 2011.
- [82] D. Zhang, J. Ma, Q. Chen, and L. M. Ni, “An RF-based System for Tracking Transceiver-free Objects”, in *Proc. of IEEE PerCom*, 2007.
- [83] D. Zhang, J. Zhou, M. Guo, J. Cao, and T. Li, “TASA: Tag-Free Activity Sensing Using RFID Tag Arrays”, *IEEE Trans. on Parallel and Distributed Systems*, vol. 22, issue 4, pp. 558-570, 2011.
- [84] S. Zhang, X. He, H. Song, and D. Zhang, “Time Efficient Tag Searching in Multiple Reader RFID Systems”, in *Proc. of IEEE GreenCom, IEEE iThings/CPSCoM*, 2013.
- [85] M. Zhang, T. Li, S. Chen, and B. Li, “Using Analog Network Coding to Improve the RFID Reading Throughput”, in *Proc. of IEEE ICDCS*, 2010.
- [86] P. Zhang, J. Gummesson, D. Ganesan, “BLINK: A High Throughput Link Layer for Backscatter Communication”, in *ACM MobiSys*, 2012.
- [87] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, “Fast Identification of the Missing Tags in a Large RFID System”, in *Proc. of IEEE SECON*, 2011.
- [88] Y. Zheng, M. Li “PET: Probabilistic Estimating Tree for Large-Scale RFID Estimation”, *IEEE Transactions on Mobile Computing*, vol. 11, issue 11, pp. 1763-1774, November 2012.
- [89] Y. Zheng, M. Li, “Fast Tag Searching Protocol for Large-Scale RFID Systems”, *IEEE/ACM Transactions on Networking*, vol. 21, issue 3, pp. 924-934, June 2013.
- [90] Y. Zheng, M. Li, “Towards More Efficient Cardinality Estimation for Large-Scale RFID Systems”, *IEEE/ACM Transactions on Networking*, in press, 2013.
- [91] Y. Zheng, M. Li, “P-MTI: Physical-layer Missing Tag Identification via Compressive Sensing”, *IEEE/ACM Transactions on Networking*, in press, 2014.
- [92] Y. Zheng, M. Li, “Read Bulk Data from Computational RFIDs”, in *Proc. of*

REFERENCES

- IEEE INFOCOM*, 2014.
- [93] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, “Evaluating and optimizing power consumption of anti-collision protocols for applications in rfid systems”, in *Proc. of ISLPED*, 2004.
- [94] Z. Zhou, H. Gupta, S. R. Das, and X. Zhu, “Slotted Scheduled Tag Access in Multi-Reader RFID Systems”, in *Proc. of IEEE ICNP*, 2007.