MOBILE TAG READING IN A MULTI-READER RFID ENVIRONMENT

By

RUPESH BHOCHHIBHOYA

Bachelor of Engineering in Information Technology

Pokhara University

Pokhara, Nepal

2005

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2008

MOBILE TAG READING IN A MULTI-READER RFID ENVIRONMENT

Thesis Approved:

Dr. Venkatesh Sarangan
Thesis Adviser

Dr. Nohpill Park

Dr. Johnson Thomas

Dr. A. Gordon Emslie
Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Venkatesh Sarangan for his constant support, understanding, regular guidance, and encouragement throughout my graduate program. I would also like to express my sincere thanks to Dr. Johnson Thomas and Dr. Nophil Park for serving on my Graduate Committee. I am indebted to Dr. Robert J. Hauenstein, Department of Physics for his continuous support in the pursuit of my Masters. I extend my thanks to all the faculty and staff of the Computer Science Dept for their constant support. I also wish to express my sincere thanks and appreciation to my family and friends for their unfailing support and constant encouragement in all my pursuits.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

RFID stands for **Radio-Frequency Identification**. The acronym refers to a form of a system where a tag that stores a data is affixed to an item. This data is transferred to a reader using electromagnetic radio frequency for an Automatic Identification and Data Capture (AIDC) purposes. A tag is a small electronic device that consists of a small chip that is capable of storing certain amount of data, and an antenna. A reader is a device that recognizes the presence of RFID tags and reads the information stored on them.

The RFID devices can be used to serve the similar purposes, but not limited to, as a bar code or a magnetic strip on the back of a credit card providing a unique identifier for that object. And, just as a bar code or magnetic strip must be scanned to get the information, the RFID device must be scanned to retrieve the identifying information. The reader can then inform another system about the presence of the tagged items. The system that the reader communicates usually runs software, called RFID middleware, that stands between readers and applications.

1.1. Motivation

In the near future, large scale RFID deployment on various applications is likely to use multiple readers as it will be hard to achieve the required read throughput with single reader. However, with the use of multiple readers, the RFID system suffers from Reader

Collision problems which we will be thoroughly discussed in section 2.7. There are numerous papers both proposed and published that offer solutions to the Reader Collision problem that we described in chapter 3. Such collision problems become complex when either or both of the reader or tags are mobile. There are numerous papers proposing novel frameworks as well as new MAC protocols which addressed the case of mobile readers in the multi-reader environment. However, it is our understanding that none of the papers that have been published so far have addressed the collision problems that are specific to the multi-reader scenario where tags are mobile. Hence this has been the motivation factor for us to conduct this research and propose a new framework that gives the solution to maximize the read throughput.

1.2. Scenario

This section defines a practical scenario that has been considered for this thesis. It is a typical manufacturing process in a supply chain where a tag affixed to an item moves on a conveyor belt like in assembly line and passes through the Readers fixed above the conveyor. Readers are aligned and installed as close as possible to each other so that tag reading process can be accomplished within shortest length possible. This causes the overlapping of interrogation regions among adjacent readers where the collision takes place if such overlapping readers operate at the same time. Herein we will call this scenario as MOBILE-SCENARIO for convenience. Figure-1.2 depicts a typical scenario that is under our consideration.

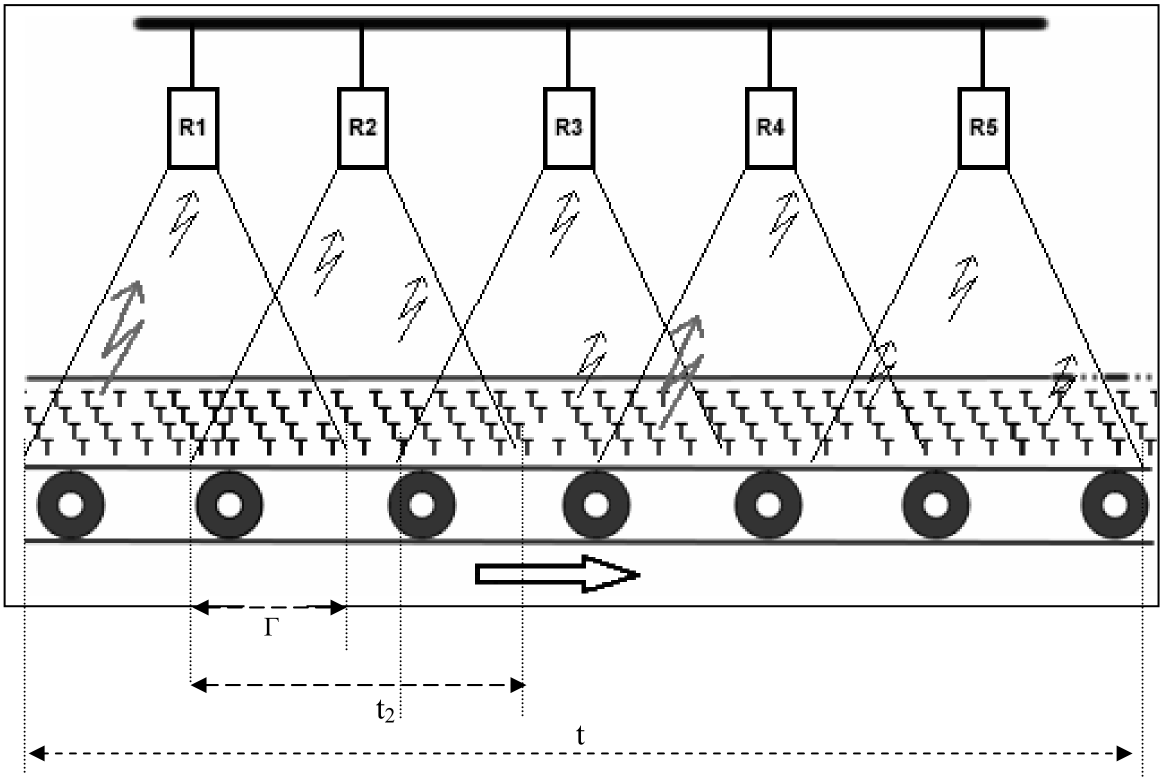

Figure-1.2: Scenario under consideration

1.3. Thesis Goal

The main objective of this thesis is to propose an efficient framework for the MOBILE-SCENARIO and prove that the time slotted model is the best approach to address the reader collision problem. We will design an offline solution that guarantees that the slotted ALOHA model can provide a better read throughput then without using such model. We will also show the how the tag arrangement on the conveyor belt can affect the overall throughput and then verify this hypothesis with simulation results. In the end, the following questions will be answered in this thesis:

- What threshold value of time (called tau; Γ that the tag spends on an overlapping region between any two adjacent readers) does the deployment of slotted time model (Duty cycle) guarantee the improvement of read throughput over the system lacking slotted model?

- What value of duty cycles (called delta; δ) for each reader or group of readers, guarantees the maximum throughput of the system?
- Since, tags can be placed on to the belt either in a distributed fashion or in a clump, what arrangement of tag placement on the belt assures better throughput?

Once the above questions are answered and verified through simulation, we could consider the completion of thesis.

1.4. Thesis Organization

This thesis is organized as follows:

- Chapter 2 discusses fundamental concepts of RFID technology and other preliminary concepts necessary for understanding the problems that may be encountered in MOBILE-SCENARIO.
- Chapter 3 reviews various approaches and protocols that have been proposed on several research papers. It explains all the important and successful MAC protocols that have contributed solving collision problem so far.
- Chapter 4 introduces the proposed solution of this thesis. It first gives detail of the framework for RFID system under the considered scenario. Then it exposes various expressions and conditions which insure the theoretical correctness of the proposed approach.

- Chapter 5 presents the simulation results to verify the statements made on our proposed solution.
- Finally, the conclusion of this thesis are drawn and included in chapter 7. This chapter also throws some lights on future research works that can be done on multi-reader RFID system.

CHAPTER 2

BACKGROUND STUDY

The use of RFID technology has brought a major paradigm shift in many businesses and applications as in supply chain replacing the bar-code technology for an instance. The advantages of RFID technology over bar codes and other automated data collection technologies are reliability and flexibility in reading the tags in a wider scanning area with greater speed, resulting significant economic, operational, technological, and logistical impacts on supply chain infrastructure. Due to the limitless possibilities and low cost, RFID supporters claim to see the integration of RFID in all businesses. This section describes the basics of RFID technology and its applications. It also details the basic components of RFID system and its infrastructure with respect to the supply chain.

2.1 Basic RFID Components

An RFID system includes three primary components: a transponder, an interrogator, and a data collection device known as a controller as shown in a figure-2.1 below. Simply put, a transponder, like a bar code, stores information but has the capacity to store considerably more information than a bar code. RFID transponder exchange information with a RFID transceiver by radio frequency (RF) signal.

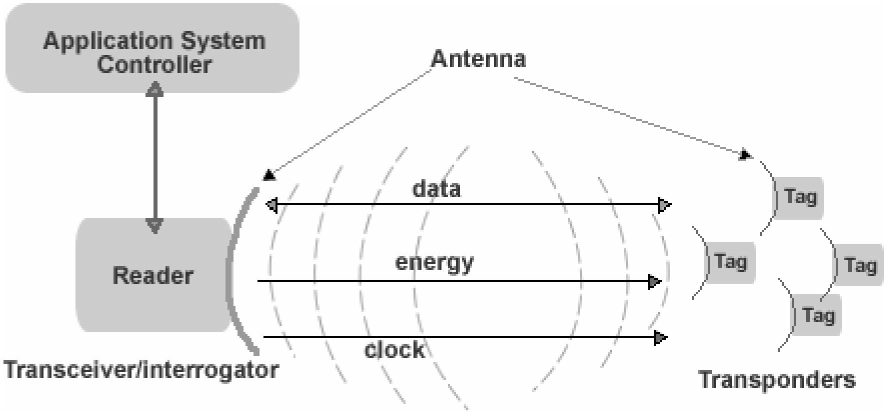

Figure-2.1: Basic RFID Components

2.1.1 Transponder

A RFID transponder, also known as RFID tag, is simply a microchip device affixed onto an item, animal or person for the purpose of identification using radio frequency. RFID tags are composed of an antenna coil for receiving and transmitting signal and an integrated circuit for storing and processing information, modulating and demodulation signal and other specialized functions. The reading range of the RFID system varies according to the type of RFID tag used. There are mainly two types of tags; the active tag and the passive tag.

Active tag is equipped with internal power supply, used for broadcasting the response signal and other circuitry operations. Communication between active tags and readers is much more reliable than passive tags and the operation range is much higher, improving the utility of the system. However, active tags have a limited life span and typically more expensive and physically larger than a passive tag.

Passive tag is the one that has no separate external power and obtains its operating power from the RF energy generated from reader. This is the type of the tag we are interested in. Such tags are the most popular as they are much less expensive and have very long lifespan. These tags are much smaller and have almost unlimited applications in consumer goods and other areas. However, the tag can be read only at very short distances, typically a few feet.

Semi-passive tag or semi-active tag is a battery-assisted tag that uses its internal battery to power its circuitry but does not power the broadcasting of a signal. The response is usually powered by means of backscattering(see section 2.2) the RF energy from the reader, where energy is reflected back to the reader as with passive tags.

2.1.2 Transceiver

Transceiver, also called a RFID reader, is a device used to interrogate an RFID tag. RFID reader acts as a bridge between the RFID tag and the controller and ha just a few basic functions [12].

- Read the data contents of an RFID tag
- Write data to the tag
- Relay data to and from the controller
- Power the tag

In addition to these functions, readers are responsible for implementing anti-collision measures to ensure simultaneous communication with many tags which will be discussed in section 2.8 and in chapter 4. Authenticating tags and data encryption are other vital functions of the RFID reader.

2.1.3 RFID Controller

RFID controllers are the brains [12] of RFID system. They are the central system in a multi-reader network where it collects all the information from readers and also send control signal to the RFID readers. Such information gathered from RFID readers is used for various purposes such as tracking products, inventory monitoring or identity verification etc.

2.2 How Does RFID Works

In a typical RFID system, the reader continuously generates an RF carrier sine wave, watching always for modulation to occur. Detected modulation of the field would indicate the presence of a tag. When a tag enters the RF field generated by the reader, it first receives sufficient energy to operate correctly then starts transmitting data back to the reader by the modulation method called **backscattering.** By repeatedly shunting the tag coil through a transistor, the tag can cause slight fluctuations in the reader's RF carrier amplitude. The RF link behaves essentially as a transformer; as the secondary winding (tag coil) is momentarily shunted, the primary winding (reader coil) experiences a momentary voltage drop. This fluctuation is detected by the reader as data bits which may be further modulated or encoded and relayed to the controller.

2.3 RFID Infrastructure

This section talks about the complete structural arrangement supporting RFID technology. The RFID infrastructure discussed below is based on a typical EPC Network model. Such model includes primarily Tag, Tag reader, EPC, Savant server, and Object Name Service (ONS) as depicted in Figure-2.3 below.

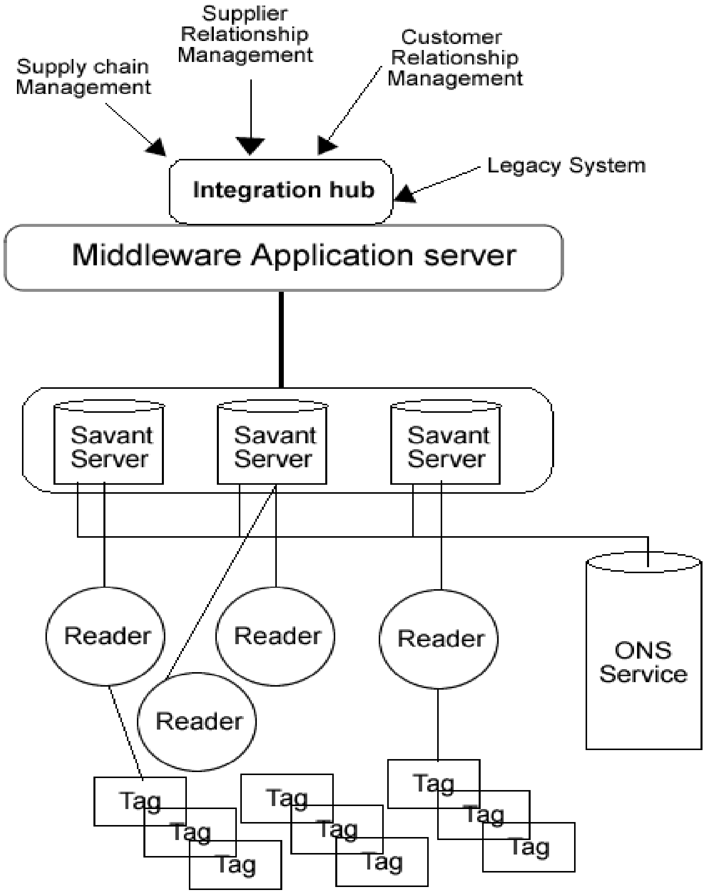

Figure-2.3: RFID Infrastructure

A savant server is a system that acts as a network gateway for tags, readers, ONS(Object Name Service) server and EPC middleware/application servers. It interacts with different types of readers, collects data then modulates into a standard format, monitors events and manages the tasks of getting, expanding, filtering, logging data, and requesting ONS service. ONS point savant server to the additional information corresponding to the EPC number stored on something called EPC Information Services. The information the savant collects for each tag includes EPC of tag read, EPC of the reader scanning the tag, time stamp of the reading, and temperature. One of the functions of the savant is to fix incorrect or duplicate data gathered from readers before it stores and forwards data to the ERP systems, including SCM and SRM via middleware technologies. The reader communicates with tags using RF, through one or more antennas. Because the reader must communicate with another device like savant server, the reader have a common network interface like serial Universal Asynchronous Receiver/Transmitters (URATs) for RS 232 and the RJ45 jack for 10BaseT or 100BaseT Ethernet cables; some readers even have Bluetooth or wireless Ethernet communications built in. Finally, to implement the communications protocols and control the transmitter, each reader must have microcontroller or a microcomputer. Within the reader, we can imagine four separate subsystems that handle different responsibilities. Figure 2.3.2 shows these logical components of a reader.

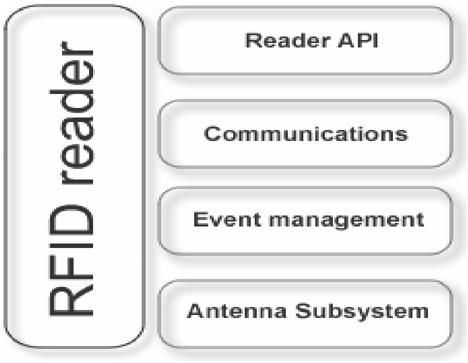

Figure-2.3.2: Logical Components of a Reader

2.4 RFID standard

It is generally understood that a technology cannot and will not be deployed pervasively and globally without a robust set of standard protocols specified between constituent entities. Two organizations are most involved in drafting standards for RFID technology: the International Organization for Standardization (ISO) and EPCglobal. ISO represents global interests and has been involved with different RFID technologies for many years.

2.4.1 ECP Standards

The Auto-ID Center at MIT has been driving towards development of a standard specification for item level tagging in the consumer goods industry called the Electronic Product Code (EPC). This has led to a new group called EPCglobal, a joint venture between the Uniform Code Council (UCC) and EAN International, which maintain the U.P.C./EAN bar code system among others. As stated in the name, a primary goal of EPCglobal is to make the final EPC standard an official global standard.

Another way of categorizing the RFID tags is by their capability to read and write data and their functionality. This leads to the following five classes as defined by EPCglobal as shown in Table-2.4.1.

Table-2.4.1 (a): EPCglobal class types

EPC class Type	**Features**	**Tag type**
Class 0	Read Only	Passive (64 bits)
Class 1	Write Once, read many (WORM)	Passive (96 bits)
Class 2 (Gen 2)	Read/Write	Passive (96 bits)
Class 3	Read/Write with battery power to enhance range	Semi-active
Class 4	Read/Write active transmitter	Active

The current thrust of EPCglobal is known as UHF Generation 2 (UHF Gen 2), a Write Once Read Many tag with more memory (96 bits vs. 64 bits) than preceding Class 0 and Class 1 tag. UHF Gen 2 will also provide a bridge to the eventual Class 2 High Memory full Read Write tag. Prior to UHF Gen 2, Class 0 and Class 1 were being utilized for EPC, but they were not interoperable.

EPC is a number made of a header and three sets of data. The header identifies the EPC's version number, allowing for different lengths or types of EPC later on.

- The second part of the number identifies the EPC Manager, most likely the manufacturer of the product.
- The third, called an object class, refers to the exact type of product, most often the Stock Keeping Unit (SKU).
- The fourth is the serial number unique to the item, which can tell us, for example, exactly to which 330 ml can of Diet Coke we are referring. This makes it possible to quickly find products that might be nearing their expiration date.

For example EPC: 02.223D2A4.64F16E.6F23CBA204

Table2.4.1 (b): EPC Frame Format

02	Version of EPC (8 bit header)
223D2A4	Manufacturer Identifier, 28 bits
64F16E	Product Identifier, 24 bits
6F23CBA204	Item Serial Number, 40 bits

2.4.2 ISO Standard

The International Organization for Standardization (ISO) is very active in developing RFID standards for supply chain operations and is nearing completion on multiple standards to identify items and different types of logistics containers. The ISO 18000 series is a set of proposed RFID specifications for item management ratified as standards during 2004. The series includes different specifications that cover all popular frequencies, including 135 KHz, 13.56 MHz, 860-930 MHz and 2.45 GHz. For most of the reference in this thesis, we follow EPC standards.

2.5 Multi-Reader System in Mobile Environment

This section describes the scenario of multiple readers in dense and mobile environments and presents benefits and difficulties in its deployments. When we say mobile, it refers to all possible cases; either readers are mobile, or tags or both readers and tags are in motion. Under such scenario, the ability to use multiple readers to read tags distributed all over the region will be more efficient and faster than by using single reader. There are numerous research papers proposing algorithm for various mobile environments. [19] proposed solution that involves multiple reader in a fixed readers fixed tags scenario or mobile tags scenario. Likewise [18], [14], [20], [21] are other papers proposing different deterministic and probabilistic algorithms for mobile environments using multiple readers. We will discuss in detail about these approaches in chapter 3. MOBILE-SCENARIO, the type under our consideration is a fixed multiple readers and mobile tags scenario in a dense environment.

2.6 RFID in Supply Chain

A supply chain with RFID technology is a global network of integration hubs [11] of suppliers and clients that create, track, and deliver RFID-tagged finished products manufactured from raw materials and semi-finished parts to multiple destinations from multiple supply sources. The process of attaching a tag to a supply item begins with entering the data into a tag via a computer and then attaching the tag to a product or a container. The tag must be positioned on the product or container so that it can be visible and at a certain distance between it and a reader. Each is assigned a unique identification number. When an items moves from manufacturer's site to inventory warehouse, following steps are taken place.

First, the item is coded with RFID tag, a radio-powered microchips that can broadcast EPC. As the item leaves the factory, reader picks up its EPC. In next step, the new location of the item is recorded in the EPC in the tag. Then, as the item is placed on a pallet one by one, a reader on the forklift picks up the EPC from the tag affixed to each unit. In the fourth step, the reader transmits the information to the warehouse manager confirming receipt of the pallet. A copy of this receipt is shown on both the warehouse manager's and manufacturer's screens. Next, the reader transmits serial numbers via Savant software that sends them to an ONS server. Once a Savant server receives EPC data, it can query an ONS server to find out where product information is stored on other servers on the Internet. This points the RFID software to a database confirming information about the product and the shipment's contents. A copy of the confirmation is shown on both the warehouse manager's and manufacturer's screens. Finally, inside the store, each time an item moves off the shelf, inventory is updated and both the warehouse manager and the manufacturer get the alerts on their screens. MOBILE SCENARIO is restricted to reading the tags moving on the conveyor belt to count the number of items moving out of store or coming into the inventory at any level of supply chain.

2.7 RFID Collision Problem

Simultaneous transmissions in RFID systems lead to collisions as the readers and tags typically operate on the same channel [13]. The *interrogation region* is the region around a reader where a tag can be successfully read in the absence of any collisions. The *interference region* is the region around a reader where the signal from the reader reaches with sufficient intensity so as to interfere with a tag response. There are three types of collisions.

2.7.1 Tag to Tag Collision

Tag collision in RFID systems happens when multiple tags are energized by the RFID tag reader simultaneously, and reflect their respective signals back to the reader at the same time. This problem is often seen whenever a large volume of tags must be read together in the same RF field. Figure-2.7.1 shows the tag-to-tag collision. The reader is unable to differentiate these signals; tag collision confuses the reader. So for the smooth tag-reader communication in multi-tag environment, we need an appropriate link-layer protocol such as framed Aloha [15] or tree-splitting [16], [17]. We describe these protocols in chapter 3.

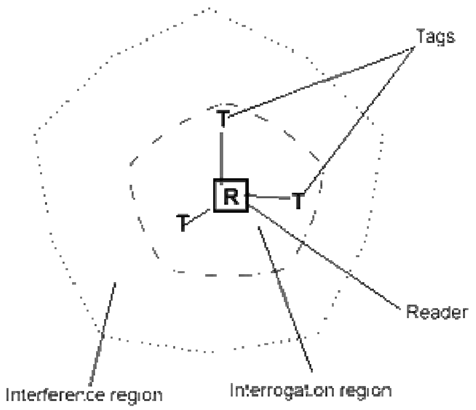

Figure-2.7.1: Tag-to-Tag Collision

2.7.2 Reader to Tag Collision

Reader-tag collision occurs when a reader or tag is in the interference region of another reader. In figure-2.7.2, interference from reader X can distort the signal from a tag T targeted for Reader Y causing Reader-to-Tag Collision. Such collision can be avoided by assigning different channels to near-by readers [18], or by scheduling the near-by readers to be active at different times.

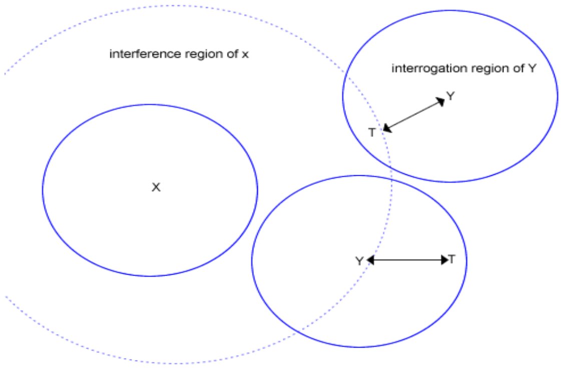

Figure-2.7.2: Reader-to-Tag Collision

2.7.3 Reader to Reader Collision

This happens when two readers with overlapping interrogation regions are active at the same time. In such a case, the tags in the overlapped region can not differentiate between the two signals, so might be unable to response to any reader at all, causing Reader-to-Reader collision. Figure-2.7.3 shows such collision. Using the different channels may not solve the collision problem as in the case of reader-to-tag collision since tag may not be capable of responding multiple readers with different frequencies at the same time. Hence only the way to avoid such collision is to activate the readers at different time.

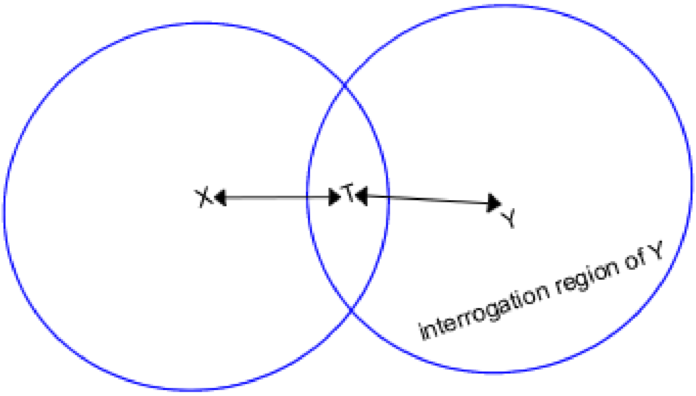

Figure-2.7.3: Reader-to-Reader Collision

CHAPTER 3

LITERATURE REVIEW

The anti-collision protocols of RFID systems are constrained by low computational capability and small memory, although they are similar to the classical multi-access communication systems. The limited power supply memory and computing capability of low-cost RFID tags rule out the use of complicated anti-collision algorithms. In addition, low-cost tags are not able to sense the medium, so the use of CSMA is also not possible. In this chapter, we will describe various anti-collision protocols that have been proposed in different research papers for solving collision problem in RFID system. Although, no papers propose a solution for the exact scenario we are considering, it is worthy to review such papers as they deal mostly with the collision problems encountered in this research.

The goal of every protocol has been to read tags efficiently. These protocols are deployed at the lowest portion of the data link layer which we call MAC sub-layer designed as **MAC protocols**. In a tag, the primary concern is passing of data over the communication medium and the detection, and optional correction of errors that may occur during data transmission, whereas in a Reader, besides handling communication, the MAC protocol also has to address the reader collision problem.

3.1. Multi-Tag Anti-Collision Algorithm

When multiple tags transmit their IDs at the same time, the tag-reader signals lead to collisions. Such collisions make both the communication overhead and the transmission delay of readers and tags often losing their usefulness. Therefore we use anti-collision algorithms to mitigate the collision affect. The majority of RFID multi-tag anti-collision protocols are time-domain based of either deterministic type or stochastic type Deterministic schemes are variants of binary search algorithms [23] which is a de-facto standard for RFID anti-collision protocols where the root-to-leaf path represents a unique tag ID. Typical polling schemes can be time exhaustive if there are many tags under the reader's interrogation area. Moreover, the length and distribution of tag IDs can affect the identification time significantly. The probabilistic approach is based on the slotted ALOHA mechanism where the channel time is split into frames and each frame again into time slots. In RFID system, all the tags are synchronized with the reader under whose interrogation they are in. All the tags have a local clock for synchronization. Here tags respond to reader's interrogation at randomly chosen time slots. A time slot is a time interval in which tags transmit their serial number or detailed information saved in them. A read cycle is a tag identifying process that consists of a frame. Figure-3.1 depicts the typical frame structure. Each tag transmits its serial number to the reader in a slot of a frame, and the reader identifies the tag when a time slot is used by one tag only.

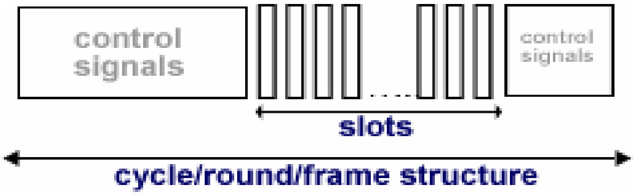

Figure-3.1: Typical Frame Structure [1]

3.1.1. Binary Search Tree Protocol

A binary search tree scanning anti-collision protocol is an implementation of a "reader talks first" methodology [23]. These protocols can have a longer identification delay than Slotted Aloha based ones, but they are able to avoid the so called tag starvation, in which a tag may not be identified for a long time when involved in repeated collisions. Here, a reader first performs identification by recursively splitting the set of answering tags. A reader queries all tags for the next bit of their IDs. After each tag transmission, the reader notifies the outcome of the query: collision, identification, or no-answer. When a collision occurs, the reader splits the queries until only one tag responds. The total number of tags in the interrogation region of a reader can be represented as a binary tree as shown in a figure-3.1.1. Most significant bit (MSB) of an EPC code of any tags starts from top excluding root to the Least Significant Bit (LSB) at the leaf. In other words, all the leaves represent tags whose EPC code can be discover simply by traversing right for 1 and left for 0 from root to leaf. The figure also shows a unique path through the tree that discovers the EPC of a particular product.

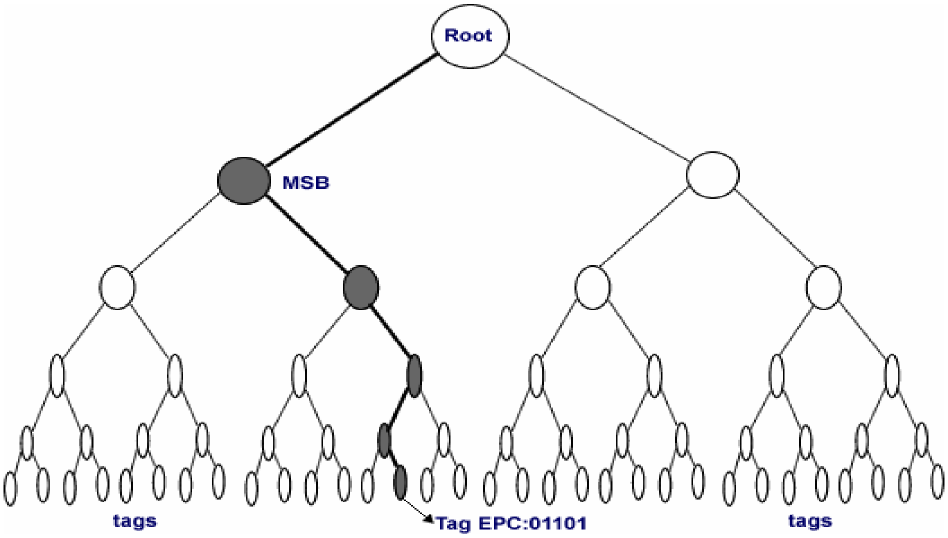

Figure-3.1.1: Binary Search Tree-based Protocol

Among tree-based protocol is a Query-Tree protocol which is an efficient memoryless scheme. In this protocol [25], the reader sends a prefix and only those tags having matching prefix ID answer. If there is a collision, the reader queries for one bit longer prefix until no collision occurs. Once a tag is identified, the reader starts a new round of queries with another prefix. The polling efficiency for this protocol is low when the tag population size is large or the ID address distribution is sparse.

3.1.2. Basic Framed Slotted Aloha

The basic Framed Slotted Aloha protocol [27] uses a fixed frame size and does not change the size during the process of tag identification. Frame size means number of slots consisted in each read cycle. In this protocol, the reader transmits to tags the frame size and each tag randomly selects the number from 0 to Q-1, where Q is the number of slots available in a frame. Then the tag transmits the information on mth slot of the frame. If more then one tag chooses same slot, the collision occurs and reader is unable to identify the both tags. There is higher number of collision with fixed frame size when there are too many tags in the region than the number of slots available. Also, if the size of the frame is too large compare the number of tags in the region, most of the slots remain idle and wasted. Hence, the read throughput depends highly on the frame size of the read cycle.

3.1.3. Dynamic Frame Slotted Aloha

After the importance of frame size in Framed Slotted Aloha protocol was realized, Dynamic Frame Slotted Aloha (DFSA) is a technique that intelligently and dynamically decides the frame size for every round in the tag identification process. Other features of

DFSA are similar to Basic Frame Slotted Aloha. To adjust the frame size, it uses statistical data such as successful slots, collision slots, and idle slots in the previous round. One way to deploy DFSA [9] is to increase the frame size if the number of collision slots reaches over the upper threshold and decrease the frame size if number of collision slots is less then lower threshold. Another way of implementing DFSA is to increase the frame size exponentially until a tag is read successfully. The process starts with the minimum frame size on next read cycle every time a successful reading occurs.
Therefore, regulating the frame size according to the number of the tags in the region is more efficient then Basic Framed Slotted Aloha. However, it is not possible to increase the frame size indefinitely and it is not possible to achieve desire performance if the number of tags is much higher than the optimal frame size.

3.1.4. Enhanced Dynamic Frame Slotted Aloha

This protocol [8] has been proposed to overcome the problem of DFSA where the number of tags is very large. After estimating the total number of unidentified tags, EDFSA divides the unread tags into a number of groups. The number of groups depends on the maximum frame size and tag population. If N is the maximum frame size and K is the number of unread tags estimated, the total tags are divided into M groups, where $M=\frac{K}{N}$. The size of the first frame is always set to 128 and then size of the next frames is set according to Table-3.1.4. The reader transmits the number of tag groups whenever it sends a request. The tags then carry out a modulo operation on a random number that they generate with the group size advertised by the reader. The tags with a zero remainder participate in that round. The tags generate a second random number to decide which the slot number they transmit their data. The reader will decrease the frame size if the estimated unread tags are below a certain threshold. Then the next group will be read provided all the tags in the former group are read thoroughly [6]. The performance of EDFSA algorithm is better than the DFSA algorithm because it restricts the maximum frame size whenever there are too many tags in the scene.

Table-3.1.4: EDFSA unread tags vs. optimal frame size [8]

Estimated unread tags (K_i)	Frame size (N_i)	Groups
1-11	8	1
12-19	16	1
20-40	32	1
41-81	64	1
82-176	128	1
177-354	256	1
355-707	256	2
708-1416	256	4
…	…	…

3.1.5. Accelerated Framed Slotted ALOHA (AFSA)

This is a novel framework proposed in [1] for reducing the average reading time of RFID tags in dense environment. The major improvement in this framework over the Enhanced Dynamic Frame Slotted Aloha and other ALOHA protocols is its ability of avoiding unnecessary bit times due to collisions and idles slots. During reservation for the slots by the tags, they generate two kinds of random numbers: first random number to select the slot and another is *n*-bits sequence number which is transmitted to reader at the slot position indicated by the first random number. Then, reader broadcast the reservation summary message in bitmaps from which the tag discovers the successful and unsuccessful slots. Then accordingly if it had successful reservation, it transmits its EPC information to the reader in appreciates slot number avoiding the broadcasting of wasted bit times due to collision and idle slots. In [1], we are, however, interested more on its solution proposed for mobile tags passing through the readers' range as in the case of assembly lines using single reader. Following is the summary from the paper for the derivation of maximum number of tags, G_i, that can participate in a round for the assurance of P reading rate, with assumption of average probability of tag being read, $P_r = 0.368$ in a round.

Let's q be the number of rounds required to guarantee P percentage of overall reading rate of the system. Then following condition holds:

$$(1-P_r)^q \le (1-P)$$

Or,
$$q = \frac{\log(1-P)}{\log(1-P_r)} \tag{1}$$

Also,
$$T = 537.5 + 199.73G_i \tag{2}$$

Where, T is a round time of the system and if *available_time* be the total time that a tag gets to pass through the active region of the reader, then using (1) and (2) we can derived following equation for calculating G_i:

$$\frac{available\_time}{q+1} = 537.5 + 199.5G_i \tag{3}$$

Later, in our proposed solution, we will compare this G_i, with the maximum number of tags that our proposed model provides for same desired reading rate P.

3.2. Multi-Reader Anti-Collision Algorithm

The large scale deployment of RFID technology has necessitated the use of multiple readers together for improving read rate and correctness. However, the tradeoff which is frequent and inevitable is reader collision problems. The anti-collision algorithms can be basically categorized under TDMA (Time Division Multiple Access) and FDMA (Frequency Division Multiple Access). Following Anti-Collision Algorithms are the countermeasures used to mitigate the reader collision problems. Several MAC layer protocols have been proposed in various research papers.

3.2.1. Colorwave

Colorwave [5] is one of the best and the first paper to address reader collisions considering a single available channel. The proposed algorithm formulated and solved the reader collision problem using a graph-based approach under TDMA. The basic idea was to randomly assign a color to the readers such that each pair of interfering readers has different colors. If each color represents a time slot, then this coloring scheme should eliminate reader-to-reader as well as reader-to-tag collision problem. If two interfering readers happen to pick the same color, then one of them again pick some other color randomly and next reader stick to the same. Colorwave concept works as following two ways.

- Distributed Color Selection (DCS)

If the transmission collides with another reader, the transmission request is discarded. Further, the reader randomly chooses a new color and reserves this color, causing all of its neighbors to select a new color, theoretically clearing the timeslot for the next time the reader needs to transmit.

  - DCS Subroutine 1 – Transmission
    - If transmission requested
      - If (timeslot_ID%max_color) = = Current_color, then transmit
      - Else idle until (timeslot_ID%max_color) = = current_color
  - DCS Subroutine 2 – Collision
    - If attempted transmission but experienced collision
      - Current_color = random(max_color)
      - Broadcast kick stating new color
  - DCS Subroutine 3 – Kick resolution
    - If kick received stating current_color
      - Randomly change to different color within max_colors.

- Variable-maximum DCS (Colorwave)

  Each reader monitors the percentage of successful transmission and uses as a criterion for varying the max color of it.

  - Colorwave Subroutine 1 – Color Change

    If collision percentage is past *SAFE threshold* AND time spent in current *max_color* exceeds *min_time* threshold, change *max_color* up or down by one respectively.

- Colorwave Subroutine 2 – Kick Resolution
  - If kick received stating *current_color*, change to random color within *max_color* other than current color.
  - If kick received stating change to new *max_color* AND collision percentage is past *TRIGGER threshold* AND time spent in current *max_color* exceeds *min_time* threshold, change *max_color* to kicked value and initiate kick to new *max_color* for next interation
- All DCS subroutines are also in use in VDCS.

Similarly [4] suggests coloring the interference graph using *c* colors, where *c* is the number of available channels. This approach eliminates the reader-to-tag collision but unable to work for reader-to-reader collision.

3.2.2. Centralized Approximate Algorithm

The solution based on Centralized Approximate Algorithm yields static, globally controlled allocations of frequency over the time. In general [3], unlike the graph-based algorithm, before reader-tag communication take place, a reader i sends a communication request r_j to the centralized controller at time t_j. The request r_j requests the use of a communication channel for the specified duration p_i. The centralized controller either accepts or rejects the request immediately. The reader uses that channel for the requested duration, and then relinquishes it for use by other readers. Requests are accepted such that no reader collisions are experienced by any reader. Readers whose requests are rejected may resubmit their requests after a random amount of waiting time. Such deterministic allocations of frequency over time however may not guarantee the optimality.

3.2.3. HiQ Learning Algorithm.

This is a hierarchical online learning algorithm [2] that finds dynamic solutions to the reader collision problem. For a given network of readers and communication pattern, HiQ learns the collision patterns of the readers during the training period then when new request comes in; it yields an optimized resource (channel and time slot) allocation scheme by effectively assigning frequencies over time. The training process determines the channel and time slot to allocate to a reader for any new request. This work considers both reader-to-reader and reader-to-tag collisions, but assumes that readers can directly communicate with each other. Moreover, they assume a fixed number of time slots, and aim at maximizing the frequency and time utilization ratio rather than the more practically important metric of total reading time. Finally, this algorithm does not provide any performance guarantee.

CHAPTER 4

PROPOSED SOLUTION

4.1. System Model

The system model that has been proposed for this thesis uses Time Division Multiple Access (TDMA) technique to regulate the operation of multiple readers. Then under each duty cycle we carefully choose a group of readers to operate simultaneously. An offline algorithm will be designed which will find the optimal number of tags that could be placed onto the conveyor belt for achieving required read throughput. Furthermore, each reader-tag communication would be regulated by Enhanced Dynamic Framed Slotted Aloha TDMA algorithm [8]. The assumptions made for such model and its characteristics under the MOBILE-SCENARIO are listed as follows.

- This is a 900 MHz UHF RFID system compatible to EPCglobal Class 1 Gen 2 standard that works in the 860 MHz – 960 MHz UHF band.
- This model uses a single channel for reader-tag communication.
- The controller handles the coordination between operations of readers as well as collects the information gathered by the readers.
- System considered six identical readers aligned to each other in a straight line fixed above the conveyor which is revolving with the speed of ***v*** meter/sec. Each reader is at the height of ***h*** meter from the conveyor belt. Reader uses directional antenna of equal strength focused vertically towards conveyor belt with maximum

- diameter of l_{max} meter. Readers are synchronized and are able to communicate with each other. Reader uses ‘0’, ‘1’ and ‘Null” for communicating with tags; 0 and 1 for constructing commands while Null for making the beginning and end of command and also to specify the end of a slot within a frame.
- Tag is affixed onto the item to be identified which moves on the conveyor belt. Tag transmits its EPC (64 bits) along with CRC (16 bits) with symbol duration of 4 *μsec* using backscatter modulation technique. We considered two types of model for the tags flow on the conveyer belts as depicted in Figure-4.1(a) and Figure-4.1(b):
  - Distributed Model: Allowing continuous flow of tags over the conveyor in a distributed way.
  - Clumped Model: Allowing discrete flow of tags in a group at one time.

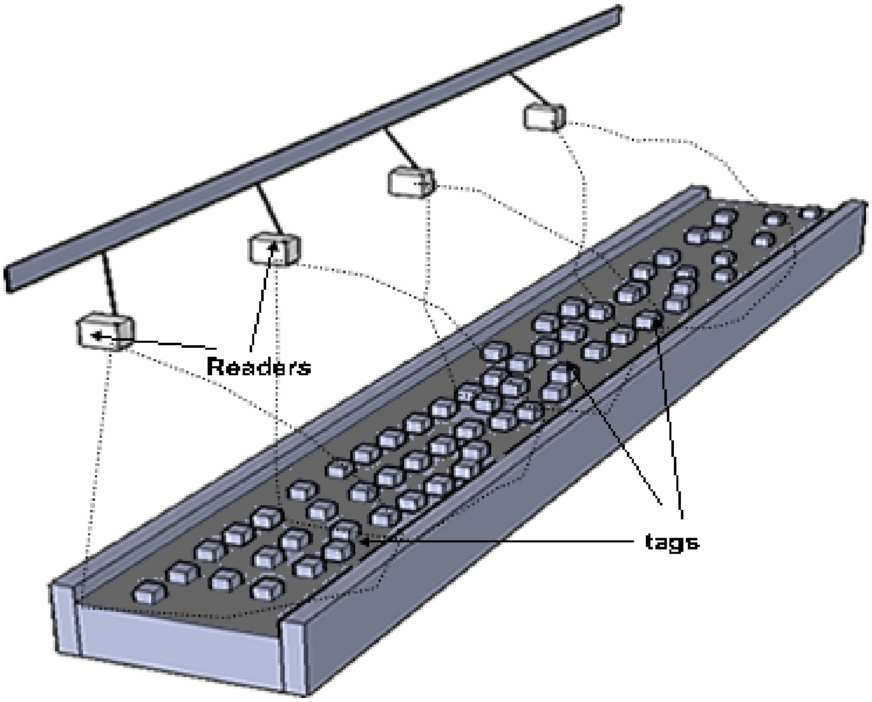

Figure-4.1(a): Distributed Model Structure

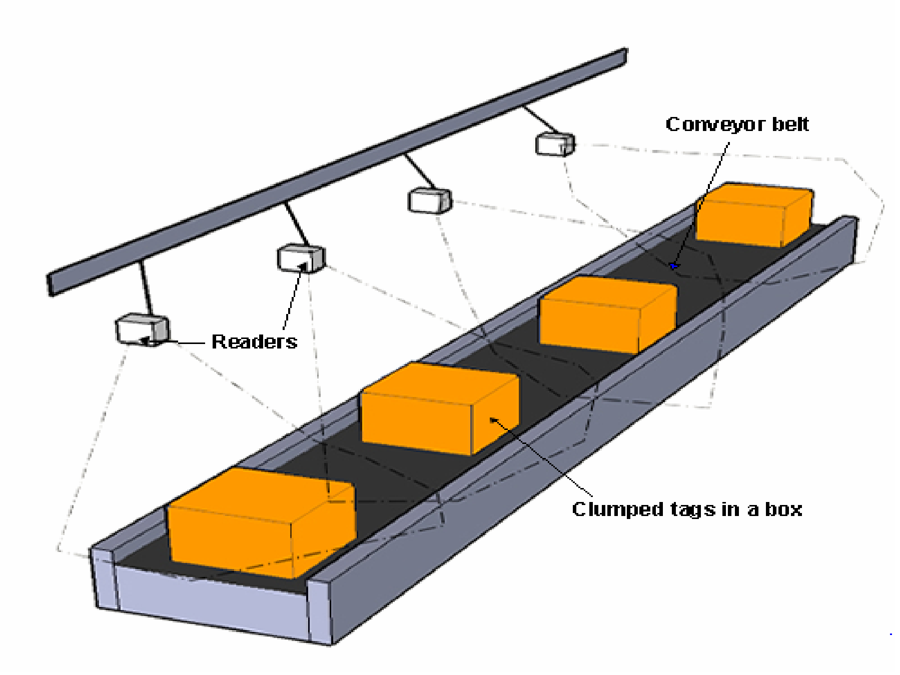

Figure-4.1(b): Clumped Model Structure

4.2. Reader-Reader Coordination

Since the readers in MOBILE-SCENARIO are in a linear topology aligned as close to each other as possible, they suffer from reader collision problem. Such collisions are mostly due to adjacent readers as their interrogation region overlaps. Therefore, the coordination between the readers is inevitable for the effective operation of multi-reader RFID system.

4.3. Reader-Tag Communication

Readers are the first to start communication with tags. The reader-tag communication take place in round structure which we call read cycle as shown in Figure-5.2(a) which is similar to the frame structure of [1]. Basically reader-tag communication proceeds in five phases each separated by a null signal, which transform a tag from one state to another as shown in state diagram figure-5.2(b). Before every round begins, the reader broadcasts the reset and calibration signals that actives and synchronize the tags under its region. This brings tags into active state from inactive state.

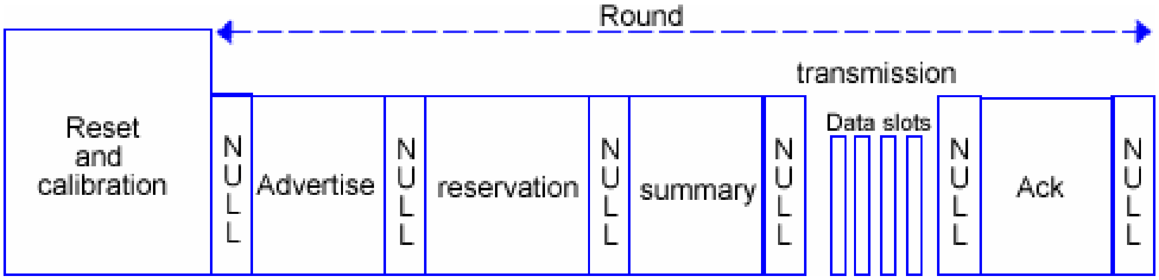

Figure-4.2(a): Round Structure of a frame[1]

The first phase of tag-reader communication is an advertisement phase where reader transmits the frame size *N*, which is equal to number of data slots that reader is supposed to allocate later. The frame size *N* and number of tags that participates in each round is estimated using the algorithm as proposed in [8]. In the reservation phase, a group of tags which are active and unread generate a random number in the range of 1 to N and transmits to the reader signifying their selection of corresponding slot. In the summary phase, the reader broadcasts the confirmation for a successful reservation. The reader considers the successful reservation for the slot if only one tag has selected its corresponding position. Any two tags selecting the same random number is considered a collision and discarded. The next important phase is the transmission phase. Once the tags get confirmation of their reservation, they move to the transmit state and start transmitting their EPC along with CRC bits. In the final phase of tag-reader communication, the reader broadcasts the signal to acknowledge receipt of the data transmitted by the tags. “1” signifies the successful reception of data at the corresponding slot which converts the tag from transmit state to muted state and “0” indicates failure to receive correctly and therefore the tags remain in the active state. The muted tags do not participate in any future tag-reader communication.

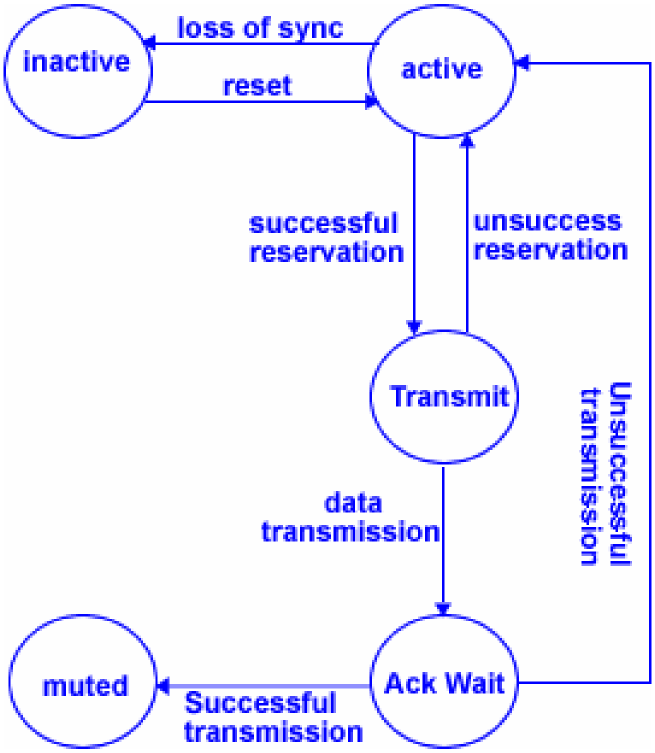

Figure-4.2(b): State Diagram of Tags

4.4. Solution Procedure

After the explanation of how reader-reader communication and reader-tag communication take place, we would like to list out the procedure for realizing the proposed model as follow. And later, this procedure will be used to build the simulation.

a. Find the number of groups (M) of readers so that all the readers in each can be operated simultaneously. Several algorithms [19] exist for solving this problem.

Because of the topology (linear) of the reader's network of our system, the simplest way is to group the readers into 2 where each group is formed by the readers that are not adjacent to each other. In our case, if *R1, R2, R3, R4, R5,* and *R6* be the 6 consecutively readers, then…

Group A = {*R1, R3, R5*} and Group B = (*R2, R4, R6*}

b. Deploy Time Division Multiple Access (TDMA) among M groups. Each group of reader will operate for $\delta = \frac{1}{M}$ duty cycle. In our case, $\delta = \frac{1}{2}$.
c. Operate reader-tag communication between the readers in an active group and the tags moving under their respective interrogation regions on conveyor belt. The communication procedure should be followed as explained above. The communication between each reader and the tags are independent to another reader's communication.
d. Handle the coordination between groups of reader as well as readers in a group by the controller. Note readers in each group are synchronized to each other.
e. Design and Simulate the model for obtaining different G_i for various value of desired reading rate P under given value of Γ. Simulation should be done for both tag flow models: clumped and distributed model
f. Compare and analyze the result from the simulation with the calculated value of G_i as derived in (3) as well as other available values from [1].

Now, for the answer to the questions listed in the thesis goal section, let us consider Γ be the overlapping region between any two adjacent readers' interrogation region and δ be the duty cycle to be deployed when TDMA is implemented. Then, to find the condition that guarantee enhanced performance of the system deploying duty cycle than those without a duty cycle, we know,

The time each tag gets to spend under each reader, $t = \frac{2\sqrt{l_{\max}^2 - h^2}}{v}$, where v is the velocity of belt, h is height of the reader and l_{max} is the maximum read diameter of the reader. Then, the total time a tag spends in non-interfering region without duty cycle $t_{ndc} = \sum_{i=1}^{n} t_i - 2(n-1)\Gamma$, where n is the number of readers in the system.

The total time tag spend in non-interfering region with duty cycle $t_{dc} = \sum_{i=1}^{n} \delta_i t_i$

or, $t_{dc} = \delta \sum_{i=1}^{n} t_i$ since δ is same for all readers.

Now, to guarantee duty cycle implementation provides better performance, following condition most be satisfied.

$$t_{dc} > t_{ndc}$$

or, $$\delta \sum_{i=1}^{n} t_i > \sum_{i=1}^{n} t_i - 2(n-1)\Gamma$$

or, $$\Gamma > \frac{1}{2(n-1)}(1-\delta)\sum_{i=1}^{n} t_i$$ or, $$\Gamma > \frac{n}{2(n-1)}(1-\delta)t,$$ assuming the same time amount t under each reader.

Hence this expression guarantees that implementing a duty cycle can provide better performance then without duty cycle if Γ is greater than $\frac{n}{2(n-1)}(1-\delta)t$. This answered the first question of thesis goal. For answering the remaining two questions, we rely on the simulation result which we discuss in the next chapter.

CHAPTER 5

SIMULATION RESULTS

A program has been built in Java for simulating the clump and distributed model in a controlled environment. The source code is listed in appendix section. The reader-tag communication is based on the Enhanced Dynamic Framed Slotted Aloha (EDFSA) [8]. For clump model simulation, we created 6 readers aligned to each other and arranged alternative reader into two groups, each with three readers. We fixed the system parameters as follows h = 1 meter, v = 5 meters/sec and l_{max} = 2 meters. We calculated the time a tag takes to cross one reader as, t = 0.6928 sec. We assigned a duty cycle δ = ½ for both group of readers. Then we carefully set up the synchronized flow for a clump containing G_i number of tags on a conveyor belt such a way that the clump will always be under the active group of readers. So in every switching of the reader group, the new clump of tags enters into the system from reader 1, and another clump of tags moves from one reader to next reader's interrogation zone. Each group of tags or tag receive a maximum of $6*(t-\Gamma)$ amount of available time for tag-reader communication. Once, the system is stable, in every switching of reader group, one clump of tags leaves the system from reader 6 as we have setup the arrival and departure of clump with same rate.

Similarly, we had a precise arrangement for the simulation of distributed model of the mobile tag reading system in multi-reader environment. The individual tags as objects continuously enter into the conveyor belt from reader 1 at specific rate such that G_i

would be the average number of tags under each reader. Since the duty cycle for both groups of readers are same, then the maximum available time for the tag reader communication is *3*t*.

Table-5.1 depicts the result of the simulation showing the maximum number of tags possible in a clumped and distributed model where the desired overall percentage reading rate as listed in the first column. Each of these values has been considered as an average of 100 simulated values. The table shows the simulation results for two different values of Γ: 0.173 and 0.3462 second.

Table-5.1: Simulation result comparison table

Reading Rate P%	Calculated Gi		Simulated Gi	
	Clumped	Distributed	Clumped	Distributed
Γ = 0.173				
97.00	1733	1154	2265	1463
98.00	1559	1038	2187	1381
99.00	1299	865	2040	1221
99.90	916	610	1630	917
99.99	708	471	1462	864
Γ = 0.3462				
97.00	1154	1154	1451	1463
98.00	1038	1038	1391	1381
99.00	865	865	1239	1221
99.90	610	610	921	917
99.99	471	471	864	864

Similarly, the table shows the calculated values of *Gi* for both models using (3). In both model, for calculated values, the average read probability was 36.8% in a round. However, in our model, we relied on heuristic approach where tags reserve the slots by generating the random numbers and multiple communications takes place before tag can actually sends the actual data to the reader. So the successful communication was fully probabilistic or random. We observed that the average read probability in a round tends to varies from 27% to 52%. Figure-5.1 depicts the chart showing the variation of G_i with respect to desire reading rate for both clumped and distributed model when $\Gamma = 0.173$.

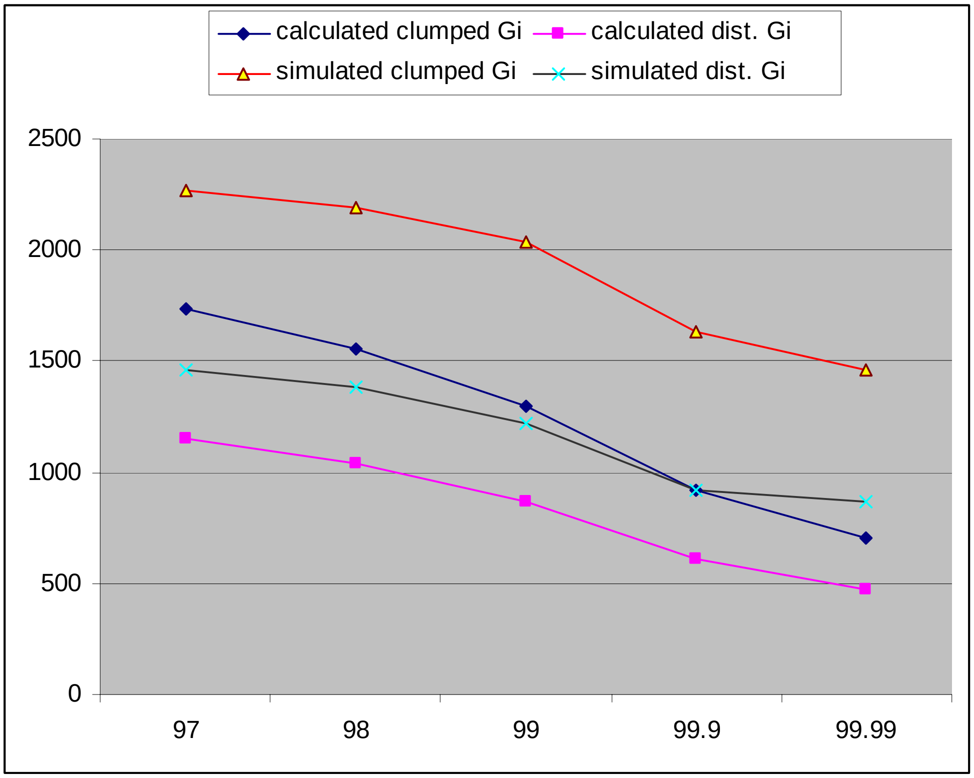

Figure-5.1: comparison chart for *Gi* vs. reading rate

Thus, the results of the simulation and comparison with calculated values shows improvement of read throughput in our proposed model as the observed values for both clumped and distributed model are greater than or equal to the respective calculated values. This simulation result also verifies that clumped model is much better option for the flow of tags on the conveyor belt providing higher throughput than distributed model when Γ = 0.173sec.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This thesis started with an objective of proposing an efficient framework for reading mobile tags in a dense multi-reader environment. We considered a practical scenario in supply chain management where the tags move continuously on a conveyor belt for getting read by the readers that are aligned and fixed above the conveyor belt. We proposed time slotted model and implemented Enhanced Dynamic Framed Slotted Aloha algorithm for tag-reader communication and deployed Time Division Multiple Access (TDMA) among readers for reader-reader communication. This implementation reduced the collision problems and was able to provide the reading probability that tends to varies from 27% to 52% in a round. We derived the expression in section 4.4 to show, for what value of Γ does the system guarantee the better performance compare to system without time slotted model, i.e. all readers operating simultaneously all the time. We had 6 readers aligned to each other for simplicity, grouped into two, each containing 3 which were not adjacent to each other. This assured the elimination of reader collision in the system. Since both groups of reader are identical, we assigned equal duty cycle, i.e. $\delta = ½$. Then, using simulation, we proved that the clumped model can provide better read throughput then distributed model. Hence, we concluded that proposed model is more efficient for reading the mobile tags in multi-reader dense environment.

Regarding future work, in this thesis, only a single channel was considered for the communication between readers and tags. Today's readers and tags are capable of communicating using more then one channel simultaneously. So, in future, we can work on a multi-reader system using multiple channels. Another possibility might be working with different number of readers with various topologies to find out which setup would be best for higher throughput.

REFERENCES

[1] Malla Reddy Devarapalli, Venkatesh Sarangan, and Sridhar Radhakrishnan. *AFSA: An Efficient Framework for Fast RFID Tag Reading in Dense Environments*, ACM/IEEE QShine 2007, Vancouver, August 2007.

[2] J. Ho, D. Engels, and S. Sarma. *HiQ: A hierarchical q-learning algorithm to solve the reader collision problem* in SAINTW, 2006.

[3] Daniel W. Engels and Sanjay E. Sarma. *The reader collision problem i*n Proc. of IEEE International Conference on Systems, Man, and Cybernetics, October 2002.

[4] V. Deolalikar et al*.* *Perturbative time and frequency allocations for RFID reader networks*. Technical report, HP Labs, 2005.

[5] J. Waldrop, D. Engels, and S. Sarma. *Colorwave: A MAC for RFID reader networks* in Proc. of WCNC, 2003.

[6] M. A. Bonuccelli, F. Lonetti and F. Martelli*, Instant Collision Resolution for Tag Identification in RFID Networks,* in Ad Hoc Networks (Elsevier).

[7] Vogt, H. 2002. *Efficient Object Identification with Passive RFID Tag,* in Proc. of the First international Conference on Pervasive Computing (August 26 - 28, 2002). F. Mattern and M. Naghshineh, Eds. Lecture Notes in Computer Science, vol. 2414. Springer-Verlag, London, 98-113.

[8] Su-Ryun Lee,Sung-Don Joo,Chae-Woo Lee. *An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag*, in Proc. of the Second Annual

International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05).2005.

[9] J. R. Cha, J.H. Kim. *Dynamic framed slotted aloha algorithms using fast tag estimation method for RFID systems*, IEEE Communications Society, pp. 768–772, 2006.

[10] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*, John Wiley, New York, 2000, pp. 200–219.

[11] Judith M. Myerson, *RFID in Supply chain, A guide to selection and Implementation*; Auerbach Publications 2007

[12] V. Daniel Hunt, Albert Puglia, Mike Puglia, RFID: A Guide to Radio Frequency Identification. John Wiley, New York, 2007

[13] Passive RFID basics, 1998. http://ww1.microchip.com/downloads/en/AppNotes/00680b.pdf.

[14] Jain, S. and Das, S. R. 2006. *Collision avoidance in a dense RFID network* in Proceedings of the 1st international Workshop on Wireless Network Testbeds, Experimental Evaluation &Amp; WiNTECH '06. ACM, New York, NY, 49-56.

[15] H. Vogt. *Efficient object identification with passive RFID tags* in Proc. of the Intl. Conf. on Pervasive Computing, 2002.

[16] D. Hush and C. Wood. *Analysis of tree algorithms for RFID arbitration* in Proc. of ISIT, 1998.

[17] C. Law, K. Lee, and K. Siu. *Efficient memoryless protocol for tag identification* in Intl. Wksp. on Dis. Algos. and Methods, 2000.

[18] J. Waldrop, D. Engels, and S. Sarma. *Colorwave: A MAC for RFID reader networks* in Proc. of WCNC, 2003.

[19] Zongheng Zhou, Himanshu Gupta, Samir Das, Xianjin Zhu, *Slotted Scheduled Tag Access in Multi-Reader RFID Systems*. Proc. of the IEEE International Conference on Network Protocols (ICNP'07), 61-70, October, 2007.

[20] J. Ho, D. Engels, and S. Sarma. *HiQ: A hierarchical q-learning algorithmto solve the reader collision problem,* in SAINTW, 2006.

[21] Zhou, S.J., Luo, Z., Wong, E.C., Tan, C.J. (2007). *Interconnected RFID Reader Collision Model and its Application in Reader Anti-collision*. IEEE RFID 2007, Texas, USA 2007.

[22] Jihoon Myung and Wonjun Lee, *Adaptive Splitting Protocols for RFID Tag Collision Arbitration* , in MOBIHOC 2006, (Florence, Italy), May 2006.

[23] [1] D. R. Hush and C. Wood*, "Analysis of tree algorithms for RFID arbitration,"* IEEE Inter. Sym. On Information Theory, 1998.

[24] M.A.-I. Center, *Draft protocol specification for a 900 MHz class 0 radio frequency identification tag*, http://www.epcglobalinc.org (February 23rd, 2003).

[25] C. Law, K. Lee, and K. Siu, *Efficient memoryless protocol for tag identification*, in Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, August 2000.

[26] ISO/IEC 18000-6: 2003(E), Part 6: *Parameters for air interface communications at 860-960 MHz*, Nov. 26, 2003.

[27] F. C. Schoute, *Dynamic frame length ALOHA*, IEEE Transactions on Communications, vol. 31, no. 4, pp. 565-568, 1983.

APPENDIX

1. Reader.java

```
import java.util.*;

public class Reader {
    public int id;
    public Vector<Tag> tgroup;
    public int N;
    public int num_grp=1;
    public int Ng; //number of tags participating in each round.
    public int gi; //no of remaining unread tags
    public Vector<Integer> successfulSlots;
    public double prob=0.0;

    public Reader(int id)
    {
        this.id=id;
        this.successfulSlots=new Vector<Integer>();
    }
    public int getUnreadTag()
    {
        int tmp=0;
        if(this.tgroup!=null)
        {
            for(Tag t:this.tgroup)
            {
                if(t.state!=MainClass.MUTED)
                    tmp++;
            }
            return tmp;
        }
        else
            return 999999999;
    }
    //Advertise frame size by assign to the read.
    public boolean advertisePhase()
    {
        OutputWriter.printLine("Reader: "+ this.id);
        gi=this.getUnreadTag();//it is to find number of unread tag in a

        System.out.println("No of unread tags in Reader: "+this.id+"  ::"+gi);

        if(gi==0)
        {
```

```
OutputWriter.printLine("No unread tags left to read. All the tags in group has been read.");
            MainClass.wasterounds++;
            return false;
        }
        else if(gi>=1 && gi<=11)
        {
            this.N=8;
            this.num_grp=1;

        }
        else if(gi>=12 && gi<=19)
        {
            this.N=16;
            this.num_grp=1;

        }
        else if(gi>=20 && gi<=40)
        {
            this.N=32;
            this.num_grp=1;

        }
        else if(gi>=41 && gi<=81)
        {
            this.N=64;
            this.num_grp=1;

        }
        else if(gi>=82 && gi<=176)
        {
            this.N=128;
            this.num_grp=1;

        }
        else if(gi>=177 && gi<=354)
        {
            this.N=256;
            this.num_grp=1;

        }
        else if(gi>=355 && gi<=707)
        {
            this.N=256;
            this.num_grp=2;

        }
        else if(gi>=708 && gi<=1416)
        {
            this.N=256;
            this.num_grp=4;

        }
        else if(this.gi>=1417 && this.gi<=2831)
        {
            this.N=256;
            this.num_grp=8;
```

```
		}
		else
		{
			this.N=this.gi+1;
			this.num_grp=(int)((double)((this.gi+this.gi/2)/this.N));
		}
		this.Ng=gi/this.num_grp;
		return true;
	}

	//selecting the random_slots
	public void reservationPhase()
	{

		Integer cho;
		for(Tag t:this.tgroup)
		{
			if(t.state!=MainClass.MUTED)
			{
				t.chooseRandomSlot(this.N);
				cho=new Integer(t.rand_slot);
				if(this.successfulSlots.contains(cho))
				{
					this.successfulSlots.remove(cho);
					t.rand_slot=99999999;
				}
				else
					this.successfulSlots.add(cho);
				this.Ng--;
				if(this.Ng==0)
					break;
			}
		}
	}

	//Reservation Acknowledgement
	public void summaryReservationPhase()
	{

		for(int i=0;i<this.successfulSlots.size();i++)
		{
			for(Tag t:this.tgroup)
			{
				if(t.state!=MainClass.TRANSMIT)
				{
					if(t.rand_slot==this.successfulSlots.elementAt(i).intValue())
					{
						t.state=MainClass.TRANSMIT;
						break;
					}

				}

			}

		}
```

```
        OutputWriter.printLine("Reserved "+this.successfulSlots.size());
    }


    public void transmitNAckPhase()
    {

        for(Tag t:this.tgroup)
        {
            if(t.state!=MainClass.MUTED)
            {
                if(t.state==MainClass.TRANSMIT)
                {
                    t.state=MainClass.MUTED;
                    t.rand_slot=99999999;
                    MainClass.total_tag_read++;
                }

            }
        }

    }
}
```

2. Tag.java

```
public class Tag {
    public static final int MUTED=0;
    public static final int ACTIVE=1;
    public static final int TRANSMIT=2;
    public static final int INACTIVE=3;
    public int EPC;
    public int rand_slot=99999999;
    public int state;

    public Tag(int id)
    {
        this.EPC=id;
        this.state=MainClass.ACTIVE;
        this.rand_slot=9999;
    }

    public void chooseRandomSlot(int N)
    {
        this.rand_slot = (int)Math.ceil(Math.random()*N);
    }
}
```

3. Controller.java

```
import java.util.*;
public class Controller {

        public Reader[] groupA;
        public Reader[] groupB;
        public double timer; //need t, tau
        private double t;
        private double tau=0.173;
        private double T;
        //Defining the System parameters which we may change for different simulation.
        public static final int h =  1; //meter height of reader.
        public static final int Lmax = 2; //meter max. range.
        public static final int v = 5; // meter/sec, velocity of belt
        public Vector<Tag> clump;

        public Controller()
        {
                this.groupA=new Reader[3];
                for(int i=0;i<this.groupA.length;i++)
                        this.groupA[i]=new Reader(2*i+1);

                this.groupB=new Reader[3];
                for(int i=0;i<this.groupB.length;i++)
                        this.groupB[i]=new Reader(2*i+2);
                this.t=(2*Math.sqrt(Lmax*Lmax - h*h))/v;
                this.timer=t-tau;
                this.clump=new Vector<Tag>();
        }
        public void startClumpModel()
        {
                int f=0;

                while(MainClass.total_tag_read<MainClass.Gi+5000)
                {
                        this.timer=t-tau;
                        this.turnOn(this.groupA,this.timer);
                        this.timer=t-tau;
                        this.turnOn(this.groupB,this.timer);
                        f++;
                }


                System.out.println("====="+f);

        }

        public void turnOn(Reader[] rgrp,double timer)
        {
                if(rgrp[0].id==1) //just to identify group, its A if id==1
                {
                        for(int i=0;i<MainClass.Gi;i++)
                                this.clump.add(new Tag(MainClass.generateEPC()));
                        rgrp[0].tgroup=this.clump;
                        rgrp[1].tgroup=this.groupB[0].tgroup;
```

```
                        rgrp[2].tgroup=this.groupB[1].tgroup;
                }
                else
                {
                        rgrp[0].tgroup=this.groupA[0].tgroup;
                        rgrp[1].tgroup=this.groupA[1].tgroup;
                        rgrp[2].tgroup=this.groupA[2].tgroup;
                }
                while(this.timer>0.0)
                {
                        int ur=0;
                        int uread=0;
                        int deno=0;
                        for(Reader r:rgrp)
                        {
                                if(r.tgroup!=null)
                                {
                                        if(r.advertisePhase())
                                        {
                                                r.reservationPhase();
                                                r.summaryReservationPhase();
                                                r.transmitNAckPhase();

        r.prob=(double)r.successfulSlots.size()/(r.gi/r.num_grp);
                                                r.successfulSlots.removeAllElements();
                                        }
                                        else
                                                r.prob=0.0;
                                        ur=r.getUnreadTag();
                                        if(ur!=0)
                                        {       uread+=ur;
                                        deno++;
                                        }
                                }
                        }
                        //for probability.
                        for(Reader r:rgrp)
                        {
                                if(r.prob!=0.0)
                                {
                                        OutputWriter.printLine("Reader "+r.id+ "-------------------------
---------------- "+r.prob);
                                        MainClass.aveprob+=r.prob;
                                        MainClass.runds++;
                                }
                        }

                        MainClass.roundcount++; //total number of rounds each reader go gets.
                        if(deno!=0)
                        {
                                uread=uread/deno; //taking mean of unread tags.
                                this.T=(537.5 + 199.73*MainClass.Gi)/Math.pow(10,6);
                        }
                        OutputWriter.printLine("Round time::"+ this.T+"Ave unread number of
tags"+uread);
                        this.timer=this.timer-this.T; //if deno is 0, a previous T is used.
```

```
                }
        }
}
```

4. MainClass.java

```
public class MainClass {

        public static int total_tag_read=0;
        //no of rounds each reader will make before terminating the system.

        public static int roundcount=0;

        public static int runds=0;
        public static double aveprob=0.0;
        public static int wasterounds=0;
        //this is the max. number of round each tag may get.==qr.
        public static int Gi;
        public static int EPCbase=100;

        public static void main(String st[])
        {
                OutputWriter ohandler=new OutputWriter();
                MainClass.Gi=915;//Integer.parseInt(st[0]);

                Controller cntrl=new Controller();
                cntrl.startClumpModel();
                MainClass.displayResults();
                OutputWriter.close();

        }
        public static void displayResults()
        {
                OutputWriter.printLine("Ave Prob. "+MainClass.aveprob/MainClass.runds);
                OutputWriter.printLine(" No of rounds: "+MainClass.roundcount);
                OutputWriter.printLine("No of tags read: "+MainClass.total_tag_read);
                OutputWriter.printLine("wasted rounds "+MainClass.wasterounds+"  Useful rounds 
"+MainClass.runds);
        }
        public static int generateEPC()
        {
                MainClass.EPCbase=MainClass.EPCbase+1;
                return MainClass.EPCbase;
        }
}
```

VITA

Rupesh Bhochhibhoya

Candidate for the Degree of

Master of Science

Thesis: MOBILE TAG READING IN A MULTI-READER RFID ENVIRONMENT

Major Field: Computer Science

Biographical:

Personal Data: Born in November 8th in Kathmandu, Nepal, son of Mr. and Mrs. Rajaram Bhochhibhoya.

Education: Graduated from Pokhara University, Pokhara, Nepal in September 2005; received Bachelor of Engineering Degree in Information Technology. Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in July, 2008.

Experience: Graduate Teaching Lab Assistant (GTLA) at Linux Lab, Department of Physics, August 2006 – May 2008. Research Assistant, Department of Computer Science, May 2007 – May 2008.

Name: Rupesh Bhochhibhoya Date of Degree: July, 2008

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: MOBILE TAG READING IN A MULTI-READER RFID ENVIRONMENT

Pages in Study: 51 Candidate for the Degree of Master of Science

Scope and Method of Study: This study will contribute in supply chain for the large scale deployment of the RFID technology. Building models and simulation was the method of study used.

Major Field: Computer Science

Radio Frequency Identification (RFID) refers to an emerging technology that intends, but not limited to replace barcode technology. RFID system assures to provide an effective inventorying, tracking and monitoring of any sorts of products in any field of applications. Recently, the large scale deployment of RFID system in supply chain management has necessities the use of multiple readers. Unfortunately, multi-reader RFID system suffers from reader collision problems that severely affect the system performance. Hence, this thesis aims to propose a novel framework for multi-reader RFID system based on Framed Slotted ALOHA protocol. The proposed framework is specific to a scenario in supply chain where a tag affixed to an item is in motion that moves on a conveyor belt and multiple readers which are fixed around the conveyor belt are supposed to read all the items. This work also determines the best pattern for the distribution of the tags on the conveyor belt.

ADVISER’S APPROVAL: Dr. Venkatesh Sarangan