UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

FRAMEWORK FOR IMPROVING PERFORMANCE OF PROTOCOLS FOR READING RADIO FREQUENCY IDENTIFICATION TAGS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

CHANDRIKA J. SATYAVOLU
Norman, Oklahoma
2014

FRAMEWORK FOR IMPROVING PERFORMANCE OF PROTOCOLS FOR READING RADIO FREQUENCY IDENTIFICATION TAGS

A DISSERTATION APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY

Dr. Sridhar Radhakrishnan, Chair

Dr. Sesh Commuri

Dr. Mohammed Atiquzzaman

Dr. Changwook Kim

Dr. S. Lakshmivarahan

To Nanna, my light at the end of tunnel
To Tatagaru, my moral compass

Acknowledgements

Over the last five years, I have received support and encouragement from several individuals. I would like to express my deepest gratitude to my advisor, Dr. Sridhar Radhakrishnan, for his excellent guidance, caring and patience. His mentorship has made this a thoughtful and rewarding journey. I would like to thank my dissertation committee of Dr. Sesh Commuri, Dr. Mohammed Atiquzzaman, Dr. Changwook Kim and Dr. Lakshmivarahan for their valuable time, and invaluable feedback.

In addition, I would like to extend my thanks to Dr. Venkatesh Sarangan and Dr. Jessica Ruyle for providing crucial insights into my research, Dr. Thomas Landers for providing me with an excellent atmosphere for doing research, Dr. Mahendran Veeramani for being a great colleague and partner for brainstorming sessions, Amlan Chatterjee, Dr. Hasan Khondker and Asif Adnan, my fellow PhD students and colleagues for listening to and validating my ideas, my husband Arun for being there through the good and bad days, and my dogs for putting a smile on my face.

Finally, I'd like to extend my gratitude and thanks to my parents, my sister and the rest of my family, without whom I would not be the person I am, and be here doing this work I love.

Contents

List of Tables

List of Figures

Abstract

Radio-frequency Identification (RFID) is a highly sought-after wireless technology used to track and manage inventory in the supply chain industry. It has varied applications ranging from automated toll collection and security access management to supply chain logistics. Miniaturization and low tag costs of RFID tags have lead to item-level tagging, where not just the pallet holding products is tagged but each product inside has a tag attached to it. Item-level tagging of goods improves the accuracy of the supply chain but it significantly increases the number of tags that an RFID reader must identify and track. Faster identification is crucial to cutting cost and improving efficiency.

Existing RFID protocols were designed to primarily handle static scenarios with both RFID tags and readers not being in motion. This research addresses the problem of inventory tracking within a warehouse in multitude of scenarios that involves mobile tags, multiple readers and high density environments. Mobility models are presented and frameworks are developed for the following scenarios: a) mobile tags on a conveyor belt with multiple fixed readers; b) mobile reader in a warehouse with stationary tags in shelves; and c) high density tag population with Near-Field (NF) communication.

The proposed frameworks use information sharing among readers to facilitate protocol state handoff and segregation of tags into virtual zones to improve tag reading rates in mobile tag and mobile reader scenarios respectively. Further, a tag's ability to listen to its Near-Field neighboring tags transmissions is exploited to assist the reader in resolving collisions and hence enhancing throughput. The frameworks discussed in this research are mathematically modeled with a probabilistic analysis of protocols employed in conjunction with framework.

With an increased number of tags to be identified, mathematically understanding the performance of the protocol in these large-scale RFID systems becomes essential. Typically, this analysis is performed using Markov-chain models. However, these analyses suffer from the common state-space explosion problem. Hence, it is essential to come up with a scalable analysis, whose computation model is insensitive to the number of tags. The following research analyzes the performance of tag identification protocols in highly dense tag scenarios, and proposes an empirical formula to estimate the approximate time required to read all the tags in a readers range without requiring protocol execution.

Chapter 1

Introduction

Radio Frequency Identification (RFID) is a wireless identification technology that uses radio frequency waves to track and manage tagged inventory in the supply chain industry. RFID tags contain electronically stored product specific information. RFID reader powers the tags through electromagnetic induction and retrieves the information stored in tags. Reader does not require the tag to be in line of sight to obtain tag information. Tags need to be within a certain distance from the reader's electromagnetic field to communicate with the reader. This property makes the use of RFID technology more convenient and effective than barcodes, especially in warehouse environments. Similar to barcode technology, the cost of RFID reader is a one time installation cost and the tags add up to recurring cost. RFID tags cost very little when purchased in large quantities and can hold large amounts of information. Also, the information on the tags can be overwritten with new data and tags can be reused. Tags are becoming smaller with advances in RFID technology. They are as small as 1 cm^2 and can be affixed to small product packaging like pharmaceuticals, cell phones, etc. Products can be tracked through various stages of manufacture to supply and retail. Miniaturization and low cost of tags have made RFID a more viable identification technology for massive retail warehouses.

1.1 Radio Frequency Bands

The RFID system can operate in certain bands in the Low Frequency (LF), High Frequency (HF), Ultra High Frequency (UHF) or Super High Frequency (SHF) ranges.

These frequency bands are placed widely separated within the radio frequency spectrum which enables RFID to choose frequencies based on the underlying application. The operating distance or range of RFID reader also depends on the frequency band chosen.

The table in Figure 1.1 lists the possible operating frequencies of an RFID system, the range of operations and applications of each frequency band [7].

RFID FREQUENCY BAND / SPECTRUM ALLOCATIONS			
RFID FREQUENCY BAND	FREQUENCY BAND DESCRIPTION	TYPICAL RANGE	TYPICAL RFID APPLICATIONS
125-134.2 kHz and 140-148.5 kHz	Low frequency	Up to ~ 1/2 metre	These frequencies can be used globally without a license. Often used for vehicle identification. Sometimes referred to as LowFID.
6.765 - 6.795 MHz	Medium frequency		Inductive coupling is used on these RFID frequencies.
13.553 - 13.567 MHz	High Frequency HF Often called 13.56 MHz	Up to ~ 1 metre	These RFID frequencies are typically used for electronic ticketing, contactless payment, access control, garment tracking, etc
26.957 - 27.283 MHz	Medium frequency	Up to ~ 1 metre	Inductive coupling only, and used for special applications.
433 MHz	UHF		These RFID frequencies are used with backscatter coupling, for applications such as remote car keys in Europe
858 - 930 MHz	Ultra High Frequency UHF	1 to 10 metres	These RFID frequencies cannot be accessed globally and there are significant restrictions on their use. When they are used, it is often used for asset management, container tracking, baggage tracking, work in progress tracking, etc. and often in conjunction with Wi-Fi systems.
2.400 - 2.483 GHz	SHF		Backscatter coupling, but only available in USA / Canada
2.446 - 2.454GHz	SHF	3 metres upwards	These RFID frequencies are used for long range tracking and with active tags, RFID and AVI (Automatic Vehicle Identification). Backscatter coupling is generally used.
5.725 - 5.875 GHz	SHF		Backscatter coupling. Not widely used for RFID.

Figure 1.1: RFID Frequency Band Spectrum

1.2 Basic Components of RFID System

An RFID system is made up of three components [16], [21], [1]:

- RFID tag or transponder

- RFID reader or interrogator
- A computer that controls the reader and manages data

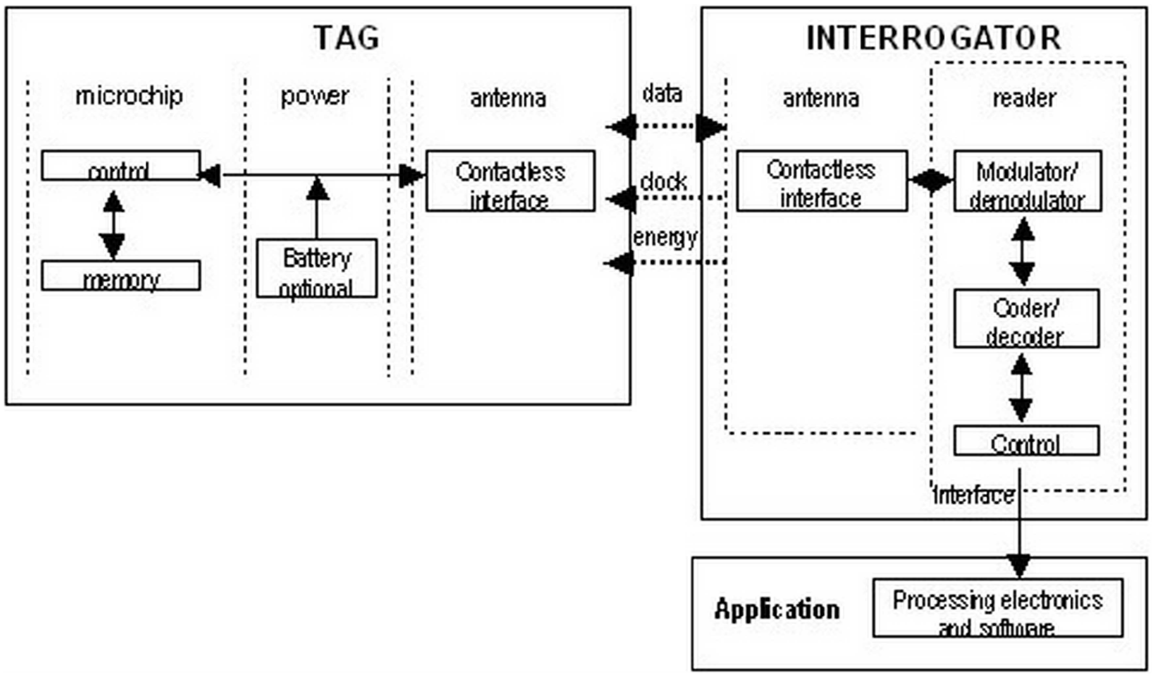


Figure 1.2: Components of RFID system

Figure 1.2 shows the components that make up an RFID system [1]. An RFID tag consists of a tiny microchip with an antenna attached. It can pick up signals from and sends signals to a reader if it is within the operating range of the reader. The tag stores a unique identification number (ID) also known as the Electronic Product Code (EPC) that corresponds to the product it is attached to. The EPC is nothing but a long binary number made up of 0's and 1's. Tag may also contain product-specific information, such as manufacture date and expiration date, that makes it easier to track the product. Tags that hold power or more functionalities cost more than the tags that do not. Most inexpensive tags used in inventory tracking need to be powered by the reader and have minimal storage and functions.

Reader or interrogator emits Radio Frequency (RF) waves that activate the tags or transponders within the range of the reader. The reader has a transmitter/receiver pair and an antenna that radiates energy to send commands and clock signals to the tags and receives IDs from the tags. The tags do not have any power of their own, they receive energy from the reader, harness that energy and respond back to the reader at an appropriate time. The reader can have one or more antennas.

The frequencies that an RFID interrogator can use to communicate with the tags depend on the application and performance with respect to reading range and/or environmental constraints as discussed in Section 1.1. RFID systems generally operate in shorter ranges.

The reader is connected to an external computer executing some application to process the data received from the tags. The reader passes on the EPCs and other information it collects from the tags to this computer.

1.3 RFID Tag Classification

The RFID tags can be broadly classified into three categories depending on their functionalities. [16], [19], [21]

- Passive tags
- Semi-passive tags
- Active tags

They can be further classified into six classes. Class 0, 1 and 2 tags come under the category of passive tags. Class 0 and 1 tags are called the identity tags. They are the basic functionality tags that store a unique EPC and need power from the reader and are essentially used for resource discovery. Class 1 tags are again classified into Generation 1 and Generation 2 tags. Class 1 Generation 1 tags are those that work with the reader that has the same manufacturer as the tag. Class 1 Generation 2 tags are those that work with any reader independent of its manufacturer. Class 2 tags are called the higher functionality tags and can store their unique EPC and also have some memory read-and-write operations on the tag.

Class 3 tags are called the semi-passive tags that can function in the absence of power from the reader.

Class 4 and 5 tags come under the category of active tags. Class 4 tags are the active ad hoc tags that can communicate with other class 4 and 5 tags. Class 5 tags known as the reader tags can power class 0, 1, 2 tags and read the data on those tags. Active tags are most expensive tags and passive tags are the least expensive tags. Passive tags are more widely used in the supply chain industry and most tag reading protocols are designed for identifying passive tags.

1.4 RFID Tag Reading Protocols

A protocol needs to be in place for the reader and tags to communicate with each other. With so many tags in the reader's range, it is essential to formulate a set of rules so that a reader can identify maximum number of tags in a given time. Some of the protocol attributes that a reader and tags need to agree on include: frequency of communication, type of modulation used for reader signals, clock signals to mark the beginning and end of periods when tags can communicate and the data that reader is requesting, etc. The tag identification protocols are mainly classified as deterministic or probabilistic.

1.4.1 Deterministic Protocols

Deterministic protocols make the assumption that tags have unique identifiers. These protocols do not need the tags to possess any more functionalities than those of a passive (Class 0 or 1) tag. Since the ID of a tag is represented by a series of 0's and 1's, these identifiers are visualized as a binary tree where each path from the root to a leaf makes up a unique ID. The tag IDs may or may not have common prefixes in the visualized binary tree. Deterministic protocols function by requesting all tags with an advertised prefix to respond with their tag ID to the reader. When there are multiple tags with common prefixes that match the reader's advertised prefix, all such tags respond to the reader with their IDs. This can cause collisions at the reader. A collision is a condition when more than one tag responds to the reader's query at the same time with different data. However, deterministic protocols guarantee to resolve these collisions in subsequent queries and hence the protocols can be completed in finite time.

1.4.2 Probabilistic Protocols

Probabilistic protocols are based on time division which is the underlying idea of the ALOHA protocols. RFID probabilistic protocols are mostly a variant of the Frame Slotted ALOHA (FSA). Each round of the protocol takes place in a period of time that is called a frame. Each frame is divided into smaller periods of equal length called slots. The tags randomly pick a slot and respond to the reader with their IDs during that slot. The probabilistic protocols do not guarantee that all tags within the reader's range will be recognized in finite time. There can be a worst case situation of repeated collisions and hence the protocol can go on for an infinite time. However, they claim to read all the tags with a certain probability. A good probabilistic protocol tries to maximize the probability of reading every tag.

Tree protocols are deterministic whereas ALOHA protocols are probabilistic. Deterministic tag reading protocols ensure that all the tags within the range of a reader are identified at the end of one round. However, they have very long identification delays in order to guarantee 100% tag identification. The probabilistic protocols have shorter identification delays when compared to deterministic protocols. Also, among the probabilistic protocols, some ALOHA protocols can be faster and more efficient than others, which would be discussed in the later sections of this dissertation. On the other hand, 100% tag identification cannot be guaranteed.

1.5 Dissertaition Agenda

The existing protocols (deterministic or probabilistic) have been constructed for static scenarios where the reader and tags remain stationary until all the tags within the reader's range are identified. Since the current RFID anti-collision protocols cater to function in such generic environments, they tend to not deliver optimal performance for the specific scenarios, such as reader-mobility (warehouse) and tag-mobility (conveyor belt), and NF-listening of UHF RFID tags. Hence the research presented in this dissertation strives to deliver environment-specific frameworks to deliver improved performance of the RFID protocols.

1.6 Dissertation Structure

The research developed in this dissertation is organized in the remaining chapters as follows.

Chapter 2 reviews the static tag reading protocols underlying the framework developed in upcoming chapters. This chapter gives an understanding of the basic Tree, ALOHA and Hybrid protocols used for tag identification in static environment. The algorithm description of the protocols is presented. The tag estimation techniques used by each protocol are also discussed. This chapter explains the existing protocols in detail that helps in understanding the design changes made in the protocols to tailor them to fit the framework developed for practical warehouse environment.

In Chapter 3, the framework designed for existing protocols to function efficiently in a mobile-tag, multi-reader environment is presented. The importance of the augmentation framework is highlighted and a mobility model with fixed multiple readers and mobile tags on a conveyor belt is developed. The information-sharing framework and corresponding augmentation to different tag-reading protocols are discussed. The protocol-specific state information that is shared between the readers, for each of the aforementioned protocols, is identified. A mathematical study of the improved performance achieved by the augmented protocols is provided. Through extensive simulations, the observations are evaluated to discuss throughput improvement of these augmented protocols with proximity readers' information-sharing framework.

Chapter 4 proposes a zonal spacing framework to improve the efficiency of the tag-reading protocol. It starts by discussing the state of research related to mobile readers and their usage in warehouse scenarios. The warehouse scenario is presented and the challenges that come with it are described. A location-learning algorithm segregating the tags into geographical groups in the warehouse is proposed. The work describing the zonal spacing framework is presented in a single-grid scenario and subsequently, the implications of this framework in multi-grid scenario are studied. Since this problem is complementary to the conveyor belt model, the framework developed in Chapter 3 is also employed and tested in conjunction with the zonal spacing framework. The zonal spacing is applied to a random walk model developed for a warehouse with multiple grids in place. The framework is studied from scalability perspective in identifying different parameters impacting protocols performance.

Chapter 5 proposes a a fundamentally different anti-collision protocol strategy, wherein the tags communicate with each other via near-field communication and assist the reader in further improving the performance of tag identification protocols. In this chapter, a protocol augmentation framework is presented to integrate the novel near-field learning functionality of tags for two different protocols, namely EPC-Class1 Gen2 (EPC-C1G2) industry standard complaint Q-adaptive and Dynamic FSA (DFSA) protocols. With a detailed performance evaluation study, the improved efficacy of the respective Q-adaptive and DFSA protocols augmented with near-field learning frameworks is demonstrated. The augmented version of the protocols outperforms the regular protocol both in terms of delay and energy costs, especially for large number of tags.

Chapter 6 provides a mathematical analysis of tag-reading protocol's performance in terms of time delays. A scalable bound-based solution to study the performance of the EPC Gen2 protocol in large-scale RFID system is proposed. This chapter highlights the fact that Q-adaptive (in its native form) is close to its maximum theoretical performance even high tag density scenarios. The functioning of Q-adaptive and DFSA protocols as they reach the highest theoretical performance is studied. The theoretical model proposed is validated with reasonable experimental results.

Chapter 7 summarizes the research presented in Chapters 3 through 6 briefly. It also highlights the results obtained from this research and shines light on how all these research problems can be further extended.

Chapter 2

Literature Review

In order to make the radio frequency identification systems most resourceful, several tag-reading protocols that aim at maximizing the number of tags read in a unit time have been proposed. These protocols try to minimize collisions and hence also have the name *tag anti-collision protocols*. The efficiency of a tag identification protocol is determined by: number of tags identified in unit time (also called throughput). Tag-reading protocols used for reading passive tags fall under two broad categories: deterministic (tree-based) protocols and probabilistic (ALOHA-based) protocols as discussed in Section 1.4. There are some protocols that are a hybrid of both deterministic and probabilistic approaches. Deterministic protocols are also known as tree protocols. As the name suggests, the deterministic protocols can be completed in finite time, however, at the cost of long identification delays. Probabilistic protocols are based on time division which is also the underlying idea of the ALOHA protocols. RFID probabilistic protocols are mostly a variant of the FSA protocol. Hybrid protocols execute tree and ALOHA protocols in parts taking advantage of their deterministic nature and shorter identification delays, respectively. There are some protocols that execute in mobile tag environments.

2.1 Tree Protocols

As mentioned above, deterministic protocols are tree protocols. The two tree protocols discussed in this section are: query tree Protocol and binary tree protocol. The execution of tree protocols focuses on acquiring tag IDs one bit at a time. They follow a highly request-response-based approach to resolve collisions and acquire tag IDs. The reader calculates and sends a prefix of some length n to the tags. A prefix is nothing but a bit string of 1's and 0's. If this prefix matches the first n bits of a tag ID, the corresponding tag replies. Depending on whether single or multiple tags respond, a new prefix is calculated by the reader and the process is repeated.

2.1.1 Query Tree Protocol

The Query Tree Protocol (QTP) is a memoryless tag identification protocol, i.e., it does not need the tags to remember any previous queries made by the reader and so it uses Class 0 or 1 passive tags that have minimum functionalities [30].

Algorithm description:

This algorithm is executed as a highly active request-response process between the reader and the tags, i.e., the reader queries and the tags respond in alternating turns. The algorithm is employed as in the following steps:

- Reader starts by pushing a 0 and a 1 into an empty queue. Reader advertises the first query in the queue (0). Tags whose prefixes match the bit sent by the reader respond with their tag IDs.

- If there are more than one tags with the advertised prefix, all the tags with the matching prefix respond to the reader with their tag IDs. This causes a collision at the reader. Due to the collision, the reader cannot decipher any of the tags that responded. The reader removes the advertised query from the queue and adds two new queries to the queue, first by appending a 0, and then a 1 to the previous prefix that resulted in the collision. A queue of such prefixes is maintained and the queries are sent in order from this queue.

- If there is only one tag with the advertised prefix, it responds to the reader with its ID. After receiving the ID, the reader sends back an acknowledgement to the tag. Upon receiving an acknowledgement from the reader, the tag mutes itself and does not participate in the reader's future queries.

- If there is no tags with the advertised prefix, none of the tags respond to the reader leading to some wasted idle time.

- Reader continues to query with the next prefix in the queue. It stops when there is no more query in the queue remaining and the queue is empty again like when it started. By this time, all the tags within the range of the reader are identified by the reader.

QTP is a deterministic protocol, therefore, its performance is measured in terms of the number of bits transmitted by the reader and tags during the execution of the protocol. If the number of bits in a tag ID is x, then the number of tags can be no more than 2^x. The length of reader's query may vary from 1 bit to $x-1$ bits. If l is the level of reader query in the tree, the number of bits in that query is 2^l. Summing the query bits at all levels from 1 to $x-1$ (depth of the tree), the total bits sent by the reader can be calculated. The tags' response is x bits long as they always respond with their entire ID.The number of bits transmitted is,

$$\begin{aligned} Total\ bits &= Bits\ sent\ by\ reader + Bits\ received\ by\ reader \\ &= \sum_{l=1}^{x-1} l\ 2^l + x \times 2(2^x - 1). \end{aligned} \tag{2.1}$$

2.1.2 Binary Tree Protocol

The binary tree protocol needs the tags to remember the last query made by the reader [14]. The tags required for this protocol need to have some memory and hence higher functionality tags are required to implement this protocol.

Algorithm description:

The query sent by the reader and the response given by the tags are both one bit long for this protocol. The algorithm is implemented in the following fashion:

- Every tag has a pointer. After the reader issues a start signal, the tags perform a reset function, pointing to the highest significant bit of their IDs. The pointer moves from the highest significant bit of the tag's ID towards the lowest significant bit during the process of the protocol.

- The reader starts by sending bit 0 after a reset. The tags whose pointed bit is the same as the inquiring bit will send back the next bit of their tag IDs to the reader. The remaining tags will go into the stand-by state, and will not answer to the remaining inquires in this round until a tag is eliminated and all the other tags are reset by the reader issuing a start command.

- If all the responding tags reply with the same bit (0 or 1), the reader uses the received bit as its next inquiring bit. If some of the responding tags reply with a 0 and the others reply with a 1, a collision is sensed and the reader uses 0 as the next inquiring bit by default. This step is repeated until a tag is identified.

- The reader always sends bit 0 as the first inquiring bit after a reset unless it gets no response from the tags at all confirming that there are no more tags with their IDs starting with a 0. When this happens, the reader will start with 1 as the first inquiring bit after a reset in all the following rounds until it gets no response for 1 as a starting bit too.

- If the reader senses a collision in the last inquiring bit, it recognizes two identifiers, one with 0 appended to the sequence of bits gathered so far and the other with a 1 appended.

- After a tag is identified, it is muted and the remaining tags are reset. When the remaining tags receive the next start signal, they change their state to active and reset their pointer so that it is pointing to the highest significant bit once again.

BTP is also a deterministic protocol, therefore, its performance is measured in terms of the number of bits transmitted by the reader and tags during the execution of the protocol. If the number of bits in a tag ID is x, then the number of tags can be no more than 2^x.The number of bits sent and received by the reader is given by, $2x$ for each round. Two tags can be identified at the end of each round. Total number of rounds required is 2^{x-1}. The total number of bits transmitted is,

$$Total\ bits = 2x\ 2^{x-1}. \tag{2.2}$$

2.2 ALOHA Protocols

RFID probabilistic protocols are mostly a variant of the FSA. Each round of the protocol takes place in a period of time that is called a frame. Each frame is divided into smaller periods of equal length called slots. The frame size is advertised only once by the reader. The tags, after receiving the frame size randomly choose a number between 1 and the frame size and start a timer waiting for their slots to send their tag IDs to the reader. The reader will either receive an ID (if only one tag responded during a slot) or experience a collision. A collision is a condition when more than one tag respond to the reader's query at the same time with different data. It is also possible for the reader to hear nothing during a slot if no tag chooses that slot. Based on the number of tags recorded, collisions and idle slots observed, a new frame size is calculated and the process is repeated.

2.2.1 Basic Framed Slotted ALOHA

The Basic Framed Slotted ALOHA (BFSA) is the simplest probabilistic protocol both for the reader to implement and the tags to follow through [58]. BFSA is similar to the FSA with the only difference being the fixed frame size. A frame size is measured in terms of the number of slots that make up the frame.

Algorithm description:

The reader's query consists of framesize (number of slots) and a start signal indicating the start of frame. The reader's query is always fixed in the BFSA protocol. The algorithm for BFSA proceeds as follows:

- The reader starts a round by sending the framesize (N) and a start signal indicating the start of the frame. The tags generate a random number in the range of the framesize ($1 - N$) and wait for their time slots to send their identifiers during that slot.
- The reader continuously listens to the channel and records the data received during each slot.
- If a slot is chosen by a single tag, its ID is recognized and recorded by the reader.
- If more than one tag choose a slot to send their IDs, a collision takes place and the data sent in the slot gets garbled (assuming no two tags have the same identifier). The reader recognizes such a collision when it gets the garbled data and discards it. If a slot is not chosen by any tag it goes empty.
- At the end of a round, the reader sends a bit string of size N acknowledging the slots that were correctly recognized with a bit 1 and those that were either empty or contained garbled signals with a bit 0.

- The tags recognize the bit meant for them in the acknowledgement string as they remember the slot number in which they sent their identifiers in the previous round. If the bit reads 1, the tags set their states to mute and do not participate in future frames. Otherwise, they set it to active and compete in the next round. The reader starts a new round until it detects no collision in one of the following rounds.

2.2.2 Dynamic Framed Slotted ALOHA

DFSA comes next in the level of simplicity of implementation after BFSA. It differs from BFSA because of its varying frame sizes in each round of implementation. The tag estimation procedures play an important role in this protocol. The tag estimation technique uses an empirical formula derived theoretically and depends on the number of collisions in the previous round [17].

Algorithm description:

Similar to BFSA, the reader's query consists of frame size (number of slots) and a start signal indicating the start of frame. However, as the name suggests, the reader's query is dynamic in the DFSA protocol and changes at the end of each round. The algorithm for DFSA proceeds as follows:

- The reader advertises the frame size (N) at the beginning of a round and then sends a start signal and starts to listen continuously on the channel.

- All the tags in the range of the reader move to the active state, generate a random number in the range of the advertised framesize and wait for their slots.

- Similar to BFSA, if a slot is chosen by a single tag, its ID is recognized and recorded by the reader. If more than one tag choose a slot to send their IDs, it causes a collision. If a slot is not chosen by any tag it goes empty.

- At the end of each round, the reader observes the number of collisions and estimates the remaining number of tags (n) using the formula:

$$n = 2.3922 \times c_k, \tag{2.3}$$

where the number 2.3922 is the average number of tags colliding in one slot and c_k is number of slots in which a collision occurred. This number is obtained by a theoretical calculation.

- The framesize for the next round is set equal to the calculated value of n. The reader continues to a new round until it detects no collision in one of the following rounds.

DFSA is a probabilistic protocol. The performance of DFSA is determined by the probability of success of each slot in a frame. The probability is calculated using a binomial distribution. If n is the total number of tags and N is the framesize advertized. the probability of success of a slot is given by:

$$P[S] = n\,(1/N)\,(1 - 1/N)^{n-1} \tag{2.4}$$

and the expected number of successful tags is given by:

$$E[S] = n\,(1 - 1/N)^{n-1}\,. \tag{2.5}$$

Finally, the throughput of the protocol is calculated after each round as,

$$Throughput = \frac{E[S]}{N} \tag{2.6}$$

2.2.3 Enhanced Dynamic Framed Slotted ALOHA

The Enhanced Dynamic Framed Slotted ALOHA (EDFSA) protocol uses a more complicated tag estimation procedure [31]. It requires the vector $< c_0, c_1, c_k >$ (empty slots, read slots, collided slots) at the end of each round to calculate the framesize for the next round. It calculates the theoretical values of the vector for different number of tags (n). The value of n that minimizes the distance between the observed and calculated vectors is the best estimate for the number of tags present [49], [50].

Algorithm description:

This algorithm is similar to DFSA where the frame size differs in each round. However, the tag estimation procedure is different in this protocol. This protocol limits the frame size beyond a certain value unlike DFSA that lets the frame size increase indefinitely depending on the tag estimation. The EDFSA protocol also limits the number of tags that can participate in a round. The algorithm is executed as follows:

- The reader advertises the framesize (N) and the number of groups it wants the tags divided into (M). The value of M is calculated using the formula:

  $$M = n/N, \tag{2.7}$$

  where n is the estimated number of tags and N is the frame size.

- The tags generate a random number and perform the modulo operation with M. All the tags that have the result of the modulo operation as 0 can participate in the following round. The tags then generate a new random number in the range 1 to the advertised frame size (N) and wait for their slots to send their ID during that slot.

Table 2.1: Number of unread tags vs. optimal frame size and Modulo (M)

Number of unread tags (n)	Framesize	Modulo(M)
.	.	.
.	.	.
.	.	.
1417 - 2831	256	8
708 - 1416	256	4
355 - 707	256	2
177 - 354	256	1
82 - 176	128	1
41 - 81	64	1
20 - 40	32	1
12 - 19	16	1
6 - 11	8	1
3 - 5	4	1
1 - 2	2	1

- Similar to BFSA and DFSA, if a slot is chosen by a single tag, its identifier is recognized and recorded by the reader. If more than one tag choose a slot to send their data, a collision takes place. If a slot is not chosen by any tag it goes empty.

- At the end of each round, the reader determines the vector $< c_0, c_1, c_k >$ (empty slots, read slots, collided slots). It then calculates the expected vector $< a_0, a_1, a_k >$ for different values of n (number of tags) using the formula:

$$a_r{}^{N,n} = N({}^nC_r)(1/N)^r(1-1/N)^{n-r}, \tag{2.8}$$

where $a_r{}^{N,n}$ is the expected number of slots occupying r tags for a given framesize. The value of n for which the distance between the expected vector ($< a_0, a_1, a_k >$) and the observed vector ($< c_0, c_1, c_k >$) is minimum is the best estimate for the number of tags and hence the frame size is determined for the next round.

- The EDFSA protocol claims that a frame size beyond $N = 256$ only decreases the efficiency of a tag identification process. It uses Table 2.1 to determine the frame size for the following round [31].

The throughput of EDFSA protocol is calculated the same way as the throughput of DFSA. The only difference here is framesize (N) has an upper bound of 256. The number of participating tags in a frame is $n' = n/M$.

2.3 Hybrid Protocols

RFID protocols that take advantages of both tree and ALOHA based approaches are called the hybrid protocols. They can be employed in the following ways: (1) retrieve partial IDs of tags and divide the tag population with matching prefix to participate in the advertised frame; (2) advertise a frame and solve collided slots using the tree protocol approach.

2.3.1 Query Tree Dynamic Frame Slotted ALOHA

This protocol employs the tree protocol followed by ALOHA protocol [37].

Algorithm description:

The reader starts by generating all prefixes of some length, say k, and adding it to a prefix pool. The reader query consists of a prefix from this prefix pool and a frame size. The algorithm is executed as follows:

- The reader randomly picks a binary suffix from a pool and advertises it along with a frame size N. Note: N is a power of 2 and does not exceed 256.

- Tags with the advertised prefix respond with their IDs.

- At the end of a frame, the remaining tag count(t) is estimated. The estimation procedure used is the same as that of EDFSA described in Section 2.2.3. If the tag count is less than 256, the next frame size is calculated as: $N = 2^{[log_2 t]}$. If it is greater than 256, the tag population is divided into M groups, where $M = 2^{[log_2(t/256)]}$.

- Prefixes from 0 to $2M-1$ are added to the prefix pool. This process is repeated until all the prefixes in the pool are exhausted.

2.3.2 Hybrid Anti-Collision Algorithm

HAC protocol first employs the ALOHA protocol followed by the tree protocol [59].

Algorithm description:

The reader starts by employing a variation of an ALOHA protocol (Q algorithm). Any collisions in the frame are resolved using Enhanced Anti-collision Algorithm (EAA) that is a variation of the tree protocol. The algorithm is executed as follows:

- It implements adaptive slot-count (Q) algorithm along with EAA.

- The reader starts by employing the Q-algorithm with a small framesize; the framesize may be varied and advertised at end of each slot as follows:

  - Collision: N = Round (N + C)
  - Idle: N = Round(N - C)
  - Read slot: no change

  where C is some constant that lies in the range $[0, 0.5]$.

- In case of a collision, the Q protocol is paused and the EAA tree based protocol is used to identify the tags involved in the collision. Once the collision is resolved, the Q protocol is resumed.

The throughput of HAC protocol is given by,

$$\eta_H = \frac{S + \sum_{j=1}^{C} n_j}{(L - C) + \sum_{j=1}^{C} 2n_j - 1}, \tag{2.9}$$

where L is the framesize, S is the number of successes, C is the number of collisions and n_j is the number of tags that collided in one slot.

2.4 Mobile Tag Protocols

The protocols discussed so far assume the RFID tags and reader to be stationary. In the supply chain industry, a more practical scenario is to be able to identify tags when the tags are in motion or the reader is in motion. There are some protocols that have been proposed to improve the tag-reading efficiency in the mobile tag environment. One of them is the Accelerated Frame Slotted ALOHA (AFSA) that employs the FSA and can be built on top of DFSA/EDFSA [39]. This framework is good for only DFSA or EDFSA protocols and cannot be extended to any deterministic protocols. The other protocol proposed in [55] uses a dynamic programming solution to generate a table of framesizes based on tag estimation and the pallet's location in the readers range. This table could get larger for denser tag populations and the reader may not afford to hold and search through so much data.

2.5 Summary

This chapter presented several existing RFID tag identification protocols such as the QTP, BTP, BFSA, DFSA, EDFSA, QTDFSA and Hybrid Anti-Collision algorithm. Tag estimation procedures used and tag functions performed in these protocols give a clear understanding of the required functionalities and performance limitations for the reader and the tags respectively, for each of the protocols discussed above. All the protocols discussed in this chapter require passive tags that can perform basic operations such as store, load, add, subtract, divide, multiply, modulo. These protocols form the foundation for the frameworks developed in the upcoming chapters. It is essential to fully understand the capabilities of the reader and the tags to design mobility models and the frameworks for mobile tags or reader and multiple readers environments. The tag identification protocols hold immense possibilities to develop mobility models in a warehouse scenario in the supply chain industry.

Chapter 3

Framework for Efficient Tag-reading in Mobile-tag, Multi-reader Environment

Mobile RFID tag reading on conveyor belt represents a practical scenario used widely in the supply chain industry. Tag reading (*i.e.*, finding the identifiers of each of the tags) when tags are non-mobile is performed using protocols that are based on ALOHA, tree, or a combination of both. The time a protocol takes to complete the reading of all tags is directly proportional to the number of tags. When tags are moving, the protocol that is initiated and coordinated by a reader will not be able to read all tags, as the tags move away from the reader's range. In this chapter, it is shown how a large number of tags moving on a conveyor belt can be read using a tandem of communicating readers placed along the axis of the conveyor belt. The tags that are unread by a reader could be read by the reader next in the sequence. Rather than restarting the protocol at the next reader in the sequence to read the unread tags, it can use the information from the previous reader (call it information sharing) to improve protocol performance and hence reduce the reading time.

This information sharing significantly enhances the tag reading performance (in terms of the number of tags read) as compared with the traditional tag-reading protocols. In experiments, the tandem reader arrangement with information sharing is performed for ALOHA, tree, and two different combinations of ALOHA and tree (a.k.a. hybrid) protocols. Performance evaluation study corroborated with extensive simulation results show that the aforementioned protocols augmented with the novel information-sharing frameworks outperform their respective primitive as-is version counterparts.

3.1 Overview

The vital supply chain management services such as product accounting, identification, and tracking require tag readings at several stages of inventory management. An RFID tag reading is a process of finding the identifiers of tags that are attached to the products of interest. These identifiers are used to retrieve the tagged'product's information. To achieve a higher degree of business efficiency, the tag-reading process needs to be carried over periodically to keep track of the continuous churning of goods, flowing in and out of the supply-chain systems, with a desired level of accuracy.

This chapter focuses on conveyor-belt-based supply-chain management systems, that require scanning of continuously moving products on a conveyor belt. The tag reading in this platform is done by suitably mounting the RFID reader on the top of the conveyor belt for scanning the moving tags.

For practical reasons, the RFID scanning system should be seamlessly integrated with product-manufacturing plane of the supply-chain system. This technically implies the following: (*i*) the conveyor belt speed is not controlled in order to enable the RFID reader to scan the tags, and (*ii*) the population of tags (products) on the conveyor belt is not adjusted so as to improve the reading efficiency of the reader.

Subject to the aforementioned practical constraints, tags on a conveyor belt can be sufficiently scanned with the help of multiple readers arranged in tandem along the axis of the moving conveyor belt. As with the wireless systems that typically suffer from interference within the overlapping communication regions, it is advised to sufficiently space the readers in such a way that the subsequent readers' reading range do not overlap with each other. The gap between the readers can be intentionally made by increasing the pathloss by either shielding or lowering the reader's transmit power level, as this decreases the strength of interfering signals from adjacent readers.

The RFID anti-collision protocols such as the industry-standard ALOHA-based, and tree-based protocols [10] are designed to improve the tag-reading efficiency, by limiting the concurrent transmissions of multiple tags competing for interaction with a (single) reader. The protocols enable this by appropriately scheduling the tags' participation with the reader. These protocols' design philosophies are monolithic and are semantically confined to handle tag reading in static single-reader, multiple-tags settings. To achieve an improved performance, these protocols need to be custom-tailored to handle mobility as well as enable a protocol-level cooperation among multiple readers. To this end, this chapter proposes a novel protocol augmentation framework to improve the tag-reading efficiency in the aforementioned conveyor-belt tandem, non-overlapping readers settings.

The current research on RFID tag reading focuses on improving throughput and addressing scalability issues under mobile scenario [55]. The work that is relatively closer to this research is [55], wherein the authors address the throughput improvement of an RFID reading protocol for a conveyor belt mobility scenario with a single reader. On the other hand, the research on multiple-readers scenario is mainly focused on devising contention-resolution mechanisms to avoid collisions caused by overlapping readers, especially by enabling scheduling among the multiple readers [12] [15]. This is the first work to consider an industry-relevant practical scenario comprising of multiple non-overlapping static sequential readers with conveyor belt mobility, for improving the throughput of RFID reading protocols.

For the considered conveyor belt mobility scenario, this research proposes novel augmentation frameworks for improving the throughput of ALOHA, tree-based, and hybrid combinations of two types of ALOHA and tree protocols. In the proposed augmentation framework, the protocol-state information between the adjacent readers is exchanged in the direction parallel to the conveyor belt motion. In this manner, the next reader in the sequence can resume the tag-reading process from a state left-off by the previous reader. Henceforth, this augmentation will be referred to as the information sharing framework.

The improved performance achieved by the augmented protocols is mathematically studied. Through extensive simulations, the throughput improvement of these augmented protocols with proximity readers, information-sharing framework is demonstrated.

The remainder of this chapter is organized as follows: Section 3.2 introduces different tag-reading protocols and motivates the need for an information-sharing framework to improve the protocol's performance. The conveyor belt system model is presented in Section 3.3. The information-sharing frameworks and the corresponding augmentation to different tag-reading protocols are discussed Section 3.4. Subsequently, a mathematical reasoning for the improvement caused by the augmentation frameworks is also discussed. The performances of the augmented protocols are evaluated by extensive simulation results in Section 6.5. Finally, Section 3.6 provides a summary of the results contained in this chapter.

3.2 Need for Protocol Augmentation

Among different types of tags, the passive tags (that energize their circuits from the reader's electromagnetic radiation) are widely used in the supply-chain systems, due to their low manufacturing costs. Tag-reading protocols for reading passive tags fall under three broad categories as follows: (*i*) deterministic tree protocols such as [30], (*ii*) probabilistic ALOHA protocols such as [17], and (*iii*) hybrid protocols that use different combinations of protocols mentioned in (*i*) and (*ii*). Invariably, the passive tag-reading protocols function in terms of discrete-timed slots, with all tags being slot-synchronized by the centralized reader.

This section highlights the importance of the augmentation framework and identifies the protocol-specific state information that is shared between the readers, for each of the aforementioned protocols.

3.2.1 Probabilistic Protocols

Probabilistic ALOHA protocols such as in [17] use a stochastic ordering to identify the tags. This protocol executes in a sequence of multiple rounds called frames. Subsequently each frame comprises of multiple time-slots. At the beginning of each frame, the reader advertises a framesize f (in terms of the number of slots) to the tags. This framesize is a function of the total number of tags in the scanning system. The reader computes the current framesize by estimating the underlying tags' population, from information obtained from the tags' response in the previous frame.

Upon receiving the frame size the tags independently generate a random number in a range between 0 and $f-1$. The in-built decremental slot counter is initiated with this random number and starts clocking by reducing the count by one in each slot. When the counter clocks to zero, the tag responds to the reader with its ID, in that slot. Therefore in each time slot the reader receives either no response (named as idle slot), or a response from a single or multiple tags leading to successful identification or causing collision, repeatedly. In this manner, the random response of tags induces a probabilistic behavior to the identification process.

For the first frame, the framesize is initialized to an arbitrary value as the number of tags under the reader's vicinity is not known at this point. To understand the effect of the framesize to the protocol's performance, Figure 3.1 shows the impact of the initial frame size on the protocol's performance in a moving tags scenario. As can be observed from Figure 3.1, too big or too small of the initial frame size would significantly affect the tag-reading performance. This loss in performance would scale up proportionately in multiple readers setting, wherein each reader sets up an

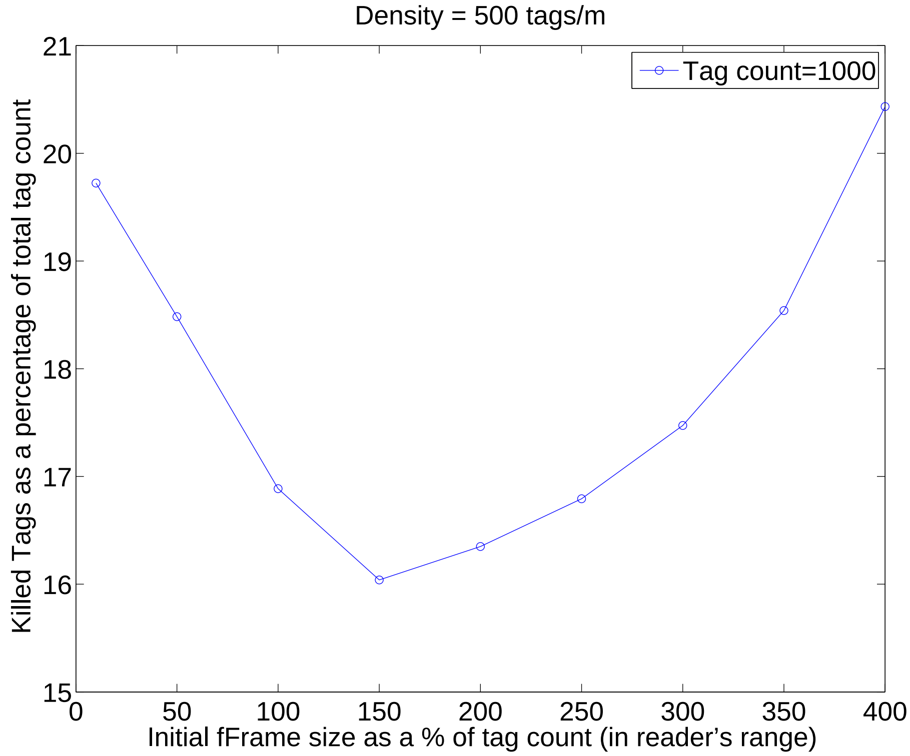


Figure 3.1: Performance of Mobile ALOHA protocol in a single reader setting with respect to various initial framesizes.

arbitrary initial frame size.

With the mobile tags settings, the framesize estimation is a continuous process. However, the underlying tandem arrangements of the reader can be exploited to share the information learnt from first reader to the subsequent reader. This would significantly enhance the protocol's performance. Therefore, an information-sharing framework is really advisable in such settings, as all the other readers except the first one can set their frame sizes based on the information shared by their respective previous readers.

3.2.2 Deterministic Protocols

The tree protocol such as QTP [30] exploits the bit-representation of tag-IDs, by a scheme called prefix matching. Each bit is represented as a node of the tree and different paths from the root to a leaf node represent different tag-IDs. Reader starting from the root node r parses each node (i) of the tree, and those tag(s) with their partial-ID(s) that match with the readers' traversed path (representing a bit-string) from the root r to the current node i will respond to the reader. Multiple tags responding/colliding will urge the reader to proceed with parsing the next node. If a single tag responds, the tags-ID of that tag is scanned. In this manner, the reader parses each node of the tree in a deterministic number of steps and identifies the tags in the system.

When QTP is applied to a multiple-reader mobile-tags setup, every reader would start parsing from the root to a leaf in order to identify the tags. This process of restarting from root in every tree would be time-consuming that drastically reduces the tag-reading throughput. In the tandem multiple-readers scenario, the information of the partially-constructed tree from the previous reader can be shared to the subsequent reader, thereby helping to start off, where the previous reader had left off. In this way, a significant amount of time could be saved in identifying tags and an information-sharing framework plays a pivotal role in improving the throughput of the tag-reading protocol.

3.2.3 Hybrid protocols

The hybrid protocols such as QTDFSA [37] and HAT [59] employ both tree and ALOHA protocols in a hierarchical manner. For instance, the QTDFSA protocol starts with QTP by randomly choosing a binary prefix string and advertises it along with a framesize. Only those tags that match their prefix-ID with the reader-chosen string will participate by picking a probabilistic slot based on the announced framesize. In this manner, each round of the protocol involves QTP hierarchically followed by ALOHA protocol. On the other hand, the protocol HAT [59] employs ALOHA protocol at a higher level, which is followed by a tree protocol.

With the help of tandem readers, each round of the hybrid protocols can be divided and distributed over the two neighboring readers. Similar to the information-sharing frameworks applied in plain ALOHA and tree protocols, the information of a partially performed round at the reader can be shared with a subsequent reader, to help it to start off where the previous reader had left off. By division of labor between the consecutive readers and with the help of information sharing, the throughput of the hybrid protocols can be improved.

Since the tags are continuously moving on the conveyor belt, the protocol state is constantly changing as well. Therefore, the information sharing between tandem readers must be conducted at regular intervals to achieve an improved performance.

3.3 System Model

The conveyor belt is assumed to move at a constant velocity v. Without loss of generality, two RFID readers can be placed in a tandem arrangement parallel to the direction of the conveyor belt at height h. The two readers are separated by a distance of $D_{IR} + D_{OR}$, where D_{IR} is the communication range of the reader within which it can interrogate the tags, and D_{OR} is the additional space between the readers that helps in not interfering with their respective RF radiations.

In a tag-reading protocol after a tag is identified it is said to be muted, by setting a flag in the tag's memory. A muted tag cannot participate in reader queries any further until the flag is reset again [22]. The same assumption can be made for the conveyor belt mobility scenario, wherein a tag once muted by a reader cannot participate in any following queries of that reader or any subsequent readers. The multiple readers in an RFID system are typically connected to a back-end database system that maintains the inventory catalogs. This connection can be wired (Ethernet) or wireless (say, WiFi).

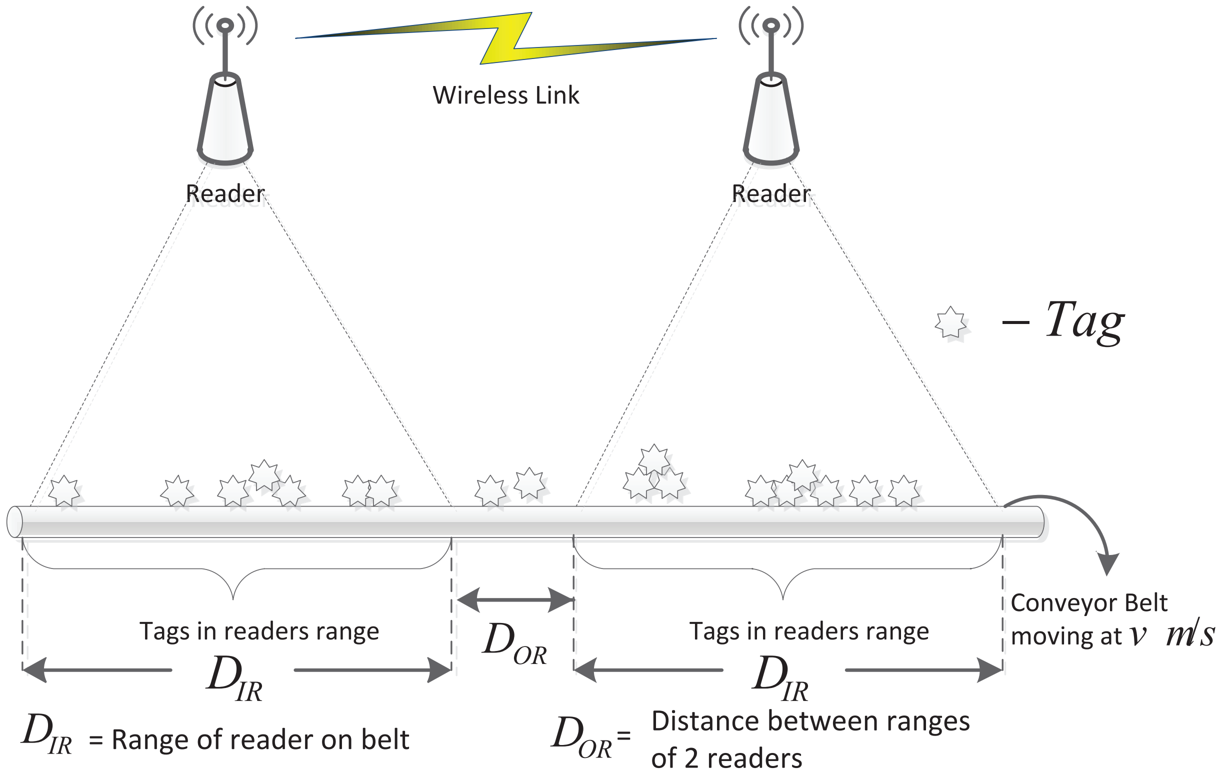


Figure 3.2: Schematic representation of mobile tags conveyor belt scenario.

3.4 Proposed Information Sharing Framework

This section describes the proposed information-sharing framework for the various tag-reading protocols.

This section describes the system model considered throughout this chapter. Figure 3.2 shows a schematic representation of the conveyor belt with tandem reader's scenario.

3.4.1 Tree-Based Protocol Augmentation

This section describes the framework being applied to a scenario where both the readers employ QTP [30]. Here, using the information sent by the first reader, the second reader picks up the prefix queries of the tree at a point where the the first reader left off. This implies that a tag that could have progressed until the last few steps of the tree-reading protocol (before moving outside the range of the first reader) will continue (at the second reader) from where it paused instead of restarting from the tree root. This improves the overall tag reading rate. The duration that a tag spends in the reader's range (T_r) is given by:

$$T_r = D_{IR}/v, \tag{3.1}$$

where D_{IR} is the reader's range of operation and v is the constant velocity of the moving conveyor belt.

Assuming that the tag IDs are x-bits long, the total number of prefixes a reader can possibly query is equivalent to not more than the total number of different paths from the root node to the leaf nodes in the tree. Therefore, the total number of possible prefixes a reader can query is no more than k, where $k = 2^x$. Let P_i denote a prefix representing a tag-ID i. The prefixes that can be advertised by the reader are in the range between P_1 (which is 0 represented in binary) and P_k (which is 2^x in binary).

The first reader starts the QTP [30]; it also starts a timer set to time T_r, as computed from equation 3.1. At the end of time T_r on the timer, the first reader sends the last prefix queried P_i (before the timer runs out) to the second reader and resets the timer back to T_r. A tag that entered the first reader's range when the timer is set to T_r witnesses that the reader advertise all queries from P_1 to P_i before it leaves the reader's range. When this tag enters the second reader's range, it gets a chance to be queried by the second reader with queries from P_{i+1} to another prefix, say, P_j. The QTP is restarted whenever the reader runs out of prefixes to advertise (*i.e.*, if the last prefix of the tree P_k is reached, it restarts the protocol from P_1). Therefore, P_j can be either of the following: (i) any prefix that comes after P_i but before P_k, if the protocol is not restarted, or (*ii*) any prefix that comes after P_1 but before P_i, if the protocol is restarted.

In case (*i*), the tag would have witnessed j unique prefixes. On the other hand, for case (*ii*), the tag has witnessed all k prefixes at least once. This cycle goes on until all tags are identified or they move out of the reader's range. The step-by-step execution of this framework is given in Algorithm 1.

Probabilistic Analysis:

The tree protocol is a deterministic protocol, *i.e.*, all the tags in reader's range would be read eventually provided there is no limit imposed on the duration for which they are within the reader's range (*i.e.*, if the tags are stationary). However, if the tags are mobile, it means that the tags have a limited time in the reader's range. Also, there is a continuous inflow (into the reader's range) of new tags that influence the protocol state due to unexpected collisions. Similarly, there is a continuous outflow (out of reader's range) of older tags that causes unexpected idle slots. Hence, the tree protocol is no longer a deterministic protocol in a mobile tag scenario.

Assuming that a tag ID is x-bits long, there can be no more than 2^x such unique identifiers. From equation (3.1), the duration of time for which a tag remains in the reader's range is given by T_r. Therefore, the number of prefixes (l) that a tag witnesses while it is within the reader's range is given by $l = T_r/(2\delta t)$, where δt is the time duration of one slot. The probability of a tag to be read (P[R]) is defined by

$$P[R] = \sum^{l}(P[p] \times P[p']), \tag{3.2}$$

where l is the cardinality of the set of all advertised prefixes in the duration T_r, P[p] is the probability of one of its prefixes being advertised, P[p'] is the probability that no other tag with the same prefix is within the reader's range.

Algorithm 1: QT Protocol with Framework

```
Reader-Reader Process:
define Queue_1;
if !FirstReader then
    while true do
        binaryPrefix = Receive Prefix from Previous Reader (R_{k-1});
        Queue_1.EnQue(binaryPrefix);
    end
end
Reader-Tags Process:
define Queue_2;
δt = Time for 1 slot;
T_r = Total time spent by a tag in reader's range;
Timer = T_r/δt;
while true do
    if Queue_1 not empty and Timer == T_r/δt then
        prefix = Queue_1.DeQue();
        Queue_2.EnQue(prefix);
    end
    else
        //Restart Protocol
        Queue_2.EnQue(0);
        Queue_2.EnQue(1);
    end
    while Queue_2 not empty do
        prefix = Queue_2.DeQue();
        Broadcast(prefix);
        Receive(tagResponse);
        if tagResponse is read then
            Send ACK;
        else if tagResponse is collision then
            Queue_2.Enque(Concatenate(prefix, 0));
            Queue_2.Enque(Concatenate(prefix, 1));
        end
        if Timer == 0 then
            Send (prefix) to Next Reader (R_{k+1});
            Timer = T_r/δt;
            break;
        else
            Timer = Timer − 1;
        end
    end
end
```

The number of prefixes of an x-bit tag ID can be no more than $x - 1$. The probability of a prefix of some tag t being advertised is given by

$$P[p] = \frac{x-1}{2^x}. \tag{3.3}$$

Say a prefix is advertised that is p-bits long, the number of tags that can have the same prefix is no more than n, *i.e.*, $n \leq 2^{x-p}$. The probability that no other tag with the same prefix is within the reader's range is,

$$P[p'] = 1 - \frac{2^{x-p}}{2^x} = 1 - \frac{1}{2^p}. \tag{3.4}$$

The equations (3.3) and (3.4) give the probability of reading for one advertised prefix. The read probability over all l advertised prefixes is

$$P[R] = \sum^{l} \frac{(x-1)}{2^x} \left(1 - \frac{1}{2^p}\right). \tag{3.5}$$

When there are two readers placed consecutively over the conveyor belt as described in the framework description, the value of l is the total number of unique prefixes advertised by the first and second readers:

$$l = l_{R1} \cup l_{R2}. \tag{3.6}$$

From equation (3.5), it is clear that greater the value of $|l|$ the greater the probability of tag read. If the readers do not share any information, the second reader would start the protocol at the root (just like the first reader). The tags will see the same set of prefixes that they came across at the first reader, so the equation (3.6) can be rewritten as $l = l_{R1} = l_{R2}$.

If readers share information as in the case of the tree protocol with framework as shown in Algorithm 1, a tag that enters a reader's range at anytime observes a fraction ($l/2^x$) of prefixes in the tree at the first reader. When it goes out of the range of the first reader and enters the second reader, because of information exchange between readers, it sees a fraction (or all) of the remaining $2^x - l$ prefixes that make up the tree.

Hence, the tree protocol with augmentation framework has a system efficiency that is less than or equal to that of tree protocol (with no framework) employed in a stationary tags scenario, but definitely greater than the efficiency of tree protocol (with no framework) employed in a mobile tags scenario.

3.4.2 ALOHA-Based Protocol Augmentation

This framework is applied to a scenario where both adjacent readers employ DFSA [17] protocol. Here, using the information sent by the first reader, the second reader calculates the frame size to be advertised in order to reduce collisions and idle slots and hence maximize tag identification. Note that, once the frame size is advertised, any newly arriving tags cannot participate in this frame. Only the tags that were present in the reader's range when the frame size was advertised can choose a slot in the following frame.

Each reader employing ALOHA protocol must estimate tags for a certain frame based on two accounts: (*i*) due to the collided slots in the previous frame, and (*ii*) due to the newly arrived tags that entered the reader's range while the previous frame is under process. The initial frame size for the reader can be suitably picked if some knowledge of the tag arrival rate is known. However, in many real-life environments the tag arrival rate is dictated by the production process throughput which can vary with time and may not be known in advance (such as the arrival rate of checked-in baggages at an airline check-in counter). A new tag estimation procedure is therefore introduced to account for the mobile nature of tags.

Tag estimation:

The first framesize (f_1) advertised can be set to any value. However, a framesize of $f_1 = 256$ is broadcast during the experimental setup because of the reason stated in [31], i.e., any framesize greater than 256 would most likely reduce the throughput of the protocol when no information about the tag population is known. Let the belt run and tags enter the reader's range and set the timer to $T_r/\delta t$. The successful slots (s_i) and the collided slots (c_i) for the very first frame (f_1) advertised by the reader must be saved for later tag estimations.

The total number of tags that participated in the first frame f_1 after the timer starts can be approximately calculated as [17]:

$$n_1 = s_1 + (2.3922) \times c_1. \tag{3.7}$$

In other words, the number of tags that entered the reader's range during the first frame time f_1 is n_1. The arrival rate of tags (ψ) can be approximated using n_1:

$$\psi = \frac{n_1}{f_1} = \frac{s_1 + (2.3922) \times c_1}{f_1}. \tag{3.8}$$

Now, the newly arrived tags at the end of the latest frame, say f_i, can be approximated using the arrival rate calculated in equation (3.9):

$$n_{i+1} = \psi \times f_i. \tag{3.9}$$

The following frame size (f_{i+1}) can be estimated as

$$f_{i+1} = n_{i+1} + (2.3922) \times c_i. \tag{3.10}$$

The same arrival rate as in equation (3.8) is used to calculate future frame sizes until the timer runs out. When the timer runs out, a new arrival rate is recalculated using equations (3.7) and (3.8).

The number of tags left unread at the end of first frame can be no more than $2.3922 \times c_1$ as given in [17]. This upper bound on the number of tags is sent to the second reader to estimate its first framesize (f'_1) using the following equation:

$$f'_1 = n'_0 = 2.3922 \times c_1. \tag{3.11}$$

The information is sent to the second reader every $T/\delta t$ slots for the framework applied to DFSA protocol. The new arrival rate is also calculated after every $T/\delta t$ slots. The following frame sizes at the second reader are calculated using equations (3.7) through (3.10). The step-by-step execution of this framework is given in Algorithm 2.

Probabilistic Analysis:

This section shows that use of this framework indeed improves the probability of success of slots in a frame. ALOHA protocol is a probabilistic tag-identification protocol, *i.e.*, all the tags in the reader's range are read with a certain probability. The probability of a tag being read in a round is also a function of participating tags (n) and advertised frame size (f) and is given by:

$$P[S] = p_s(n, f) = n\,(1/f)\,(1 - 1/f)^{n-1}\,, \tag{3.12}$$

where n is the number of tags participating in the frame and f is the frame size advertised.

Similarly, the probability of idle and collision slots, respectively, for tags can be calculated as follows:

$$P[I] = (1 - 1/f)^{n}\,, \tag{3.13}$$

Algorithm 2: DFSA Protocol with Framework

```
Reader-Reader Process:
if !FirstReader then
    while true do
        (F) =Receive(f_1) from PreviousReader (R_{k-1});
        Queue.EnQue(F);
    end
end
else
    define f_1 = 256;
end
Reader-Tags Process:
define f_0 = 256, i = 1, δt = Time for 1 slot;
T_r = Total time spent by a tag in reader's range;
Timer = T_r/δt;
while true do
    Broadcast(f_i);
    Timer = Timer − f_i;
    for j ← 1 to f_i do
        Receive(tagResponse);
        if tagResponse is read then
            Send ACK; s_i ++;
        else if response is collision then
            c_i ++;
        end
    end
    if Timer == 0 then
        Send (2.3922 × c_i) to NextReader (R_{k+1});
        n_i = s_i + (2.3922) × c_i;
        ψ = n_i/f_i = (s_i+(2.3922)×c_i)/f_i;
        if Queue not empty then
            f_{i+1} = Queue.DeQue();
        end
        else
            f_{i+1} = 256;
        end
        Timer = T_r/δt;
    end
    n_{i+1} = ψ × f_i;
    f_{i+1} = n_{i+1} + (2.3922) × c_i; //new frame
    if Timer < f_{i+1} then
        f_{i+1} = Timer;
    end
    i ++;
end
```

$$P[C] = 1 - P[S] - P[I]. \tag{3.14}$$

From equations (3.12) through (3.14), it can be assumed that the number of tags in the first reader's range is n. However, some tags are identified by the reader while under the first reader's range and these tags do not participate when they move to the next reader's range. The number of tags identified at the first reader can be calculated by finding the expected number of successfully read tags, which can be derived from the binomial distribution function as follows:

$$E[S] = n\left(1 - 1/f\right)^{n-1}. \tag{3.15}$$

Since, $E[S]$ tags have been muted and will not participate under the second reader's range, the number of leftover tags is

$$n' = n - E[S]. \tag{3.16}$$

From equations (3.12) through (3.14), one can see that the probabilities of successful, idle and collision slots are the functions of the framesize (f) and estimated number of tags (n). In a two-reader system, if there is no information exchange, these probabilities will be a function of $< n, f >$ at the first reader and $< n', f >$ at the second reader.

The tag estimation for ALOHA protocol in case where the tags are not moving takes into account only the collisions that happened in the previous frame. This is because all the tags remain within the reader's range indefinitely until removed and no new tags enter or leave the reader's range mid way through the identification process. The framework calculates the arrival rate of tags regularly (once every unit time T) and shares this information with the second reader. The second reader re-evaluates the remaining unread tags ($n_{opt} = n'$ from equation (3.16)) periodically after a time lapse of T and advertises a more accurate framesize ($f_{opt} = f'_1$ from equation (3.11)) to read tags. These $< n_{opt}, f_{opt} >$ values improve the probability of successful slots at the second reader and hence improve the overall success probability of the two readers.

The tag estimation procedure introduced in the framework for ALOHA protocol makes sure that the frame size advertised is approximately equal to the number of participating tags in the reader's range ($f \approx n$). With this relation plugged into equation (3.12), the probability of successful read can be arrived at $p_s = 0.368$. This is also in accordance with the analysis given in [4] that proves that the most optimal framesize to advertise is the number of tags participating in the frame ($f_{opt} = n$).

3.4.3 Hybrid ALOHA-Tree Protocol Augmentation

This framework is applied to a scenario where both adjacent readers employ HAT protocol [59]. The HAT protocol is a combination of Q Algorithm (ALOHA-based protocol) and EAA (tree-based protocol). The hybrid protocol employs the Q Algorithm and whenever a collision occurs, it is resolved immediately using the EAA approach.

The adapted framework assigns the first reader to employ the Q algorithm variation of the ALOHA protocol. As explained in [59], the Q algorithm starts with a small frame size f (typically 16 to 32 slots in the frame.) The reason behind choosing a smaller frame size is that the frame size continuously varies based on the outcomes of each slot as described in Section 3.2 to adapt itself to match the number of tags in the reader's vicinity. The incoming tags participate in a number of frames at the first reader. A tag stays within the reader's range for a time period T_r as shown in equation (3.1). As stated above, the tag must get a certain number of chances to take part in the reader's frames. However, in a dense tag environment, if the tags are allowed to participate in every frame during their entire duration of T_r in the reader's range, it may lead to increased collisions and therefore unproductive frames. If δt is time taken for one time slot, the number of time slots that a tag sees within a reader's range is $T_r/\delta t$. In order to ensure that the tags do not crowd the frames exponentially, after a duration of T_r/k, they cannot participate at the first reader. They need to wait until they reach the second reader to be identified. Algorithm 3 and 4 give the step-by-step execution of the framework applied to the Hybrid ALOHA Tree protocol as employed by reader and tags.

Since the frames are only 32 slots long, they can be referred to as mini-frames denoted by f_i in the algorithm 3. A group of such mini-frames over a period of $T_r/(\delta t \times k)$ time slots duration make up a bigger frame whose count is calculated by F in Algorithms 3 and 4.

The tags record the big frame number (F) and the slot number (S_i) in the last mini-frame in which they participated. Note that the tags can participate in all the mini-frames ($f_i's$) of one big frame (F) but they store the slot information (S_i) of only the last mini-frame they participated in. In this way, tags get several chances to be read by the first reader.

The unread tags have the big frame number (F_i) and the slot number (S_j selected in the last mini-frame of the big-frame F_i) stored in their memory as they move out of the reader's range. The first reader records collided slots in the last mini-frame advertised for a certain big-frame number and passes the vector consisting of the big-frame number, slot number pair ($< F_i, S_j >$) to the next reader.

The second reader accumulates the $< F_i, S_j >$ pairs that it receives from the first reader in a receive-queue. The second reader initiates two more queues (Q1 and Q2). It adds all the entries with the same F_i value (big-frame number) from the receive-queue into the Q1. It adds the initial prefixes 0 and 1 into Q2. As the tags begin to enter the reader's range, the second reader starts by advertising the first entry

$< F_i, S_j >$ from Q1 followed by initiating the tree protocol starting with the prefix 0 of Q2.

All the unread tags in the reader's range compare the advertised $< F_i, S_j >$ to the (frame number, slot number) pair stored in their memory. If there is a match, they can participate in the tree protocol executed by the second reader. Since a frame number (F_i) is advertised for $T_r/(\delta t \times k)$ slots at the first reader, the vectors $< F_i, S_j >$ will have a TTL of $T_r/(\delta t \times k)$ slots in the second reader's queue (Q1) after which all vectors with the frame number F_i will be purged and the second reader moves on to advertise the vectors with the next big-frame number ($< F_{i+1}, S_j >$) in the queue (Q1).

In the case where Q1 runs out of vectors (associated with a big-frame F_i) before the TTL of the big-frame expires, the second reader will run through the same set of vectors ($< F_i, S_j >$) once again until the TTL expires. Note that the second reader starts executing the EAA protocol after waiting for a time period T_r/k. This ensures that all the tags that participated in a big-frame F_i at the first reader are all within the second reader's range when the collision slots corresponding to the same frame F_i are resolved.

Probabilistic Analysis:

The Hybrid ALOHA-Tree Protocol is a combination of the Q and EAA protocols. The throughput of this protocol is also a combination of the throughputs of Q and EAA. The throughput of a protocol is given by $\eta_H = S_H/L_H$, where S_H is the number of successfully read tags and L_H is the total number of slots advertised.

However, in this protocol, some slots are advertised when the reader is executing Q algorithm and other slots are advertised during the EAA. The total number of slots advertised can be broken down for the respective protocols as $L_H = L_Q + L_{EAA}$. The Q algorithm gives a frame (L), and it is executed until a collision is observed in the

Algorithm 3: Hybrid ALOHA-Tree Protocol with Framework - Reader 1 (ALOHA)

```
define f_1 = 32;
i = 1, F = 1;
Timer = T_r / (δt × k);
while true do
    Broadcast(F, f_i);
    Timer = Timer − f_i;
    for S ← 1 to f_i do
        Receive(tagResponse);
        if tagResponse is read then
            Send ACK;
            s_i ++;
        else if tagResponse is collision then
            c_i ++;
            if Timer == 0 then
                Send (F, S) to NextReader (R_{k+1});
            end
        end
    end
    if Timer == 0 then
        n_i = s_i + (2.3922) × c_i;
        ψ = n_i / f_i = (s_i + (2.3922) × c_i) / f_i;
        Timer = T_r / (δt × k);
        F ++;
    end
    n_{i+1} = ψ × f_i;
    f_{i+1} = n_{i+1} + (2.3922) × c_i;
    if Timer < f_{i+1} then
        f_{i+1} = Timer;
    end
    i ++;
end
```

Algorithm 4: Hybrid ALOHA-Tree Protocol with Framework - Reader 2 (Tree)

```
Reader-Reader Process:
define RecvQ;
while true do
    (F, S) = Receive from PreviousReader(R_k);
    RecvQ.EnQue(F, S);
end
Reader-Tag Process:
define Q1, Q2;
Timer = T_r / (δt × k);
while true do
    F' = 0, F = 0, S = 0;
    while F == F' and RecvQ not empty do
        (F, S) = RecvQ.Head;
        if F' == 0 then
            F' = F;
        end
        if F == F' then
            Q1.EnQue(RecvQ.DeQue());
        end
    end
    Timer = T_r / (δt × k);
    while Timer != 0 and Q1 not empty do
        Q2.EnQue(0);
        Q2.EnQue(1);
        (F, S) = Q1.DeQue;
        while Timer != 0 and Q2 not empty do
            prefix = Q2.DeQue();
            Broadcast((F, S), prefix);
            Receive(tagResponse);
            if tagResponse is read then
                Send ACK;
            else if tagResponse is collision then
                Q2.Enque(Concatenate(prefix, 0));
                Q2.Enque(Concatenate(prefix, 1));
            end
            Timer = Timer − 1;
        end
        Empty Q2;
    end
    Empty Q1;
end
```

frame when the Q algorithm takes a break and switches to EAA until the collision is resolved. The frame size of the Q algorithm is given by $L_Q = L - C$, where L is the frame given by Q algorithm and C is the number of collisions in this frame after each of which the protocol is switched to EAA. The slots for resolving the collisions using EAA is given by

$$L_{EAA} = \sum_{j=1}^{C} 2n_j - 1. \tag{3.17}$$

Similarly, the number of successfully read tags is the sum of tags read during the execution of the Q algorithm and EAA, respectively:

$$S_H = S_Q + S_{EAA} = S + \sum_{j=1}^{C} n_j, \tag{3.18}$$

where S_Q is the number of tags read by the reader during the Q algorithm and S_{EAA} is the number of tags involved in the collisions of Q algorithm and are eventually resolved by the EAA.

The Q algorithm is an ALOHA-based protocol and hence the number of successful slots (S) can be derived from the binomial distribution function given in equation (3.15). At the first reader, there will be probabilities associated with tags moving out of reader's range similar to equation (3.16). However, the Q algorithm insists on small framesizes ($L = 32$) because of which the probabilities of tags moving out of range and missing their slots can be considered to be negligible. This problem never occurs at the second reader because the protocol starts executing on the second reader after a time lapse of $T/2$, making sure all the tags that participated in a big-frame are within the reader's range before the big-frame corresponding slots are advertised at all. Hence the throughput of the Hybrid protocol, as given in [59] is,

$$\eta_H = \frac{S_Q + S_{EAA}}{L_Q + L_{EAA}} = \frac{S + \sum_{j=1}^{C} n_j}{L - C + \sum_{j=1}^{C} 2n_j - 1}. \tag{3.19}$$

Equation 3.19 sufficiently models when the tags are stationary and the hybrid protocol is executed until all tags in the reader's vicinity are read. However, if the tags are mobile and have a limited time in a reader's range, it is highly unlikely to achieve the throughput computed in equation 3.19. The value of L_H is nothing but the total time spent by the reader to read a given number of tags. Also, from equation (3.1), the total time spent by a tag in a reader's range is T, i.e., $T/\delta t$ slots. In order to achieve the above throughput in a mobile scenario, the total time taken by the Hybrid protocol (L_H) should not be greater than T, so,

$$L - C + \sum_{j=1}^{C} 2n_j - 1 \leq \frac{T}{\delta t}. \tag{3.20}$$

In the case where $L_H > T$, there will be unread tags because of two reasons: (i) the tags that selected a slot in the frame L will leave the reader's range before they get a chance to participate and (ii) all tags that encounter a collision in the frame cannot be resolved as some of these tags may move out of the reader's range. By running the Hybrid protocol in conjunction with the framework, the Q protocol is assigned to the first reader and EAA is assigned to the second reader. This would require the following two conditions to be met by the first reader and second readers, respectively: $L_Q \leq T/\delta t$ and $L_{EAA} \leq T/\delta t$. Hence the framework applied to the Hybrid ALOHA-Tree protocol considerably improves upon the successful tag read percentage in a mobile tag scenario with two readers.

3.4.4 QTDFSA Protocol Augmentation

In this section, the QTDFSA protocol [37] is modified to work with the conveyor belt mobility scenario. The QTDFSA protocol, as the name suggests, is a combination of QTP [30] and DFSA [17] protocols.

The QTDFSA protocol generates all prefixes of a certain length. It then advertises each prefix with the optimum framesize ($f = 256$) according to the EDFSA protocol [31]. The collisions observed in the advertised frame not only help determine the next frame size but also assist to generate more prefixes. To apply framework to the QTDFSA protocol, the reader does not start its queries from the root (0,1) like in QTDFSA protocol [37]. Instead, the first reader generates all prefixes P of length k. The generated prefixes are stored in a queue and advertised one at a time with a framesize f. When a framesize f is advertised with a prefix P, only the tags with the matching prefix pick a slot in the frame. The first slot in the frame is reserved. The reader expects all the tags that picked a slot in the following frame (or all the tags that have a prefix matching the advertised prefix query P) to respond during the first slot of the frame. If the first slot is a collision, it means there are more than two tags with the matching prefix (P). The first reader goes on with the rest of the frame (if the first slot is a collision); otherwise the frame is terminated and the reader moves to the next prefix in the queue. Each prefix is advertised with only one frame. At the end of the frame, the collisions are recorded and new framesize and the prefix pair ($< F_i, P_i >$) are passed on to the second reader. In the case where the frame is terminated, no information is shared with the second reader.

The second reader accumulates the $< F_i, P_i >$ pairs that it receives in a queue. As the tags start coming into the second reader's range, it advertises the $< F_i, P_i >$ pairs from the queue one at a time. When a $< F_i, P_i >$ pair is advertised, all tags with the matching prefix respond in the following frame. The second reader computes the next framesize to be advertised from the observed collisions in the this frame. In case there are collisions, the same prefix is advertised again with the calculated framesize. A prefix can be advertised multiple times with different framesizes until all tags with the prefix in the second reader's range are identified. Algorithm 5 gives the step-by-step execution of this framework.

Algorithm 5: QTDFSA Protocol with Framework

```
Reader-Reader Process:
define Queue;
if !FirstReader then
    while true do
        (F, P) = Receive from PreviousReader (R_{k-1});
        Queue.EnQue(F, P);
    end
end
else
    Queue.EnQue(Generate all prefixes of length k);
end

Reader-Tags Process:
RepeatPrefix = false;
while Queue not empty do
    if !RepeatPrefix then
        (F, P) = Queue.DeQue();
    end
    else
        RepeatPrefix = false;
        F = F_next;
    end
    Broadcast(< F, P >);
    for S ← 1 to F do
        Receive(tagResponse);
        if tagResponse is read then
            Send ACK;
        else if tagResponse is collision then
            c++;
        end
    end
    F_next = 2.3922 × c;
    if !FirstReader & c > 0 then
        RepeatPrefix = true;
    end
    else
        Send (< F_next, P >) to NextReader (R_{k+1});
        if Queue is empty then
            Queue.EnQue(Generate all prefixes of length k);
        end
    end
end
```

Probabilistic Analysis:

The main motivation behind the QTDFSA protocol is to improve the throughput of DFSA protocol by grouping the tags together on the basis of prefixes. The QTDFSA protocol starts by using the shortest prefixes (0,1) to group the tags. It then adds more prefixes to the pool depending on the outcome of the frame. If the prefix length is small, the number of tags with that prefix will be large and the advertised framesize ($f = 256$) will have a lot of collisions. On the other hand, if the prefix is too long, the number of tags with that prefix will be less and the advertised framesize ($f = 256$) will have a lot of idle slots. The prefix size must be appropriate so that the number of tags with the matching prefix is close to the advertised framesize ($f = n$).

The QTDFSA protocol in a stationary tags situation has enough time to start at the root with prefixes 0, 1 and slowly calculates the right prefix length at which $f = n$ is achieved. However, when the tags are mobile, the reader does not have sufficient time to experimentally calculate the appropriate prefix length. That is the reason why all prefixes of a certain length (k) are calculated and the QTDFSA protocol starts by advertising the framesize with the $k-$length prefix.

The prefix length k is chosen before experimenting with any other prefix P of length $|P| < k$. This may lead to a possibility where some prefixes from the prefix pool (of length k) may have no tag with a matching prefix at all. The frames for such prefixes will go completely empty. In order to prevent this situation, the first slot of every frame is reserved to find out if there are some tags with the matching prefix. All the tags that choose a slot in the following frame must respond in the following frame and respond with their IDs in the first slot. If there is no response, the frame for the prefix is canceled and the reader moves to the next prefix. If there is only one response, the one tag that responded is recognized and acknowledged and the frame is canceled. If there is a collision, the reader continues with the rest of the slots in the frame.

The prefix length k is chosen appropriately so that the frames advertised with the prefix are most productive. Suppose that the tag IDs are x bits long and the number of tags can be no more than 2^x. If P is a chosen prefix of length k, the number of tags with a matching prefix can be no more than 2^{x-k}. As suggested in DFSA protocol [17], the framesize is most efficient if the number of matching prefixes is approximately equal to the framesize advertised as shown in the following equation:

$$2^{x-k} \leq f. \tag{3.21}$$

From equation (3.21), the appropriate prefix length k can be determined as

$$k > x - log_2 f. \tag{3.22}$$

In QTDFSA, both the prefix and frame size are generated at the end of a frame. In the case of mobile tags with new tags moving into the range and older ones moving out continuously, changing both prefix and frame size may make the execution of the protocol highly unstable. In other words, a bad frame (one with too may collisions or idle slots) could be either because of incorrect frame size or prefix calculation or simply because some of the older tags left the reader's range and new tags moved in creating disorder. In order to fix that, the initial prefix length (k) is calculated using equation (3.22) with the condition $f \leq 256$. The calculated prefix length k is kept fixed for every query but the associated framesize is varied in accordance with previous frame information.

The framework handles these situations by fixing the prefix length. The protocol starts by generating all prefixes of length k. Each prefix is advertised only once at the first reader. The number of collisions help to determine the next framesize for a given prefix. At the second reader, a prefix is advertised multiple times until all tags in the reader's range are picked up. Now, keeping k fixed makes QTDFSA in the mobile scenario similar to the ALOHA in mobile scenario, as shall be described in Section 4.2. The only difference is that the frames are grouped further on the basis of prefixes. The throughput for the QTDFSA in mobile scenario with multiple readers can be given by

$$\eta_{QTDFSA} = \frac{1}{2^k} \sum_{j=1}^{2^k} p_s(n, f) \tag{3.23}$$

where $p_s(n, f)$ is the probability of successfully reading a tag as given in equation (3.12). This throughput is achieved by employing the QTDFSA protocol in conjunction with the framework and hence the read percentage for QTDFSA is higher when compared to the same without framework.

3.5 Performance Evaluation

This section studies the performance of the proposed frameworks for different protocols, with the number of killed tags as the metric. This is a complementary metric in a way that lower the number of killed tags and betters the protocols' reading throughput or efficiency. The efficiency of a protocol can be measured as,

$$Efficiency = 1 - \frac{Number\ of\ tags\ unread}{Total\ tag\ population}. \tag{3.24}$$

Table 3.1 gives the tag-reading percentages (Efficiency %) of protocols and their augmented counterparts.

It is to be noted that those tags that come out of the second reader's range unidentified are considered as killed tags. The interrogation range D_{IR} is set to $4m$ for the reader mounted on a height h of $1m$. For the considered settings, the arrival rate ψ of the tags can be computed as follows: $\psi = \rho \times v = 4$ tags/ms . The simulations are done in custom built C++ programs.

Tag Count	1000	1500	2000	2500	3000	3500	4000
TP-M	95.05	75.24	64.80	58.39	54.23	51.39	49.21
TP-M-F (**)	98.86	81.38	70.05	63.65	59.50	56.00	53.78
AP-M	100.00	99.99	98.31	94.20	90.85	87.17	85.43
AP-M-F (**)	100.00	100.00	100.00	99.74	97.72	94.23	90.60
HAT-M	86.13	68.84	59.81	55.01	50.66	47.84	46.04
HAT-M-F (**)	69.34	66.17	63.26	56.95	52.89	54.24	53.61
QTDFSA-M	100.00	96.95	86.71	80.68	75.62	73.17	71.13
QTDFSA-M-F (**)	93.89	92.57	87.77	81.74	78.38	76.73	74.85

Table 3.1: Tag reading percentages (Efficiency %) of different protocols and their augmented counterparts.

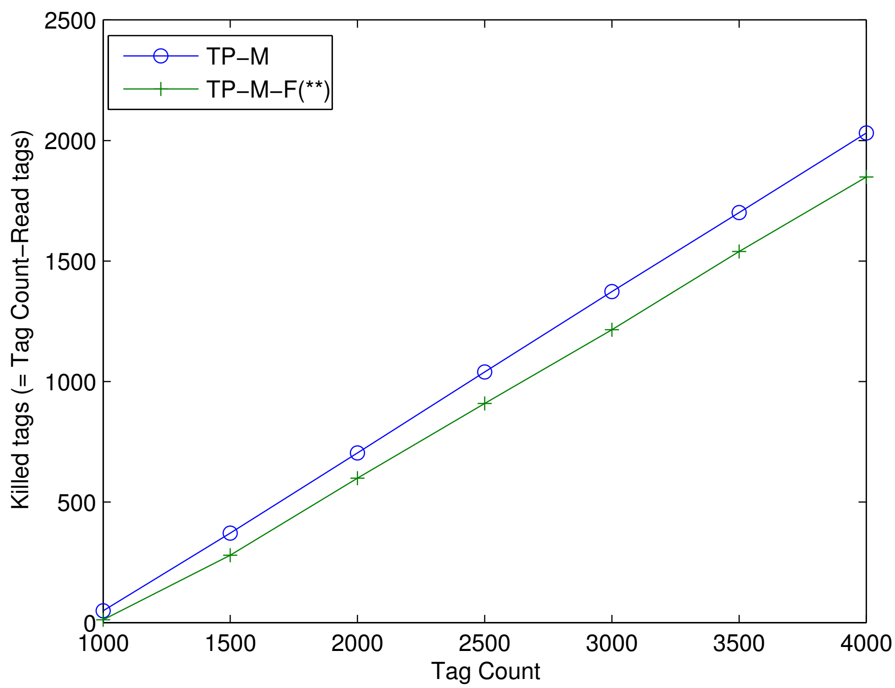


Figure 3.3: Protocol performance with and without framework: Tree protocol (TP-M) vs. Tree protocol with framework (TP-M-F(**))

Figure 3.3 shows the performance of a simple tree protocol compared with its augmented framework counterpart. It is intuitive that an increase in tag count increases the number of killed tags in a linear fashion.

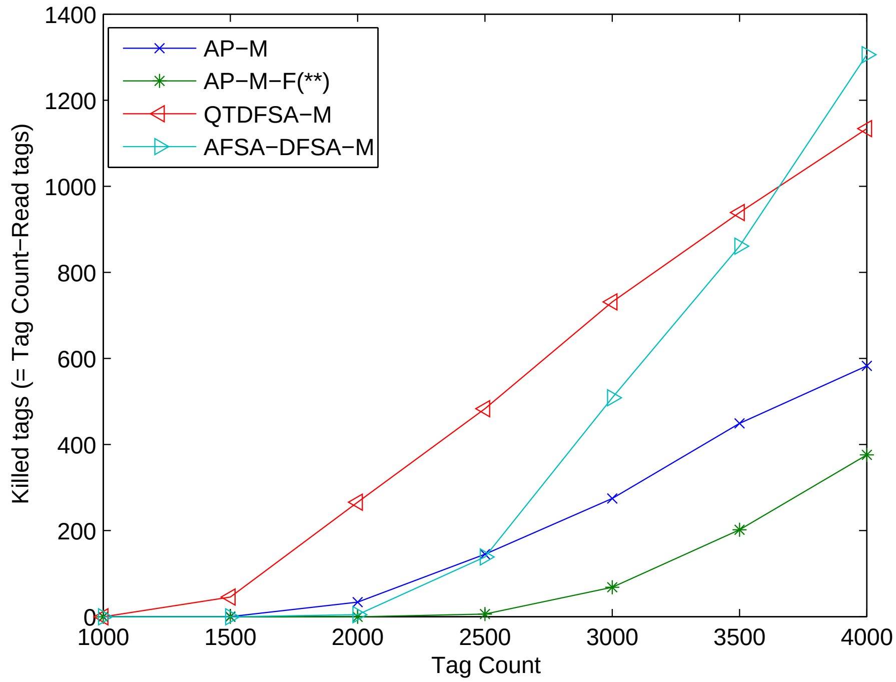


Figure 3.4: Protocol performance with and without framework: ALOHA Protocols (AP-M, AFSA-DFSA-M), Hybrid Protocol (QTDFSA-M) vs. ALOHA Protocol with framework (AP-M-F(**))

Figure 3.4 plots the performance of ALOHA protocol (AP-M) and its framework augmented version (AP-M-F) along with different variants of ALOHA protocol such as QTDFSA-M and AFSA-DFSA-M. The framework augmented with variant of the simple ALOHA protocol (AP-M-F(**)) significantly reduces the number of killed tags to about 50% more than its unmodified version (AP-M). The framework augmented with simple-ALOHA protocol outperforms all other variants of ALOHA as well. These ALOHA variants are two-phased protocols. For instance, the first phase of QTDFSA involves an initial prefix generation phase and AFSA-DFSA uses minislots for slot reservation for the following phase. The second phase of both protocols is similar to

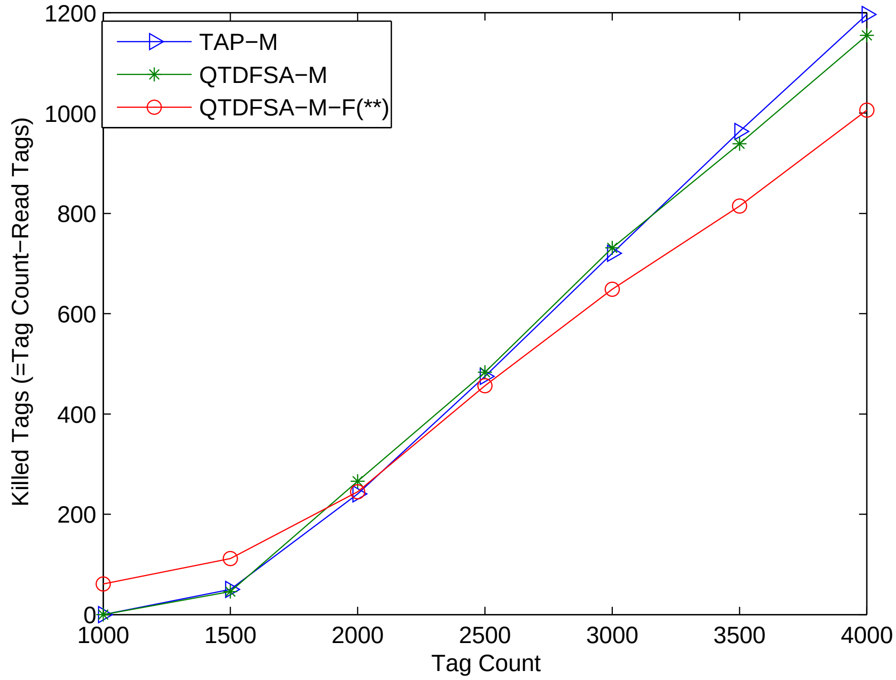


Figure 3.5: Protocol performance with and without framework: QTDFSA Protocol (QTDFSA-M) vs. QTDFSA Protocol with framework (QTDFSA-F(**))

the framed ALOHA. And the variants execute both the phases in a cyclic fashion one after another. And with an increase in tag count, it takes more time to complete each phase and leads to more collisions. Due to this fact, the number of killed tags increases in a higher rate than the simple ALOHA protocol. Hence the protocols that involve multiple phases perform poorly in a mobile scenario and the information sharing is seldom useful.

Hybrid ALOHA-Tree protocol and QTDFSA protocol perform fairly well in a mobile 2-reader scenario without the framework for lower initial tag counts as evident from the figs 3.5 and 3.6. However, when tag count goes over 2500, the performance of both Hybrid ALOHA-Tree protocol and QTDFSA protocol increases considerably when executed with framework. This performance difference between higher and lower initial tag counts can once again be attributed to the two phased nature of the hybrid protocols. In other words, the overhead brought about by employing the

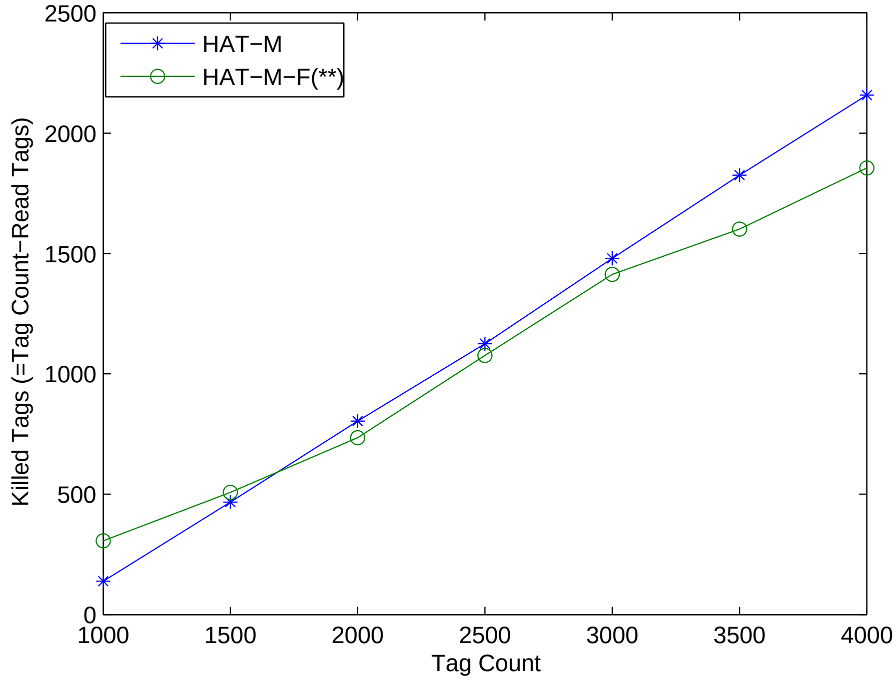
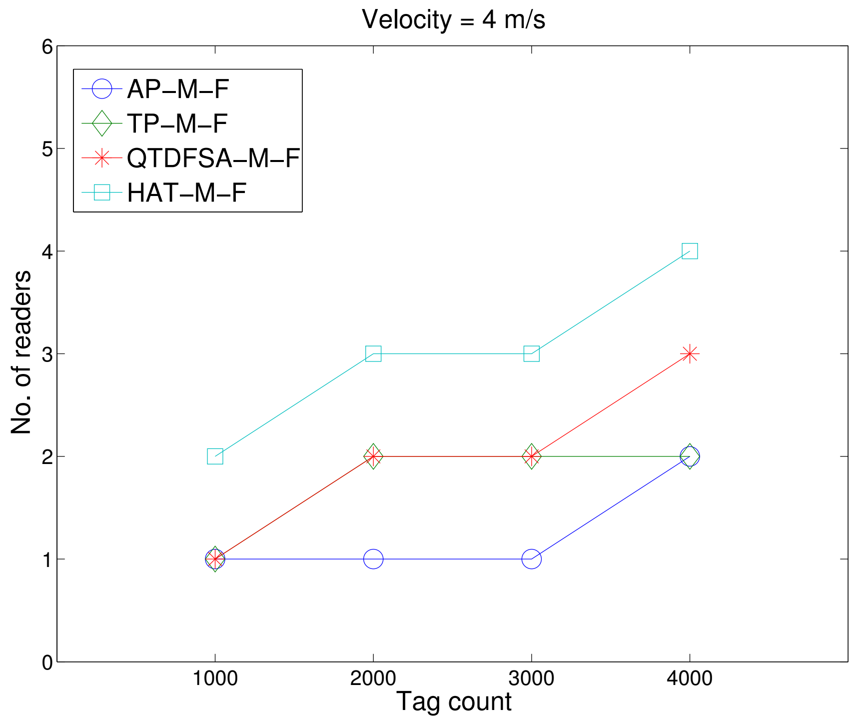


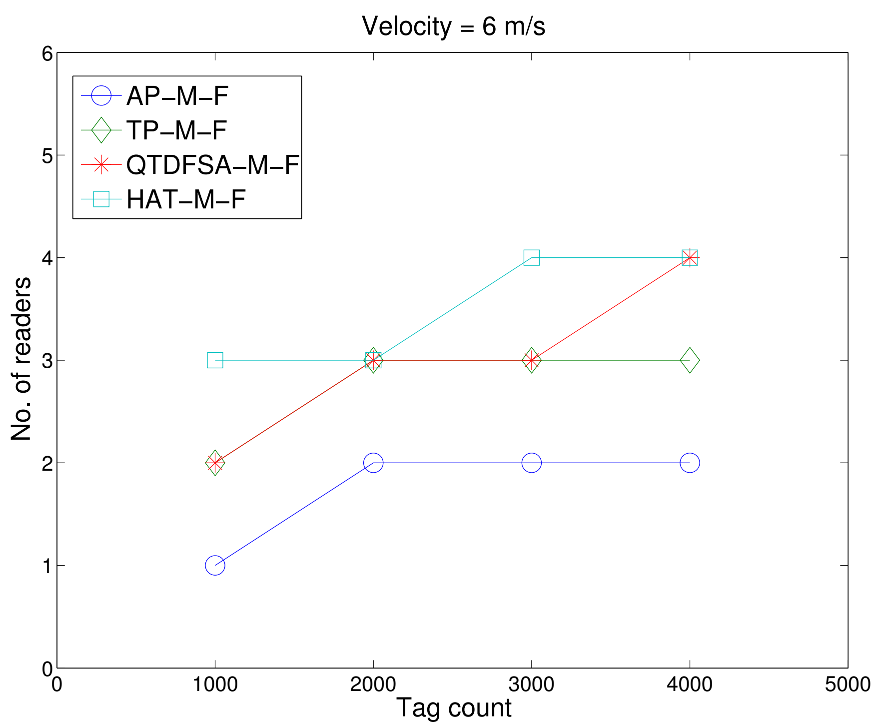
Figure 3.6: Protocol performance with and without framework: Hybrid ALOHA-Tree Protocol (HAT-M) vs. Hybrid ALOHA-Tree Protocol with framework (HAT-F(**))

framework is significantly high for lower tag counts making both the hybrid protocols perform about the same, whether framework is applied or not. As the initial tag count increases, the overheard of employing the framework becomes negligible compared to the overhead caused due to increased number of collisions in the Hybrid protocols.

The velocity of conveyor belt is directly related to the number of unread tags at the end of second reader. Also, if more readers are added after the second reader over the conveyor belt, leftover tags can be identified. The number of readers in tandem arrangement required to read at least 95% of the tags, initial tag population and velocity of conveyor belt are all dependent on one another. Figures 3.7(a), 3.7(b) and 3.8(a) show this relationship between varying tag population and number of readers required for 95% coverage for the velocities 4m/s, 6m/s and 8m/s, respectively. Figure 3.8(b) gives the number of readers required for 95% coverage with varying the velocity of conveyor belt for a fixed tag population of 4,000 tags. From the figures, ALOHA

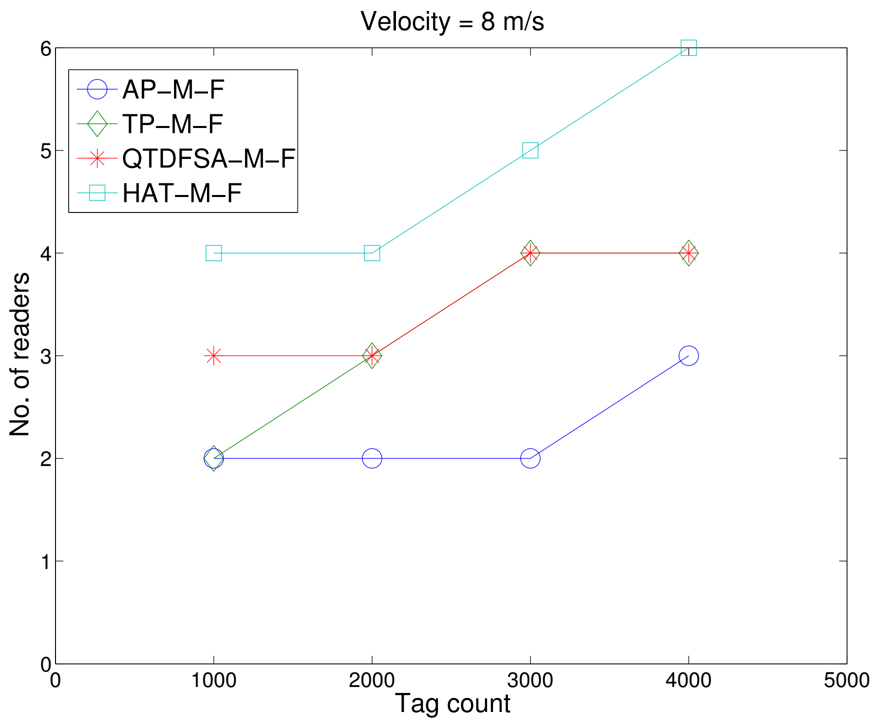


(a) Varying tag population, Velocity = 4m/s

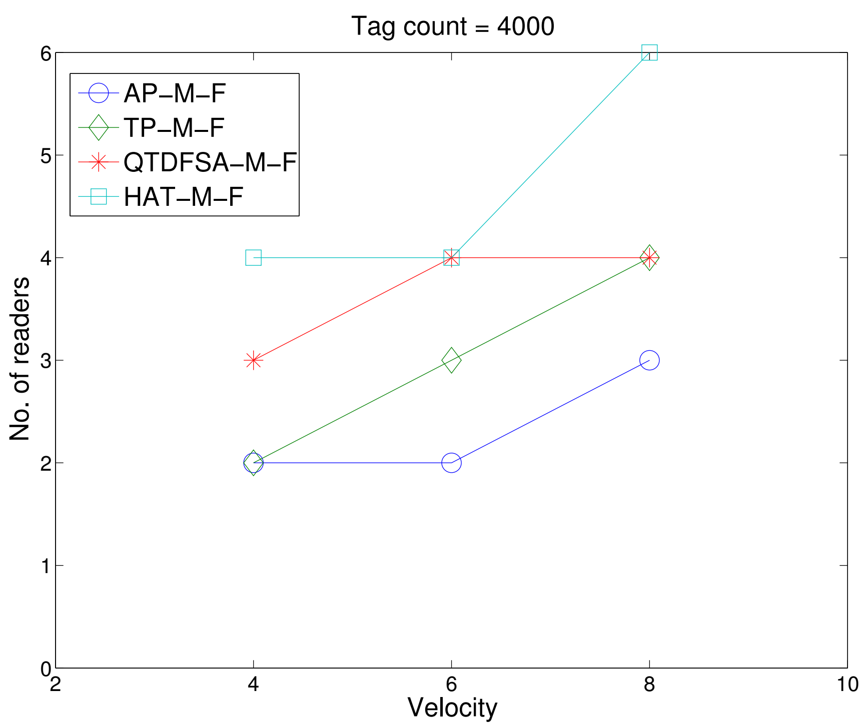


(b) Varying tag population, Velocity = 6m/s

Figure 3.7: Number of readers required to achieve $\geq$ 95% tag reads, for different number of tags population, on a conveyor belt with varying velocities from $4m/s$ to $8m/s$.



(a) Varying tag population, Velocity = 8m/s



(b) Varying velocities, Tag poulation = 4000

Figure 3.8: Number of readers required to achieve $\geq 95\%$ tag reads, for different number of tags population, on a conveyor belt with varying velocities from $4m/s$ to $8m/s$.

protocol requires the least number of readers over all velocities and tag populations.

The protocols show an improvement with the framework. However, the tags used with framework may need more storage capacity than regular passive tags. They may need to store their state information from the previous reader such as, frame or slot number they last participated in or whether they were muted (identified) at the first reader. This information is required for the framework to work. There will be a tradeoff between using tags with more memory that are more expensive and the efficiency that the framework offers for the underlying protocol.

3.6 Summary

This chapter proposes consecutive reader's information-sharing frameworks to improve the tag-reading efficiency of the protocols functioning in a conveyor belt mobility scenario. The proposed augmentation frameworks for ALOHA, tree, and two different combinations of hybrid tag-reading protocols identifies different types of information-sharing for each one of them. The augmented protocols outperformed from their respective native counterparts, with the augmented ALOHA protocol showing a maximum improvement of 50%. The improvement gained due to the augmentation has been mathematically studied. Furthermore, the scalability problem of determining the number of readers required to identify almost all tags in the system has also been addressed. The research in this chapter has been adapted from [40].

Chapter 4

Zonal Spacing Framework with Mobile Reader in a Warehouse

Storage warehouse plays a crucial role in managing the logistics of a supply-chain industry. Maintaining the sheer volume of goods in a warehouse requires a daunting task of bookkeeping. While a manual counting is impossible, automating inventory management with the help of RFID technology can be a viable and scalable solution. The warehouse facility has a typical arrangement of a grid pattern, which provides an easy access for storage and retrieval of goods. The movement of RFID reader is therefore restricted along the aisles between the storage units. The standard RFID tag-reading protocols with their generic design philosophies would abstain from rendering their best performances in the typical warehouse scenario, where the reader moves along the aisles.

This chapter proposes a novel zonal spacing framework to improve the efficiency of the tag-reading protocol. In this framework, the tags in a storage unit (*a.k.a.* grid) are envisioned as being packed logically into multiple non-overlapping compartments called zones. These zones are strategically demarcated based on the footprint traces of the passing-by reader's signal. The tags' participation in the reading process is then dictated by their respective zonal mappings. With adequate mathematical reasoning, the improved efficiency of the RFID reading protocol augmented with the zonal spacing framework is demonstrated. The performance improvement due to the proposed framework is corroborated with the help of extensive simulation, which shows up to three orders of magnitude of improvement in performance.

Note to Practitioners: An RFID reading protocol enables a reader to uniquely identify tags in a group and inventory them. This research proposes to redesign a rudimentary reading protocol in order to improve its efficiency while functioning in a warehouse grid scenario. The candidate reading protocol conforms to industry standard EPC-C1G2, which is also ratified as ISO 18000-C. The proposed augmentation framework for the tag-reading protocol is still in compliance with the industry standards, without requiring additional (specialized) hardware components.

4.1 Overview

RFID wireless identification technology uses radio frequency transmissions to track or identify tagged objects. The products (or goods) of interest are pinned with tiny inexpensive electronic tags that respond to one or more passing-by reader(s) that scan them. These tags are mostly passive as they do not have their own power source such as battery, which eventually makes them cheap, miniaturized in form-factor, and less in weight. Passive tags are energized by the reader's Electro-Magnetic (EM) field and communicate with them by backscattering the EM signal.

The RFID reader executes a light-weight tag-identification protocol to query the tags under its radio range, looking for unique embedded IDs called EPCs. In a one-time initial setup phase, the EPCs of tags are mapped to the corresponding products to which they are tagged/pinned. The EPC information therefore helps in retrieving the product's information (such as expiry date and price), as maintained in a back-end database.

The performance of a tag identification or reading protocol is quintessential in determining the speed of the scanning process. Several RFID tag-reading protocols such as FSA protocol [17] and QTP [30] are naively designed, by mainly targeting

Figure 4.1: Inside an Amazon warehouse. Contiguous storage units (grids) and constrained aisle-based pathways for the reader mobility.

on the static tags and readers scenarios. Of late, the RFID applications landscape is potentially shaping up with the inclusion of mobility of tags (eg., conveyor belt) or readers (eg., warehouse). To cope up with the performances in such mobile scenarios, these native and rudimentary tag-reading protocols need to evolve and optimize their functionalities by considering the mechanics of the tags' or readers' locomotion.

In this line of work, a protocol augmentation framework is developed for a warehouse scenario. A typical snapshot of an Amazon warehouse is shown in Figure 4.1, highlighted with a schematic overlay representing the mobility of reader and storage arrangements of tags. Figure 4.1 shows the restricted aisle-based pathways for the readers' mobility. It highlights the storage units (also known as grids) with an array of contiguous assembled cartons of products in each grid. Cartons in a storage grid could potentially have pallets of products pinned with RFID tags.

The novel framework presented in this chapter exploits both the aisle-based mobility of the reader and the typical storage arrangement of static tags in a warehouse. The protocol augmentation is henceforth called as a zonal spacing framework, wherein the tags in a storage unit (*a.k.a.,* grid[1]) are logically segregated into different zones. These zones are demarcated based on the exposing intensity patterns of the passing-

[1]henceforth, the terms grid and storage-unit will be used interchangeably.

by reader's signal. The tags' participation in the reading process is then dictated by their respective zonal mappings. As the partitioning is based on the mobile reader's footprint traces of its EM field, different patterns of strong and weak exposure areas (zones) of tags can be identified and logically partitioned from other parts of the grid. Restricting tags' participation from certain regions of exposure would help reduce collisions and at the same time would not stress much on the reading protocol's performance. This way of enabling strategic participation of tags based on their logical partitioning would help in reducing collisions at the reader, thereby improving its reading performance.

The research contained in the current chapter has been gradually developed with an incremental approach by demonstrating improvement of the tag-reading framework on a single storage unit with the mobile reader looping around it. Subsequently, the proposed framework has been extended to demonstrate its veracity on a multi-grid mobility scenario (*i.e.*, a scenario with reader's mobility spanning the entire warehouse, involving multiple storage units). With adequate mathematical reasoning and with the help of extensive simulations, the prowess of the proposed framework has been validated. The scalability aspect of the system has been analytically studied by identifying the suitable design parameters that impact the reading protocol's performance.

The remainder of this chapter is organized as follows. Section 4.3 discusses known research/results related to the mobile readers and their usage in warehouse scenarios, especially from the context of improving anti-collision protocol's performance. A brief background information is provided on the nature of tags and anti-collision protocols considered in Section 4.4. Section 4.5 introduces the warehouse problem considered throughout in the current chapter. The crux of the work describing the zonal spacing framework is presented in Section 4.6 and the implications of this framework in multi-grid scenario in Section 4.8. Section 4.8 also studies the framework from

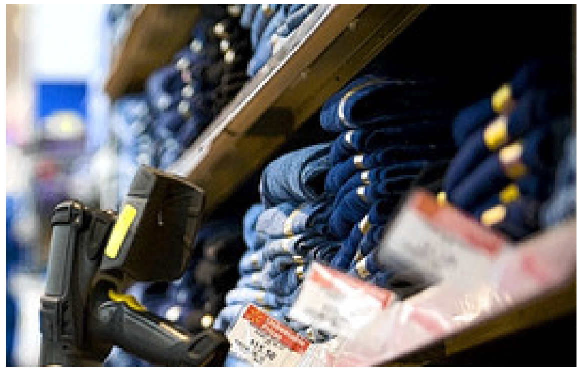


Figure 4.2: RFID handheld scanner used to read RFID tag labels for inventory checking at a Wal-Mart Supercenter. Image source: Wall street journal

scalability perspective in identifying different parameters impacting on the protocol's performance. Section 4.10 summarizes the research results contained in the current chapter.

4.2 Motivation

Warehouse is an integral part of the supply-chain industry [23]. Large volumes of goods are stocked in warehouse making it a massive buffering place that helps in smoothening the end-to-end fluctuations of a supply-demand chain. Warehouse-style of goods storage is also popular in retail store businesses such as apparels. The stocked goods in warehouses need to be monitored periodically for essential purposes such as inventory management and theft prevention and control. The sheer volume of goods forbidding the usage of manual accounting process therefore calls for automation. To this end, RFID technology is considered as a promising solution to improve the efficiency of a supply chain.

Researchers are actively studying the implications of RFID item-level tagging in retail businesses [38]. Retail majors such as Wal-Mart have rolled out RFID-enabled inventory management technology (shown in Figure 4.2) into their businesses. The ground truth is that the RFID-tags implementation has helped Wal-Mart improve sales by keeping shelves better stocked [9]. Usage of RFID in the stores has reduced the out-of-stock merchandise by 16% [9]. A business angle of this improvement could be perceived from the following: By the Wal-Mart sales in fiscal year 2003, a 1% improvement in the out-of-stock issue could generate nearly $2.5 billions [8].

4.3 Related Work

Research on RFID pertaining to the warehouse inventory applications [57], [34], [20], [54] have mainly focused from a software architecture perspective, especially in developing or improving the middleware, to enable efficient access of data from the RFID readers to the servers. Independently, as a promising step towards smart-retail shopping with item-level RFID tagging, usage of mobile RFID readers have been encouraged. Especially attempts have been made to use pervasive mobile phones as mobile RFID readers [26], [11], [35]. For instance, the authors in [35] propose the development of prototypes of low power, low cost, small size RFID reader chip that can fit on a mobile phone.

Recently, the research encouraging the potential use of mobile readers in warehouses has gained traction. The authors in [18] propose a secure channel for mobile RFID reader to the server (say, to a cash register at the sales counter), to encourage M-commerce. The authors in [32] consider a warehouse scenario with active static tags forming a mesh network for the purposes of data delivery and recommend the usage of mobile readers due to their most power-efficient way to collect data.

There has not been much research on improving anti-collision protocol focusing on mobile reader and static passive tags scenario. The composite problem involving mobile reader operating in a warehouse scenario (containing static passive tags) from the perspective of improving tag reading protocol is a practical problem to be addressed.

Several tag-identification protocols have been developed in order to handle tag collisions. Examples include QTP [30], BTP [14], and DFSA [17]. These protocols are rather devised for a generalized setting that are mainly intended to work in static scenarios representing static reader and tags. These protocols have been discussed in detail in Chapter 2.

Lately, a few research works have focused on mobile environments dealing with mobile tags and stationary (multiple) readers scenario. Some of them are ASAP [27], AFSA framework [39], and framework with cooperative readers for mobile tag environments [40]. ASAP and AFSA provide frameworks to accommodate moving tags scenario (say, on a conveyor belt) to work more efficiently by tailoring protocols that are actually designed for static scenarios. In a similar mobile-tag setting, the framework with cooperative readers uses protocol state information-sharing mechanism among them in order to improve the tag-reading performance.

The mobile tags-and-static reader problem is significantly different from a mobile reader-and-static tags scenario. While in the former case, a reader interrogates the tags for only one continuous episode of time (say, a conveyor belt scenario) whereas in the latter, a reader makes multiple sporadic visits to interrogate the tags from the specific region. Especially in this work of warehouse scenario, the unique storage structure of a grid-partitioned warehouse is exploited to improve the tag-reading performance of a revisiting mobile reader.

From the perspective of zonal-based partitioning of tags, the authors of [13] have proposed the use of micro-zones within the interrogation area of the reader. However, this is realized with specialized multiple antennas at the readers and special hardware (similar to a wireless sensor node), namely, Fielder. The fielder clusters the group of co-located tags' response and communicate to the reader, via a Zig Bee protocol. Orthogonal to their work, the proposed way of logical zone partitioning is possible with the off-the-shelf tags and readers, thereby making it industry-ready.

4.4 Background

Tags store their IDs or EPCs in the form of a binary number. While there are different types of tags available in the market, EPC class C tags are the most industry-preferred tags. They offer simple and essential computing and storage resources yet remain inexpensive. For instance, they support read and/or write memory access. This gives flexibility of assigning tag IDs at the end-user's place, not only at the manufacturing time. The write operation serves to store user-defined protocol information as well. Moreover, the EPC class C tags are passive tags that harvest energy from the proximity readers. This eliminates the need of a separate power source, which tremendously reduces the manufacturing cost, size, and weight of the tags. The passive tags are main consideration for performance study owing to their relevance and popularity.

A variant of FSA is considered as the tag identification protocol for the study contained in this chapter, especially the DFSA [17]. The FSA-class of reading protocols are industry preferred as they conform to EPCglobal C2G1 standards. The FSA protocols are more light-weight than the tree protocols (such as QTP [30]), as the latter require additional prefix-matching operations in the tag's hardware. The functionality of DFSA protocols works in a time-slotted manner. A sequence of time-slots is clumped together, and is called a frame. The behavior of the protocol happens in terms of multiple sequences of non-overlapping frames. In the beginning of each frame, the reader advertises a frame size (*i.e.*, the number of slots in the following frame). Each tag with its minimal hardware can perform two basic functions, namely the random number generation and the counting operation. With the advertised framesize, each tag independently chooses a random integer within the range of the framesize. And this number determines a specific slot the corresponding tag should respond to the reader. The counter initialized to this random number would clock (in the scale of a time slot duration) until it becomes zero. On reaching zero, the tag will respond to the reader in order to transmit its EPC ID. In this manner, the multiple tags' response are stochastically distributed in time. This naturally reduces the tags collision to some extent.

When a single tag responds in a slot, the reader can recognize the tag and its ID successfully, in which case the slot is called a singleton slot. This process of tag identification is also called singulation. A slot is deemed as collided when multiple tags respond simultaneously in that slot, thereby failing the reader to recognise them. The other possibility would be no tags responding to the reader making it an idle slot. In subsequent frames, the identified tags would remain muted and therefore do not participate in later rounds of identification process. Moreover, the subsequent framesizes are determined by the number of collisions and idle slots in the previous frames. Owing to this nature of changing framesizes, this protocol is named as DFSA.

The collisions and idle slots degrade the performance of tag-reading protocol. In a system with reasonable tag density within the reader's interrogation range, collision slots predominately occur and degrade the protocol's performance. Much of the research effort is therefore being invested in devising efficient protocols to mitigate tag collisions.

4.5 Warehouse Problem

Similar to the observed storage layout of an Amazon warehouse from a snapshot shown in Figure 4.1, the following research considers a generic layout of a warehouse scenario consisting of multiple storage areas separated by aisles as shown in Figure 4.3 (a). This aisle space around the storage areas allows for easy movement of goods in and out of the area. An RFID reader mounted on a vehicle (*e.g.*, forklift) or carried by a person can move around the aisle space for scanning tags within the communication range. A warehouse layout with square grids is opted for study, as shown in Figure 4.3

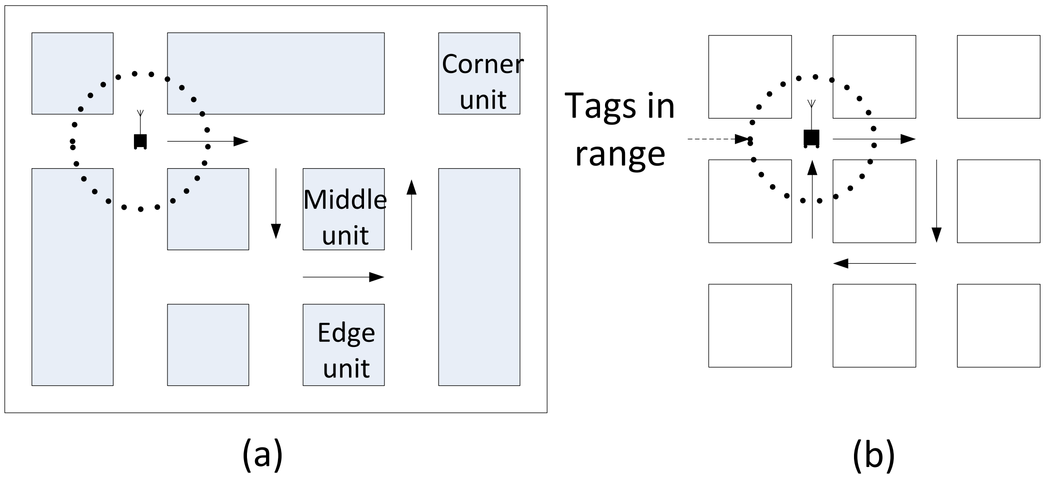


Figure 4.3: Warehouse schematic with a mobile reader

(b). The reader makes its way along the aisles between the grids and moves with a constant velocity. The passive tags are assumed to have a minimal storage (a few bits) to store the zonal information (the detail of which is deferred to Section 4.6). In each grid, the tags are assumed to be uniformly distributed inside the grid storage area. Without loss of generality, a 2-dimensional spatial structure of grid storage is considered in this work. Nevertheless, a 3-dimensional storage grid can be envisioned as a stack of multiple racks piled upon one another. The entire scanning process can be perceived as repeated 2D plane scanning on each of the racks.

Initially, the improvement caused by zonal spacing technique is studied on a single grid with a reader looping around it, as shown in Figure 4.3 (b). Subsequently, the study is extended by incorporating the proposed framework to a multi-grid setting.

The reader is assumed to be aware of the protocol-state information at any point in time. This can be made feasible by either of the following two ways:

- Centralized connectivity - The reader is assumed to be connected to an in-house WiFi infrastructure[2], which typically serves to send the scanned tags information to a central database. This communication channel is utilized to

[2] A communication infrastructure connecting the readers to the backend database server is an integral part of the inventory management.

store protocol state information as well and this helps enabling multiple readers to participate in multiple rounds of a single reading process.

- Distributed connectivity - In absence of a centralized connectivity, special control tags can be placed permanently at different strategic locations inside the storage grid. These are plain passive tags that are exclusively meant for storing the protocol-state information in each frame cycle. A reader picks up this information upon revisiting the same place and continues the reading process.

4.6 Zonal Mapping Framework

The research starts with focus on a single grid with a looping mobile scenario and extends to a composite scenario involving a multi-grid with the reader's mobility spanning entire warehouse. In the considered single-grid setting, the reader laps around the storage area as shown in Figure 4.3 (b), for multiple times, in order to identify all the tags (products). The single-grid scanning scenarios have practical importance such as requiring the end-user to scan a particular storage grid that has seen a recent arrival of new goods and at the end-of-day bookkeeping of selective grids in a retail store that has been observed with product purchases.

During this scanning process, the number of neighboring storage areas also affects the protocol's performance. Therefore in reality, a looping reader sees tags participating from neighboring areas as well.

Based on the relative spatial location in the grid, as shown in Figure 4.3 (a), the storage areas can be categorized into three different types as follows:

- Corner unit - having neighboring storage units only along two edges,

- Edge unit - having neighboring storage areas along three edges, and

- Middle unit - having neighboring storage areas all around it.

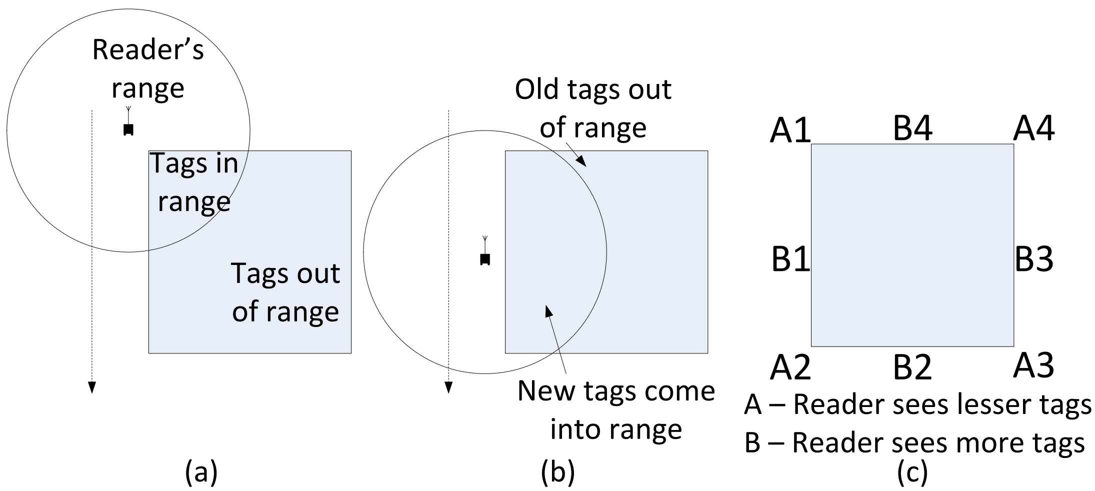


Figure 4.4: (a,b) Mobile reader with tags going in and out of its range, (c) Reader location

For obvious reasons, the number of candidate tags participating in the protocol is maximum for the middle units and minimum for the corner units. The way the tags get exposed as the reader passesby is shown in Figs. 4.4 (a) and 4.4 (b).

The area of storage space within the reader's range in Figure 4.4 (a) is much smaller than the area shown in Figure 4.4 (b). In a similar way, considering a regular spacing layout of neighboring grids, the number of neighboring, participating tags in the corner is less than those in the middle. Unless otherwise mentioned, the middle unit of a grid is primarily considered for this study. It is important to note that the large number of tags leads to more collisions, thereby depleting the performance of the tag-reading protocol.

Figure 4.5 (a) shows the radio-signal footprints marked by a reader while lapping around a storage grid. These footprints show a definite pattern of different overlapping regions of tags' exposure within the reader's interrogation range. The tags in corners fall in the overlapping zones as opposed to the tags along the sides, especially in the middle of the grid.

Initially all tags within the grid are oblivious to their relative locations in terms of their higher granular levels (such as the corners and sides of the grid). In each strategic

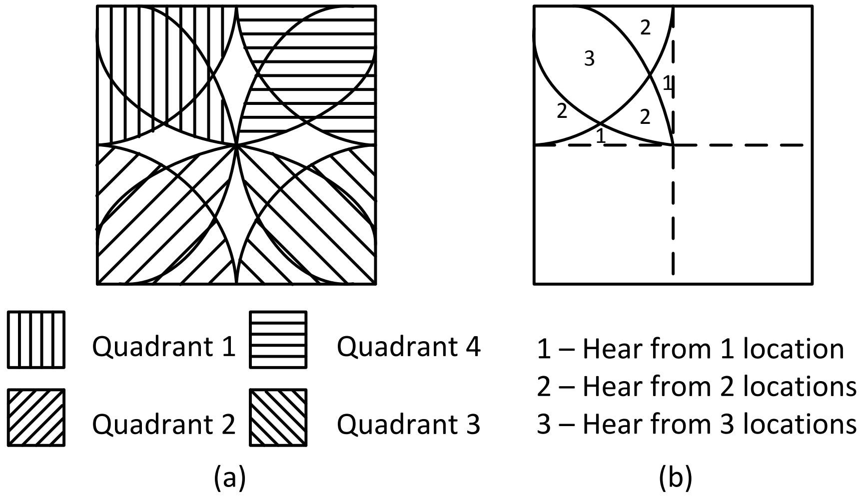


Figure 4.5: (a) Tag location quadrants, (b) Reader coverage

locations around the grid, as indicated by different labels in Figure 4.4 (c), around the grid, the reader advertises specific control information to the tags within its read range. Subsequently each tag tends to hear one or more of these advertisements based on their placements within the grid. In this way, the tags based on their differential number of hearings can align themselves into different logical zones.

Figure 4.5 (b) shows such a logical zone segregation of tags based on their hearing capabilities at different locations of the passing-by reader. For instance, the tags in zone 3 can hear the reader from three different passing segments, namely B1-A1, A1, and A1-B4. In a similar manner, the zonal demarcation and the overlapping regions of exposure for a rectangular grid is shown in Figs. 4.6 and 4.7, respectively.

Once the tags are designated to their respective zones, their participation to the reading protocol is decided based on their assignments. As shown in Figure 4.5 (a), the square grid shows symmetric patterns along the four different quadrants of the grid. Let these quadrants form a set Q comprising all the quadrants $Q1, Q2, Q3, Q4$. Those tags that hear the reader from more than one locations can be assigned to their respective quadrants. For instance, tags from logical zones in Figure 4.5 (b) labeled by 2 and 3 can be assigned to quadrant $Q1$. On the other hand, those tags that hear

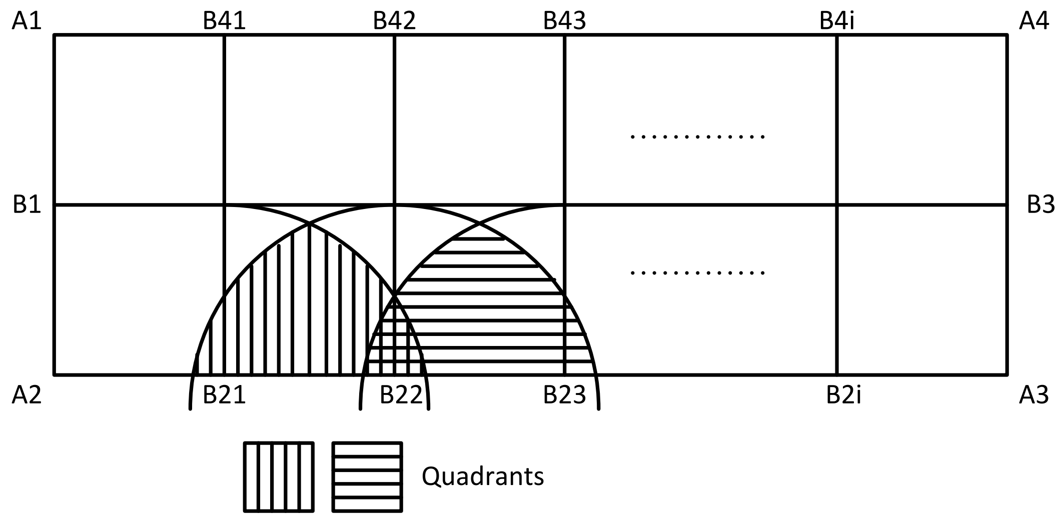


Figure 4.6: Tag location quadrants for rectangular storage area

from exactly one location can be assigned to both of the neighboring quadrants. For instance, the logical zone in Figure 4.5 (b) labeled by 1 can be assigned to both the neighboring quadrants $Q1$ and $Q2$.

The tags participate in the scanning protocol based on the reader's current quadrant of residence. For illustration, consider a case where the reader is moving in the direction from $B1$ to $A1$, albeit its radio range falling in both the quadrants $Q1$ and $Q2$; only those tags from zones 2 and 3 from quadrant $Q1$ and the respective zones labeled 1 from both neighboring quadrants $Q1$ and $Q2$ would participate in the scanning process. In other words, those tags from zone 2 and zone 3 from quadrant $Q2$ are secluded from participation even if they fall within the reader's radio range.

In this manner, the tags from contentious region along the mid-sides of the grid are effectively mitigated from collisions. Those tags from zone 1 that would have otherwise got fewer chances of reading from a plain scanning protocol (due to the high contentious area) are now allowed to participate in multiple quadrants as enabled by the framework. This multiple participation helps offset their otherwise lower read probabilities. Also the zonal-level location information requires a minimal number of storage bits (n) in the tags' memory as follows: $n = \log_2(|Q|)$, where Q is the set of

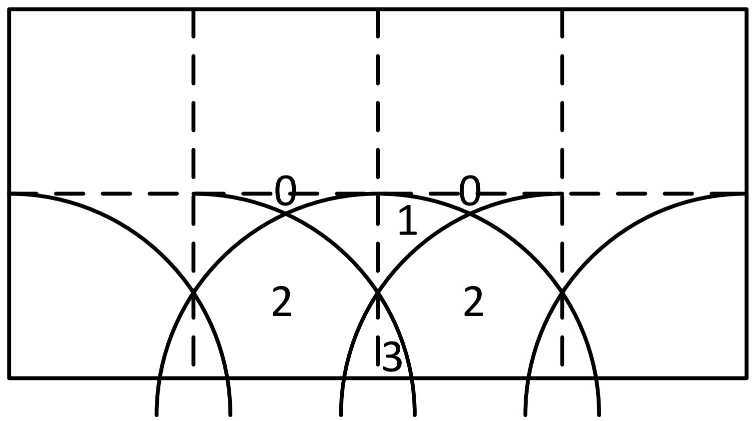


Figure 4.7: Reader coverage for rectangular storage area

quadrants in a grid. This meagre information does not stress much on the protocol's overhead.

4.6.1 DFSA Augmentation to Mobile Reader Scenario

Having discussed the zonal-spacing framework of participation, the reading protocol has to be conditioned to function in a mobile reader, static tags scenario. The DFSA protocol is a popular tag-reading protocol that fits well within the inventory standards of EPCglobalC1G2 air-interface protocol, which is subsequently ratified as ISO 18000-6C standards. The DFSA protocol is a canonical version that functions very well in a stationary reader-tags setting. Owing to the stateful nature of the protocol design, there are several research attempts to tailor this protocol to function in different settings, especially to exploit the specifics of the operating environment. For instance, in earlier work, the FSA protocol [40] was augmented to work efficiently in a conveyor belt scenario characterized by multiple static readers scanning the mobile tags.

In this section, the canonical version of the FSA reading protocol is extended to make it efficiently function in the considered warehouse scenario. The reader loops around the storage grid for the first time during which logical zones are demarcated and the tag assignments are made as per the framework described in the previous section. After the first round, the subsequent rounds will see the actual functioning of the DFSA protocol. As described in Section 4.4, the DFSA protocol functions in terms of multiple rounds of time-slots called frames. At the beginning of each frame, the reader needs protocol state information, which is required to decide the length of its next frame, namely the Framesize (FS). This FS is a crucial information that the reader constructs from its past experience of protocol execution. Moreover, the FS information is valid only for the specific tag's profile placed within the reader's range.

This protocol information must be made available for a revising reader depending on its current position around the grid. The protocol information is therefore stored based on the infrastructure support facilitated by the operating environment, either in a centralized or distributed manner, as described in Section 4.5. When a reader passesby the marked location, the protocol state information is fetched on-the-fly from the backend communication infrastructure. The infrastructure support makes the protocol information-sharing framework to be scalable in a manner that supports multiple mobile readers seamlessly. Fetching of this protocol state information would help readers to restart where they had left-off by recalculating the current FS, which minimizes collision and thereby effectively increases the reading throughput.

The state information is stored in the form of a tuple: $< a_0, a_1, a_k >$, where a_0, a_1, a_k represent the number of idle slots, successful slots, and collision slots, respectively. And the FS for the next round of inventory reading is computed as a function of tag estimation process as follows:

$$FS(L) = TagEstimation(< a_0, a_1, a_k >), \tag{4.1}$$

where $FS(L)$ denotes the FS at the specific location L.

4.6.2 Zonal Demarcation and Assigning Algorithm

In this section, the algorithm for the zonal spacing framework is described.

Phase I: Tag-location learning phase

This is a zonal construction phase, which happens when the reader makes its initial first pass revolving around the storage grid. The reader in this phase learns about the location as mentioned in Algorithm 6.

As observed in Figure 4.5, the different zonal patterns form regular symmetrical patterns along the four quadrants of the grid. The location set of these quadrants is defined by, $Q = \{Q_i : i \leq 4\}$, where

$$Q_i = \begin{cases} \{B_{i-1}, A_i, B_i\}, & \text{if } 2 \leq i \leq 4 \\ \{B_{i+3}, A_i, B_i\}, & \text{if } i = 1 \end{cases} \tag{4.2}$$

Q is the set of all quadrants of a storage area and Q_i, $1 \leq i \leq 4$ represents the i^{th} quadrant as defined by a set of strategic location points enclosing it. The reader advertises while reaching these strategic location points to all the proximity tags. In the process, each receiving tag stores the audible location information of the reader, in a set TL_i for tag i. Using this information, the tags assign themselves to different zones they belong to.

Let TQ_i represent a set of quadrants a tag i belongs to, where the mapping is done as follows:

$$\begin{aligned} TQ_j = \; & \{Q_i : |TL_j| \geq 1 \text{ and} \\ & TL_j \in Q_i \text{ for some } Q_i \in Q\} \end{aligned} \tag{4.3}$$

$$|TQ_j| = \begin{cases} 1, & \text{if } |TQ_j| \geq 1 \\ 2, & \text{if } |TQ_j| = 1 \end{cases} \tag{4.4}$$

Algorithm 6: Location Learning Algorithm

```
Reader Process:
i = 0, Round = 1;
while Round = 1 do
    LocationArray = A1, B1, A2, B2, A3, B3, A4, B4;
    if Position(Reader) == LocationArray[i] then
        Advertise(LocationArray[i]);
        i++;
    end
    if i == 7 then
        Round++;
    end
end
Tag Process:
TQ = ∅, TL = ∅;
Q1 = B4, A1, B1, Q2 = B1, A2, B2;
Q3 = B2, A3, B3, Q4 = B3, A4, B4;
TL.Add(Receive(ReaderLocation));
if TL.size() ≠ 1 then
    Q = Q1, Q2, Q3, Q4;
    while !Q.End do
        if TL ⊆ Q.First then
            TQ.DeleteList;
            TQ.Add(Q.First);
            break;
        end
        Q = Q.Next;
    end
end
else
    if TL.First= B1 then
        TQ.Add(Q1, Q2);
    end
    else if TL.First= B2 then
        TQ.Add(Q2, Q3);
    end
    else if TL.First= B3 then
        TQ.Add(Q3, Q4);
    end
    else
        TQ.Add(Q4, Q1);
    end
end
```

All tags that listen from two or more locations in a quadrant set Q_i, mark their mapping to a single quadrant as per equation 4.2. On other hand, the tags that hear from a single location mark their locations to multiple quadrants (Q_i's) based on the advertised location (A_i or B_i), as given in equation 4.3.

By the nature of the DFSA protocol, the FS is reset every time the reader moves a distance that is equal to its range. During this reset, the reader stores Protocol State Information (PSI), especially the number of collisions, idle and successful slots–corresponding to the location address of the latest reset. If the set of reader locations around the storage area is represented by the set RL, then the reader location (RL_i) and PSI will be stored in the reader's memory as follows:

$$\begin{aligned} &RL = \{A_i, B_i : 4 \geq i \geq 1\} \\ &PSI[RL_i] =< a_0, a_1, a_k >, \text{ where } 8 \geq i \geq 1. \end{aligned} \tag{4.5}$$

The tuple $< a_0, a1, ak >$ refers to the number of slots with 0 (idle slots), 1 (successful slots), k (collision slots) tags, respectively.

Phase II: Zonal-Based Tag Reading

This phase of the protocol starts after the reader completes its first lap around the storage area. In this phase, the mobile reader fetches the protocol state information which is based on its current location. This state information is used to estimate the tag population in the reader's current vicinity. Finally, the frame size is estimated from the estimated tags' population. There are several known tag estimation techniques [50], [27], [17]. In this chapter, the estimation procedure adapted from [17] is used, owing to its simplicity and lower computation time.

At the beginning of each frame, in addition to the FS advertisement, the current quadrant location (Q_i) information of the reader is also advertised. Only the tags that match with the advertised quadrant will participate in the protocol.

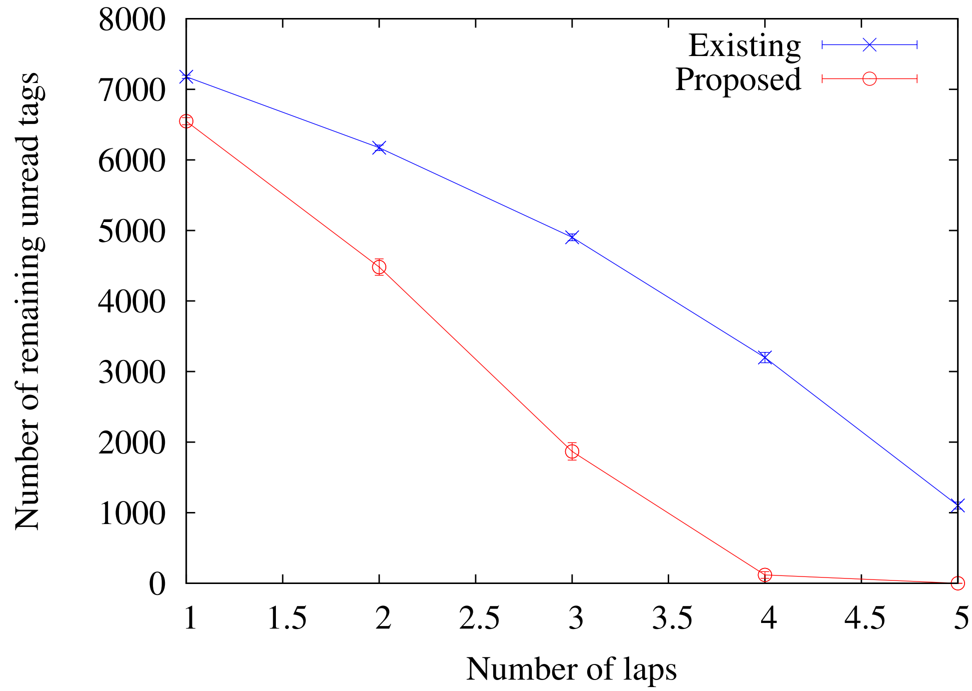
4.7 Performance Study of Single-Grid Zonal Spacing Framework

This section discusses the performance evaluation of the tag-reading protocol for the proposed framework and compare it to its rudimentary protocol counterpart. The reader is assumed to be moving with a constant velocity of $v = 2m/s$. The range of the reader is set at $R = 6m$. The storage area considered is a $10m$ by $10m$ square area. The inter-grid spacing is about $2m$. With the range of the reader and its movement parallel to the edges (at a distance of $1m$) of the storage area, it is safe to assume that reader's range is good enough to reach the middle of the storage area and hence all tags will be in the reader's range at some point of time. The storage is uniformly deployed with $8,000$ tags. The reader performs the DFSA tag-reading protocol for inventorying the tags. All results are plotted with 95% confidence.

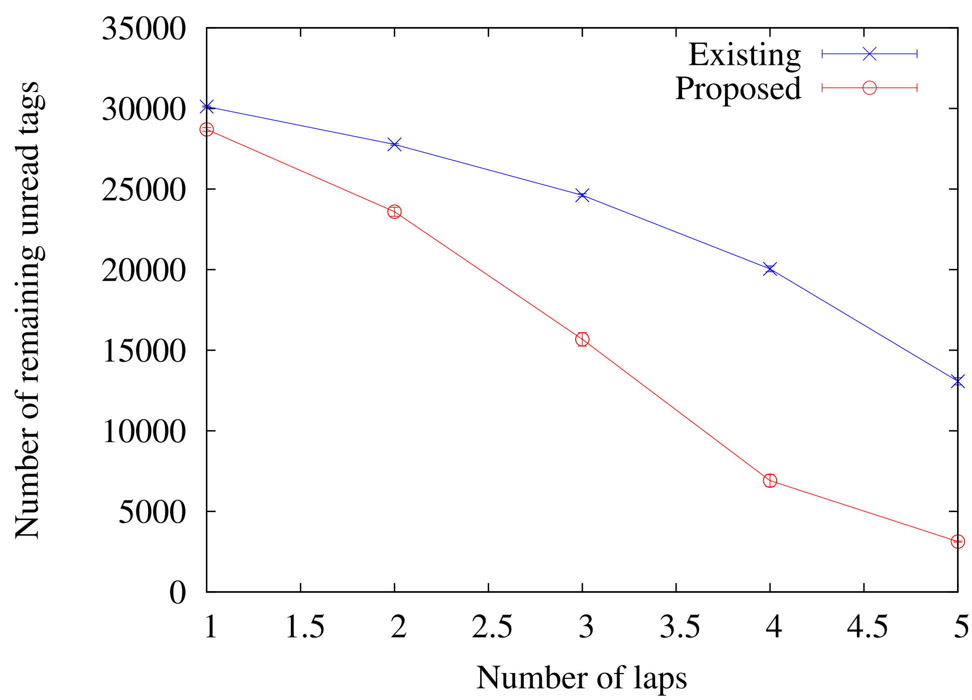
Figure 4.8(a) shows performance of the DFSA protocol augmented with the framework and compares it with performance of rudimentary protocol. The experiment is performed by observing the performance over different numbers of laps. The number of unread tags over each lap is the performance metric under consideration.

It is evident from Figure 4.8(a) that the number of tag reads increases with an increasing number of laps. Furthermore, the zonal-spacing framework with its mobile-reading feature outperforms the rudimentary DFSA. The zonal-spacing framework performance shoots up to 96% more than that of the rudimentary DFSA. At the end of the 5^{th} lap, this protocol has completed the identification process, while the rudimentary protocol is way behind with three orders of magnitude of tags left for scanning.

Figure 4.8(b) shows the overall performance results by observing overall unread tags of the storage area and their neighboring areas. The overall performance results are also at its best for the proposed framework. The framework achieves up to 76%



(a) Number of tags read in the middle storage area by the reader at the end of each lap around the storage area



(b) Number of tags read in the middle storage area and all neighboring storage areas by the reader at the end of each lap around the storage area.

Figure 4.8: Protocol performance with and without zonal spacing framework

improvement over its rudimentary counterpart. At the 5^{th} lap, the rudimentary protocol is left with three orders of magnitude of tags left behind for scanning. Moreover, as compared with the relative performances as shown in Figure 4.8(a), the tags from the neighboring grids make a substantial impact on the protocol's performance. For detailed information on the performance evaluation with regard to the single-grid looping reader setup, the readers are referred to [41].

4.8 Zonal Spacing Tag Reading with Multi-Grid Aisle-based Mobility

The previous section presented the performance improvement of the framework on a single grid with a mobile-reader looping around it. In this section, the study is extended by demonstrating the prowess of the framework on a multi-grid scenario with a mobile reader roaming around the entire warehouse by spanning multiple storage-grids. In particular, consider a casual reader making random movements along the aisles of multiple grids. This research shows analytically that the corners of the storage units (or grids) are revisited by the reader with a higher probability than the mid-sides of the grid. Therefore, these heterogeneous exposure patterns are exploited in the grid and applied to the zonal-spacing framework for performance improvements. Additionally, the scalability problem of the revisiting mobile reader is analyzed. The analysis gives insights by identifying appropriate design parameters of the system, as the problem scales. These parameters can be adequately tuned to achieve the required scanning speed in the system.

For analysis, consider a random walk grid mobility in a boundary unconstrained space by considering a torus. A close practical application would be a person with the reader making a random trip inside a large warehouse overseeing the products. For analytical convenience, the boundary effects in this model are not considered.

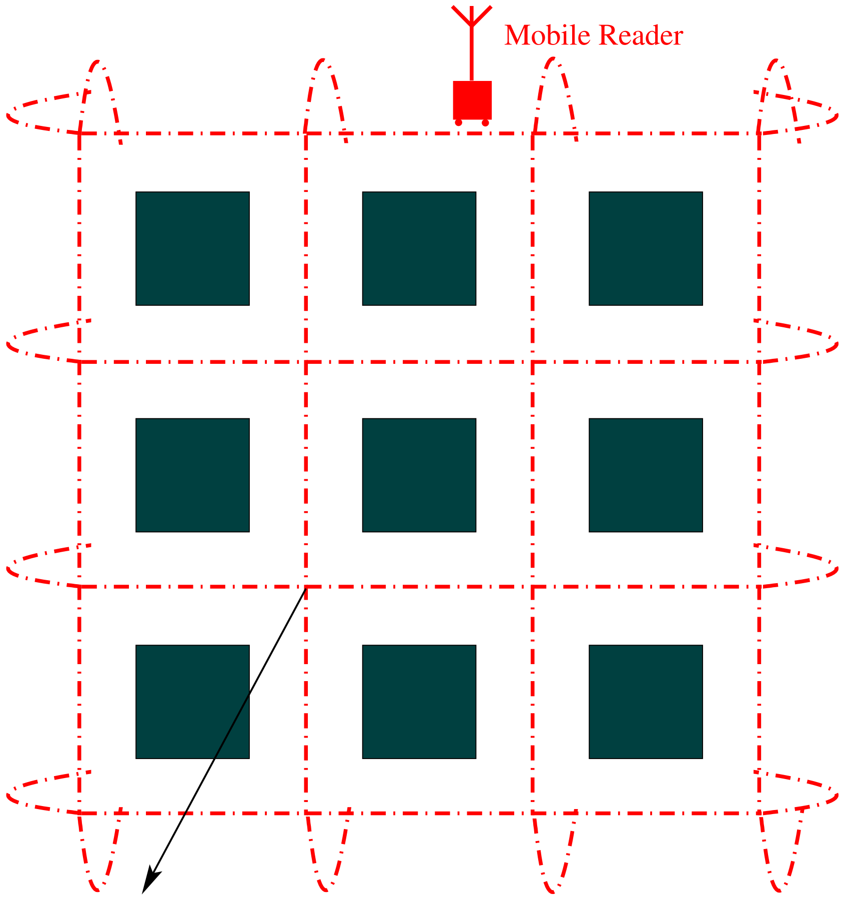


Figure 4.9: Multi-grid warehouse with aisle-based, wrap-around mobility. The dotted lines show the aisle-based path for the mobile reader

In an abstract sense, the storage warehouse aisle-based mobility is represented as an N-grid random walk on a 2D-torus, as shown in Figure 4.9. Here N represents the number of unique intersection points in the torus. An intersection point is a center junction point in the aisle surrounded by the corners of the four neighboring grids, as shown in Figure 4.9. At an intersection point, the mobile reader chooses, with equal probability, one of its four branches as its next leg movement.

From Lemma 4.8.1 given below, it is clear that the casual movement made by a reader in the considered multi-grid setting revisits the intersection points (*i.e.*, the corners of the four surrounding grids) with a higher probability than the mid-sides of the grids.

Lemma 4.8.1. *The visiting probability of a reader to the grid corners is twice than that of visiting the mid-sides of the grid.*

Proof. Without loss generality, consider a corner of the grid and one of its adjoining sides to base the proof. Let P_{A1} and P_{A2} represent the probabilities of visiting the respective corners $A1$ and $A2$ of a typical grid as shown in Figure 4.4 (c). Let P_{A1-A2} represent the probability of a reader visiting the side $A1$-$A2$ of the grid.

By virtue of symmetric random walk on a 2D torus with 4-branch intersection points, the probability of a reader choosing one of the edges (from the adjoining intersection point) is uniformly random with probability 0.25. The probability of a reader passing through the edge $A1$-$A2$ depends on either of the following two events:

- The reader starting from $A1$ chooses $A2$ as its next visiting intersection point, or

- The reader starting from $A2$ chooses $A1$ as its next visiting intersection point.

Therefore, P_{A1-A2} is given by

$$P_{A1-A2} = P_{A1}\frac{1}{4} + P_{A2}\frac{1}{4}. \tag{4.6}$$

In an unbiased symmetric random walk, the probability of visiting any intersection points are the same. In other words, $P_{A1} = P_{A2}$. Therefore,

$$P_{A1-A2} = \frac{1}{2}P_{A1}. \tag{4.7}$$

□

From Lemma 4.8.1, it is clear that the tags in the corners of the grid would get double the chances of being read compared to those along the sides.

4.8.1 Scalability Problem of Mobile Reader in the Warehouse

The essential role of a warehouse in a supply-chain industry is to serve as a buffering place that effectively contains the fluctuations of the end-to-end supply-demand requirements. The warehouse facility therefore sees a continuous churning of products that move in and out of the storage grids. The product inventory cycle involving RFID scanning must not be affected by the continuous churning of goods in a warehouse. In other words, inventorying cycle-rate should outrun the churning rate of the products. From the context of the RFID scanning problem, the inventorying cycle (say, to read a bunch of tags on a particular grid-side) depends on multiple revisiting episodes of the mobile reader. To this end, it is essential to identify the governing design parameters of the system that guide the designers to tune on the RFID scanning cycles.

In this section, the scalability problem of multiple revisiting episodes of the mobile reader is studied and shown that it is governed by the following two parameters: (*i*) the physical dimensions of the warehouse within which the reader is moving, and (*ii*) the velocity of the reader.

Furthermore, this scalability study is important in this context as the proposed framework that strives to improve the tag-reading performance over multiple revisiting episodes of the reader is also sensitive to the aforementioned physical parameters. In other words, the stored protocol state information at each place of the grid is updated with respect to the frequency of a revisiting reader to that point.

Addressing the scalability of revisiting time of a reader addresses the following question: How does the revisiting time scale with respect to different physical parameters of the system such as velocity of the reader and physical space of the warehouse?

In the considered scalability problem, the return time is computed in terms of the number of steps a mobile reader takes in order to return to any grid-point. It should be noted that the number of steps represents the normalized version of the distance traveled by the reader from which the return time can be computed.

The mobility of the reader makes a simple random walk on a 2D plane represented by the intersection points in the warehouse. This random walk can be modeled by a Discrete-Time, finite state Markov Chain (DTMC). The state space of this Markov chain represents the set of all intersection points in the warehouse. And the size of this state space is N, namely the total number of intersection points in the warehouse.

During the course of movement and upon reaching an intersection point, the reader picks one of its neighbor intersection points uniformly at random and moves towards that point. Therefore, the transition matrix $\mathbf{P}$ of the underlying Markov chain is given by

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{A} \tag{4.8}$$

where $\mathbf{D}$ is the diagonal matrix of the set of all intersection points with diagonal elements $D_{i,i} = 1/4$, and $\mathbf{A}$ is the adjacency matrix of a graph formed with intersection points as nodes and their inter-connections as edges.

The random walk of the reader is symmetric in nature as it chooses any of the four different directions uniformly in random. Therefore, the transition matrix $\mathbf{P}$ from equation 4.8 is also symmetric. An interesting property of the Markov chain (representing a random walk) with its transition matrix being symmetric is that the stationary distribution π tends to become a uniform distribution, which is given by

$$\pi = \left(\frac{1}{N}, \cdots, \frac{1}{N}\right), \tag{4.9}$$

where N is the total number of intersection points (*i.e.*, the size of the sample space).

In the considered random walk, a reader starting from any intersection point can reach any other points in a finite number of steps. This property makes the underlying Markov chain to form a single communicating class with all of its states.

This property makes the Markov chain irreducible. Moreover, revisiting any state (or an intersection point) in the chain does not always take multiples of the same number of steps, thereby making the Markov chain aperiodic. By the fundamental theorem of Markov chains, if a Markov chain is aperiodic and irreducible, then it holds for all states j within the sample space that

$$\lim_{t \to \infty} \mathbf{P}^t_{j,j} = \pi_j = \frac{1}{E[R_{j,j}]} \tag{4.10}$$

where $\mathbf{P}^t_{j,j}$ is the probability of a reader starting from state (or an intersection point) j and reaching the state j again in t steps and $R_{j,j}$ denotes the return time to state j, which is mathematically defined as follows:

$$R_{j,j} = \min\{t \in : X_t = j, X_0 = j\} \tag{4.11}$$

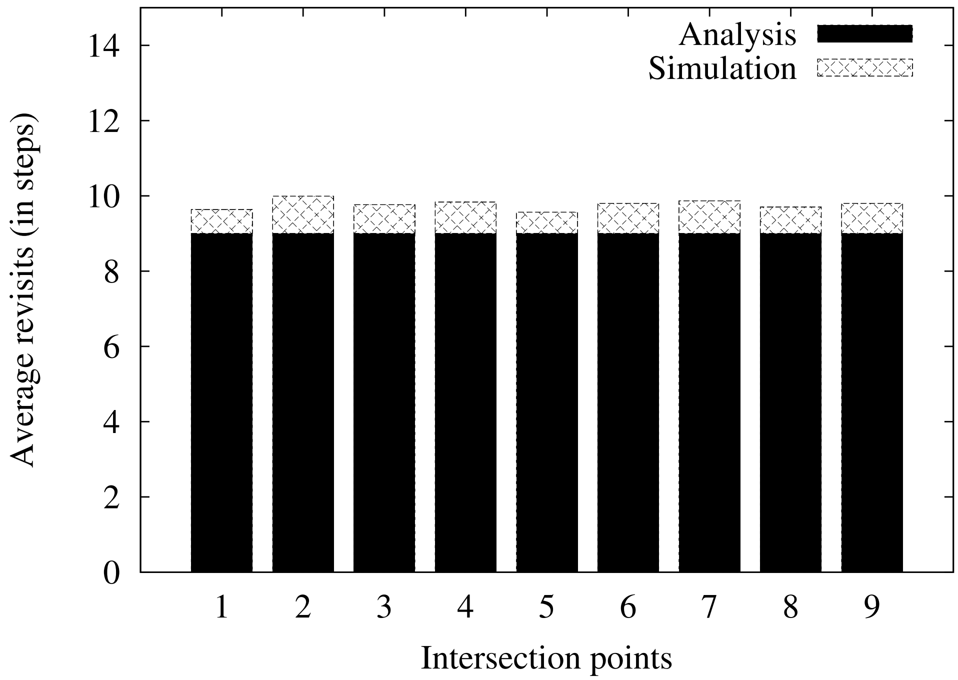
From equation 4.10, the average return time to any intersection point is given by the inverse stationary distribution of the underlying Markov chain. Therefore from equation 4.9, the average return time is given by

$$E[R] = \frac{1}{\pi_i} = N. \tag{4.12}$$

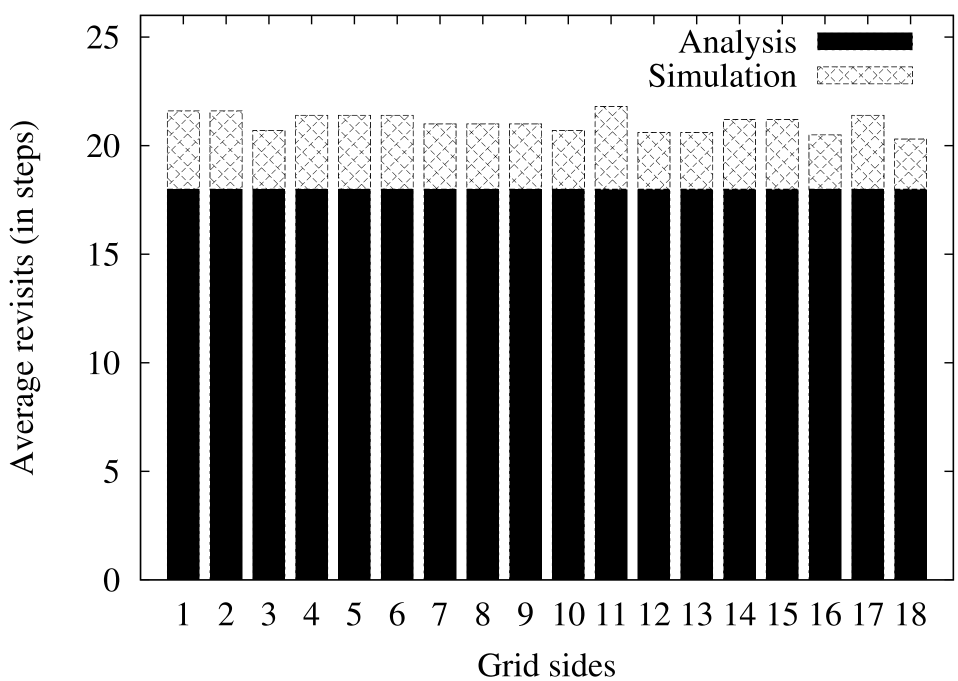
Equation 4.12 is validated with the simulation results as shown in Figure 4.10(a). The results show the average revisiting times for each of the nine intersection points in a 3×3 grid.

From equation 4.12, it is clear that the average return time is bounded by the total number of intersection points in the grid. Therefore, by converting steps to time-scale, the average time taken to revisit an intersection point $E[\overline{R}]$ is given by

$$E[\overline{R}] = \frac{E[R] \times G}{V} = \frac{G}{NV}, \tag{4.13}$$



(a) Average number of revisits for all the intersection-points of a 3×3 grid setup.



(b) Average number of revisits for all the grid sides (edges) of a 3×3 grid setup.

Figure 4.10: Number of visits to nodes and edges in a Random Walk.

where G is the distance between two adjacent intersection points and V is the velocity of the moving reader. Equation 4.13 shows that the average return time is linearl to the velocity and physical space of the grid.

Grid-side is an edge connecting the two adjacent intersection points in the considered graph. A grid-side invariably refers to the mid-sides of the two nearby grids along the referred edge. In other words, the revisiting probability of the reader to the mid-sides of the grid is studied by semantically studying the probability of a reader revisiting the associated edge in the graph.

The following lemma shows that the average revisit time to reach any grid-side also depends on the total number of intersection points in the warehouse facility.

Lemma 4.8.2. *The average number of steps needed to return to any grid-side depends on the number of intersection points in the warehouse.*

Proof. Consider two adjacent intersection points, i and j. The probability of reader moving in a direction from i to j is given by

$$\pi_i \times P_{ij} = \frac{1}{4N}. \tag{4.14}$$

Equation 4.14 provides the stationary probability of a reader to visit the intersection point i and subsequently selecting the edge $i - j$, out of the four different choices (edges) at its disposal.

The probability of visiting a grid-side connected by adjacent intersection points i and j depends on both directions of the reader's movement on that edge, *i.e.*, it depends on both the probabilities of traversing the edge from i to j and vice versa.

By the time reversibility property of the underlying Markov chain, the following equality can be obtained:

$$\pi_i \times P_{ij} = \pi_j \times P_{ji}. \tag{4.15}$$

Therefore, from equations 4.14 and 4.15, the probability of visiting a grid-side is given by

$$\pi_i \times P_{ij} + \pi_j \times P_{ji} = \frac{1}{2N}. \tag{4.16}$$

Therefore, the average number of steps a reader takes to revisit any grid-side is given by $2N$. □

Figure 4.10(b) validates the analysis of equation 4.16 with the help of simulation. Eventually in this case, the total number of grid-sides (edges) can be computed from the total number of intersection points. The considered grid setup can be represented as an r-regular graph with order N, where r represents the degree of each node and N represents the total number of nodes in the graph. Eventually edges of the graph represent the grid-sides of the warehouse. Therefore r and N in this case are 4 and 9, respectively. The size of an r-regular graph of order N can be computed as $N_r/2 = 2N$. Hence from equations 4.12 and 4.16, it is again noted that the grid-corners are more often revisited than the grid-sides, doubling the times.

Figure 4.11 shows the expected revisiting time in the time scale for different reader velocities for the respective intersection points and grid-sides. For obvious reasons, an increase in the velocity of the reader yields a decrease in the average revisit time. Moreover, this difference reduces at higher velocities.

4.9 Performance Study of Zonal Spaced Tag Reading in a Multi-Grid Scenario

This section studied the performance of the zonal-spaced tag reading protocol's performance in the multi-grid scenario. Unless otherwise specified, the simulation settings are the same as discussed in Section 4.7. The performance metric considered is the Tag reading throughput, which represents the average number of tags read in a unit

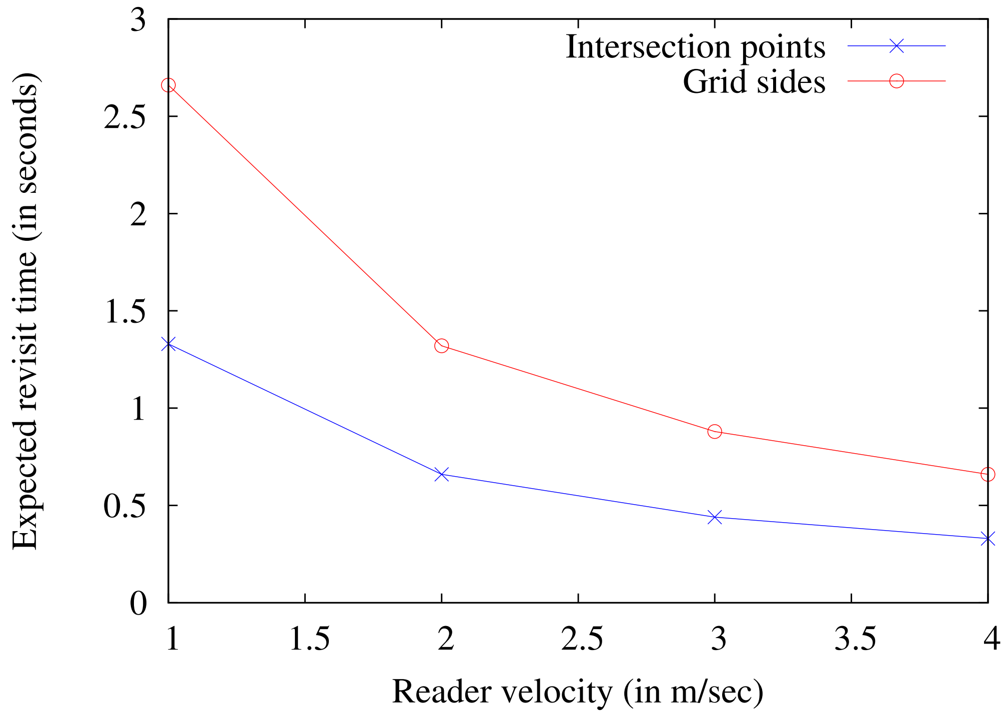
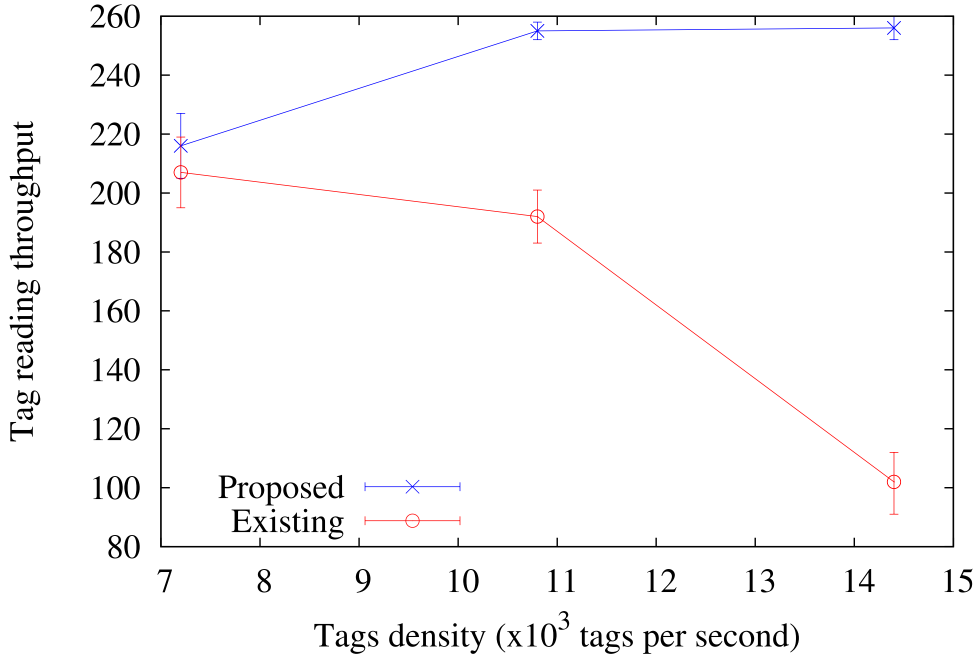


Figure 4.11: Expected revisiting time of the reader for different velocities for a 3×3 grid setup

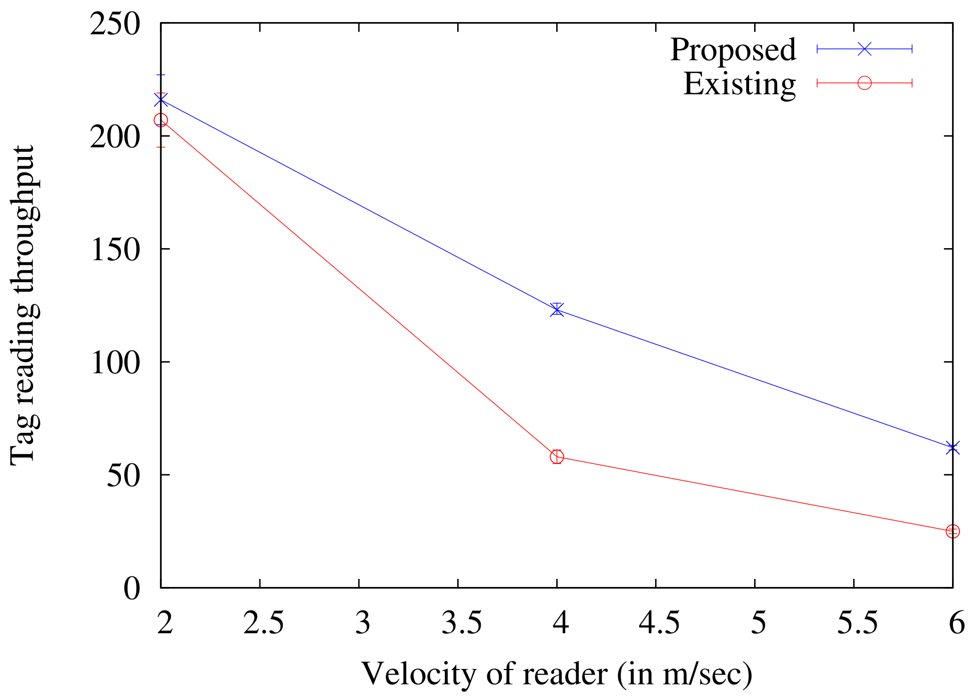
time. All performance results are plotted with 95% confidence.

Figure 4.12(a) shows the tag-reading throughput performance over different tag densities. The tag density represents the number of tags in a unit square meter. The tag-reading performance of zonal-spacing framework outperforms the traditional protocol, especially when the tag density is larger. This is intuitive as more tags in the same grid space would collide often, thereby making zonal-based participation to be fruitful. The throughput improvement due to the proposed framework is observed to increase up to 1.5 times compared to that of the traditional protocol.

Figure 4.12(b) shows the protocol performance of the existing and proposed frameworks over different reader velocities. An increase in reader's velocity decreases the tag-reading performance with the proposed framework showing better performance. As observed in Figure 4.11, the proposed framework shows better results with an increase in the reader's velocity. The velocity of the reader being an important de-



(a) Tag reading performance over various tag densities.



(b) Tag reading performance over various reader velocities.

Figure 4.12: Tag reading performance.

sign choice cannot be improved forever. Because after a certain extent the physical velocity would not be ideal for tag reading, in which case the designers can deploy multiple readers in the grid to compensate for the performance.

The protocols show an improvement with the segregation mechanism used. Limiting the participating tags is the only viable approach for existing protocols in highly dense scenarios. The tags used with zonal-spacing framework may need more storage capacity than regular passive tags. They will need to store the location IDs that they hear. They also have to run a location learning algorithm to identify the zone they belong to. This would require the tags to have more complicated circuitry than what a passive tag offers. This information is required for the framework to work. There will be a tradeoff between using tags with more memory that are more expensive and the efficiency that the framework offers for the underlying protocol.

4.10 Summary

This chapter addresses an important RFID tag-identification problem in a widely applied warehouse scenario. In this regard, a novel zonal-spacing framework to improve the throughput of an RFID warehouse system is introduced, consisting of a mobile reader moving along the aisles between the grids. The proposed framework augments the industry-standards compliant DFSA to yield better performance. This framework provides an three-orders of magnitude of performance improvement over the untailored protocol implementation.

Moreover, the analytical scalability study provides useful insights into this warehouse problem domain, in identifying the essential parameters of the system that affects the protocol's performance. The analysis with validation from simulation results enable the system designers to make strategic design decisions during the deployment phase. The analysis provided is generic enough to be extended for multiple readers setting, which helps to further the research along the similar lines. The research in this chapter is adapted from [41].

Chapter 5

Framework Exploiting Near Field Tag-to-Tag Communication

Reducing collisions caused by the concurrent transmissions of RFID chips (tags) is a crucial research topic in designing efficient air-interface protocols in passive-tag RFID systems. Different anti-collision protocols are proposed to improve the tag-reading efficiency, by ordering the tags' participation over a period of time, by stochastic or deterministic manner. All existing anti-collision protocols follow a single line of thought, wherein the onus for mitigating collisions is always on the reader, with no active role being played by the tags. This chapter proposes a fundamentally different strategy of enabling communications among the tags, especially to assist the reader in further resolving collisions.

Overthrowing the conventional wisdom of limited RFID communication allowed only between the reader and tags, recent experimental studies have shown the possibility of Tag-to-Tag (T2T) communication in industry popular UHF RFID systems. Two tags within the Near-Field (NF) of each other can electromagnetically couple with and therefore communicate as long as they are receiving energy from the reader. A critical property of NF communication systems is their small reading range, about 30mm for a 900MHZ UHF RFID system. This provides two spheres of communication and enables concurrent T2T communications to happen within the long range (around 6m) of a UHF RFID system. In this NF UHF passive tag RFID system, a tag can promiscuously listen to its NF neighboring tags' collided-transmissions and give them a second chance for identification. In this chapter, two different variants of frame-slotted ALOHA protocols will be augmented to support promiscuous learning of NF tags and study the change in reading efficiency in terms of delay and energy costs. A detailed performance evaluation study will demonstrate the efficacy of the proposed augmented protocols, both in terms of delay and energy costs.

5.1 Overview

Cost-effective volume production of RFID passive tags have gained a prominent place in the logistics and supply-chain industry. The quest for 100% inventory accuracy and several other factors such as demonstrable return of investments in certain retails like apparel and pharmaceuticals have pushed the retailers to use the RFID tagging beyond from pallet-level to item-level [5]. The item-level tagging significantly increases the number of tags spread in a given area. This imposes a significant challenge for a tag-identifying reader that scans them as the underlying air-interface protocol for tag reading is sensitive to the number of tags (within a reader's identification range).

The increasing number of tags significantly stresses on the reader's performance such as tag-reading delay as the number of collisions increases proportionately. The tag-reading delay creates a cascading effect on the energy spent by the reader in identifying the tags. This is due to the fact that the passive tags harvest energy from the reader's EM field. With widespread usage of mobile handheld readers, the energy efficiency becomes a critical protocol design factor to be considered.

Different tag-reading protocols have been proposed in the literature. Among them the FSA protocol [24] [48] is considered as the industry-recommended standard. Different protocols differ by the way they resolve collisions. Henceforth tag-reading protocols are also known as anti-collision protocols. Despite the protocols' functional differences, they have a common design philosophy: the onus for mitigating collisions is always on the reader. This unilateral reader-based design is due to the inherent restriction of the RFID systems that allow communications to happen only between a reader and the tags.

Exploiting the NF-based T2T communications, a fundamentally different anti-collision protocol strategy is presented, wherein the tags communicate with each other and assist the reader in further resolving collisions. NF-based T2T communications with appropriate change to the anti-collision protocol helps the tags to learn about their respective NF neighboring tags' that have collided with the readers. A tag, upon gaining the channel with the reader, can resolve its own identity and give its NF collided neighbor(s) a second chance for participation. In this manner, certain collided tags get a beforehand chance of getting resolved early in time, thereby significantly reducing potential collisions in the subsequent rounds of identification. This chapter proposes a protocol augmentation framework to integrate the novel NF learning functionality of tags for two different protocols, namely EPC-Class1 Gen2 (EPC-C1G2) industry standard [2] complaint Q-adaptive and plain DFSA. The Q-adaptive is the most preferred protocol widely implemented in the industry [48]. In spite of being from an FSA family of protocols the plain-DFSA is not preferred over Q-adaptive due to the time- and energy-efficient features of the latter.

The reason for considering DFSA as one of the candidate protocols for this study is given as follows: A tag listening in its NF region needs a mechanism to identify its potential neighbor that collided with the reader. This identity is necessary so as to pinpoint the required NF tag (at a latter point in time) and enable it to participate for the second time (if such an event occurs). In DFSA protocol, every tag responds to the reader with its unique identity, also known as EPC. Therefore in DFSA protocol, a tag without additional control message exchanges can promiscuously overhear their NF tags collision transmissions and learn their identities via their advertised EPCs. It should be noted that a tag's collision with the other tags in the far-field might not have a collision with the neighbors at the same tags' NF region.

In the industry popular Q-adaptive protocol, a tag before gaining the channel and getting identified by the reader would send different 16-bit random numbers (RN16) in different transmitting instances. This is an important industry requirement which is reasoned by the fact that the tags can flexibly be hard-coded with unique EPCs at the clients' place rather than from the manufacturers' place. The Q-protocol therefore enables the clients to identify and count tags before they are assigned with unique EPCs. To necessitate the tags to uniquely identify the NF neighbors, a node identification mechanism is proposed which incurs a significant delay overhead to the protocol's performance.

With a detailed performance evaluation study, the efficacy of the respective Q-adaptive and DFSA protocols augmented with NF learning frameworks is demonstrated. The Q-adaptive protocol, despite its augmentation delay-overhead, outperforms both in terms of delay and energy costs, especially for large number of tags.

The chapter is organized as follows: Section 5.2 discusses on the research works that focus on improving the performances of different anti-collision protocols. The necessary background on the functioning of anti-collision protocols is briefed in Section 5.3. The different antenna transmitting regions and necessary conditions for successful NF communications are discussed in Section 5.4. The crux part of the work proposing NF-based learning of tags and their augmentation to the anti-collision protocols is discussed in Section 5.5. The veracity of the proposed framework is discussed via extensive simulation results in Section 5.6. Finally, the research in this chapter is concluded in Section 5.7.

5.2 Related Work

Anti-collision protocols in passive tag RFID systems still remains an active research area [24] [48] [29] [28] [52]. Among several anti-collision protocols proposed, the EPC

Gen-2 standard complaint Q-adaptive FSA variant is considered as an important benchmark anti-collision protocol owing to its wide-spread practical applications. The authors of [24], augment the EPC Gen2 Q-adaptive protocol with a physical layer collision recovery mechanism. In their work, when a collision is detected at the reader due to two or three tags, a signal separation method is adopted at the reader to decode and identify each of the collided tags from the transmission. The reader then sequentially queries each of the three candidate collided tags and resolves their identities.

While research attempts have been made to improve the quality of Q-adaptive protocol [24], attempts have also been made to replace Q-adaptive protocols with radically different MAC-level protocols such as CDMA-based protocols [48]. However, the authors of [48] have analytically shown that the Q-adaptive protocol outperforms CDMA protocols in terms of the number of bits transmitted. This result is very critical from the context of mobile RFID readers that run on batteries.

5.3 Background

This section discusses the anti-collision protocol considered in this chapter and briefly explains the NF and Far-Field (FF) zones surrounding the tag's antenna.

5.3.1 Functioning of FSA Protocol

The Q-adaptive FSA protocol works in slotted times, as shown in Figure 5.1, with tag identification attempt being made in each slot. At the beginning of the slot, the reader advertises a value $Q \in [0, 15]$ (with the very first value of Q being 4). Each tag within the reader's identification region chooses a random value uniformly in the range of $[0, 2^Q - 1]$. The tags then start a decremental slot counter initialized by this random value. Those tags that have (randomly) chosen the value 0 would respond

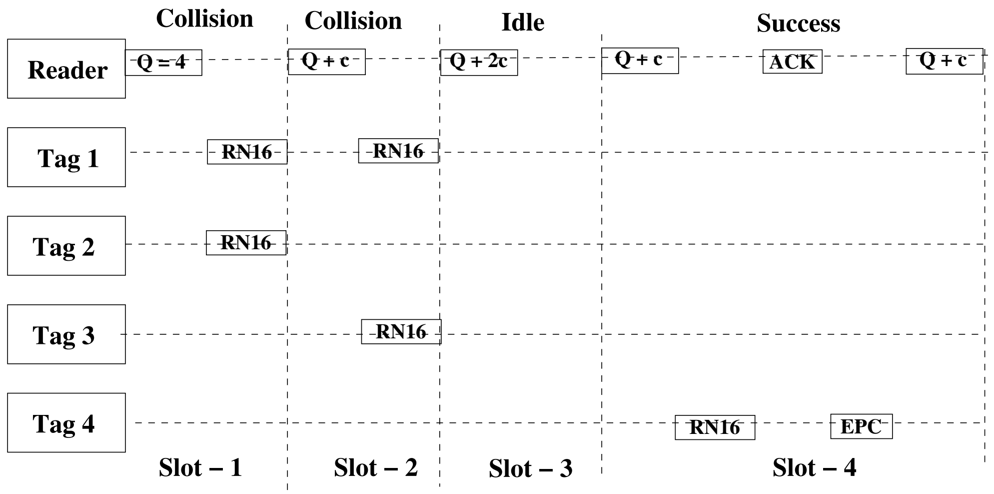


Figure 5.1: A schematic of Q-adaptive FSA protocol

to the reader with a new 16-bit random number (RN16). Subsequently, either of the following three events could happen from a reader's perspective:

- Idle slot: No tags have chosen the value 0. The reader reduces Q value by a constant $c \in (0, 0.5)$ and announces the new, reduced value of Q in the beginning of the next slot. The tags would reinitialize their decremental slot-counters with a new random value using the new Q.

- Successful slot: Out of all tags, exactly one tag has chosen 0 and transmits RN16 to the reader. The reader sends acknowledgment (ACK) to the specific tag using RN16 sent by the tag as the handle. Upon receipt of the ACK message, the corresponding tag responds with its unique EPC-code to the reader after which the tag is considered as muted and refrains itself from participating in the subsequent rounds of identification. The reader starts the next slot with the same Q value. Remaining tags decrement their slot counter by one and those tags that had previously chosen a random value one would now turn out to be zero in the next slot, and therefore participate by responding to the reader.

- Collision slot: Two or more tags respond to the reader in the slot, leading to collision. The reader increments Q value by c and proceeds to the next slot.

In general, the plain-DFSA algorithm functions in a fashion similar to the Q-adaptive algorithm, but with certain differences listed as follows: (*i*) In each slot, the successful tag responds with their EPCs (not with RN16 as in Q-adaptive protocol), (*ii*) The Q-value equivalent of DFSA, namely the $\log_2(framesize)$, is not limited to a specific range such as $[0, 15]$, and (*iii*) A new framesize in the DFSA protocol is chosen as a function of the number of collisions and idle slots occurred in the previous round of identification (which is typically a few number of slots). In contrast, the Q-value is immediately updated (*i.e.*, added or subtracted by a constant), which is based on the collisions or idle events occurred in the previous slot. In other words, the Q-protocol is more reactive than DFSA, as its Q-value is checked for a necessary update in every slot.

5.4 NF T2T Communication

Consider an NF UHF RFID system with a single reader scanning a bunch of passive tags deployed randomly within its identification zone. In a UHF system, the tags communicate with the reader in a zone known as FF zone. For a 900MHz system this range extends to about 6m. In a UHF system, recent studies have demonstrated the possibility of T2T communication with in NF region. The range of the NF region (R_{NF}) is computed as $R_{NF} = 0.62 \times \sqrt{l^3/\lambda}$, where l is the geometric dimension of the antenna and λ is the wavelength of the operating system.

Between the NF and FF regions of a tag's antenna lies the Fresnel Zone (FZ). Any communication between a pair of tags within the NF region would happen successfully when no other transmission happens in the FZ and FF zones of the receiving tag's antenna. The FZ zone begins after the NF region and extends up to a range (R_{FZ}) which is given as follows [36]: $R_{FZ} = 2l^2/\lambda$. The tags FF region starts after the FZ region.

5.5 Protocol Augmentation with NF T2T Based Learning

This section presents the protocol augmentation framework to utilize the NF-based T2T communications in an appropriate way to assist the reader in resolving collisions. To this end, the reader should facilitate an NF T2T communication phase at the end of each collision slot in order to enable the selective NF tags to pick up information from the neighboring collided tags.

By the nature of the Q-adaptive protocol, uniquely identifying the collided tags in a future time is not possible as the tags send random values (RN16) as handles in a collision slot. Therefore, a mechanism is proposed to identify the NF collided neighbors in a Q-adaptive protocol. Since the tag-reading protocol is arbitrated by a centralized reader, the tags are perfectly time-synchronized at the slot-level. Therefore, the slot number is used as a handle to identify the NF tags. Without loss of generality, each tag is assumed to have sufficient memory to store the details of its NF neighbors. Each tag uses this memory to maintain a neighbor-table storing one entry of slot-number for each of its collided NF neighbor tags.

Each tag remembers the slot-number of its previous collision slot. In the future, a tag upon facing a fresh collision with the reader, would advertise its previous collided slot number to the NF tags. On the other hand, a tag receiving a slot number from its NF neighbor, would search its neighbor-table for an entry with this slot number. If such a slot number exists, the corresponding entry would be updated with the current slot number. This prevents tags from storing redundant information from the same NF tag collided for more than once with the reader. If no such entry is found, a new entry is made in the neighbor-table to represent a new NF neighbor.

This mechanism therefore requires the reader to allot $\log_2(q)$ extra bits at the end

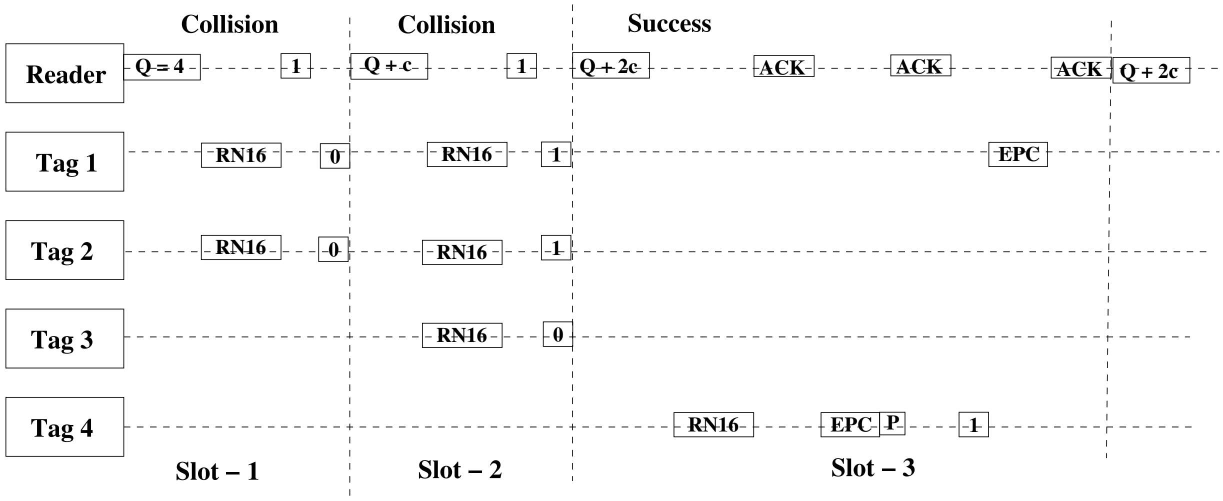


Figure 5.2: A schematic of Q-adaptive FSA augmented with promiscuous NF learning

of each collided slot, where q is the current slot number of the protocol. An illustration of promiscuous Q-adaptive is shown in Figure 5.2. Tags 1 and 4 are assumed to be within their NF regions, Tags 2 and 3 are outside the NF and FZ regions of both Tags 1 and 4. As shown in Figure 5.2, Tags 1 and 2 involved in the collision with the reader in slot 1. The reader subsequently allocates one bit for collided tags to transmit their respective previous slot numbers. In this case, the previous slot number is 0. The Tag 4 in the NF region of Tag 1 listens to this slot number, and creates a new entry in its neighbor table with the current slot number, say 1. Subsequently, Tags 1 and 2 are involved in the collision for the second time. This time Tag 4 receives slot number 1 from Tag 1. Tag 4 looks up for an entry in its neighbor table with value 1 if there exists one and it replaces the entry 1 by the current slot number, say 2. Finally, when Tag 4 gains a successful slot with the reader along with its EPC message, it sends a special promiscuous-bit P set to 1, indicating the reader that it has an NF collided neighbor waiting for identification. After receiving ACK from the reader, Tag 4 advertises its neighbor table entry, *i.e.*, the slot number 1. Hearing this, Tag 1 that has collided in this slot number would respond to the reader with its EPC. In this manner, Tag 1 is also identified by the reader.

The following lemma shows that the slot-number-based storage is sufficient to uniquely identify the tags.

Lemma 1. *Maintaining slot-number-based storage is necessary and sufficient to uniquely identify the neighboring tags.*

Proof. In order for an NF T2T communication to happen successfully, out of all tags within the NF and FZ regions of a receiving tag, one and only tag is allowed to transmit. In all other cases, the transmission becomes unsuccessful. Therefore, given a successful NF heard by a tag in a particular slot, there is one and only tag associated with that transmission. Therefore, there exists a one-to-one mapping between the slot numbers and the NF tags. □

The slot-number-based storage mechanism provides a way to bound the tag's memory only to a certain extent. The following lemma proves that this mechanism suffers from a memory leakage problem.

Lemma 2. *Indexing and updating neighbor-table based on the previous slot number advertisements is necessary but not a sufficient condition to keep the table memory bounded.*

Proof. Consider three tags, namely A, B, and C located within the NF regions of each other. Tag C has another Tag D within its FZ region. Also, Tag D is in the FF regions of all other tags, say A and B. Consider a slot X in which Tag A has been involved in the collision. As per the proposed mechanism, B and C will have their respective neighbor-table updated with a new entry say X. In a future slot, say Y, let tags A and D get involved in the collision. Now by virtue of the NF communications, B would successfully update its neighbor-table entry by replacing X by the slot number Y. Tag C would not have received a successful transmission from A due to the simultaneous transmissions of tags A and D. The entry X in the neighbor-table of Tag C, therefore, remains not discarded, causing memory leakage condition in Tag C. □

As per Lemma 2, there needs to be a trade-off between the tag's memory and the performance of the protocol, which is a design factor to be considered. In order to study the maximum benefit an NF T2T communication could offer to the performance of Q-adaptive protocol, the tags are assumed to have sufficient memory to hold all the entries.

The NF T2T augmentation for the plain-DFSA implementation is trivial as the tags naturally send their unique EPCs in a collision slot. Any NF tag can therefore promiscuously overhear the neighboring tags' EPCs and use this information to identify them at a suitable point in time and gives them a second chance.

5.6 Performance Evaluation

This section provides an evaluation and comparison of the performances of the Q-adaptive and plain-DFSA protocols with their respective NF T2T augmentation framework counterparts. A custom-developed RFID simulator written in C++ is used to demonstrate this. Both delay and energy costs are taken into consideration for performance evaluation study. Without loss of generality, energy costs do not consider the control messages that are common to both the native and augmented protocol versions. The overall performance results are shown with 95% confidence.

Figure 5.3 shows the delay cost studied in terms of the number of time slots, for native and augmented versions of the respective Q-adaptive and DFSA protocols. Intuitively, the performances of protocols increase with an increase in the number of tags. The NF T2T Q-adaptive protocol shows a significant improvement over its native counterpart. The augmented DFSA performs up to 80% better than its native counterpart. On the other hand, the augmented Q-adaptive shows a marginal improvement over its native counterpart with performance improvement of up to 72%. This is due to the additional variable-sized message overhead (*i.e.*, slot number

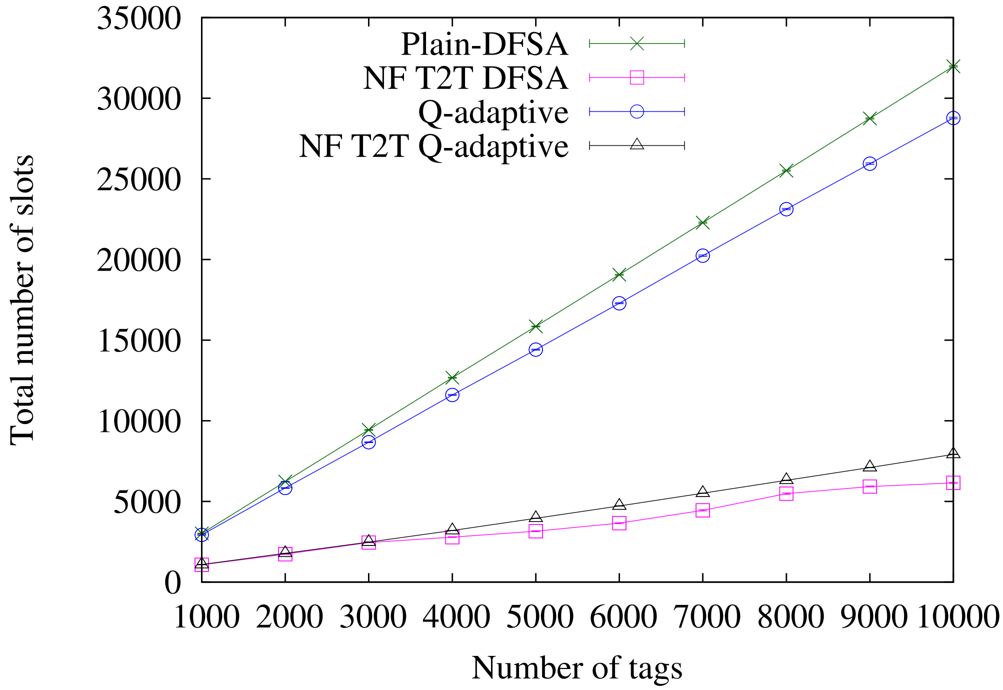


Figure 5.3: Delay performance in terms of number of bits studied over different number of tags.

exchange) incurred in each collision and successful slots of the augmented Q-adaptive protocol. Figure 5.4 shows the performances of different protocols frame sizes vs the number of slots required to identify all the tags. It is evident that the respective NF T2T communication-based variants are indeed beneficial, as the collisions are recovered at the earliest.

The energy efficiency is defined as the ratio of bit-rate transmission of the reader to the reader's transmitted power. For EPC Gen2 systems, the bit-rate is typically around 40 Kbps. By the FCC limits a UHF reader's transmission power can be allowed up to 1 watt. Therefore, energy efficiency of a typical UHF RFID system is around 40 $kbps/1\ watt$ = 40 Kbits per Joule.

From energy consumption perspective, the path loss is the dominant factor in the typical long-range UHF RFID systems, especially the transmitted power of reader in a passive tag RFID system suffers from two-times path-loss as it takes a full-round-

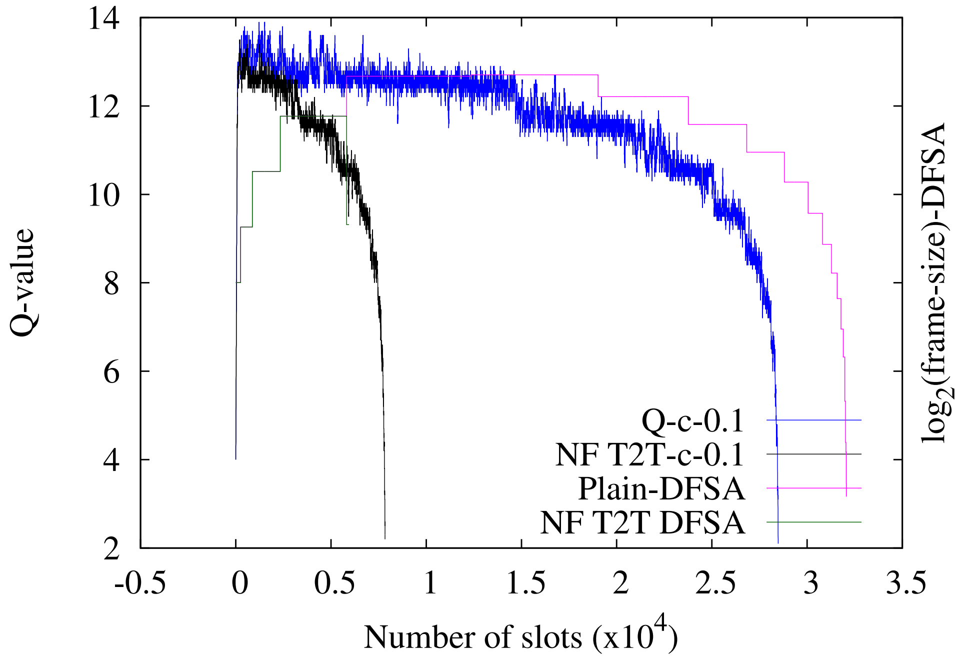


Figure 5.4: Frame size vs number of slots for various protocols. The *X-axis* is appropriately shown to highlight the starting frame-size values

trip to reach the reader [3]. As the NF T2T phase message exchanges/transmissions are confined only within the tags' NF regions, the reader can scale down its transmission power to half as compared to its power in other phases. Therefore, the energy efficiency during the NF T2T phases can be given as, 40 $kbps/0.5\ watt = 80$ Kbits per Joule. Figure 5.5 shows the performance of different anti-collision protocols, with total number of bits expressed in terms of energy consumption. NF-T2T versions of both DFSA and Q-adaptive outperform their native versions. Moreover, the NF-T2T-based Q-adaptive is better than its augmented DFSA counterpart. NF-T2T Q-adaptive is better than native Q-adaptive by up to 33% at the minimum number of tags. The NF-T2T DFSA is better than its native counterpart by up to 47% when the number of tags is large.

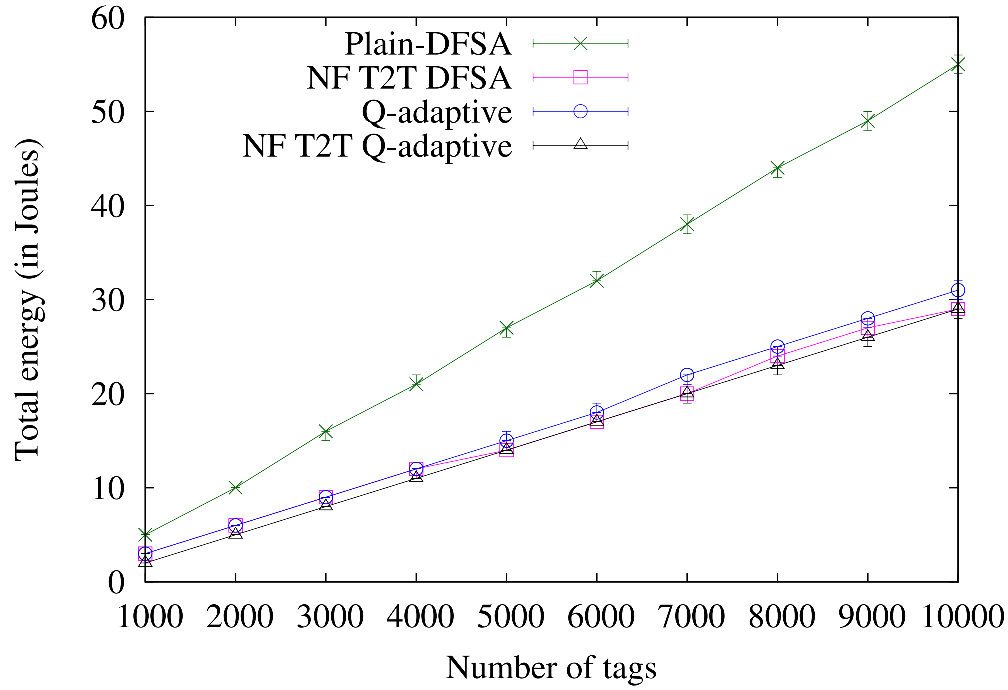


Figure 5.5: Energy expenditure performance in terms of number of energy normalized bits, studied over different number of tags.

5.7 Summary

This is the first research in the literature that makes use of NF T2T communications to assist the reader in improving the tag-reading performance. This research proposes the NF T2T communication-based augmentation frameworks for the respective Q-adaptive and the plain DFSA protocols. The Q-adaptive protocol, even with additional communication overheads, still remains as the best-in-class protocol in terms of energy expenditure. The NF T2T variant of the plain-DFSA protocol outperforms in terms of delay, showing an improvement of up to 80%. The NF T2T variant of Q adaptive performs close to the respective DFSA counterpart with an improvement of up to 72%. Hopefully, this work would provide the research community a novel direction of using NF communications for further improving the efficiency of the anti-collision protocols. The research in this chapter is adapted from [43].

Chapter 6

Analysis of ALOHA-based Protocols

The RFID system is a highly sought-after technology prominently used in the automation, logistics and supply-chain industries. EPC Gen-2 is the current standard RFID air-interface protocol in the industry. Underlying this standard is the Q-adaptive MAC protocol, which is used to identify and read the tags, by stochastically arranging them over time. The Q-adaptive protocol's performance is sensitive to the number of tags placed under the vicinity of the reader. With the current era of item-level tagging witnessing an increased number of tags in a given area, mathematically understanding the performance of the protocol in these large-scale RFID systems becomes essential.

Recent works have established the importance of the Q-adaptive protocol by analyzing its performance with the help of Markov-chain models. However, these analyses suffer from the common state-space explosion problem as the state-space increases quadratically with the increasing number of participating tags. Hence it is essential to come up with a scalable analysis, whose computation model is insensitive to the number of tags. This chapter proposes a scalable bound-based solution to study the delay performance of Q-adaptive protocol. The delay bound is derived by exploiting the fact that, in the large number of tags regime, the Q-adaptive protocol rapidly reaches to the theoretically maximum performance and stays reasonably close in that optimal region for most of the time. Extensive simulation results validate this bound-based solution with a reasonable accuracy.

6.1 Overview

RFID is a promising technology that is popular in the retail automation supply-chains such as apparel and grocery industries. Passive tag RFID systems are prominent in the fields of inventory tracking and identification and unlike the bar-code systems they are not restricted by the line-of-sight constraints.

Among different operating frequencies, the UHF RFID systems are quite popular as they cover relatively long ranges. While the UHF RFID systems are commonly used at the pallet- or case-level, tag manufacturers such as Impinj have recently focused on producing tags conducive for item-level tagging [5]. The item-level tagging with the philosophy of one tag per item essentially improves the accuracy and quality of the supply-chain, benefiting both the retailers and the customers. To this end, the EPCglobal Inc, a subsidiary of the global not-for-profit standards organization GS1, has come up with the EPCglobal standards for item-level tagging [4]. Item-level tagging is seen as a promising solution to the industries handling perishable goods such as the grocery industry [6].

Item-level tagging significantly increases the number of tags within the RFID reader's reading range. This poses a significant performance overhead to the air-interface protocol, that helps in identifying these tags. The tag-identification protocol's performance is sensitive to the increasing number of tags. Especially, the delay incurred by the protocol increases as the number of tags increases. Moreover, with the advent of hand-held portable RFID readers, the energy efficiency also becomes a critical performance metric to be studied. In passive tag RFID systems, there are direct relationships between energy efficiency, delay, and the number of tags.

It is therefore necessary to mathematically analyze the tag-identification protocol's performance, especially the delay. The current industry standard RFID air-interface MAC protocol, namely EPC Gen-2 Q-adaptive protocol, is based on the DFSA mechanism which uses probabilistic ordering of tags for identification. Of late, efforts have been made to analyze the EPC Gen2 protocol's performance using Markov chain models [48] [47] [51]. It is known that the Markov models suffer from the common state-space explosion problem. For instance, the absorbing Markov chain model [48] for N tags requires a state space of about $((16/c)+N)^2$, where c is the protocol-dependent parameter constant. Hence the Markov model sees a quadratic increase in its state space, for a proportionate increase in the number of tags. For instance, for a nominal 1000 tags system, the associated Markov-chain model requires around one million states. This makes the model intractable for a relatively large number of tags.

Driven by the item-level tagging, a scalable mathematical solution is necessary to study the performance of the EPC Gen2 MAC protocol. To this end, the research in the current chapter proposes and studies the following:

- Proposes a scalable bound-based solution to study the performance of the EPC Gen2 protocol in large-scale RFID system.
- Highlights the fact that the Q-adaptive in its native form is close to its maximum theoretical performance, even for the scenario in which the number of tags is very large.
- The Q-adaptive protocol is aggressive in reaching to a state of maximum theoretical performance and continues to stay stable in that state with a small margin of error.
- Extensive simulation results validate the proposed model and demonstrate the tight bound with reasonable accuracy.

This chapter is organized as follows: novelty and necessity are motivated for a scalable analysis in Section 6.2. A brief background on the functioning of the

Q-adaptive protocol is given in Section 6.3. The bound-based scalable solution is presented in Section 6.4. The proposed model is validated with the help of extensive simulation in Section 6.5. Finally, a conclusion to this work is provided in Section 6.6 by highlighting the importance of the proposed model.

6.2 Motivation and State-of-the-Art

Among different tag-identification protocol variants in the family of DFSA protocols, the Q-adaptive protocol gains a special attention as it is officially industry standardized as EPC Gen2 protocol [2]. Recently, attempts have been made to examine the efficiency of the Q-adaptive with a different family of protocols, such as the CSMA-type protocols, and the work in [48] has analytically showed that the Q-adaptive protocol stands out to be the energy-efficient protocol. The performance modeling of Q-adaptive protocol has been extensively studied using Markov-chain models [48] [51]. However, these models suffer from state-space explosion problem, wherein the modeling computation complexity quadratically increases with the increasing number of tags. On the other hand, the works such as [45], [44], [56] have focused on improving the EPC Gen2 protocol's performance by optimally estimating the tags' population. However, the tags' population estimation involves combinatorial computations as a function of the number of tags.

While the existing works such as [48], [47], [51], [45], [44], [33] evaluate the Q-protocol using simulations involving a small number of tags, say about 1000 tags. Orthogonal to them, this work focuses on analytical understanding of the behavior of the Q-adaptive protocol for much large number of tags in the order of 10^4 and even beyond. However, the Q-adaptive protocol effectively handles the maximum tags count of up to 2^{16}, as the protocol parameter in the current standard is restricted to a maximum length of 16 bits. Therefore, the proposed validation experiments are restricted to consider up to $60,000$ tags. Without loss of generality, this bound-based solution would work for a larger number of tags, even beyond 2^{16}, because it can be expected for the standard to evolve in future and increase the protocol parameter to include more than 16 bits in order to effectively identify the large population of tags. The authors in [56] have attempted to optimize EPC Gen2 Q protocol by a DFSA-type tags' population estimation technique as proposed in [49]. Using simulations, they have evaluated for up to $30,000$ tags, where the performance metric considered is the time (in micro-seconds) required to complete the tag identification process. The maximum throughput improvement using the proposed population estimation technique, even with a fine granular performance metric (*i.e.*, in microseconds rather than in the number of slots) is found to be around 4% for $30,000$ tags. This chapter provides mathematical reasons for the cause of this minimal improvement, for the large number of tags regime.

With the emerging trend of item-level tagging, the number of tags can easily reach a quantity that eventually makes the existing analyses intractable [48] [51]. Recent research works actively focus on minimizing the tag's antenna dimension, a dominant factor to the tags' form factor. Of late, the authors in [25] have proposed a novel planar spiral antenna design for passive tags in a UHF RFID system and fabricated tags with a maximum dimension of 3cm. They observed experimentally a maximum reading range of 6m in the presence of a 1W reader antenna characterized with circular polarization. Assuming the 3D radiation pattern of the reader to be roughly shaped as a hemisphere with the maximum radius of 6m, this volume can accommodate a large number of tags. However, other parameters such as physical product sizes and the tags' EM propagation factors restrict the maximum reading depth of the stacked tags. Recently, the authors in [53] proposed to improve the reading efficiency of stacked-tags with the help of a novel design of the multibeam UHF RFID reader with 2×2 antenna array. In a real experiment with the fabricated 2×2 array antenna, they used a stacked arrangement of $5 \times 4 \times 8$ boxes with six passive RFID tags in each box. Moreover assuming six tags arranged in a $2 \times 2 \times 2$ fashion inside each box, as described in the authors' base paper, [46]; roughly eight to ten tags stacked one behind the other are read. A 3cm $\times$ 3cm dimensioned tag [25] pinned to a boxed-in product not more than 3cm thick and 3cm breadth, and height not more than 600/10cm, can be arranged in a (non-overlapping) tiled fashion. This can be done inside the largest-square that fits inside the reader's cross-sectional circular area of radius 600cm, and can accommodate as many as $(2 \times 600)^2/(2 \times 3 \times 2) = 80,000$ tags. With a depth of ten stacked tags, the total number of tags within a reader would be about $80,000 \times 10 = 8 \times 10^5$ tags. However, as per the EPC Gen2 protocol limitations, the Q value roughly representing $\log_2(n)$ tags (where n representing number of tags) is limited to a maximum of 16 bits, therefore the protocol can effectively accommodate to a maximum of $2^{16} = 65,536$ tags for scanning. Therefore, in the present situation, considering both the factors of potential maximum possible tags count and current protocol parameter limitation, a maximum of about 2^{16} tags can be effectively identified by the EPC Gen2 protocol. At this scale, Markov models would fail. For instance, according to the work in [48], the state space of the Markov model would consists of about $65,536^2 = 4,000$ million states, deeming it to be computationally intractable.

6.3 Background

Figure 6.1 shows the schematic operation of EPC Gen2 Q-adaptive protocol. This protocol belongs to the family of DFSA protocols that orders tags' response in a probabilistic manner to reduce collisions. In (passive tag) RFID systems, the tag-identification protocol is centrally arbitrated by the reader in a reader-talks-first fashion. The time is slotted by the reader and all tags are synchronized to these time slots. At the beginning of each slot, the reader advertises an integer Q valued

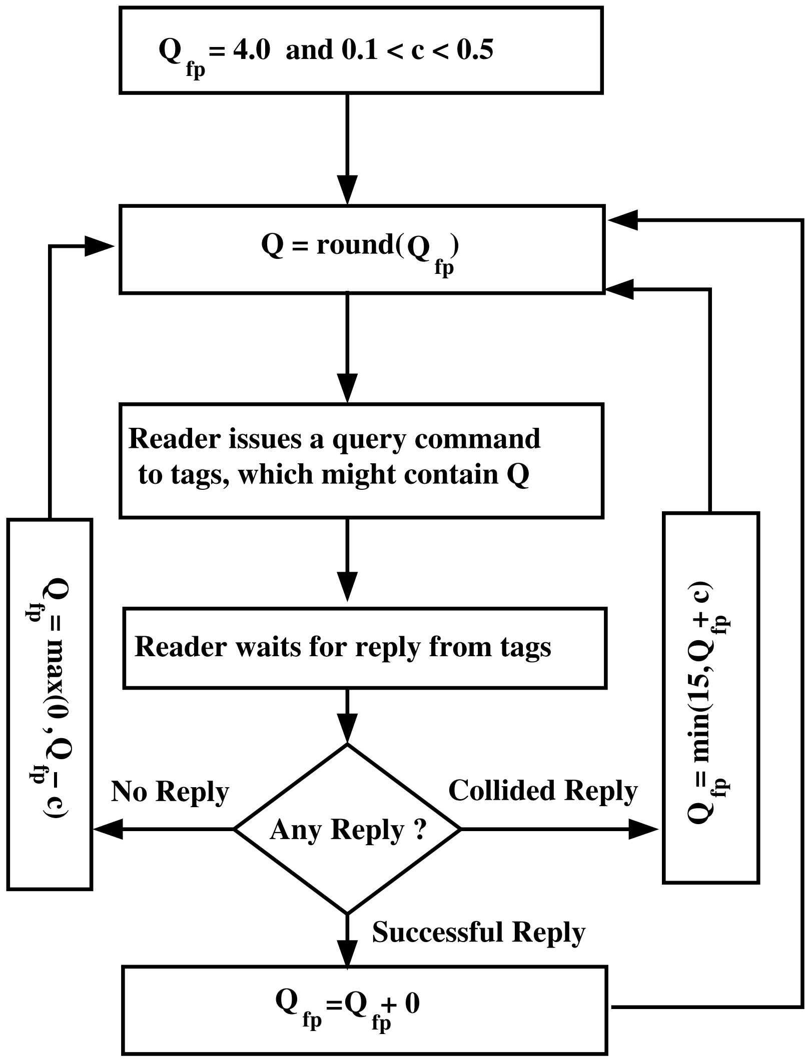


Figure 6.1: EPC gen2 Q-adaptive protocol

in the range of [0, 15]. Followed by this, the tags choose a value randomly in the range of $[0, 2^Q - 1]$. Those tags that select 0 would choose to respond to the reader during that slot by sending a 16-bit random number (RN16). All other tags that have chosen a value $r \in [1, 2^Q - 1]$ would initiate a decremental-slot counter with this value r. In every slot, the reader instructs all tags to either reinitialize their counters to a different value or clock to decrement their present counter-value. The reader makes this decision based on the tags' response in the previous slot. Based on one or more tags' response, either of the following events is bound to happen in a slot:

- Idle slot: No tags have chosen the value 0 leading to no transmissions in that slot, therefore the reader reduces the floating point representation of Q, namely Q_{fp}, by $c \in (0, 0.5)$ where c is a protocol design parameter. If the integer-rounded version of updated Q_{fp} is the same as Q, then all tags are instructed to decrement their respective slot counters by one. On the other hand, if the integer-rounded version of updated Q_{fp} represents a new Q, the tags are then instructed to reinitialize their counters with a different r generated randomly by using this new Q-value.

- Successful slot: Out of all tags, exactly one tag has chosen 0 and subsequently transmits RN16 to the reader. The reader sends an acknowledgment (ACK) to the tag by using the RN16 number sent by the tag. Upon the receipt of the ACK message, the corresponding tag responds with its EPC to the reader[1]. After identification, the associated tag is considered as muted and is refrained from participating in the subsequent slots of the identification process. Every tag decrements their slot counter by one and those tags that had previously chosen a random value 1 would now turn out to be 0 in the next slot and respond to the reader in a similar manner.

- Collision slot: Two or more tags respond to the reader in the same slot, leading to collision. The reader increments the Q_{fp} value by c and starts the next slot by instructing the tags to either restart or reduce their slot counters depending on the new state of the rounded Q value.

Some distinguishing features of EPC standard Q-adaptive protocol include the following: Q-adaptive uses a tunable-design parameter c by which Q-value is incremented, decremented or unchanged in the events of the collision, idle or successful slots. This decision changing the Q-value is made after every slot. The parameter c can be in

[1]EPC is the unique identification code for the tags

the range between 0.1 and 0.5 with no specific usage of value recommended by the standard. On the other hand, generic DFSA protocols make decisions only at the end of each frames composed of multiple time-slots. For instance, the current frame size is computed explicitly based on the number collisions, successes and idle slots, occurred in the previous frame [17] [49]. In this manner, the Q-adaptive protocol is more reactive than the other DFSA variants.

6.4 EPC Gen2 Q-Adaptive - Scalable Approximated Performance Analysis

The probability of a successful slot p_s is a binomial process, which can be computed as follows:

$$p_s = n \times \frac{1}{2^Q} \times \left(1 - \frac{1}{2^Q}\right)^{n-1}, \tag{6.1}$$

where 2^Q is the frame size in the protocol and n is the number of tags in the system.

From the well-known theory in ALOHA-based protocols, the protocol's performance is optimal when the frame size (*i.e.*, 2^Q in Q-adaptive) equals the total number of tags in the system [44]. This can be obtained from differentiating equation 6.1 with respect to Q and equate it to 0:

$$\begin{aligned} \frac{dp_s}{dQ} &= \frac{n(n-1)\ln(2)}{2^{2Q}}\left(1 - \frac{1}{2^Q}\right)^{n-2} - \frac{n\ln(2)}{2^Q}\left(1 - \frac{1}{2^Q}\right)^{n-1} \\ &= \frac{-n\ln(2)}{2^Q}\left(1 - \frac{1}{2^Q}\right)^{n-2}\left[\left(1 - \frac{1}{2^Q}\right) - \frac{n-1}{2^Q}\right] \\ &= \frac{-n\ln(2)}{2^Q}\left(1 - \frac{1}{2^Q}\right)^{n-2}\left[1 - \frac{n}{2^Q}\right]. \end{aligned} \tag{6.2}$$

For maximum throughput, equating equation 6.2 to 0, we have:

$$\begin{aligned}\left[\frac{dp_s}{dQ} = 0\right] &\implies \left(1 - \frac{n}{2^Q}\right) = 0 \\ &\implies n = 2^Q \\ &\implies Q = \log_2(n)\end{aligned} \tag{6.3}$$

Therefore, when $Q \approx \log_2(n)$ the EPC Gen2 Q-adaptive protocol achieves its theoretical maximum performance. Therefore success probability at maximum throughput can be computed from equation 6.1 as

$$p_s = n \times \frac{1}{2^{\log_2(n)}} \times \left(1 - \frac{1}{2^{\log_2(n)}}\right)^{n-1} = \left(1 - \frac{1}{n}\right)^{n-1}. \tag{6.4}$$

Therefore,

$$p_s \approx e^{-\frac{n-1}{n}} \approx e^{-1}. \tag{6.5}$$

By Chebyshev's inequality [49] the outcome of a random variable is most likely somewhere near to its expected value. The objective of the anti-collision protocol is to identify all of the n tags in the system. This incurs the tag identification experiment, a total time of n successful events to happen. Each of these events is independent and identically distributed according to the aforementioned binomial distribution, as shown in equation 6.1. Let T be the total number of slots required to identify n tags in the system. By Chebyshev's inequality and from the expected value of binomial distribution, the total number of slots for identifying n tags for a theoretically optimal protocol whose Q value is always $\log_2(\overline{n})$, where $\overline{n}$ is the number of remaining unidentified tags in the system, can be computed as follows:

$$T \times e^{-1} = n. \tag{6.6}$$

Therefore, the total number of slots T a theory-optimal protocol requires is as follows:

$$T = \frac{n}{e^{-1}} \approx 2.72 \times n. \tag{6.7}$$

Hence, a theoretically optimal protocol incurs roughly $2.72n$ slots to identify n tags in the system.

The following lemma shows that the standard Q-protocol that starts by a default Q-value of 4 converges at a faster rate to $log_2(n)$, which is the state of theoretical maximum.

Lemma 3. *The standard Q-protocol that initiates by default with $Q = 4$ converges quickly to theory optimal frame-size especially for large values of n.*

Proof. The idle-slot event p_i follows a binomial distribution and is computed as follows:

$$p_i = \left(1 - \frac{1}{2^Q}\right)^n \approx e^{-(\frac{n}{2^Q})}. \tag{6.8}$$

The collision probability p_c can be computed as follows:

$$p_c = 1 - p_s - p_i. \tag{6.9}$$

From equation 6.1 and equation 6.8, p_c can be written as follows:

$$p_c \approx 1 - \left[\frac{n}{2^Q} \times e^{-(\frac{n-1}{2^Q})}\right] - \left[e^{-(\frac{n}{2^Q})}\right]. \tag{6.10}$$

Let T_A denote the transient phase of the protocol, that covers the duration as number of time-slots of the protocol from its starting state with $Q = 4$ to the state it reaches a theoretically optimal state, *i.e.*, $Q = \log_2 n$. For the case of a large n, the value of 2^Q is relatively small during the initial transient period of the protocol, thereby causing the e^{-x} terms in equation 6.10 negligible. In other words, $e^{-x} \rightarrow 0$ as $x = n/2^Q >> 1$

and this applies to the terms in equation 6.10. Therefore, $p_c \to 1$ in the initial transient period of the protocol. The initial transient period ends when $x \approx 1$, *i.e.,* when $2^Q \approx n$ the theoretical maximum performance.

By the nature of Q-adaptive protocol, during collision-slot events, the Q-value is incremented by an offset of c, where c is a real number between 0.1 and 0.5. In other words, during a collision slot, we have

$$Q = Round(Q_{fp} + c). \tag{6.11}$$

where Q_{fp} is the floating-point representation of Q. During the initial transient phase, with a very high probability of collision events occuring, the total number of slots it takes to reach the stable phase is given by

$$(Q/c \times p_c) - 2^4 \approx (Q/c - 2^4). \tag{6.12}$$

In equation 6.12, the 2^4 term reduction is due to the fact that the standard Q protocol starts with the initial value of 4 instead of 0.

Figure 6.2 shows the number of slots in the transient phase. The simulation results are compared with the derived upper-bound representing the starting point of the theoretically stable state, *i.e.*, $((\log_2(n)/c) - 2^4)$. The simulated number of slots in the transient phase is upper-bounded by the theoretical bound. Therefore,

$$\lim_{n\to\infty} \frac{T_A}{T} \to 0. \tag{6.13}$$

Hence the transient phase of the Q-adaptive protocol converges at a relatively faster rate. □

From Lemma 3, it is clear that the Q-adaptive protocol rapidly reaches a stable state. Figure 6.3 shows this behavior of Q for $30,000$ tags. It should be noted that

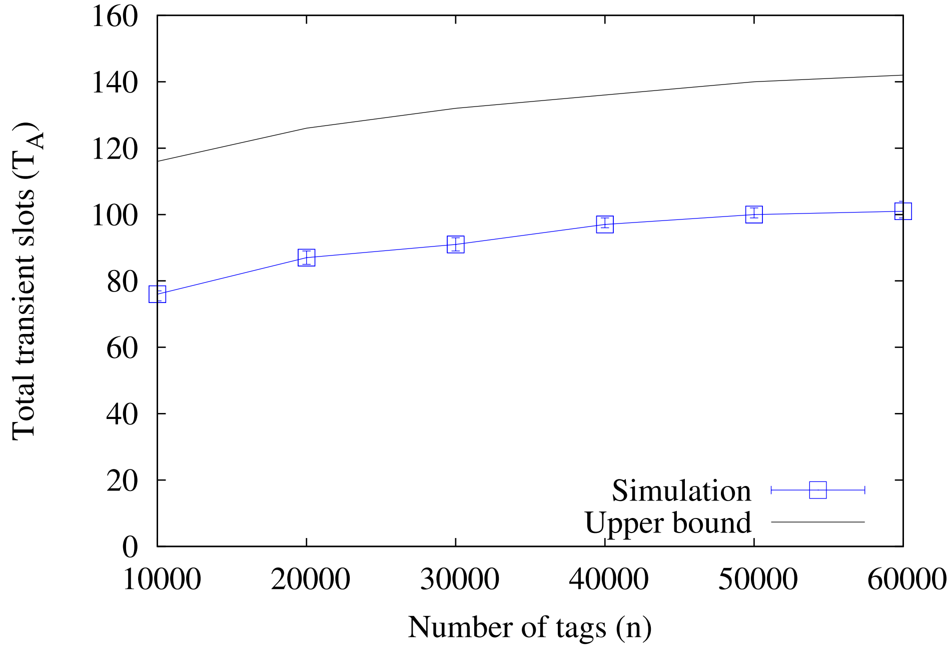


Figure 6.2: Number of slots in the transient phase T_A vs total number of tags, compared against Lemma 3 bound shown in equation 6.12

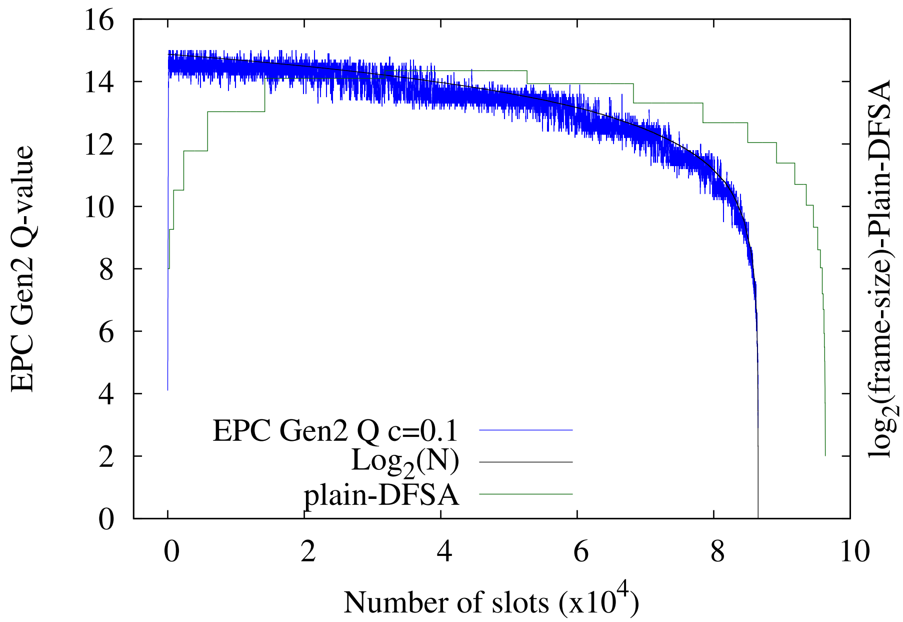


Figure 6.3: Frame size vs number of slots for $30,000$ tags. The *X-axis* is appropriately shown to highlight the starting frame-size values

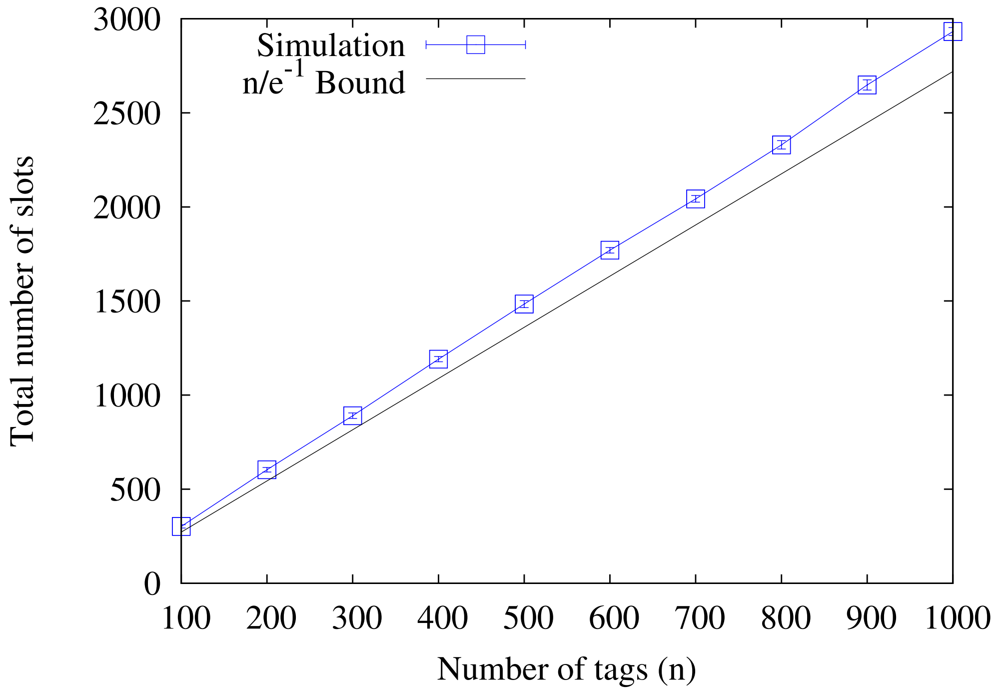
the simulation values of Q are bounded by the optimal frame-size values of Q, which is $\log_2(\overline{n})$, and also lower-bounded by the theoretically optimal protocol which is $(\overline{n})/e^{-1}$. Figure 6.3 also shows the slot-wise evolution of frame-size in a plain-DFSA protocol. The step-wise decrease of frame size exemplifies the slow adaptive nature of the plain-DFSA over the aggressive nature of the Q-adaptive protocol. The transient phase T_A of plain-DFSA incurs a significantly larger number of slots than that of the Q-adaptive protocol. Therefore, Q-adaptive is comparatively the most dynamic and fast-responsive protocol.

It is clear from Lemma 3 that the Q-protocol converges at a faster rate to the theoretical optimal, especially when n is large. Subsequently, after reaching the optimal stable state, the protocol maintains its stable state for the entire remaining duration of the tag-identification process. Therefore, the protocol's performance is largely dominated by its theoretical maximum performance. Hence the performance of this protocol can be closely approximated to the performance of a theory optimal protocol. The bound n/e^{-1} is therefore a reasonably valid approximation that predominantly characterizes the delay of the protocol.

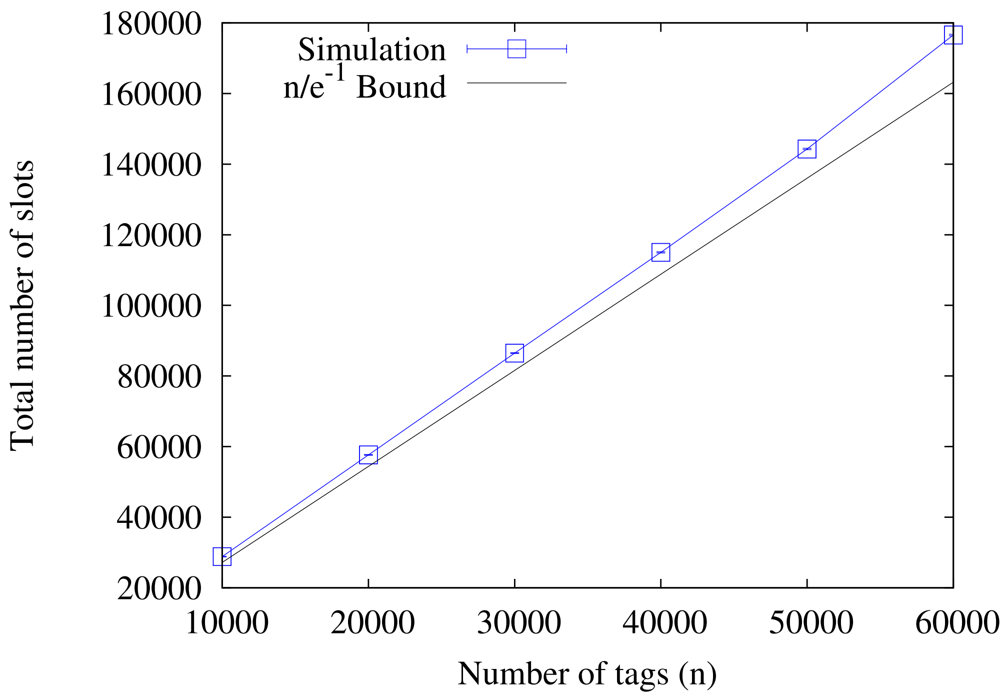
6.5 Performance Evaluation

This section provides an evaluation of the ALOHA-based protocols using simulation experiments, and demonstrates the tightness of the proposed bound-based solution. For this, the EPC Gen2 Q-adaptive protocol is simulated in a custom-developed C++ program. All performance results are plotted with 95% confidence.

The latest works that model the EPC Gen2 Q-adaptive protocol's performance using an absorbing Markov chain [48] [47] evaluated the protocol's performance in their experiments by considering a maximum number of tags of up to 200, for two different c values, say 0.2 and 0.4, in respective separate experiments. The previ-



(a) EPC Gen2 performance with $c = 0.1$, for number of tags varied from 100 tags to 1000 tags, compared against the theoretical maximum bound



(b) EPC Gen2 simulation performance with $c = 0.1$, for large number of tags varied from 10K to 60K, as compared to the theoretical maximum bound

Figure 6.4: EPC Gen2 performance

ous work [51], which is seen as an approximate Markov chain model (over the latest work [47]) incurs a relatively less state-space have evaluated the protocol's performance for a maximum number of $1,000$ tags, for the respective 0.2 and 0.4 different c-values. It is not evident from these works, as they have not explicitly mentioned the maximum number of tags their model can tractably capture for a given system's computing configuration.

To highlight the tightness of this bound in the real large-scale regime of the system, which is beyond the scope of the Markov models, Figure 6.4(b) shows the performance results for the number of tags varied between $10,000$ and $60,000$ in the increments of $10,000$ tags. The approximation fits well with the simulation counterpart with a reasonable accuracy. The difference in the accuracy of the model with respect to the simulation results is shown in Figure 6.5. Let T_s denote the number of simulated slots incurred in the EPC Gen2 Q-adaptive protocol in the simulation environment. Let T_o denote the number of slots incurred by the theoretically optimal protocol, $i.e., n/e^{-1}$. Then the accuracy (in percentage) is computed as follows:

$$Accuracy = \left(1 - \frac{T_s - T_o}{T_s}\right) \times 100. \tag{6.14}$$

The accuracy is maintained reasonably high throughout the entire ranges of the tags. The accuracy is better for a larger number of tags, with a maximum of up to 95% for $10,000$ tags and 92% for $60,000$ tags. Therefore, the n/e^{-1} can be considered as a good approximation in characterizing the EPC Gen2 protocol's performance for a large number of tags in this system.

6.6 Summary

This chapter studied the performance of EPC Gen2 Q-adaptive protocol and its sensitiveness to a large number of tags. It has been demonstrated that the Q-adaptive

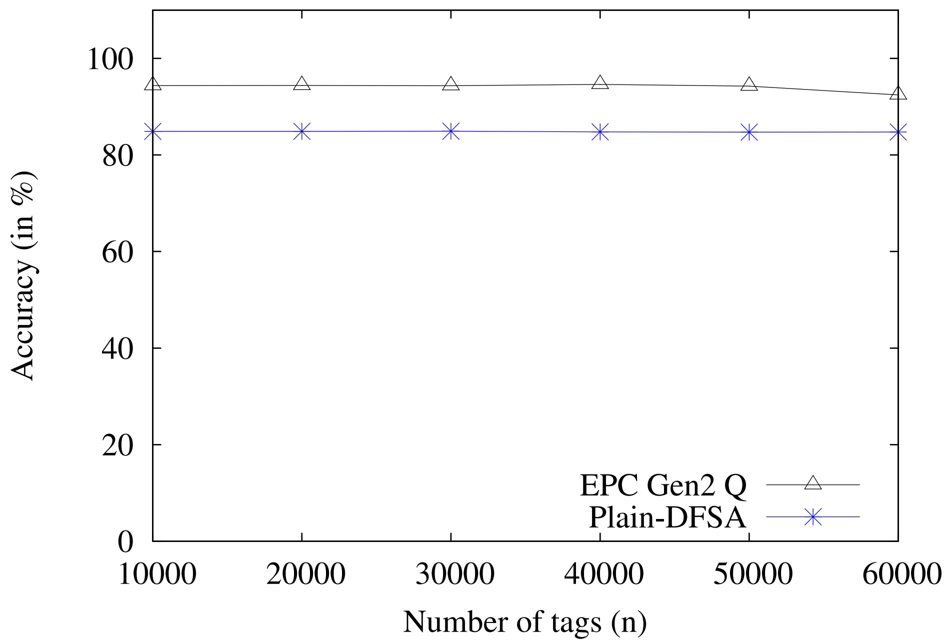


Figure 6.5: Accuracy of the respective EPC Gen2 ($c = 0.1$) vs Plain-DFSA simulation performance results, with respect to the theory-optimal bound

protocol aggressively reaches the theoretically maximum performance and maintains around this optimal state for most of the time. Hence this stable optimal phase predominantly characterizes the protocol's delay performance. Therefore, with a reasonable accuracy, the protocol's performance can be bounded by its optimum efficiency of n/e^{-1} time slots, where n is the total number of tags in the system. Extensive simulation results validate this standpoint and, for a large number of tags, the accuracy was found to be a maximum of 95% for $10,000$ tags and 92% for $60,000$ tags. This exemplifies the fact that the bound holds good for the scenario of a large number of tags. The research in this chapter is adapted from [42].

Chapter 7

Conclusions

7.1 Summary

This chapter summarizes the research presented in this dissertation. Chapter 1 provided a quick look into various components of an RFID system. It also discussed the range of frequencies used by RFID technology in the frequency spectrum along with applications and distance of operation pertaining to each frequency band. The different categories of tags that exist are introduced with the functionalities that come with them. The broad classification of the existing RFID protocols is described and the layout of the dissertation structure is given.

Chapter 2 presented several existing RFID tag-identification protocols like the QTP, BTP, BFSA, DFSA, EDFSA, QTDFSA and Hybrid Anti-Collision algorithms. The tag estimation procedures used and tag functions performed in these protocols give a clear understanding of the required functionalities and performance limitations for the reader and tags, respectively. All the protocols discussed in this chapter require passive tags that can perform basic operations such as store, load, add, subtract, divide, multiply and modulo. These protocols form the foundation for the frameworks developed in later chapters. It helps to fully understand the capabilities of reader and tags to design mobility models and frameworks for mobile tags or reader and multiple readers environments. The tag identification protocols hold immense possibilities to develop mobility models in a warehouse scenario in the supply chain industry.

Chapter 3 proposes the first research problem where consecutive readers share information to improve the tag-reading efficiency of the protocols functioning in a conveyor belt mobility scenario. The proposed augmentation frameworks for ALOHA, tree and two different combinations of hybrid tag-reading protocols identifies different types of information-sharing for each one of them. The augmented protocols outperformed from their respective native counterparts, with the augmented ALOHA protocol showing a maximum improvement of 50%. The improvement gained due to the augmentation is mathematically studied. Furthermore, the scalability problem of determining the number of readers required to identify almost all tags in the system is also addressed.

Chapter 4 addresses an important RFID tag-identification problem in a widely applied warehouse scenario. In this regard, a novel zonal-spacing-framework to improve the throughput of an RFID warehouse system is introduced, consisting of a mobile reader moving along the aisles between the grids. The proposed framework augments the industry standards compliant DFSA to yield better performance. This framework provides an three orders of magnitude of performance improvement over the untailored protocol implementation. Moreover, the analytical scalability study provides useful insights into this warehouse problem domain in identifying the essential parameters of the system that affects the protocol's performance. The analysis with validation from simulation results enables the system designers to make strategic design decisions during the deployment phase. The analysis provided is generic enough to be extended for multiple readers setting, which helps to further the research along the similar lines.

Chapter 5 provides research that is the first one to make use of NF T2T communications in order to assist the reader in improving the tag reading performance. This research proposes the NF T2T communication-based augmentation frameworks for the respective Q-adaptive and plain DFSA protocols. The Q-adaptive protocol even with additional communication overheads still remains as the best-in-class protocol in terms of energy expenditure. The NF T2T variant of plain-DFSA protocol outperforms in terms of delay, an improvement of up to 80%. The NF T2T variant of Q adaptive performs close to the respective DFSA counterpart with an improvement of up to 72%. Hopefully, this work would provide the research community a novel direction of using NF communications for further improving the efficiency of the anti-collision protocols.

Chapter 6 presents an analysis of the performance of EPC Gen2 Q-adaptive protocol and its sensitiveness to a large number of tags. It is demonstrated that the Q-adaptive protocol aggressively reaches the theoretical maximum performance and maintains around this optimal state for most of the time. Hence this stable optimal phase predominantly characterizes the protocol's delay performance. Therefore, with reasonable accuracy, the protocol's performance can be bounded by its optimum efficiency of n/e^{-1} time-slots, where n is the total number of tags in the system. Extensive simulation results validate this standpoint and for a large number of tags, the accuracy was found to be the maximum of 95% for $10,000$ tags and 92% for $60,000$ tags. This exemplifies the fact that this bound holds good for the scenario of a large number of tags.

7.2 Future Directions

The research in this dissertation can be expanded in several avenues, some of which are discussed below.

1. The mobility model studied in Chapter 3 can be implemented with the multiple overlapping conveyor belt scenario, where new tags join the system in the middle of adjacent readers. This would require measures to be taken to prevent reader-to-reader interference alongside efficient tag-identification.

2. The warehouse model presented in Chapter 4 can be extended to one or multiple

readers in the warehouse. Also, the implementation can be extended to grids of different shapes and sizes.

3. The near-field tag-to-tag communication considered in Chapter 5 can be studied from the implementation and analytical perspective. This concept can be used to solve problems in other areas such are RFID security. Some commonly encountered attacks are denial-of-service attacks, integrity and privacy attacks that compromise the effective functioning of RFID tags. The NF communication can detect and prevent such security issues.

4. The analysis discussed in Chapter 6 can be extended by understanding the behavior of protocol that includes the capture effect. The capture effect is the physical layer mechanism that helps the protocol to identify tags during collisions, to a certain extent.

Bibliography

[1] Basic components of an RFID system.

[2] EPC radio frequency identity protocols generation-2 UHF RFID standards specification.

[3] Five factors for success - UHF Gen2 RFID readers.

[4] Gs1 epcglobal standards for item-level tagging.

[5] Impinj launches UHF tech for item-level tags.

[6] Item-level tagging in the grocery industry - are we there yet.

[7] RFID frequencies and frequency bands.

[8] RFID in the supply chain: The Wal-Mart factor.

[9] Wal-Mart: A case study: Implementation of RFID in supply chain management.

[10] Taxonomy and survey of RFID anti-collision protocols. *Computer Communications*, 29(11):2150–2166, 2006.

[11] S.-Y. Ahn, J.-S. Park, Y.-R. Seong, H-.R. Ohd, and J.-M. Kim. BoMR: A booster for mobile RFID readers. In *ICTC '11: Proceedings of the International Conference on ICT Convergence*, pages 161–165, 2011.

[12] J. J. Alcaraz, J. V. Alonso, and J. Garc´ía-Haro. RFID reader scheduling for reliable identification of moving tags. *IEEE Transactions on Automation Science and Engineering*, 10(3):816–828, 2013.

[13] K. Ali and H. Hassanein. Distributed receiving in RFID systems. In *LCN '09: Proceedings of the International Conference on Local Computer Networks*, pages 69–76, 2009.

[14] N. Amin and P. W. Lin. A sequential study of anti-collision protocol development for passive RFID tags. In *International Journal of Circuits, Systems, and Signal Processing*, volume 1, pages 161–165, 2007.

[15] M. V. Bueno-Delgado, R. Ferrero, F. Gandino, P. Pavón-Mariño, and M. Rebaudengo. A geometric distribution reader anti-collision protocol for RFID dense reader environments. *IEEE Transactions on Automation Science and Engineering*, 10(2):296–306, 2013.

[16] MIT Auto-ID Center. Draft protocol specification for a 900 mhz class 0 radio frequencyidentification tag. 2003.

[17] J.-R. Cha and J.-H. Kim. Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID system. In *CCNC '06: Proceedings of the International Consumer Communications and Networking Conference*, pages 768–772, 2006.

[18] C.-L. Chen, J.-K. Jan, and C.-F. Chien. Based on mobile RFID device to design a secure mutual authentication scheme for market application. In *BWCA '10: Proceedings of the International Conference on Broadband, Wireless Computing, Communication and Applications*, pages 423–428, 2010.

[19] D. W. Engels and S. E. Sanna. Standardization requirements within the RFID class structure framework. *tech. rep., Auto-ID Center*, 2005.

[20] Y. Feng. Design and implementation of middleware for RFID warehouse management. In *ICICEE '12: Proceedings of the International Conference on Industrial Control and Electronics Engineering*, pages 1258–1260, 2012.

[21] K. Finkenzeller. RFID handbook: Radio-frequency identification fundamentals and applications. *John Wiley and Sons*, 2000.

[22] K. Finkenzeller and D. Muller. RFID handbook: Fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication, 2010.

[23] Institute for Local Self-Reliance. Store size caps. [Online; March 15, 2012].

[24] L. Fu, L. Liu, M. Li, and J. Wang. Collision recovery receiver for EPC Gen2 RFID systems. In *IOT '12: Proceedings of the International Conference on Internet of Things (IOT)*, pages 114–118, 2012.

[25] S.K. Goudos, K. Siakavara, and J.N. Sahalos. Novel spiral antenna design using artificial bee colony optimization for UHF RFID applications. *IEEE Antennas and Wireless Propagation Letters*, 13:528–531, 2014.

[26] J.-W. Jung, K.-K. Nae, J. Kim, and J.-S. Park. RF-IPD directional coupler for mobile RFID handset applications. *Electronics Letters*, 43(13):719–720, 2007.

[27] G. Khandelwal, A. Yener, K. Lee, and S. Serbetli. ASAP: A MAC protocol for dense and time constrained RFID systems. In *ICC '06: Proceedings of the International Conference on Communications*, pages 4028–4033, 2006.

[28] Y.-C. Lai, L.-Y. Hsiao, H.-J. Chen, C.-N. Lai, and J.-W. Lin. A novel query tree protocol with bit tracking in RFID tag identification. *IEEE Transactions on Mobile Computing*, 12(10):2063–2075, 2013.

[29] H. Landaluce, A. Perallos, and I.J.G. Zuazola. A fast RFID identification protocol with low tag complexity. *IEEE Communications Letters*, 17(9):1704–1706, 2013.

[30] C. Law, K. Lee, and K. Y. Siu. Efficient memoryless protocol for tag identification. In *DIAL-M '00: Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 75–84, 2000.

[31] S.-R. Lee, S.-D. Joo, and C.-W. Lee. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. In *MobiQuituous '05: Proceedings of the International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 166–172, 2005.

[32] M. Lin, H. Rowaihy, T. Bolbrock, G. Cao, and T. La Porta. Data collection using RFID and a mobile reader. In *IEEE GLOBECOM '08: Proceedings of the IEEE Global Telecommunications Conference*, pages 1–6, 2008.

[33] Y. Maguire and R. Pappu. An optimal Q-algorithm for the ISO 18000-6C RFID protocol. *IEEE Transactions on Automation Science and Engineering*, 6(1):16–24, 2009.

[34] L. Minbo, G. Shengxi, C. Guangyu, and Z. Zhu. A RFID-based intelligent warehouse management system design and implementation. In *ICEBE '11: Proceedings of the International Conference on e-Business Engineering*, pages 178–184, 2011.

[35] S. Nakao, T. Norimatsu, T. Yamazoe, T. Oshima, K. Watanabe, K. Minatozaki, and Y. Kobayashi. UHF RFID mobile reader for passive- and active-tag communication. In *RWS '11: Proceedings of the Radio and Wireless Symposium*, pages 311–314, 2011.

[36] P.V. Nikitin, K.V.S. Rao, and S. Lazar. An overview of near field UHF RFID. In *RFID '07: Proceedings of the International Conference on RFID*, pages 167–174, 2007.

[37] Y. X. Qing, Y. Z. Ping, and X. Y. Lun. A query tree dynamic frame slot ALOHA collision resolution protocol for RFID tags. In *FGCN '08: Proceedings of the Internationa Conference on Future Generation Communication and Networking*, volume 1, pages 198–201, 2008.

[38] G. Roussos. Enabling RFID in retail. *Computer*, 39(3):25–30, 2006.

[39] V. Sarangan, M. R. Devarapalli, and S. Radhakrishnan. A framework for fast RFID tag reading in static and mobile environments. *Computer Networks*, 52(5):1058–1073, 2008.

[40] C.J. Satyavolu, S. Radhakrishnan, V. Sarangan, and T.L. Landers. Framework for co-operative RFID tag reading in mobile environments with multiple readers. In *ICNC '12: Proceedings of the International Conference on Computing, Networking and Communications*, pages 1172–1177, 2012.

[41] C.J. Satyavolu, S. Radhakrishnan, V. Sarangan, and T.L. Landers. On RFID tag reading by a mobile reader in a warehouse. In *ANTS '13: Proceedings of the International Conference on Advanced Networks and Telecommuncations Systems*, pages 1–6, 2013.

[42] C.J. Satyavolu, M. Veeramani, and S. Radhakrishnan. Item-level tagging sees more tags: Analyzing the performance of epc gen-2 protocol in large-scale RFID systems. In *GLOBECOM '14: Global Telecommunications Conference*, 2014.

[43] C.J. Satyavolu, M. Veeramani, S. Radhakrishnan, and J. Ruyle. Close-coupled chips can coordinate to contain collisions. In *VTC Fall '14: Vehicular Technology Conference*, 2014.

[44] P. Solic, J. Radic, and N. Rozic. Algorithm for deriving optimal frame size in passive RFID UHF class1-gen2 standard using combinatorial model boundaries. *Automatika: Journal of Control, Measurement, Electronics*, 51(3):255–263, 2010.

[45] P. Solic, J. Radic, and N. Rozic. Linearized combinatorial model for optimal frame selection in gen2 RFID system. In *RFID '12: Proceedings of the International Conference on RFID*, pages 89–94, 2012.

[46] W. Strauss, S. Kraus, and J. Bernhard. Statistical analysis of UHF RFID bulk measurements. In *RFID Sys Tech '10: Proceedings of the European Workshop on Smart Objects: Systems, Technologies and Applications*, pages 1–8, 2010.

[47] E. Vahedi, R.K. Ward, and I.F. Blake. Analytical modeling of RFID generation-2 protocol using absorbing Markov chain theorem. In *GLOBECOM '12: Proceedings of the International Global Communications Conference*, pages 385–390, 2012.

[48] E. Vahedi, R.K. Ward, and I.F. Blake. Performance analysis of RFID protocols: CDMA versus the standard EPC Gen-2. *IEEE Transactions on Automation Science and Engineering*, PP(99):1–12, 2014.

[49] H. Vogt. Efficient object identification with passive RFID tags. In *Pervasive '02: Proceedings of the International Conference on Pervasive Computing*, pages 98–113, 2002.

[50] H. Vogt. Multiple object identification with passive RFID tags. In *SMC '02: Proceedings of the International Conference on Systems, Man and Cybernetics*, pages 6–9, 2002.

[51] C. Wang, M. Daneshmand, K. Sohraby, and B. Li. Performance analysis of RFID generation-2 protocol. *IEEE Transactions on Wireless Communications*, 8(5):2592–2601, 2009.

[52] S. Wang, W. Hong, L. Yin, and S.-F. Li. A novel fast tag estimate method for dynamic frame length Aloha anti-collision algorithms in RFID system. In *VTC FALL '12: Proceedings of the International Vehicular Technology Conference*, pages 1–5, 2012.

[53] L. Weisgerber and A.E. Popugaev. Multibeam antenna array for RFID applications. In *EuMC '13: Proceedings of the European Microwave Conference*, pages 84–87, 2013.

[54] J. White and N. Banerjee. Mirage: Mitigating illicit inventorying in a RFID enabled retail environment. In *COMSNETS '11: Proceedings of the International Conference on Communication Systems and Networks*, pages 1–9, 2011.

[55] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li, and D. Chen. Efficient tag identification in mobile RFID systems. In *INFOCOM '10: Proceedings IEEE*, pages 1 –9, 2010.

[56] X.-Q. Yan, J. Bai, Y. Xu, and B. Li. An optimized schema to improve the time efficiency of the EPG gen-2 protocol. In *iThings '11, Proceedings of the International Conference on Cyber, Physical and Social Computing*, pages 123–126, 2011.

[57] X. Zhang and X. Lian. Design of warehouse information acquisition system based on RFID. In *ICAL '08: Proceedings of the International Conference on Automation and Logistics*, pages 2550–2555, 2008.

[58] B. Zhen, M. Kobayashi, and M. Shimizu. Framed ALOHA for multiple RFID objects identification. In *IEICE Transactions on Communications*, volume 88-B, pages 991–999, 2005.

[59] M. Zheng, J. Xie, Z. Mao, and Y. Zhu. A hybrid anti-collision algorithm for RFID with enhanced throughput and reduced memory consumption. In *EUC '08: Proceedings of the International Conference on Embedded and Ubiquitous Computing*, pages 259–265, 2008.