Library Tracking System Based on Radio Frequency Identification Technology

Ali Alwadi

Thesis submitted in partial fulfillment of the degree of the Master of Engineering

Auckland University of Technology

Auckland

New Zealand

October 2016

Acknowledgment

I wish to express my gratitude to my supervisor, Jeff Kilby for his patience, support, guidance and advice throughout this research. I also would like to thank Professor Amjad Gawanmeh from Khalifa University, Abu Dhabi, United Arab Emirates for his assistance during this research.

I would also like to thank my wife Haneen, my son Tamim, and my daughter Yasmin for their patience, and the long nights that they spent up supporting me during the work on this research. I would also like to dedicate this to my parents Abdulaziz Alwadi and Nawal Alwadi for their hard work raising my, my brothers and sisters, and also their continuous support.

Special thanks are also extended to Dr. Hakilo Sabit, and all staff of the School of Engineering for their kindness and encouragement during my study period at the Auckland University of Technology.

Abstract

This research evaluates the latest algorithms and available hardware that can be used to automate a library system using Radio Frequency Identification (RFID) technology as an infrastructure. This will allow the library management system (LMS) to track and manage real-life library utilizing the existing library resources.

The research provides an overview of the radio frequency identification technology, its history, radio frequency principle and how distance and frequency can affect the signal coverage. Then the radio frequency identification structure and types are explained, along with RFID infrastructure and middleware. The issue of radio signal collision is discussed, and the corresponding anti-collision algorithms are briefly explained.

The research then discusses implementing Location-Aware RFID service based on RFID technology as a communication technology. Which shows that automating a library system with passive RFID tag infrastructure is feasible and can be achieved with even distribution of antennas, and a well-designed RFID network. Selecting the appropriate middleware and a library application with an accurate error minimization function that minimizes the detected location error down to 35 cm.

The current library system architecture is discussed next, with the effect of using RFID on the upper application and system layers will also be explored. A simulation of random-distributed tags with different antenna range with the feasibility of adding a '*Location-Aware system layer component*' to identify the availability of each resource, also the physical location of each resource from the main computer.

RFID concept was utilized to build a secure implementation in which the library middleware application senses the availability of each resource in the network, locates that resource at runtime, and updates the library system interface with any dislocated items in the library. In order to prove the efficiency of the proposed solution, and the simplified design of the library system, the RFID system was simulated first using Wireless Sensor Network simulation tool, which was not suitable for RFID systems for several reasons. This raised the need to develop a new simulation software that translates the hardware components into java code methods and types, which is then

executed to deploy the selected antennas and tags. The measured values such as power loss, actual, estimated distance, actual tag location in pixels, which are detected by each antenna in the application are generated in text files, which made it easy for this application to be customized in order to be used in future to integrate with other applications, or middleware.

The research investigated aspects involved in automating a library system using RFID technology as a communication protocol for the underlying RFID infrastructure. The library middleware application is a smart location-aware service that is used to identify and track library objects automatically and dynamically at runtime. Initially, various aspects were developed more in an outlined manner rather than a strong focus in one area, using enhanced techniques utilized from the literature review.

The second phase of the research is software development, the goal was to build a custom-made software simulator, that locates the RFID tags from fixed, and identified points in the simulation window, which holds the antennas. Data collection and acquisition were carried out alongside the completion of the programs.

RFID Simulator was built to provide a user-friendly interface to the user, and customizable simulation parameters that can be updated and deployed at runtime. Developing the RFIDSim from scratch has proven to be more versatile, easy to use application and more importantly the application achieved all the conditions and simulation requirements that the other simulations software could not achieve.

Table of Contents

List of Figures

List of Tables

Statement of Originality

'I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for qualification of any other degree or diploma of a university or other institution of higher learning, except where due acknowledgment is made in the acknowledgments section.'

Ali Alwadi
26th October 2016

Chapter 1
Introduction

Radio Frequency Identification (RFID) today is widely used in many different industries and sectors. RFID is more commonly used than barcode technology today, because of its contactless nature that enables multiple-item detection. The memory capability of RFID makes it more than just an identification technology, but also a data carrier that can update and transfer information momentarily [1].

Previously many libraries had already installed barcode systems, where a barcode is placed on each book and the Library Management System (LMS) uniquely identifies the book by reading the barcode, using a barcode scanner. Where barcodes simplify the identification of items for library circulation and archives [2, 3]. Since RFID system has a lot in common with the existing barcode system, RFID has been introduced in library systems starting from late 1990s [1], more and more library systems have adopted RFID technology to automate their systems.

Until recently, barcode technology was the most popular library system, because of its simplicity and low cost. It also has some important limitations, for instance, it requires line of sight for operation and also the data on a barcode is very limited in size, which cannot be modified or added. Modern application processes such as patient care or supply chain integration, and libraries, need more advanced capabilities, which a barcode system cannot achieve [4, 5].

The use of RFID in libraries, for developing self-service applications dates back to the late 1990's [1]. Since then, RFID has generated more and more interest among the libraries as a technology to enhance self-service for improving productivity and user satisfaction [1]. By 2003, more than 300 libraries in the United States had implemented RFID technology [6]. According to the 2004 library report in [6], the collections total nearly 1,000,000 monographs, 8500 serials subscriptions, 12,000 serial titles accessible through aggregated databases, over electronic indexes, databases or reference sources, and 27,000 media materials [6].

The jobs carried out in a library are quite repetitive like [6]:

- Check-in and check-out.
- Sorting or re-shelving.
- Inventory management.

When libraries first started to use RFID, high frequency (HF) was the only available technology and thus naturally became the perfect choice by the system suppliers at that time. The HF frequency band of spectrum at 13.56 MHz is available in most countries throughout the world. The use of HF ensures that a supplier's RFID system will be commercially viable in most countries. HF RFID has been used in libraries for over 15 years, but RFID technology continues to evolve and there exist other options [1].

RFID systems at Ultra High Frequency (UHF) has started gaining more attention in recent years, it has been widely deployed in the logistics and warehouse management industry. The number of industries that started to use UHF RFID is increasing. RFID vendors supplying goods directly or indirectly to the US Department of Defense (DOD) are required to integrate UHF RFID into their shipping procedures by phases. The use of UHF in RFID was triggered in airports, shopping malls, retail shops, and libraries [1].

The library is a unique environment, that combines technological innovation with sustained need for customer service, and highly efficient information management. It has been an area of interest, that attracts emerging technology for patron services, where they are frequently trialled [7, 8].

In recent years, several RFID-based case studies in journals and books, examining implementation in educational institutions, have been published in the subject of warehouse and library management [9], commercial supply chain [10] and in healthcare settings, among others [9]. There is only a limited number of publications that cover the details of organizational and innovative potential of RFID technology in academic, public or private libraries [8].

There is a major technological shift in product traceability that began with the transitioning to RFID technology from bar-code. This has contributed to the ability of the RFID technology to resolve tracking problems in more effective and faster way. Also this has resulted in significant economic, operational, technological and logistical impacts on supply chain infrastructures. The advantages of the RFID technology over bar-code and any other data collection technology, can be summarized in [5]:

- Reading the tags in heavy moisture, noisy, or dirty environments.
- Greater flexibility in reading the tags with wide scanning area.

Operating and managing libraries, involves undertaking numerous repetitive tasks as shown in Figure 1.1, which include the following tasks:

- Item tagging.
- Shelving and user-related borrow/return procedures.
- Alerting system for the late returns.
- Managing inventory and anti-theft protection.

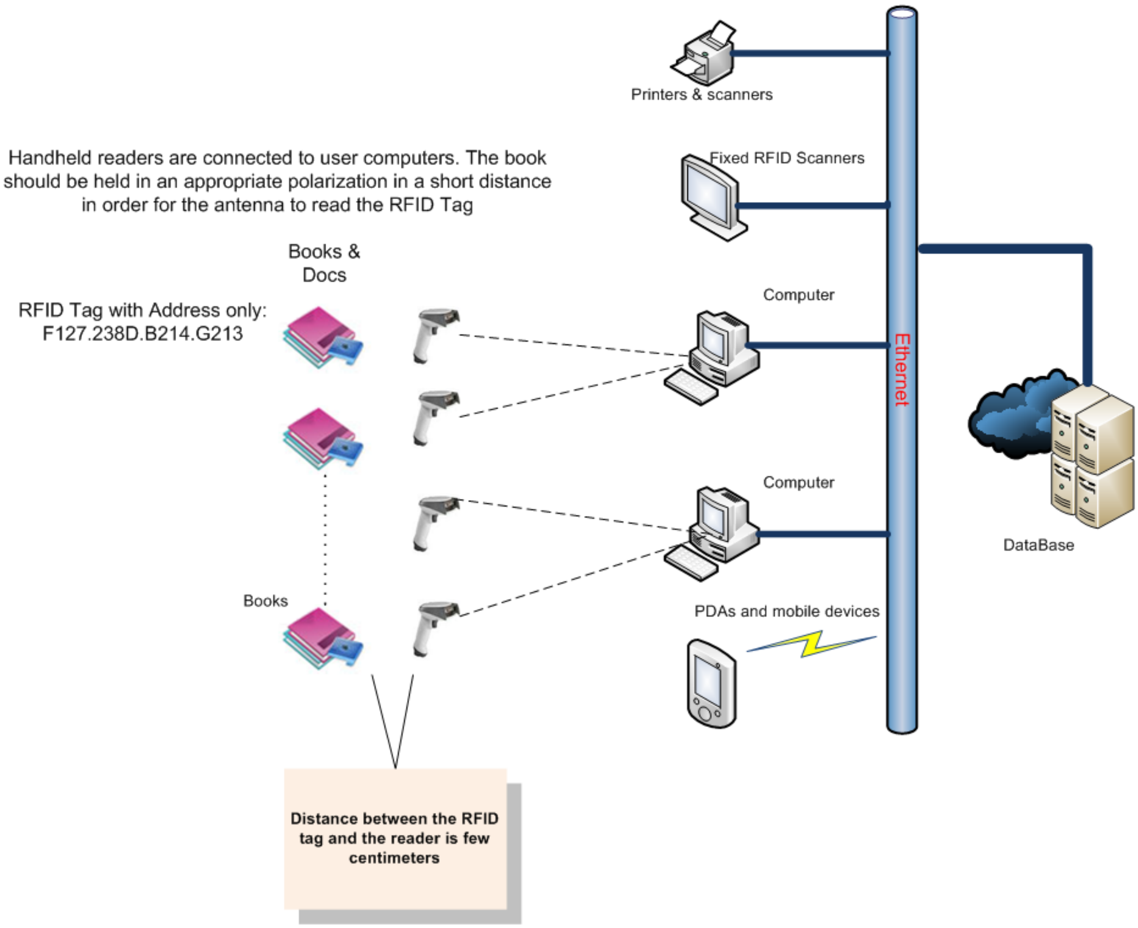


Figure 1.1 Traditional Library System: Manual handheld readers are used to scan the RFID Tags (Figure created by the student)

These tasks are repetitive, and require attention to detail, careful and precise, which involves considerable amount of labour/effort and time. These tasks require high levels of resources in terms of number of administrators involved, and the amount of time required, which are also implies a high factor of human errors, therefore it is affecting both, the efficiency and effectiveness of a library system. In order to increase efficiency and effectiveness, many libraries are adopting the automation of these tasks. By doing this, the RFID technology is being deployed to play the vital role of efficient management in library automation [11, 12]. The RFID technology has revolutionized the item identification and tracking system. Other results have been quicker check-in and check-out, better inventory management, better monitoring of usage, and better knowledge of collection development status. Furthermore, RFID utilization has freed up staff members to work at important tasks around the library rather than repetitious work. The use of RFID technology in the library system was internally focused with a well-defined, narrow set of objectives. No significant application combined technologies, hence the process was narrow in scope [6].

In this chapter, several recent publications were examined on the use of RFID in libraries, which focus on the change management process around RFID implementation and the effects of RFID technology on people, processes and technology in a library environment. The literature also addresses the divergent effect of RFID on the key areas of influence of RFID implementation and operation on clients, staff and processes [8].

1.1 Background

By carrying out this research, it is expected to prove the potential of enhancing the existing library passive RFID infrastructure to automate a library system, and designing a location-aware library management system. This research focuses on chipless RFID in terms of tag design and identification protocols employed in a library. Knowing the location of an item in a library using the attached RFID tag will add extra flexibility and opens up new applications for chipless RFID systems. The significance and application of localization using RFID tags are summarized as follows [13]:

- Localization facilitates multiple tag reading in chipless RFID systems. On knowing the location of all the tagged items. The reader can read them one by one by beam steering.
- Knowing the location of the tag, the reader can direct the antenna beam towards precise directions.

- Tracking and detection of objects or people carrying chipless RFID tags enables automatic responses such as, automatic door operation, turning on/off lights, or triggering alarms. Chipless RFID can support these application within low implementation cost.

Radio Frequency (RF) is the most widely used communication protocol that serves the IoT concept, by connecting real-life objects [13, 14]. RF is a wireless communication technology that uses a radio signal to transfer data between the '*sender*' and the '*receiver*'. In an RFID System, the sender is referred to as '*RFID Reader*', and the receiver is '*RFID Tag*'. The tags are attached to objects, with the RFID reader connected to the same system, and senses the information stored on the tag [13, 14]. The objects can arbitrarily move either proactively (e.g., vehicles or humans) or due to external forces (e.g., goods or tools carried by people), so do the associated RFID tags [15]. A simplified diagram of a conventional RFID system is shown in Figure 1.2, The RFID reader connects the antenna with the main computer. The antenna activates the tag by sending the power up signal, and the tags reply in return.

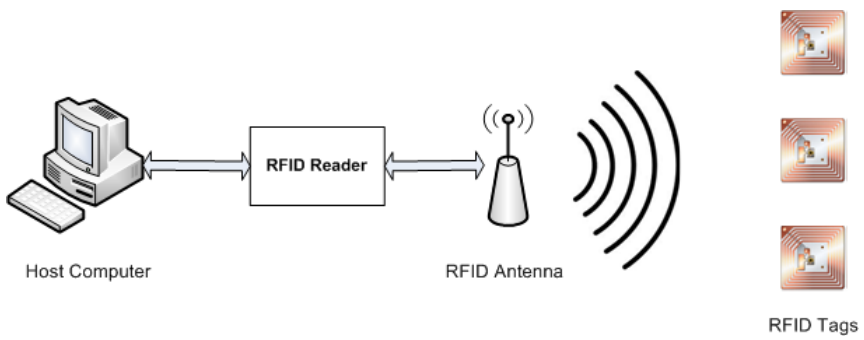


Figure 1.2 Conventional RFID System

RFID technology today is widely used in automating the industrial, educational and home processes. The reason behind this is attributed to two main factors [16]:

- The first is the dramatic decrease of the market price for the tags.
- The second is the increasing demand on this technology from both private and public sectors.

Based on the tag type, RFID systems can be divided into two main categories depending on their tag specifications:

a) Passive RFID tags, they do not have an internal power source, they depend on the power transmitted from the reader's antenna RF signal to power up the tag.

b) Active RFID tags, they have an internal power source, which is normally a small battery that provides the integrated circuit of the tag with the required power [16].

Active tags have greater reading distance, greater memory capacity than passive tags and they are programmable, which enables the system administrator to reprogram the tag whenever necessary [16-18].

RFID is used in many diverse industries, such as logistics, supply chain management, and quick response systems, this allows it to be used to identify and track location of items in warehouses [19]. Hence this research is to utilize this technology for tracking and/or locating library items that can be found in a library environment.

1.2 Radio Frequency Identification Classification

With the mass production and the falling cost of tags, passive RFID is becoming a widespread method for inventory tracking and item identification. This section briefly explains how RFID system transfers data between the tag and the reader, based on the type of the RF wave used in transferring power from the reader to the tag [20]. RFID can be classified into two main categories:

- Near Field uses magnetic induction.
- Far-Field uses electromagnetic wave capture.

Both types use load modulation to transfer information between the antenna and the tag, and vice versa. Load modulation procedures using subcarrier are primarily used in inductively coupled systems for data transfer between the tag and the reader [21]. The tag's antenna is responsible of capturing the energy emitted from the reader, also transferring the tag's modulated information that is programmed in an integrated chip. For each chip there is a tag reader that powers up the tag from the energy captured by the antenna. Receiver circuit is a key component of RFID reader, which is the main component that has been rapidly developed during recent years [22].

The underlying technology is the same for both types, they both take advantage of the electromagnetic (EM) characteristics associated with an RF signal to power up the tag. Both transfer enough power to remote tags to sustain their operation and its typically between 10 μW and 1 mW, depending on the tag type [18]. Both types describe certain electromagnetic areas formed by a radio frequency signal transmitted by an antenna. There is a thin line between both terms, and also there is a transition area between them as well that has the characteristics of both regions, see to Table 1.1. Electromagnetically far-field is commonly used whenever a long reading range is required, and typical RFID reader antenna works with a pure far-field characteristic.

Table 1.1 Radio Frequency Bands

RF Band	ADVANTAGES/ APPLICATIONS	Disadvantages	RF Range
Low Frequency (LF)	*Relatively Inexpensive* *Good Penetration* *Used in Access Control*	*Short read range* *Slow read speed*	*100–500 kHz*
High Frequency (HF)	*Good Penetration* *Medium read range* *Medium speed* *Used in Smart Cards*	*Expensive*	*10–15 MHz*
Ultra High Frequency (UHF)	*Long Distance* *High Speed* *Used Entry Control Vehicle ID*	*Expensive* *Requires a Line-of-Sight to read the tag*	*850–950 MHz*
Microwave (High UHF)	*Long Distance* *High Speed* *Used in Wi-Fi*	*Expensive* *Bad Penetration* *Line-of-Sight reading*	*2.4–5.8 GHz*

The boundary between near-field and far-field regions is inversely proportional to the carrier frequency and approximately equals to [18]:

$$d = \frac{c}{2\pi f} \tag{1.1}$$

where d is the distance (metres) between the reader and the tag, c is the speed of light (c = 3×10^8 metres per second), and f is the frequency in Hertz (Hz). Therefore, only low or high carrier frequencies are used in near-field coupling tags; the two most commonly used are 128 kHz (Low Frequency) and 13.56 MHz (High Frequency), see table 1.1 [23].

1.2.1 Near-Field RFID

Near-Field uses a load modulation technique that uses the coupling type to transfer RF signal between the reader and the tag. Inductive coupling uses the magnetic field generated by the reader to induce an electric current through a coupling element, which is consisted of an antenna and a capacitor. The current from coupling charges a capacitor which provides voltage and power to the tag. Near-Field coupling is the result of Faraday's principle magnetic induction [18]. The magnetic induction that is a result of a reader passes a large alternating current through the reader coil, producing an alternating magnetic field. A tag incorporates a smaller coil, when placed in this field, an alternating voltage will appear across it, see Figure 1.3. By rectifying and coupling this voltage to a capacitor, a reservoir of charge accumulates, which can then be used to power up the tag chip [18].

The EM field in the near-field region is reactive, the electric and the magnetic fields are orthogonal. Depending on the type of the antenna, one field (such as the electric field for a dipole or magnetic field for a coil) dominates the other, see Figure 1.3 [18].

Inductive coupling technique is preferred in near-field RFID applications mainly for one reason, most of the reactive energy is in the magnetic field. The load modulation techniques can be used to transfer information between the tag and the antenna and vice versa [24].

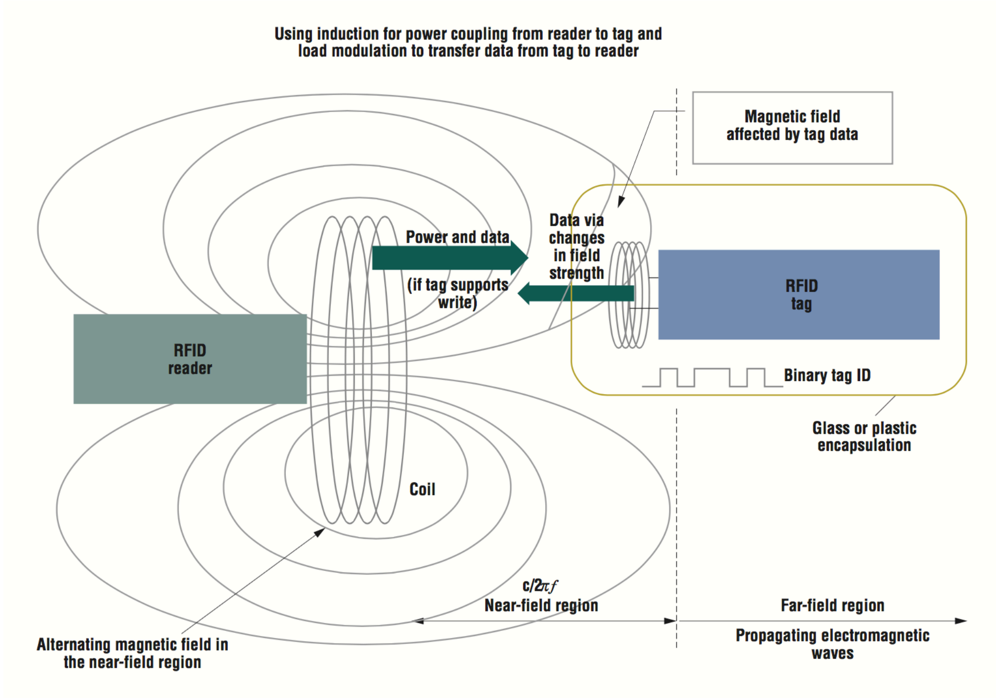


Figure 1.3 Near-Field Coupling [18]

Near-Field coupling was one of the first approaches to be chosen for Passive RFID implementations around the world because of its design simplicity despite its physical limitations [18, 24]. The range used for magnetic induction is highly dependent on the tag's frequency. So, if the frequency of operation increases, the distance over which near-field coupling can operate decreases. Another limitation is the energy available for induction as a function of distance from the reader coil. The magnetic field drops off at a factor, which can be calculated using equation (1.2):

$$F = \frac{1}{r^3} \tag{1.2}$$

where F is the drop factor, r is the distance in metres between the tag and the reader, along a centre line perpendicular to the coil's plane. As applications require more information to be stored on the tag's integrated circuit, it is also required to maintain a fixed read rate by distinguishing multi-pole tags in the same read range. A tag that requires a higher data rate will need a higher operating frequency. These design considerations have led to new passive RFID designs based on far-field communication [18].

As the signal range for this type of coupling is small compared to the other types of RFID techniques, will result in designing the readers with larger than normal antenna. This is to enlarge the interrogation zone to cover up the short read range for the RFID tag, for instance 150 × 150 mm^2 will be required [18, 25]. The conventional solid-line loop with the perimeter comparable to one operating wavelength cannot produce even magnetic field distribution in the near-field zone of the antenna. As the current distribution along the loop experiences phase-inversion and the current approaches zero. The magnetic field is relatively weak in certain region of the interrogation zone, which degrades the reliability of RFID tag detection [24].

1.2.2 Far-Field RFID

Far-Field system uses electromagnetic radiation to transfer information between a reader and tags unlike Near-Field which utilizes capacitive coupling to transfer information. Far-Field RFID systems operate at ultra high frequency (840-960 MHz) or microwave frequency (2.45 GHz - 24 GHz) with a long reading range compared to that for the Near-Field, see to Table 1.1 [26]. Far-Field RFID tags capture EM waves propagating from a dipole antenna attached to the reader. A smaller dipole antenna in the tag receives this energy as an alternating potential difference that appears across the arms of the dipole. A diode can rectify this difference and link it to a capacitor, which will result in an accumulation of energy in order to power its electronics. Unlike the inductive designs, the tags are beyond the range of the reader's near field, information can't be transmitted back to the reader using load modulation [18].

The EM field in the far-field region is radioactive in nature, see Table 1.2, coupling captures EM energy at a tag antenna as a potential difference. Part of the energy incident on a tag's antenna is reflected back due to an impedance mismatch between the antenna and the load circuit. Changing the mismatch or loading on the antenna can vary depending on the amount of reflected energy, which is also called '*Backscattering*' [27].

Table 1.2 Near-Field Vs. Far-Field

Property	***Near-Field***	***Far-Field***
Tag Read Range	*Between 5mm to 10 cm depending on the frequency and antenna*	*Can reach up to 22.1 metres for some frequencies*
Reader Antenna	*Small, Omni-directional*	*Resonant, directional* *Small antenna size for high frequencies*
Usage	*Metal or liquid surrounded objects*	*Whenever a long reading range is required*
Modulation	*Load Modulation using capacitive coupling*	*Electromagnetic radiation*
EM Signal	*Radiative-like signal*	*Radiative-like signal*

A far-field system's range is limited by the amount of energy that reaches the tag from the reader, and also by how sensitive the reader's radio receiver is to the reflected signal, attenuation occurs as EM waves radiate from the reader to the tag [18]. The returned signal to the antenna is very weak, since it is the result of two attenuations, each based on an inverse square law, once as the EM waves radiate from the reader antenna to the tag, the second when the reflected waves travel back to the antenna.

The drop factor of the returning energy is:

$$F = \frac{1}{r^4} \tag{1.3}$$

where r is the distance in metres between the tag and the reader in metres [18].

1.3 Literature Review

Near-Field and Far-Field RFID systems have been introduced earlier in this chapter and Table 1.2 summarizes the characteristics and differences of these systems.

These days, as the size of the semiconductor-based electronics is getting lower and lower, the energy required to power up the tag continues to decrease, which serves the purpose of RFID very well. Customized RFID tags can be designed and manufactured at a very low cost, they can be read from a distance that might reach up to 4-6 metres with average power consumption 100 dBm (in a frequency of 2.4 GHz).

Typically far-field reader antennas can successfully interrogate tags from distance up to 3m, and some RFID manufacturers claim that their products can achieve read ranges of up to 6m and sometimes much more [18]. Some electrically large loop antennas have been reported to generate strong and even magnetic field. Dobkin et al. [28] presented a segmented magnetic antenna consisting of a number of segments and each segment is composed of a metal line and a series of lumped capacitor. S.-Y. Chen and P. Hsu [29] presented a miniature folded-slot antenna with RF performance suitable for RFID tag use at 5.8 GHz.

A segmented loop antenna with a diameter of 10 cm performed a desirable performance at a frequency of 915 MHz. Oliver R. A. [24, 30-32] proposed the broken-loop antennas, three broken-loop antennas using different coupled lines, namely triple line, double line and single line. Xianming et al. [24] presented a segmented loop antenna that generates a strong and even magnetic field distribution in a large near-field zone for broadband UHF near-field RFID applications.

S.K. Padhi et al. [33] designed a dual linearly polarized aperture coupled circular microstrip patch antenna. The antenna uses a novel configuration of symmetric and asymmetric coupling slots. Gaetano Marrocco [34] presented a new meander line antennas with improved gain, as low-profile, self-resonant tags for application, in passive radio frequency identification. Antenna shape and size is optimized by genetic algorithm taking into account the conductor loss. M. Hirvonen et al. [35] proposed a small and low-cost antenna solution for RFID tags. The impedance of the antenna is independent of the platform, which makes the antenna valid for use in many different environments. Finally, Xianming et al. [36] presented a novel folded dipole antenna with a very simple configuration, with achievable input impedance, which can vary by choosing suitable geometry parameters.

To achieve the necessary range, an antenna with the suitable design must be chosen depending on the designated tag size, reader antenna size and operating frequency. The designed antenna will generate a magnetic field evenly distributed across the space to cover all the passive RFID tags, which conventional line loop antennas cannot achieve.

In this chapter, a technology background has been explained earlier. Chapter 2 will describe the current system architecture of a traditional RFID-based library. It also explains the points of modifications. At the end of the chapter, the error minimization function is explained. Chapter 3 explains the software simulation part of the research. This is divided into two parts, (a) the Wireless Sensor Network (WSN) simulation application, and then (b) the negative sides of using WSN, then the development of a newly developed simulation software, RFIDSim. The proposed solution is explained in Chapter 4, all the aspects of the change and describing the chosen hardware to do the simulation. Chapter 5 covers the conclusions, discussions and the future work related to this research.

Chapter 2

Literature Review and Related Work

The traditional library system architecture was explained in chapter 1. This chapter will explore the library system architecture. The RFID components required to automate the system are described, and their functionalities are explained. Finally, the collision issue will be briefly mentioned, and a few researches around the anti-collision algorithms are to be mentioned. A general description of the middleware layers is provided, and two examples for two middleware providers that support our reader antenna infrastructure, one of which is open source middleware solution.

Like other ubiquitous systems, the overall system performance and scalability of the library RFID system is highly dependent on the system architecture. For instance, the distance between the RFID Antennas and the tags affects the tag coverage, which must be taken in consideration when planning the system [37].

The tag selected in this research is the passive RFID for a simple reason, is that passive RFID is more common and its widely used. They are found in a wide variety of areas like, airports, libraries, shopping malls, home automation applications, and much more. For instance as of 2009, 1500 libraries employ RFID applications in 2,500 facilities [37, 38].

The basic component of an RFID system is the tag, as shown in Figure 1.1. The tags are attached to the library items. Each tag is identified by a unique number that uniquely describe it, this makes it easy for the middleware to communicate and store information about the objects. Every tag has its own internal antenna that is used to transmit information to the RFID Reader, also it participates in powering up the tag by receiving the signal from the reader antenna. The second component is the RFID reader antenna, which is referred to as the antenna, usually it contains two parts. Normally these two components are packaged separately into two component sets. The third and last component is the Backend system, which is the application that contains the information about all the tags in the network and manages the flow of information between the tags and the readers.

In the following sections, all these three components are studied and investigated closely, with one more section that was added to the end of this chapter to explain a very important concept in RFID, which is collision and later on, in the same section, a few anti-collision algorithms are discussed.

2.1 RFID Tag

RFID tags are either (a) passive, (b) active, (c) semi-passive. The part passive, or active describes the power source that the tag uses to extract the required energy to receive and transmit signals. Passive tags retrieve the necessary energy from the signal transmitted by the reader's antenna; depending on the antenna type. The tag retrieves the required energy from the interrogating wave or from a constant signal that some antennas transmit to power up the tag. Passive tags are the less expensive to manufacture, do not require maintenance, are more compact, and lighter [16, 39]. The second type of tags is the active tag, which includes a battery as a power source, to enhance reading range. Semi-passive tag operates similarly to the passive tag, using backscattering technique to reply to the reader. The main difference is that semi-passive tag has a battery to power a circuitry that is embedded in the tag. They are used in conjunction with externally integrated electronic components such as sensors. Other than that semi-passive tag has the same characteristics of the passive tag in terms of reading range, and operating frequency [40].

Backscattering in RFID tags do not use a radio transmitter; instead, they use modulation of the reflected power from the tag antenna, the current flowing on a transmitting antenna leads to a voltage induced on a receiving antenna. If the antenna is connected to a load, which presents little impediment to current flow, it seems reasonable that a current will be induced on the receiving antenna [41].

Each RFID tag contains an RF transponder with a digital memory chip that is uniquely identified by an Identifier. The interrogator, an antenna packaged with a transceiver and decoder, emits a signal activating the RFID tag in order to read the information saved on this tag, as shown in Figure 2.1 [16].

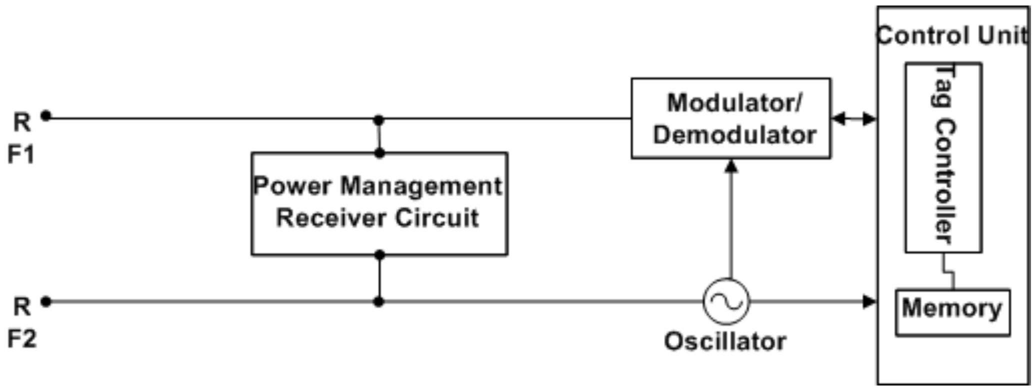


Figure 2.1 Basic Passive RFID Tag

A brief description of each RFID tag component is mentioned below:

- RF Interface, it is the radioactive component of the tag, which does the following, see Figure 2.1:
  - Supplying RFID transponders with power by generating the energy required to power up the tag.
  - Modulating the signal for the transponder to transmit.
  - Reception and demodulation of signals received from the transponders [42].
- Control unit, the controller part of the reader that performs the following functionalities [42, 43]:
  - Communication and execution of the application software commands.
  - Signal coding and decoding.
  - Communication control with a transponder.
  - Some RFID readers have additional functionalities like anti-collision, and encryption.
  - Decryption of transferred data, transponder-reader authentication.
  - The data capacity for the RFID tag depends mainly on the manufacturers' specifications. With the current semiconductor revolution, modern tags capacity can save up to 2048 bits of information [16, 42, 43].

2.2 RFID Reader Antenna

The transmitted signal from the RFID reader antenna is electromagnetic wave. Based on the characteristics of the transmitted signal, antennas can be divided into two different types:

1 ***Linear Polarization***: The transmitted electromagnetic wave propagates in one direction, either vertical or horizontal, depending on the antenna's orientation, which is why they are also called *Dipole* antennas. This type of antennas is the best to use when the tag orientation is known. This is why it is used in the hand-held readers in libraries and factories. One disadvantage of this antenna is that the reader and the tag must be aligned for the reader to read the tag [44-46], as shown in Figure 2.2 (Top) .

2 ***Circular Polarization***: The electromagnetic wave covers the two planes when propagating in a circle-like motion, which looks like the motion of a screw. Which makes this type more efficient than the first one for two simple reasons, it covers a wide area, and the tag orientation does not have to be in line with the reader. One disadvantage of this antenna is the energy loss, since most of the transmitted signal energy is within the first few waves, as shown in Figure 2.2 (Bottom). The top figure demonstrates the signal flow of linear polarization antenna. While the bottom figure depicts a signal for Circular polarization antenna, the blue arrows represent the direction of the magnetic field, and the red line is the electromagnetic signal flow [44-46].

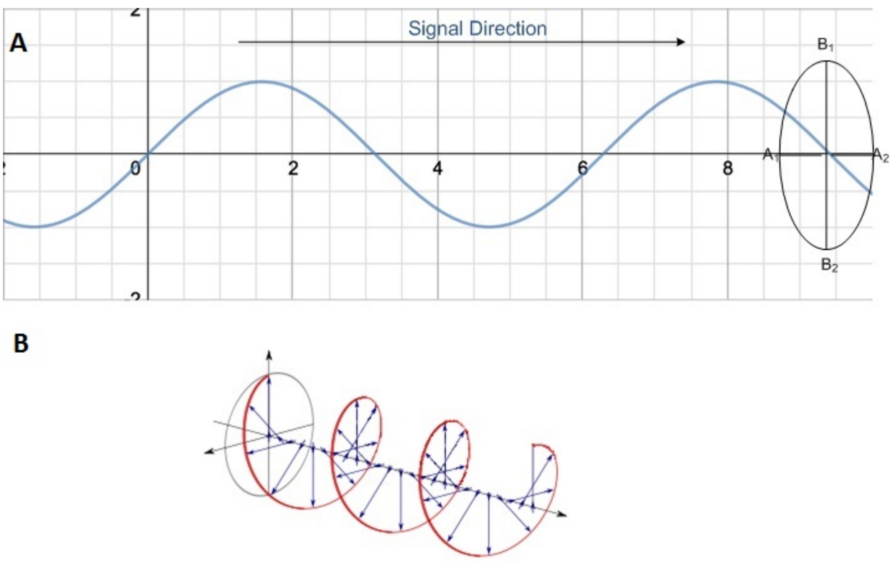


Figure 2.2 Signal flow of linear polarization antenna (top), circular polarization antenna (bottom)

Commercial, antennas are classified into two main groups, based on the number of ports available in the antenna, and the transmission direction of the signal on each port:

- Monostatic Circular: This antenna uses a common port for transmission and receiving of the RFID signal [45].
- Bistatic Circular: This antenna has a dual ports, one for transmission and the other one for receiving signals; this type is common, but its more expensive [45].

Both antennas can be commercially available with an Listen Before Talk (LBT) port, which is a dedicated port that listens for signals before sending RFID signals [45]. The read range for both types depends on:

- The power available at the reader/interrogator to communicate with the tag, or tags [47].
- The power available within the tag to respond [47].
- The environmental conditions and structures, including signal to noise ratio [47].

The delivered wave from an antenna propagates in space, the signal strength diminishes as the travelled distance increases. The antenna design determines the shape of the wave delivered, so the read range and the positive identification will be affected by the distance and the orientation between the antenna and the tag. In space free of obstructions or absorption objects, the strength of the field decay in inverse proportion to the cube of the distance, given in Equation (1.2) [47].

An important factor that heavily affects the efficiency of the identification is collision. When an antenna transmits a power signal, or a regular RF signal, all the tags in the range will respond with their identification signal. So, for a large number of tags in the same area, this will cause a collision.

2.3 Collision and Anti-Collision algorithms

Collision is a technical term that describes an event of interference between two or more RFID signals, this normally happens as a result of two or more tags replying to the reader simultaneously. This is very important in RFID networks; since all tags in the interrogation zone respond blindly to the reader once the reader sends the power up signal, or the read signal.

As a normal result for the collision, the reader will not be able to read and identify the tag's signal, this implies that a solution must be provided to '*detect*' or isolate each tag's signal to allow the reader to read those signals separately. Such a solution enhances the system efficiency by reducing the identification time. Various anti-collision algorithms have been developed based on two categories:

1. ALOHA-based algorithms, which divides the frequency range into time slots and assigns a time slot for each tag.
2. Tree-based algorithms, which divides the tags into tree-based subsets and iterates through these tags for identification [48, 49].

ALOHA algorithm is simply a probability-based algorithm, which belongs to control algorithms of electronic label. A collision resolution algorithm based on Time Division Multiple Access (TDMA) [48].

ALOHA-based algorithms are the most common anti-collision techniques, which is why it is the main focus in this research. The one algorithm in particular covered in details, Dynamic Frame slotted ALOHA (DFS-ALOHA), which is one of the most commonly used anti-collision algorithms in RFID systems, this technique calculates the probability of the occurrence of the tag-collision to estimate the number of the passive tags, also the frame size is decided which makes this algorithm the most efficient technique. In DFS-ALOHA algorithm, each tag transmits its data in a frame at a random slot to avoid collisions [47, 50, 51].

ALOHA divides the bandwidth into frames, each frame is divided into slots, as shown in Figure 2.3. The system efficiency depends on the number of tags, and the dynamic frame size that is decided based on the number of tags in the RFID network [25, 52, 53]. The main disadvantage of ALOHA-based algorithms is '*starvation*', in which, a tag is not allocated a time slot, or a frequency slot, depending on the division multiplexing, to transfer its signal. This makes it '*indiscoverable*' from the reader, in this case, this particular tag will not be identified for a long time, and may be forever [48].

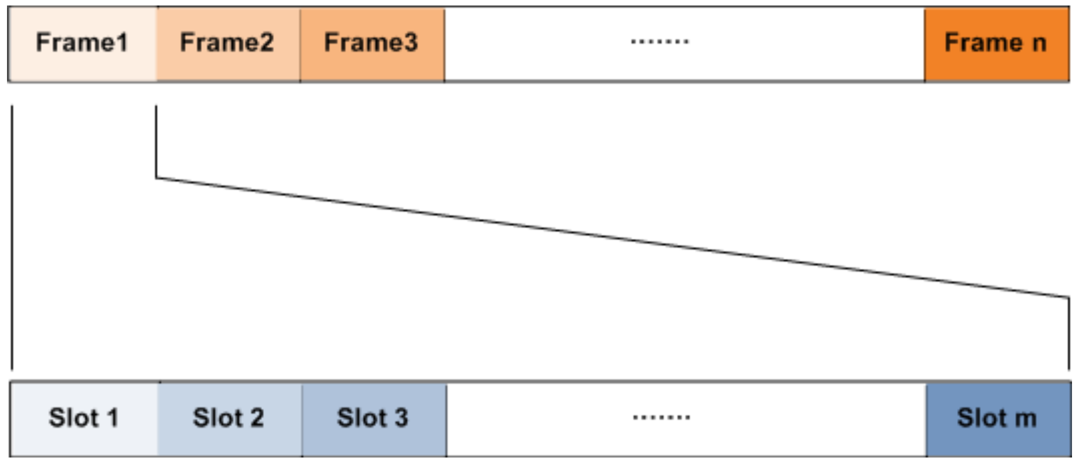


Figure 2.3 ALOHA-based anti-collision algorithm

Tree-based algorithms are deterministic, which identify the tags that lie in the interrogation zone of the antenna using a search tree. This classifies all the tags in the zone into a tree of subsets and iterates through these subsets until all tags are identified. Tree-based algorithms can be divided into three main categories: (a) binary tree (BT), (b) query tree (QT) algorithms, (c) and a composite of both, binary and query tree [39, 52, 54].

QT uses the tag ID to split the tags into two group of tags, which are called '*subsets*', which then iterates through all tags by splitting all tags in two groups until all tags are identified. The identification efficiency is significantly affected by the ID distribution. BT uses random numbers to identify the tag splits, which results in tags needing reprogrammable memory to store their assigned number that makes it more efficient, but the cost and size of the tag are higher [55, 56].

2.4 RFID Middleware

The middleware refers to software or device(s) that connect RFID readers and the data they collect, to enterprise information systems. RFID middleware helps making sense of RFID tag reads, applies filtering, formatting and logic to tag data captured by a reader, and provides this processed data to back-end applications (Burnell, 2008). RFID middleware is the core engine that manages the flow of data between tag readers and enterprise applications, and is responsible for the quality, and therefore usability of the information. It provides readers connectivity, context-based filtering and routing, and enterprise/B2B integration [39, 42]. It is responsible of operating the integrated components, and managing the flow of information in and out of the repository [42]. It provides readers connectivity, context-based filtering and routing, and interface integration. For efficient tag identification, there are few conditions the middleware should meet in order for the solution to meet the requirements, which can be listed in [55]:

- Real-time handling of incoming data from the RFID readers: The RFID Middleware should support a wide range of RFID readers, and allow the system to interact with these readers in a timely manner [42].
- The middleware must provide a common interface to access different kinds of hardware offering different features.

RFID middleware is generally composed of five major layers [39]:

- Reader Interface: The lowest layer of the RFID middleware, which handles the interaction with the RFID hardware. It maintains the device drivers of all the devices supported by the system, manages all the hardware related parameters like reader protocol, air interface, and host-side communication.
- Data Processor and Storage: Processing and storing the raw data coming from the readers.
- Application Interface: Provides the user with the Interface required to configure, manage the RFID Middleware, and the components attached to it.
- Middleware Management: Add, configure, modify connected RFID readers and modify application level.
- Parameters: Such as filters, duplicate removal timing window, add or remove services supported by the RFID middleware.

2.4.1. IBM WebSphere RFID Middleware

WebSphere RFID middleware solution, which has been developed by IBM. It consists of three main components, as shown in Figure 2.4 are: (a) RFID devices. (b) WebSphere Premises Server and (c) WebSphere Business Integration Server [57].

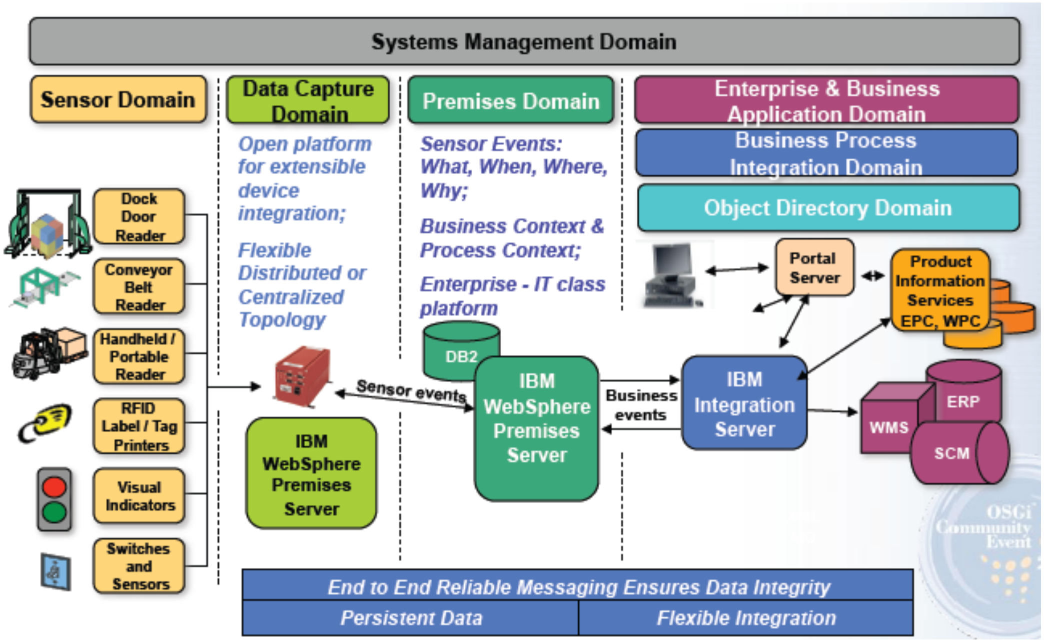


Figure 2.4 Websphere Middleware Architecture [51]

IBM WebSphere is a sensor-enabled product, meaning that it allows sensor data aggregation and analysis, deriving insights from sensor data and integrating those insights with the Service Oriented Architecture (SOA) business processes. The software provides the use of intelligent business rules that manage complex event identification and processing. This adds more capabilities to the middleware application to integrate with different type of interfaces by just expanding the functionalities of the middleware APIs. Websphere application server has an advantage over the other types of servers that are available in the market, it supports a dynamically integrated APIs. This allows the application to integrate with different sorts of sensor hardware, which will deliver new and enhanced capabilities to create a robust, flexible, and scalable platform for capturing new information from sensor data [57].

2.4.2. Rifidi Edge Server

Rifidi Edge is a complete RFID middleware platform with an edge server and development tools to enable the development and deployment of highly customized RFID applications. The aim of the product is to provide an open source alternative to popular RFID platforms such as IBM Premises Server and Microsoft Biztalk RFID. Built on a cutting edge Java and integrated with a powerful open source rules engine

Esper Rifidi Edge can build complex applications that interact with the most popular RFID and sensor devices available today [58].

The important function of Rifidi Edge Server is collecting data from sensors and deliver them to systems that use the data for business processes [58]. The server filters out all the noises and distorted signals that the sensors deliver to the middleware, which is important in the RFID area to filter out all the undesired tag signals. Figure 2.5 shows a high-level description of how data is collected, and the flow of data through the edge server.

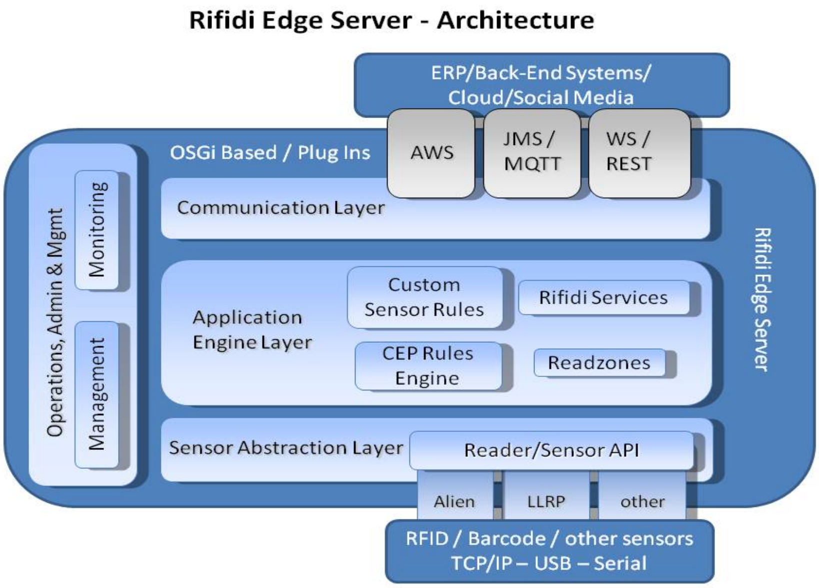


Figure 2.5 The data flow in Edge Server [58]

Figure 2.5 contains a high-level description of how data is collected and the flow through the edge server. The input data are produced by sensors, which are normally hardware RFID readers, such as Alien 9800, Symbol XR400, and many others. These data might also be produced by a legacy barcode reader, a database, or even another edge server. The Sensor Abstraction Layer provides a way for users to develop their own programming interface for their custom sensors [58]. The data collected from the sensors, they are inserted into a high-speed internal message bus, through which other internal edge server components can access them. As sensors have the ability to produce an enormous number of events, which is necessary for the middleware application to confine and filter the incoming data before processing [58].

The Application Layer Events (ALE) provide a standard Application Programming Interface (API) for collecting and filtering RFID data. Rifidi Edge Server has an implementation of the ALE version 1.1 specification. Internally ALE layer uses an event stream processor called Esper to collect data according to the ALE rules [58, 59].

2.4.2.1 Application Engine Layer

For the majority of applications, it is not desirable to save every event that the sensors produce. RFID tags reply to the nearest antennas once they are powered up, which produce thousands of events in a second, a large number of which might be duplicates [58].

Complex Event Processing (CEP) is a platform of viewing data as ephemeral events, i.e. a stream consisting of non-persisted events, identifying meaningful events from the stream using rules. Rifidi Edge Server uses a Complex Event Processor called Esper. This allows the integrated application to write queries using a Structured Query Language (SQL) like syntax, which makes it easier to query information about the detected tags from the deployed middleware application, and display this information if necessary [58].

The application layer enables writing a custom business logic that uses Esper to filter, aggregate and process events produced by the antennas. The application in this layer will be responsible of performing the calculations to estimate the distance for all the detected tags in the antenna range, build the mean square error function for each tag, and minimize the error function to reach the lowest possible error value which means the nearest estimation.

2.4.2.2 Edge Server Middleware Application Server

The server can be started by running the Rifidi Server start-up file, which initializes the server modules, loads the necessary classes and JAR files. The server starts listening on port 8111, see Figure 2.6.

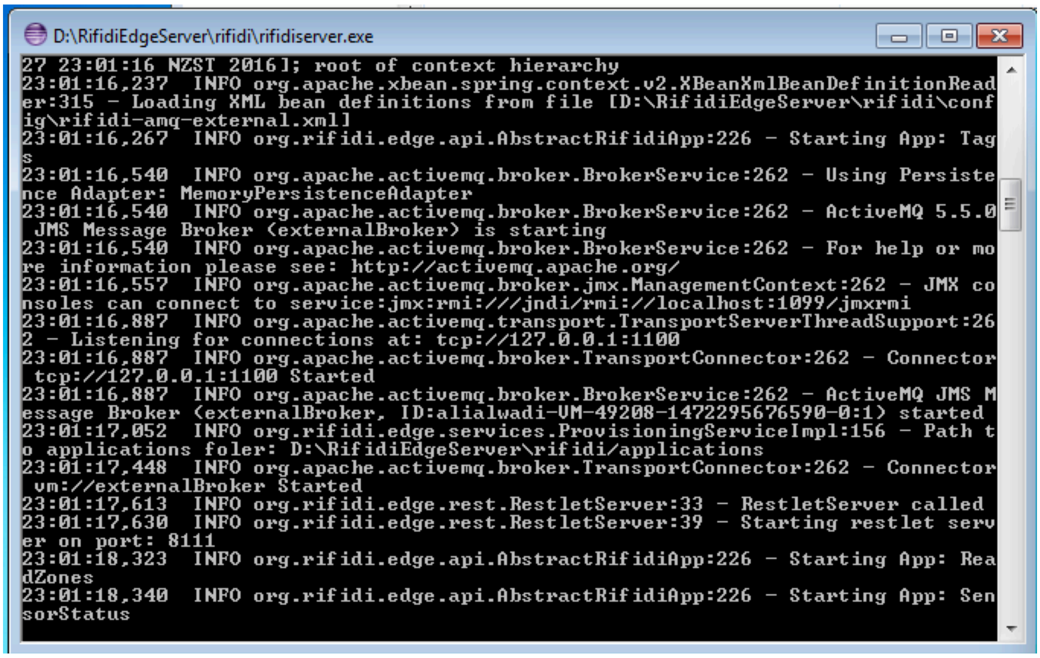


Figure 2.6 Rifidi Edge Server start-up

Once the server is up and running, the client can be started, see Figure 2.7 below a screenshot for the initial window of Rifidi workbench.

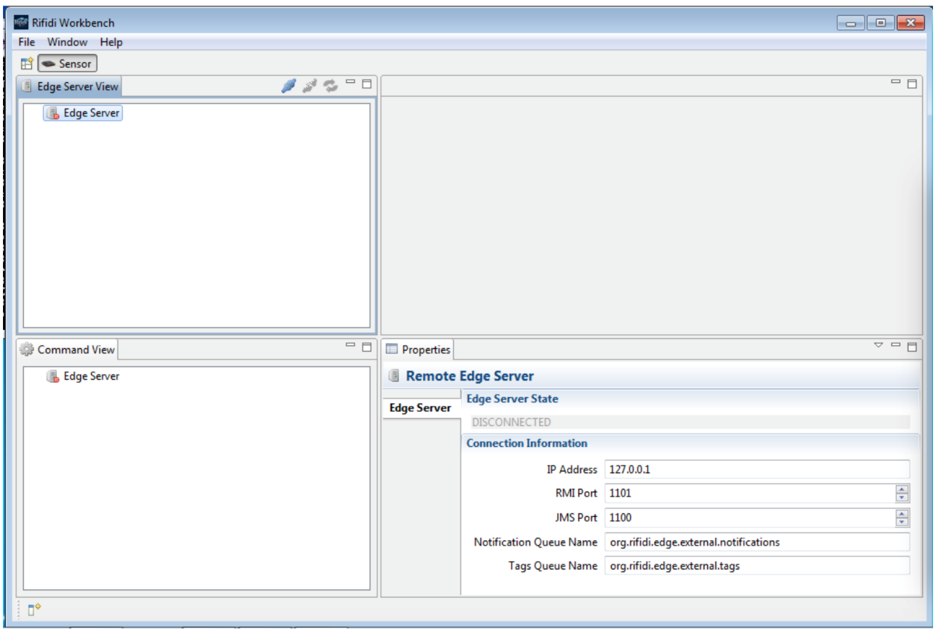


Figure 2.7 Rifidi Edge Server Workbench

When the workbench connects to the edge server, the command window is activated with all the per-configured list of commands supported by default, see Figure 2.8.

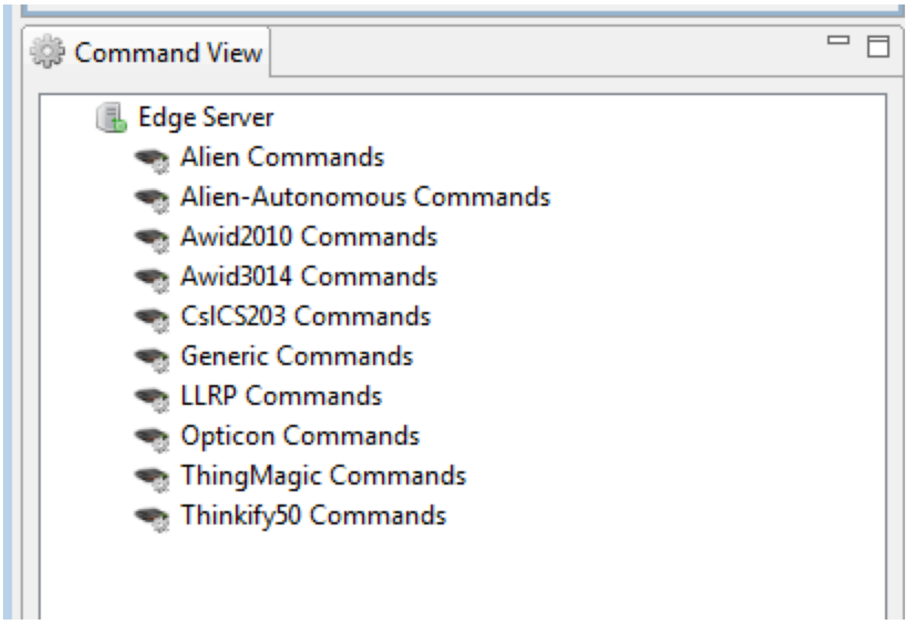


Figure 2.8 Edge Server Command View

The appropriate set of commands for the designated reader will be configured in the above list of commands, which then can be used by the middleware application to trigger antenna read commands.

Below are the steps taken to get an Alien reader up and reading tags on Rifidi Edge. The supported Alien readers are the 9800, 9900, 8800, and 9600, which have been pre-configured and tested with Rifidi Edge server:

1. Start edge server and the edge client. Connect the client to the server (Refer to the steps mentioned previously in this section).
2. Right click on the 'Edge Server' icon in the ‘Edge Server View’ and ‘New Reader’ is to be chosen.
3. Select ‘Alien’ from the drop-down and click ‘Next’.
4. The IP and port of the reader is inserted in their respective boxes. There is a discovery tool in Alien software to find out what the IP and port are for the reader which makes it easy to find the IP and port that we need to integrate with.
5. Set the username and password of the reader. The default is ‘alien’ and ‘password’, respectively, this is manufacturer specific and may vary between models.

6. The reconnect interval (in milliseconds) is chosen. And we want Edge Server to keep pooling the reader for connection if the reader loses connection while running.
7. The maximum number of times the server will try to connect to an unresponsive reader then can be chosen. This is set to -1, which means the server will keep trying to get a connection.

2.4.2.3 Reading an RFID Tag

When the reader is set up, the middleware can actually connect to it and read the tags detectable by its antennas [58]. Below are the steps taken to read the signal of a configured reader:

1. Once the reader is recognized by the server, the reader icon will appear as an element configured with the server which will appear under the '*Edge Server'*. It will be given a unique name like '*Alien_1*'.
2. By opening the '*properties*' tab at the bottom. There are many read-related properties, some of which can be adjusted. Properties can be adjusted by typing in the new value, then right click on the reader and select '*Commit Property Changes*'. More about the properties can be found in Alien documentation.
3. Right clicking the reader, then '*create session*', establishes a connection with the reader, and enables the middleware application to start the communication with it.
4. From the '*Command View'*. Right click on the '*Alien-Poll'* folder and click '*create command template*'.
5. The command created in the previous step will be displayed, and can be used here to set the options.
6. Right clicking the session created under the reader and then clicking on '*Submit Job*' will trigger the read event, continuous polling can be triggered by selecting '*Recurring Execution'*, and selecting the interval to read tags for (default is 1 second) [58].

2.4.2.4 Mean Square error function

Mean square error is the most common technique to forecast the accuracy depending on a data set of observations. It is the average squares of the difference between the actual previous actual observations and the predicted. The measure of the centre of distribution is associated with the value of error. Supposing that we are measuring the quality of t, as a measure of the centre of distribution, the mean square error is given in equation (2.1) [60]:

$$MSE(t) = \frac{1}{n}\sum_{i-1}^{k} f_i (x_i - t)^2 = \sum_{i=1}^{k} p_i (x_i - t)^2 \tag{2.1}$$

where x is a vector of i predictions, p is the power loss value, t is the assumed good measure of the centre (In our case; t can be the first estimated distance value).

The best measure of the centre, relative to this measure of error, is the value of t that minimizes MSE. Translating the equation into java results in the code fragment in **Appendix A**.

Data sets generated from the reader antennas will be applied to this function for each tag ID in the network. Results from this function will be stored in the database along with the other information related to the book.

2.4.3 Security Constraints

The core RFID privacy problem is the unauthorized tag readout. With the help of wireless communication, third parties can read the tags of personal items from large distances, without any indication that such readout is taking place. Controlling access to tag data is of prime importance [52, 53]. By default, most RFID tags are indiscriminate, i.e. the tag replies to any reader that transmits a signal that is strong enough to power up this tag [19, 55, 61].

A few security enhancements that should be implemented for secure communication between all library components are:

- ***Repository Lookup*** of items: the system implements a database, in which the item's checkout status is logged in a database, with all tag information). When a customer carries an item through the security gate, the gate identifies the item, accesses the database and confirms that the item has been checked out. This approach requires that each item full identification number be accessed and relayed to the server for verification, at check out time. This condition will be met when using any of the middleware servers mentioned in this paper, because they all use unified repository [19, 55].
- ***Application Family Identifier*** (**AFI**): is code assigned to all the RFID tags in the library. When a library security system uses AFI, the gate will request a response from the checked in library item. When an item is checked out, the AFI code is modified to disable response to this signal [19, 55].
- ***Electronic Article Surveillance*** (**EAS**): proprietary code is assigned to all RFID Tags inside a library, which should be unique across other libraries and other industries [55, 62].

In summary, this chapter has described the system architecture of a conventional library system, and also described the different hardware components that can be utilized to automate the library system. Starting from the basic component of the library system, which is the tag, and ending with middleware application server. the different types of antennas was also explained, and the characteristics of the transmitted signal for each type. Lastly a common issue with radio frequency interface was explained, collision, and a few anti-collision algorithms have been explained.

Chapter 3
Wireless Sensor Network Simulation Tool

With the increasing demand on the wireless networks and their increasing popularity in terms of the implementations in almost every electronic, mechanical, and electrical system has raised the necessity of virtualizing the wireless networks for various reasons, with importance to study the efficiency of the design or to prove the compatibility or operability of the selected hardware. Applications need to achieve a high degree of realism and simulation capability, this requires the use of the simulation models need to include a physical description of the environment where the network will be deployed. As well as a definition of the physical features and operational parameters for the hardware components that the researcher has chosen to implement in their final design concept [46].

This chapter will cover Wireless Sensor Network (WSN) simulation application, along with some the advantages and disadvantages of using WSN simulation tool to simulate the library system will be discussed. Finally, a newly developed simulation software along with the main coding components are briefly explained.

The objectives of the software simulation part of the research are as follows:

- Extract the main features of an RFID System, and use them to build an abstracted simulation system to describe how the system would behave in reality without having the need to purchase any hardware.
- Test and verify the capability of the integration between Rifidi Edge server bench and the Edge server in order to communicate with the readers to gather the RFID tag datasets.

3.1 Introduction to Wireless Sensor Network Simulation

A wireless sensor network is a network of nodes that work in cooperation to acquire data and send it to a main sink node(s), which work as central gateway station that pick up signals from all the individual sensors [63]. This research is related to RFID networks, the terms '*sensor*' and '*tag*' are often used alternately. Later in this chapter, the RFID reader antenna will be introduced as the '*sink*'.

RFID networks in particular are hard to simulate using the existing simulation tools for two main reasons:

1. The first is that RFID system is a short distance identification technology, that replaced the bar code to identify library objects, which required a reading distance of no more than few centimetres; hence the requirement to simulate such a network, is either does not exist or insignificant, as the main components of the existing library infrastructure, is the middleware computer, and the RFID reader.
2. The second reason is the variety of the operating frequencies, and the dropping price of the RFID components has made it easy for a researcher to implement a hardware prototype to prove the concept and test the design.

The concept of RFID system was explained in Chapter 1, which can be summarized as: *RFID is a wireless communication technology that uses the radio frequency signal to transfer data between the RFID Reader and tag, where RFID Tags are attached to objects. There is an RFID reader connected to the same system, which 'sense' the information stored in the RFID Tag [13, 14].*

Choosing the right tool for simulating an RFID network is a tricky task for two main reasons:

1. The first factor is the radio frequency in which the RFID system operates is variant and dependent on the chosen readers, antennas, and tags. Which will require altering the simulation window to reflect the characteristics of the chosen hardware.
2. The second factor is localization using RFID is a new feature that is not implemented in most of the existing simulation software. The chosen algorithm for localization is triangulation, which requires three readings from three antennas to estimate the location of the tag depending on the power loss for the three signals received from the three tags individually.

While a single reader antenna may be able to roughly determine the range of a tag based on the above model, it requires multiple spatially diverse reader antennas to estimate the coordinates of a tag unambiguously [20].

3.2 Wireless Sensor Network Simulation Tool (WSN)

With the development of embedded system and network technologies, there has been growing interest in providing fine-grained controlling of living environments using low power devices. WSN consist of spatially distributed self-configurable sensors, perfectly meet the requirement. The sensors provide the ability to monitor physical or environmental conditions, such as temperature, humidity, vibration, pressure, sound, motion, with very low energy consumption [64, 65].

The sensors must also have the ability to transmit and forward sensing data to the base station. Most modern WSNs are bi-directional, enabling two-way communication, which could collect sensing data from sensors to the base station as well as disseminate commands from base station to end sensors. The development of WSNs was motivated by military applications such as battlefield surveillance, WSNs are widely used in industrial environments, residential environments and wildlife environments. Structure health monitoring, healthcare applications, home automation, and animal tracking become representative WSNs applications [65].

A typical WSN is made up of several hundreds, or sometimes thousands of '*sensor nodes*'. The topology of WSNs can vary between star network, tree network, and mesh network. Each node has the ability to communication with every other node wirelessly, thus a typical sensor node has several components:

- A radio transceiver with an antenna which has the ability to send or receive packets.
- A micro-controller which could process the data and schedule relative tasks.
- Several kinds of sensors sensing the environment data.
- Batteries providing energy supply –In most caes- [65].

Wireless Sensor Network Localization Simulator is designed for localization of sensor nodes is a simulation task that requires a networks of sensors, sinks. The program comes with eight localization algorithms, while can be implemented when required. Numerous parameters that define network topology include: network size, locators deployment strategy and antenna type, as well as the path loss and node mobility can be configured using this tool, see Figure 3.1, the Graphical User Interface (GUI) for the WSN [64].

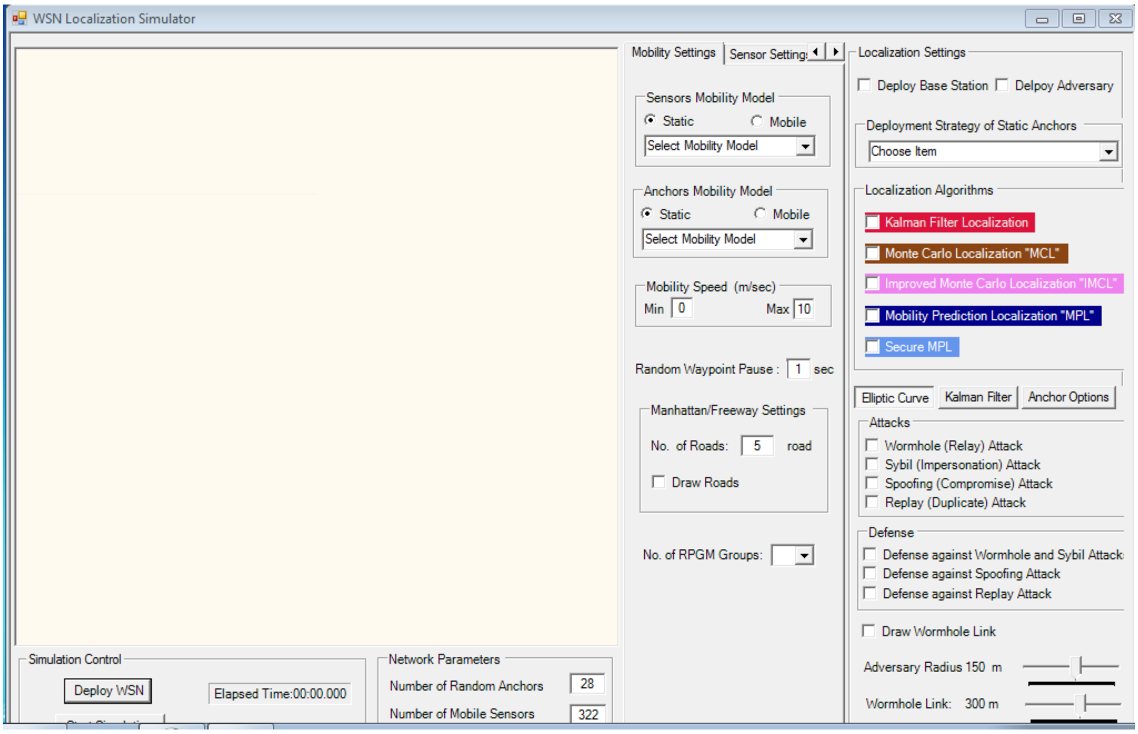


Figure 3.1 WSN Simulator screen

One advantage of WSN Simulator tool is that Kalman Filter can be introduced to simulate signal noise at runtime. WSN simulator draws a rectangular box to paint the simulation in, once the simulation frame is set, WSN deploys the configuration set on the simulation window, and generates the result files, which contains the estimates figures for the following:

- The amount of residual energy read from each sensor.
- Monte Carlo Localization result file.
- Mobility Prediction Localization.

Figure 3.2 shows a screen capture for the simulation window for the deployment of 20 sensors and 8 antennas, where the sensors randomly distributed in the simulation window, and the antennas have fixed locations in the corners.

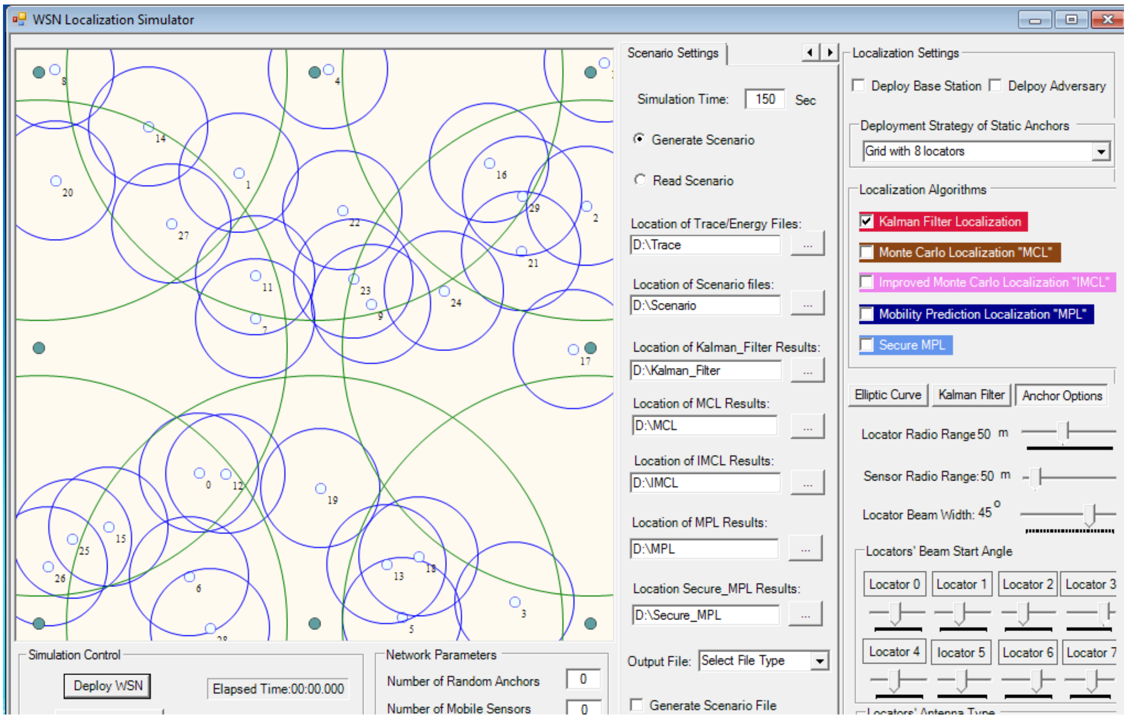
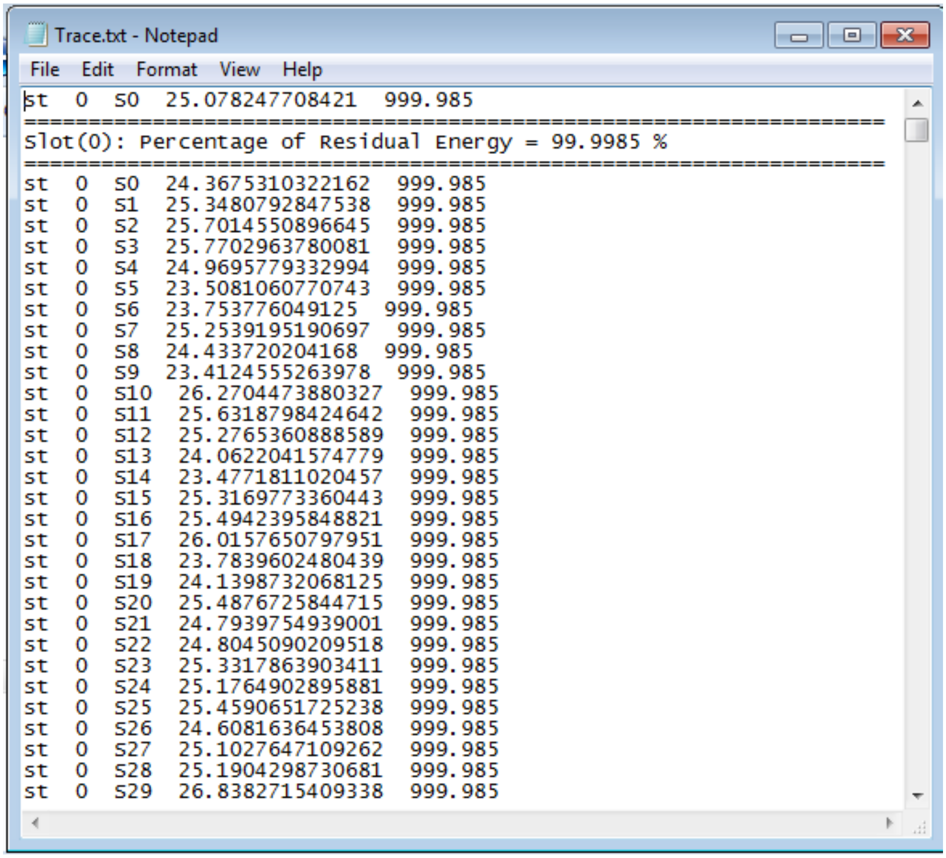


Figure 3.2 WSN Simulator - 8 Locators deployed

Figure 3.3 shows a sample of the resulted trace file that contains the amount of residual energy that is read from each sensor. It can be clearly seen that WSN generates the reading from all sensors in blocks that represent the time slots for every iteration in the simulation.




```
st   0   S0   25.078247708421   999.985
====================================================================
Slot(0): Percentage of Residual Energy = 99.9985 %
====================================================================
st   0   S0   24.3675310322162   999.985
st   0   S1   25.3480792847538   999.985
st   0   S2   25.7014550896645   999.985
st   0   S3   25.7702963780081   999.985
st   0   S4   24.9695779332994   999.985
st   0   S5   23.5081060770743   999.985
st   0   S6   23.753776049125   999.985
st   0   S7   25.2539195190697   999.985
st   0   S8   24.433720204168   999.985
st   0   S9   23.4124555263978   999.985
st   0   S10   26.2704473880327   999.985
st   0   S11   25.6318798424642   999.985
st   0   S12   25.2765360888589   999.985
st   0   S13   24.0622041574779   999.985
st   0   S14   23.4771811020457   999.985
st   0   S15   25.3169773360443   999.985
st   0   S16   25.4942395848821   999.985
st   0   S17   26.0157650797951   999.985
st   0   S18   23.7839602480439   999.985
st   0   S19   24.1398732068125   999.985
st   0   S20   25.4876725844715   999.985
st   0   S21   24.7939754939001   999.985
st   0   S22   24.8045090209518   999.985
st   0   S23   25.3317863903411   999.985
st   0   S24   25.1764902895881   999.985
st   0   S25   25.4590651725238   999.985
st   0   S26   24.6081636453808   999.985
st   0   S27   25.1027647109262   999.985
st   0   S28   25.1904298730681   999.985
st   0   S29   26.8382715409338   999.985
```

Figure 3.3 WSN- The simulation result file

From Figure 3.3, WSN does not have a feature to estimate the distance between the antenna and the sensor based on the power loss factor.

As WSN is an open source tool, the source has been imported to implement the power loss-based distance estimation function. Figure 3.4 shows the file generated:

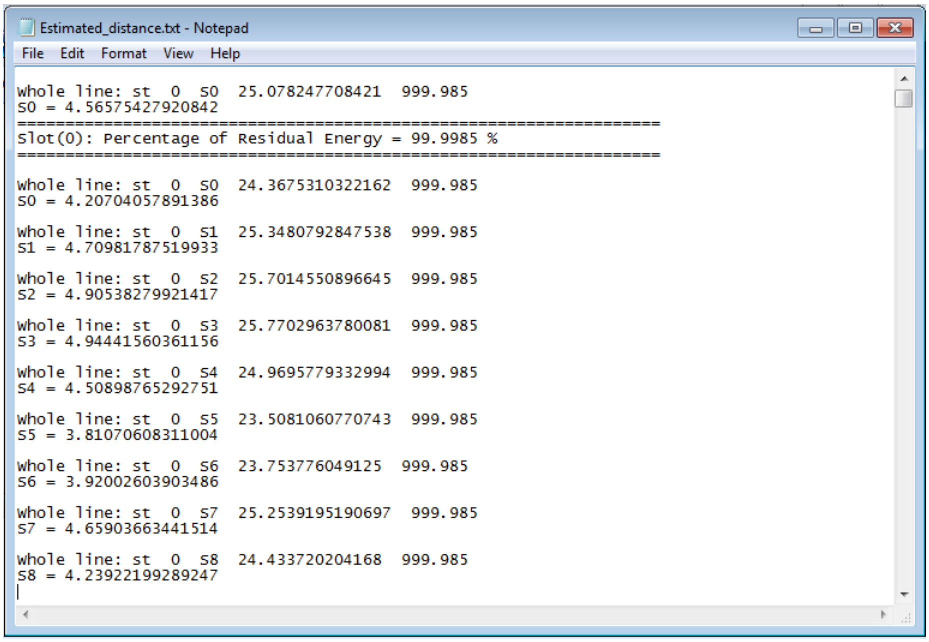


Figure 3.4 WSN Simulator- Estimated Distance result file

The distance is calculated using the power loss values, the antenna frequency (which has been assumed to be fixed at the beginning of the simulation based on the chosen hardware). The distance can be measured using the power loss equation 3.1 [66]:

$$P_{Loss} = 20.\log\frac{4\pi.d}{\lambda} \tag{3.1}$$

where λ is the wavelength, d is the distance, P is the power loss.

From equation 3.1, the estimated distance can be calculated if the other variables in the equation can be determined, the distance formula is shown in equation (3.2):

$$d = \frac{\lambda.e^{\frac{P_{Loss}}{20}}}{4.\pi} \tag{3.2}$$

Figure 3.5 shows the C# code of translating the above equation:

```
private double calculate_Distance_PowerLoss(double powerLoss) {
    double estimatedDistance = 0;
    double pi = Math.PI;

    // wavelength = speed of light / frequency
    double waveLength = 0.32397408207343414;
    double temp = 0, temp2 = 0;
    Console.WriteLine("Value of Pie: " + pi);
    Console.WriteLine("Wavelength: " + waveLength);

    temp = waveLength / 4 * pi;
    temp2 = Math.Pow (10, (powerLoss / 20));
    estimatedDistance = temp * temp2;
    return estimatedDistance;

}
```

Figure 3.5 WSN Simulator- Calculate distance function based on power loss value

The frequency used in the code is the frequency for the chosen antenna, SkyRFID SKYA902RHP9. Figure 3.5 shows the wavelength was calculated by dividing the speed of light by the frequency.

```
private void estimate_Location() {
    double estimated_distance = 0;
    double powerLoss = 0;
    string path = @"D:\Trace\Trace.txt";

    System.IO.StreamWriter file =
        new System.IO.StreamWriter(@"D:\Trace\Estimated_distance.txt");
    using (StreamReader sr = File.OpenText(path))
    {
        string s = "";
        while ((s = sr.ReadLine()) != null)
        {
            if (!s.StartsWith ("=") && !s.StartsWith ("Slot")) {
                Console.WriteLine ("Whole line: " + s);
                string[] words = s.Split (' ');
                Console.WriteLine ("Residual Energy for " + words[6] + " is : " + words [6]);
                words [6] = words [6].Replace (" ", String.Empty);
                    //file.WriteLine ("location 3:" + words [3] + "location 6:" + words [6] +"loc
                powerLoss = Double.Parse (words [6]);

                //powerLoss = powerLoss / 100;
                estimated_distance = calculate_Distance_PowerLoss (powerLoss);
                file.WriteLine (words [4] + " = " + estimated_distance);
            } else {
                file.WriteLine (s);
            }
        }
    }
    file.Close();
}
```

Figure 3.6 WSN Simulator- The location estimate function

Figure 3.6 shows the function that reads the estimated energy amount and pass it to the power loss calculation function. In order to simulate the library shelving process, the code has been modified to order the sensors in rows, as shown in Figure 3.7. As the number of sensors, which represent the RFID tags increases, the distance between the tags decreases, they are aligned more closely.

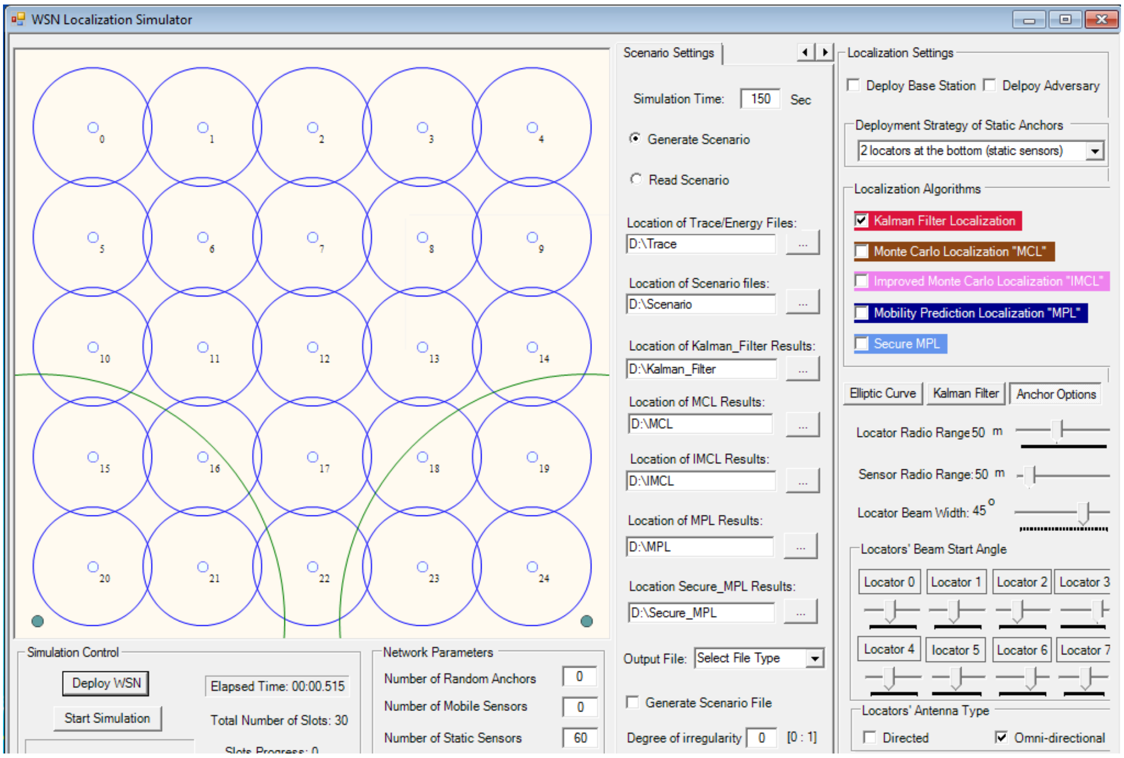


Figure 3.7 WSN simulation window - Post-Code modifications to locate tags in columns

The two challenges that were faced in this simulation part can be summarized as following:

1. The software does not identify the sensors detected by every antenna. The sample text file, the result files show that all tags have been identified per iteration, which implies that the simulation tool does not categorize the tags detected by the deployed antennas.
2. The result files show that all tags are identified, which implies that there is no tag miss. WSN assumes that the antennas are spatially distributed in the simulation window, the tags are completely covered by the deployed antennas.

As the above challenges could not be overcome with the existing WSN simulation tool, because of missing library source code. A new simulation software has been designed and developed which will be explained in details in the following section.

3.3 Developing a New Simulation Software

The main focus of this part of the research was to develop an intelligent and user friendly Wireless Sensor Network simulation software. This section is aiming to explore the main and most important components of a newly developed WSN to simulate a network of randomly distributed sensors, fixed antennas, which can be controlled dynamically at runtime via drop-down boxes, buttons and text-boxes.

The new software must achieve the following:

- The simulation window must draw distinctively antennas, sensors, and all items on the window must be uniquely numbered.
- The coverage area of each antenna's radio wave must be outlined and coloured distinguishably from sensor areas.
- The simulation tool must generate at least two result files: (a) A file contains the actual distances between the sensors, which represent the RFID tags, and the antennas. Which will be used to validate the simulation results. (b) A file contains the estimated distance measured by each antenna for each detected sensor. This file should contain the estimated distance in pixels, in metres, and the power loss values.

According to Chhimwal et al. [67], the following components need to be considered when building a WSN system:

- Nodes: each node is a physical device monitoring a set of physical variables. Nodes communicate with each other via a common radio channel [67].
- Environment: the main difference between classical and WSN models are the additional 'environment' component. This component models the generation and propagation of events that are sensed by the nodes, and also triggers sensor actions, i.e. communication among nodes in the network. The events of interest are generally a physical magnitude as sound or seismic waves or temperature [67].
- Radio channel: it characterizes the propagation of radio signals among the nodes in the network. Very detailed models use a '*terrain*' component, connected to the environment and radio channel components. The terrain component is taken

into consideration to compute the propagation as part of the radio channel, and also influences the physical magnitude [67].

- Sink nodes: these are special nodes that, if present, receive data from the sensors, and process it. They may interrogate sensors about an event of interest. The use of sinks depends on the application and the tests performed by the simulator [67].
- Agents: the agent may cause a variation in a physical magnitude, which propagates through the environment and stimulates the sensor. This component is useful when its behaviour can be implemented independently from the environment, e.g., a mobile vehicle. Otherwise, the environment itself can generate events [67].

The developed code comprises a number of different java methods related to painting corners, nodes, sensors, circles and so on, as well as many other methods that are called internally within the other methods to deploy the simulation and generate the result files. The chosen programming language is java, the reason for choosing Java over the other programming languages is explained in the next section.

Java is designed to enable development of portable, high-performance applications for a wide range of computing platforms possible. By making applications available across heterogeneous environments, can provide more services and enhance end-user productivity, and dramatically reduce the cost of ownership of both enterprise and consumer applications. Java has become invaluable to developers by enabling them to [68]:

- Write software on one platform and run it on virtually any other platform.
- Create programs that can run within a web browser and access available web services.
- Develop server-side applications for online forums, stores, polls, Hyper Text Markup Language (HTML) forms processing.
- Combine applications or services using the Java language to create highly customized applications or services.
- Powerful and efficient applications for wireless modules, sensors, gateways, consumer products, and practically any other electronic device [68].

Depending on the solution design; the basic simulation of RFID system requires covering the radio frequency signals transmitted by at least two main RFID components, RFID reader antenna, and the tag. Which must be clearly represented in the simulation window. The results of the simulations are dependent on few factors, which can be listed in the following:

- The assumption that the programmer makes usually to describe the physical layer objects in the simulation application, which may deviate from the actual description of the hardware in real life which may cause a variance in behaviour between the object in simulation and real life [67]. This must be fixed by the programmer at the early steps of building the simulator, the programmer needs to describe the hardware based on the manufacturer documented specifications, this should be as accurate as possible.
- The noise introduced to the RFID signal in real-life must be taken in consideration, which is an important factor for the programmer to get accurate results that are close to real hardware test. This can be solved by introducing a Kalman Filter to the simulation results.
- Polarization is another important consideration that every researcher needs to take into consideration when simulating RFID reader antennas and tags. For maximizing tag range, antenna polarization of the tag must match that of the reader antenna, both must be described in the simulation.

Figure 3.8 shows the simulation window initially at start-up. The dimensions of the simulation window are fixed during the application run, which is important to preserve fixed points to draw the antennas on, which simulates the antennas' fixed-point positions in reality, which results in building a robust estimation procedures.

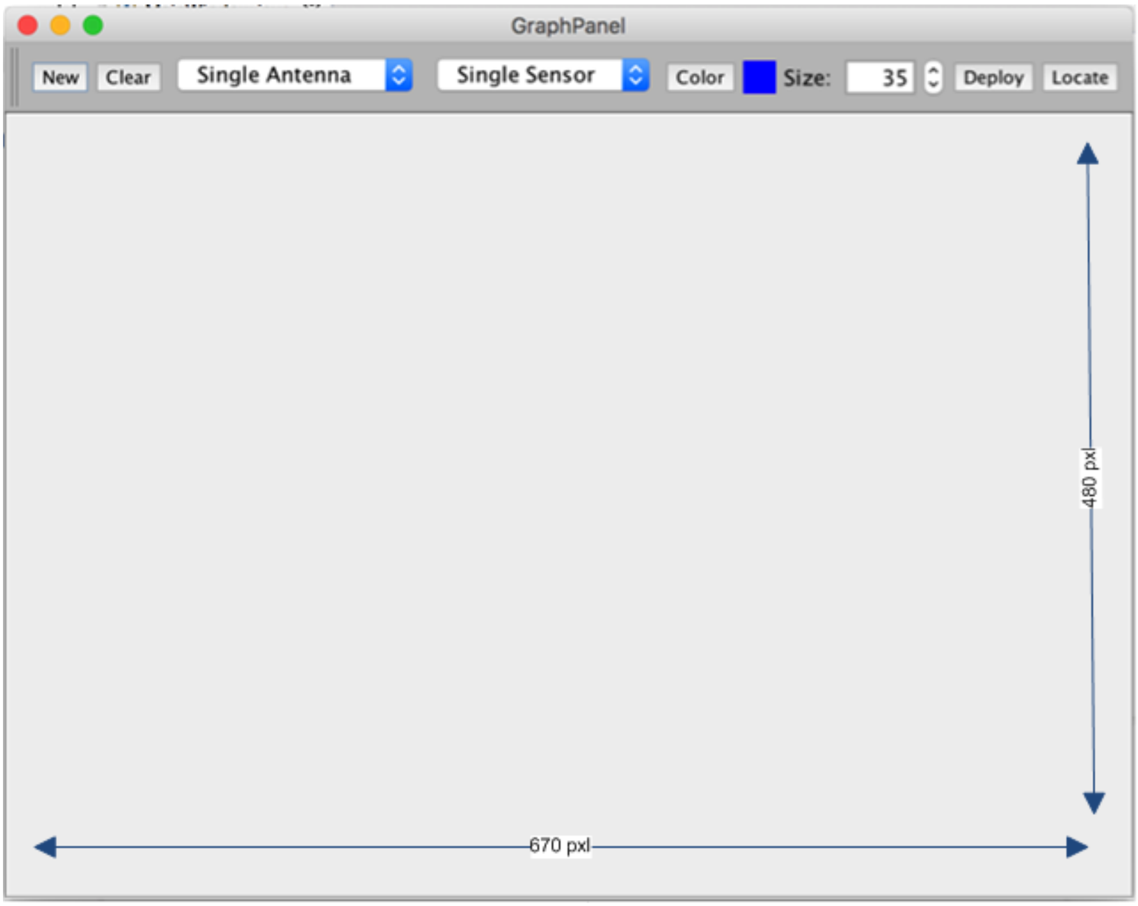


Figure 3.8 RFIDSim application home screen

The suitable simulation window is (670×480 pixels). The first antenna is placed on the exact coordinates (20×20 pixels), which represents point (0, 0) in reality, which is the reference point that will be used to locate objects from, for objects localized from antenna 1. The second antenna is located on the opposite corner of antenna 1, which has the coordinates of (650, 450), which means that the detectable simulation window size is ($650 - 20 = 630$ pixels), the sensors lie within that area are detectable from the nearest antenna(s). The simulation distance in pixels is 630 pixel, which represent 20 metres in real life; i.e. 1 metre in real life is equal to 31.5 pixel in our simulation.

According to the manufacturer of SkyRFID, the minimum detection range of antenna SKYA902RHP9 is 10 metres in a worst case scenario. So, the simulation space is divided so all the sectors in the simulation space is detectable by at least 3 antennas.

To simplify the process, all the items drawn in the application window are called '*Nodes*', including antennas, window corners, sensors, and drop boxes. The application starts by initializing buttons, drop-boxes, text-boxes, and the window main frame. Then the application creates and attach an event handler for each item with a changeable value in the window. This allows the user to see the effect of changing the value of the item that was changed.

The antenna's drop-down box has an event handler associated to it, which picks up the chosen number of antennas from the drop-down box, the following method in Figure 3.9 shows the antenna event handler.

```
        public AntennaComboAction(String name) {
            super(name);
        }

        public void actionPerformed(ActionEvent e) {
            System.out.println("Antenna Action performed");
            JComboBox combo = (JComboBox) e.getSource();
            String antennaSt = (String) combo.getSelectedItem();
            for (Antenna a : Antenna.values()) {
                if (a.ONE.toString().equals(antennaSt)) {
                    Node.updateAntenna(nodes, Antenna.ONE);
                    antenna = Antenna.ONE;
                } else if (a.TWO.toString().equals(antennaSt)) {
                    Node.updateAntenna(nodes, Antenna.TWO);
                    antenna = Antenna.TWO;
                } else if (a.THREE.toString().equals(antennaSt)) {
                    Node.updateAntenna(nodes, Antenna.THREE);
                    antenna = Antenna.THREE;
                } else if (a.FOUR.toString().equals(antennaSt)) {
                    Node.updateAntenna(nodes, Antenna.FOUR);
                    antenna = Antenna.FOUR;
                } else if (a.EIGHT.toString().equals(antennaSt)) {
                    Node.updateAntenna(nodes, Antenna.EIGHT);
                    antenna = Antenna.EIGHT;
                }
            }
            System.out.println("Repainting after adding antennas");
            repaint();
        }
    }
```

Figure 3.9 RFIDSim - Antenna configuration

This method retrieves the chosen item from the combo box, and assign that value to a global variable that holds the number of antennas to be drawn.

The next drop box item sets the number of sensors to be drawn in the simulator, the event handler here sets a global variable that holds the number of sensors to be drawn, see Figure 3.10:

```java
private class SensorComboAction extends AbstractAction {

    public SensorComboAction(String name) {
        super(name);
    }

    public void actionPerformed(ActionEvent e) {
        JComboBox combo = (JComboBox) e.getSource();
        String sensorSt = (String) combo.getSelectedItem();
        System.out.println("Action performed");
        for (Sensor a : Sensor.values()) {
            if (a.ONE.toString().equals(sensorSt)) {
                Node.updateSensor(nodes, Sensor.ONE);
                sensor = Sensor.ONE;
                identifiedBy = new String[1];
            } else if (a.THIRTY.toString().equals(sensorSt)) {
                Node.updateSensor(nodes, Sensor.THIRTY);
                sensor = Sensor.THIRTY;
                identifiedBy = new String[30];
            } else if (a.SIXTY.toString().equals(sensorSt)) {
                Node.updateSensor(nodes, Sensor.SIXTY);
                sensor = Sensor.SIXTY;
                identifiedBy = new String[60];
            } else if (a.HUNDRED.toString().equals(sensorSt)) {
                Node.updateSensor(nodes, Sensor.HUNDRED);
                sensor = Sensor.HUNDRED;
                identifiedBy = new String[100];
            } else if (a.HUNDRED_TWENTY.toString().equals(sensorSt)) {
                Node.updateSensor(nodes, Sensor.HUNDRED_TWENTY);
                sensor = Sensor.HUNDRED_TWENTY;
                identifiedBy = new String[120];
            }
        }
        System.out.println("Repainting after adding sensors");
        repaint();
```

Figure 3.10 RFIDSim - Sensor Configuration

By clicking on '*Deploy*' button; the deploy action handler is called, which loops on all the graphic nodes in the application, decides the colour, width, height, and the type of the node, see Figure 3.11:

```java

private class DeployAction extends AbstractAction {

    public DeployAction(String name) {
        super(name);
    }

    public void actionPerformed(ActionEvent e) {
        // for (int i = 0; i < 16; i++) {
        Point p = new Point(rnd.nextInt(getWidth()),
                rnd.nextInt(getHeight()));
        nodes.add(new Node(p, radius, new Color(rnd.nextInt()), kind,
                antenna, sensor));
        // }
        repaint();
    }
}
```

Figure 3.11 RFIDSim- Deploy simulation action

The application retrieves the number of sensors to be deployed from the index of the enumeration chosen by the user during the simulation setup. A Random generator class is instantiated, which is used in a loop to generate random numbers to represent the coordinates of the sensor. Once the sensor is drawn, the number corresponding to that sensor is also drawn using the graphic method '*drawString*', see Figure 3.12.

```
if (this.sensor.equals(Sensor.SIXTY)) {
    Random randomGenerator = new Random();
    g.setColor(Color.DARK_GRAY);
    for (int i = 1; i <= 60; i++) {
        int randomX = randomGenerator.nextInt(650);
        int randomY = randomGenerator.nextInt(550);
        System.out.println("Painting sensor " + i + "(X,Y): ("
                + randomX + ", " + randomY + ")");
        g.fillRoundRect(randomX, randomY, 5, 5, r, r);
        g.drawString(String.valueOf(i), randomX + 5, randomY + 8);
        xCoordinates
                .put(String.valueOf(i), String.valueOf(randomX));
        yCoordinates
                .put(String.valueOf(i), String.valueOf(randomY));
    }
}
```

Figure 3.12 RFIDSim - Using Random Generator class

When the deploy action finishes the drawing section, it generates two Hash Maps that contain the sensor numbers and the corresponding locations of the distributed sensors in order to be used for location estimation.

Locating the sensors in the simulation starts by clicking on '*Locate*' button; The Locate class is a subclass that is called within the main class; it contains all the logic that does the locating and the result file generation.

When the Locate action is performed, the below action handler is called, which creates a file handle, opens the file, and calls the method '*generateTagLocationFile*', which does the calculation for all the sensors, see Figure 3.13.

```
public void actionPerformed(ActionEvent e) {
    //
    // Each antenna covers 450 Pi. which covers a space of 20 meters.
    // Each 1 meter in real life is 22.5 Pi in Simulation.
    //
    System.out.println("Estimating Distance and writting to file");
    PrintWriter writer;
    try {
        writer = new PrintWriter(
                "/Users/alialwadi/Documents/workspace/RFIDSimulator/Bin/Estimated_distance.txt",
                "UTF-8");
        writer.println("Estimated distance:");
        writer.println("-----------------------------------------------------------");
        writer.println("-----------------------------------------------------------");
        writer.println("-----------------------------------------------------------");
        writer.close();
    } catch (FileNotFoundException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    } catch (UnsupportedEncodingException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    generateTagLocationFile();
}
```

Figure 3.13 WSN – Generate Tag location file

Figure 3.14 shows '*generateTagLocationFile*' method; the method instantiates and initializes an iterator that holds the items of each hashmap entry in order to be fetched and evaluated against the distance estimation method.

```
private void generateTagLocationFile() {
    double distance;
    double powerLoss = 0;
    System.out.println("Generating tag location file");
    PrintWriter writer;
    try {
        writer = new PrintWriter(
                "/Users/alialwadi/Documents/workspace/RFIDSimulator/Bin/Tag_Location.txt",
                "UTF-8");
        writer.println("Tag locations: ");
        writer.println("-----------------------------------------------------------");
        Iterator it = xCoordinates.entrySet().iterator();
        Iterator it2 = yCoordinates.entrySet().iterator();
        while (it.hasNext()) {
            Map.Entry pair = (Map.Entry) it.next();
            Map.Entry pair2 = (Map.Entry) it2.next();
            System.out.println("Tag " + pair.getKey() + " : "
                    + pair.getValue() + ", " + pair2.getValue());
            writer.println("Tag " + pair.getKey() + " : "
                    + pair.getValue() + ", " + pair2.getValue());
        }
```

Figure 3.14 RFIDSim- Generate tag location file method

First, the method evaluates the distance between the antenna and the sensor by comparing the x,y coordinates of the sensor against the x,y coordinates of the deployed antennas, as shown in Figure 3.15.

MainWindow.java

```java
        writer.println("-----------------------------------------------------------");
        if (antenna.equals(Antenna.ONE)) {
            System.out.println("One Antenna only");
            Iterator i = xCoordinates.entrySet().iterator();
            Iterator i2 = yCoordinates.entrySet().iterator();
            writer.println("Tags identified by Antenna 1 :");
            while (i.hasNext()) {
                Map.Entry pair = (Map.Entry) i.next();
                Map.Entry pair2 = (Map.Entry) i2.next();
                System.out.println("Tag " + pair.getKey() + " : "
                        + pair.getValue() + ", " + pair2.getValue());
                // Antenna 1:
                if (Integer.valueOf(pair.getValue().toString()) >= minCoordinates[0][0]
                        && Integer.valueOf(pair.getValue().toString()) <= maxCoordinates[0][0]
                        && Integer.valueOf(pair2.getValue().toString()) >= minCoordinates[0][1]
                        && Integer.valueOf(pair2.getValue().toString()) <= maxCoordinates[0][1])
                    distance = calculateTagAntennaDistance(
                            Integer.valueOf(pair.getKey().toString()),
                            20,
                            20,
                            Integer.valueOf(pair.getValue().toString()),
                            Integer.valueOf(pair2.getValue().toString()));
                    double actualDistance = calculateActualDistance(distance);
                    double randomPowerLoss = generatePowerLoss(actualDistance);
```

Figure 3.15 RFIDSim- The evaluation of the distance between the antenna and sensors

If the sensor location lies within the radio wave range of the antenna, the actual distance is calculated and generated using Equation (3.3):

$$d = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{3.3}$$

where X_1, Y_1, and X_2, Y_2 are the coordinates of the sensor and the antenna.

The conversion of the above equation into java is shown in Figure 3.16:

```java
    private double calculateActualDistance(double pixDistance) {
        double distance = 0;
        int simulationSpace = 630;// Simulation window is a representation
                                   // of 20 Meters. 650 - 20 = 630
        simulationSpace = simulationSpace / 20; // Divide by 20 meters to
                                                 // get the distance in
                                                 // meters.

        distance = pixDistance / simulationSpace;
        return distance;

    }
```

Figure 3.16 RFIDSim- Calculate the actual distance method

A random power loss value is calculated, which is proportional to the distance of the sensor from the closest antenna shown in Figure 3.17.

```
    private double generatePowerLoss(double distance) {
        distance = (int) distance;
        double pi = Math.PI;

        // wavelength = speed of light / frequency
        // Speed of light: 299,792,458
        // frequency: 868 000 000
        double waveLength = 0.345622119815 6682;// double waveLength =
                                                // 0.32397408207343414;
        double temp = 0, temp2 = 0;
        System.out.println("Value of Pie: " + pi);
        System.out.println("Wavelength: " + waveLength);
        System.out.println("Random distance value: " + distance);

        temp = (4 * pi * distance) / waveLength;
        // temp2 = Math.log10(powerLoss / 20);
        temp2 = 20 * Math.log10(temp);
        System.out.println("Power Loss value before Kalman Filter: "
                + temp2);

        return temp2;
    }
```

Figure 3.17 RFIDSim- Generate random power loss value

The power loss value is generated based on Equation (1.1). The distance value of the sensor is substituted in the equation to get the power loss value. When the power loss value is calculated, it is passed to the method that estimates the distance between the antenna and the tag, based on the power loss, as shown in Figure 3.18.

```
    private double calculate_Distance_PowerLoss(double powerLoss) {
        double estimatedDistance = 0;
        double pi = Math.PI;

        // wavelength = speed of light / frequency
        // Speed of light: 299,792,458
        // frequency: 868 000 000
        double waveLength = 0.345622119815 6682;// double waveLength =
                                                // 0.32397408207343414;
        double temp = 0, temp2 = 0;
        System.out.println("Value of Pie: " + pi);
        System.out.println("Wavelength: " + waveLength);
        System.out.println("Power Loss value: " + powerLoss);

        temp = waveLength / 4 * pi;
        // temp2 = Math.log10(powerLoss / 20);
        temp2 = Math.pow(10, (powerLoss / 20));
        System.out.println("Temp2: " + temp2);
        System.out.println("Temp: " + temp);
        estimatedDistance = temp * temp2;
        estimatedDistance = estimatedDistance/10;
        return estimatedDistance;

    }
```

Figure 3.18 RFIDSim- Estimate the location method

The simulation results for four antennas and 30 sensors (which represent RFID tags in this research). Figure 3.19 shows the purple lines represent the maximum coverage that the radio frequency signal of the antennas can cover. The antennas are fixed at the corners of the simulation window, as recommended by the manufacturer of the chosen antenna.

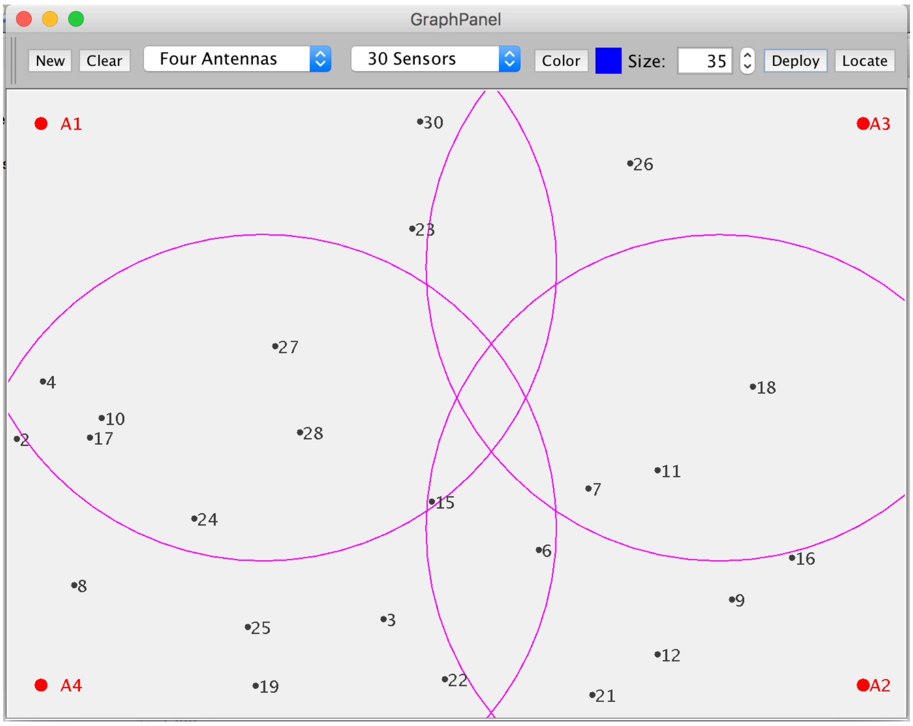


Figure 3.19 RFIDSim- Deployed simulation for 4 antennas and 30 sensors

A sample result file is shown in Figure 3.20. The position of the tag is displayed for validation purposes, the distance in pixels comes next, then the actual distance is calculated by converting the pixel distance to metric value, which is then divided by the simulation window dimensions.

Antenna_Detection.txt

```
Identified Tags for each Antenna:
-----------------------------------------------------------
Tags identified by Antenna 1 :
Tag 17 identified, Position:(60, 263), Distance(Pixel): 246.27017683836587, Actual Distance: 7.944199252850512, PowerLoss: 48.11412748986365, Est. Distance:6.9087230807625515
Tag 15 identified, Position:(322, 312), Distance(Pixel): 420.0809445809224, Actual Distance: 13.55099821228782, PowerLoss: 53.49103373571525, Est. Distance:12.83048572141617
Tag 23 identified, Position:(307, 103), Distance(Pixel): 298.7607738643077, Actual Distance: 9.637444318203473, PowerLoss: 50.29701687836501, Est. Distance:8.88264396098042
Tag 24 identified, Position:(140, 325), Distance(Pixel): 327.75753233144775, Actual Distance: 10.572823623595088, PowerLoss: 51.21216668957851, Est. Distance:9.869604401089358
Tag 27 identified, Position:(202, 193), Distance(Pixel): 251.10356429170812, Actual Distance: 8.100114977151875, PowerLoss: 49.273966429417385, Est. Distance:7.895683520871489
Tag 28 identified, Position:(221, 259), Distance(Pixel): 312.28512612674973, Actual Distance: 10.073713746024184, PowerLoss: 51.21216668957851, Est. Distance:9.869604401089358
Tag 2 identified, Position:(4, 264), Distance(Pixel): 244.5240274492468, Actual Distance: 7.88787185320151, PowerLoss: 48.11412748986365, Est. Distance:6.9087230807625515
Tag 10 identified, Position:(69, 248), Distance(Pixel): 233.20591759215716, Actual Distance: 7.522771535230876, PowerLoss: 48.11412748986365, Est. Distance:6.9087230807625515
Tag 30 identified, Position:(313, 21), Distance(Pixel): 293.00170647967224, Actual Distance: 9.45166795095717, PowerLoss: 50.29701687836501, Est. Distance:8.88264396098042
Tag 4 identified, Position:(24, 220), Distance(Pixel): 200.0399960007998, Actual Distance: 6.452903096799993, PowerLoss: 46.77519169725139, Est. Distance:5.921762640653618
--------------------------------------------------
Tags identified by Antenna 2 :
Tag 18 identified, Position:(568, 224), Distance(Pixel): 584.7392581313486, Actual Distance: 18.862556713914472, PowerLoss: 56.31761679164463, Est. Distance:17.76528792196084
Tag 15 identified, Position:(322, 312), Distance(Pixel): 420.0809445809224, Actual Distance: 13.55099821228782, PowerLoss: 53.49103373571525, Est. Distance:12.83048572141617
Tag 16 identified, Position:(598, 355), Distance(Pixel): 668.063619724948, Actual Distance: 21.550439345966065, PowerLoss: 57.6565525842569, Est. Distance:20.726169242287664
Tag 11 identified, Position:(495, 288), Distance(Pixel): 545.3888521046246, Actual Distance: 17.593188777568535, PowerLoss: 55.821145117144, Est. Distance:16.778327481851917
Tag 12 identified, Position:(495, 429), Distance(Pixel): 626.8221438334801, Actual Distance: 20.220069155918715, PowerLoss: 57.232766602858135, Est. Distance:19.739208802178716
Tag 22 identified, Position:(332, 448), Distance(Pixel): 529.648940336899, Actual Distance: 17.085449688287067, PowerLoss: 55.821145117144, Est. Distance:16.778327481851917
Tag 7 identified, Position:(442, 302), Distance(Pixel): 507.5509826608555, Actual Distance: 16.372612343898563, PowerLoss: 55.294566342697, Est. Distance:15.791367041742964
Tag 6 identified, Position:(404, 349), Distance(Pixel): 505.6649088081948, Actual Distance: 16.311771251877254, PowerLoss: 55.294566342697, Est. Distance:15.791367041742964
Tag 9 identified, Position:(552, 387), Distance(Pixel): 646.3072024973883, Actual Distance: 20.848619435399623, PowerLoss: 57.232766602858135, Est. Distance:19.739208802178716
--------------------------------------------------
Tags identified by Antenna 3 :
Tag 18 identified, Position:(568, 224), Distance(Pixel): 584.7392581313486, Actual Distance: 18.862556713914472, PowerLoss: 56.31761679164463, Est. Distance:17.76528792196084
Tag 11 identified, Position:(495, 288), Distance(Pixel): 545.3888521046246, Actual Distance: 17.593188777568535, PowerLoss: 55.821145117144, Est. Distance:16.778327481851917
Tag 26 identified, Position:(474, 53), Distance(Pixel): 455.1977592211983, Actual Distance: 14.683798684554784, PowerLoss: 54.13472740314327, Est. Distance:13.817446161525107
Tag 7 identified, Position:(442, 302), Distance(Pixel): 507.5509826608555, Actual Distance: 16.372612343898563, PowerLoss: 55.294566342697, Est. Distance:15.791367041742964
Tag 6 identified, Position:(404, 349), Distance(Pixel): 505.6649088081948, Actual Distance: 16.311771251877254, PowerLoss: 55.294566342697, Est. Distance:15.791367041742964
--------------------------------------------------
Tags identified by Antenna 4 :
Tag 17 identified, Position:(60, 263), Distance(Pixel): 246.27017683836587, Actual Distance: 7.944199252850512, PowerLoss: 48.11412748986365, Est. Distance:6.9087230807625515
Tag 15 identified, Position:(322, 312), Distance(Pixel): 420.0809445809224, Actual Distance: 13.55099821228782, PowerLoss: 53.49103373571525, Est. Distance:12.83048572141617
Tag 22 identified, Position:(332, 448), Distance(Pixel): 529.648940336899, Actual Distance: 17.085449688287067, PowerLoss: 55.821145117144, Est. Distance:16.778327481851917
Tag 24 identified, Position:(140, 325), Distance(Pixel): 327.75753233144775, Actual Distance: 10.572823623595088, PowerLoss: 51.21216668957851, Est. Distance:9.869604401089358
Tag 25 identified, Position:(181, 408), Distance(Pixel): 420.0773738253466, Actual Distance: 13.550883026624085, PowerLoss: 53.49103373571525, Est. Distance:12.83048572141617
Tag 28 identified, Position:(221, 259), Distance(Pixel): 312.28512612674973, Actual Distance: 10.073713746024184, PowerLoss: 51.21216668957851, Est. Distance:9.869604401089358
Tag 3 identified, Position:(285, 402), Distance(Pixel): 464.91827238773914, Actual Distance: 14.997363625410939, PowerLoss: 54.13472740314327, Est. Distance:13.817446161525107
Tag 2 identified, Position:(4, 264), Distance(Pixel): 244.5240274492468, Actual Distance: 7.88787185320151, PowerLoss: 48.11412748986365, Est. Distance:6.9087230807625515
Tag 10 identified, Position:(69, 248), Distance(Pixel): 233.20591759215716, Actual Distance: 7.522771535230876, PowerLoss: 48.11412748986365, Est. Distance:6.9087230807625515
Tag 6 identified, Position:(404, 349), Distance(Pixel): 505.6649088081948, Actual Distance: 16.311771251877254, PowerLoss: 55.294566342697, Est. Distance:15.791367041742964
Tag 4 identified, Position:(24, 220), Distance(Pixel): 200.0399960007998, Actual Distance: 6.452903096799993, PowerLoss: 46.77519169725139, Est. Distance:5.921762640653618
Tag 8 identified, Position:(48, 376), Distance(Pixel): 357.0994259306503, Actual Distance: 11.519336320343557, PowerLoss: 52.04002039274301, Est. Distance:10.85656484119829
```

Figure 3.20 RFIDSim- A sample result file for a simulation deployment

The newly developed simulation software fits the requirements of our RFID system in terms of the relation between the power loss, the estimated distance, and more importantly, it allows the user to choose the number of antennas, tags, at runtime. The potential to customize all the parameters is also there, which made this tool the perfect candidate to simulate RFID networks.

At this stage of the research; a network model was built for an RFID system to simulate the antenna, tag distribution in reality, and the radio frequency signal flow between these components. For a randomly chosen simulated model, the maximum distance detected between an antenna and an RFID tag, was '11.843525281', while the actual distance was '12.36094349'. The percentage of the error to the total distance is (12.36094349/11.843525281) × 100%, the total is 1.043 %, which can be considered an acceptable error range considering that this will be entered in a mean square error minimization function that will also gather more readings from the all the antennas that can reach the negotiated tag including that antenna that was used in the original reading.

The results in this chapter prove that implementing a location-aware service employing Radio Frequency Identification as a communication technology automating a library system with passive RFID tag infrastructure is feasible and can be achieved with even distribution of RFID antennas, well-designed RFID network, the appropriate middleware, and a library application with an accurate error function that minimizes the detected location error to 35 cm.

Chapter 4
Proposed Library Tracking System Based on Radio Frequency Identification

This chapter will cover the principles of implementing Location-Aware RFID service to automate a library system using the existing passive RFID tag infrastructure. It also covers the necessary hardware components required to automate a library system with, and the appropriate middleware that can be used to integrate with the RFID readers are all going to be explained. First; a high level discussion about the current library system architecture is discussed, the effect of using RFID on the upper application and System layers. Also the feasibility of adding a '*Location-Aware functionality'* to all system layers to identify the availability of each resource, and the physical location of each resource from the main computer.

Using radio frequency operated smart '*tags*' allows real-life objects to become '*smart objects*' that will actively communicate with computer applications connected to the network. The information '*sensed*' about their identities is exchanged while reacting to the '*physical world*'. This information is driven by robust middleware applications that activate the tags in the required areas and securely process tag information [31].

Automating a traditional library system can achieve the following:

- Reduce the human error to the minimum by automating the library return procedure using categorization of the library sections. This will be explained in details in the System Architecture section.
- Speed up, automate the borrow/return process by automating the administration process in borrow, return, and the automated arrear book alarms.
- Reduce the maintenance work in the library by categorizing books in the library according to the subject, and alarming the administrator of any misplaced book.
- Maintain the library items efficiently by tracking objects in the library, and alarm the administrator in case of any unauthorized object removal upon leaving the library gates.

The automation of tasks in a library is achievable based on concepts such as, data gathering, warehouse management, tagging, reporting or custom application processes. The library items will be tagged in order to be activated and interrogated at any time, regardless of the orientation of the object or the location of the RFID tag. The main computer must be able to identify and locate tags, based on a location abstraction depending on the individual tags for each resource in a fault-tolerant manner.

This research presents the feasibility, and applicability of adding location-aware services to the core layer (without affecting the running services). In order to enable the upper application layer to track and locate the smart entities. The existing tag readers must remain operational after adding the new service; this is essential to any library system. The main challenges in reusing the existing UHF RFID infrastructure are as follows:

- The solution must be compatible with the existing commercial passive UHF tags, which requires designing the library system based on antennas compatible with the existing RFID tag specifications.
- The reader antenna must be simple enough for very low-cost mass production, easy integration into different shapes and sizes of store structures, i.e. shelves, tables, security gateway, inclusive of existing structures.
- The reader antenna must be able to activate and read at least 20 books per second.

This chapter will also present a study of the library system components in order to redesign the system to automate the library business processes, taking advantage of the capabilities of the passive RFID reader antennas, and relying on a smart middleware application to handle the calculations to locate library objects at runtime. The revised library system will automate the following processes:

- The borrow and return processes can be automated, the system will use main reader antennas on the library gates to automatically scan in/out the books that readers take in/out the library.
- Accurately locating books inside the library on request with an accurate to 35 centimetres (cm). The librarian must be able to use an interface to request the location of a specific book.
- Based on the library book categories, the librarian must be warned if there is a book that has been returned to wrong shelf, if the book belongs to a different

subject category, the system should be able to retrieve this category from the database and warn the librarian about the location of the book and where it belongs.

- The communication between the RFID antennas and the main computer(s) should be secure.

4.1 System Architecture of Traditional Library System

An RFID tag may contain only a unique identification number a barcode or an accession number, which identifies that copy of the book, that is the only piece of data that can be communicated between the RFID system and the LMS system, this has made the inter-operation easier and more efficient. The required data are held in the LMS, which can be accessed using this key identifier [69].

The existing RFID tag infrastructre can be utilized to hold more information about the book, e.g. bibliographic information, the shelf that the book is held on, its previous lending history, supplier information. This information will have to be obtained from somewhere and loaded onto the tag. This data would normally come from the LMS, i.e. the interface between the two sets of hardware (RFID and LMS), it must be able to exchange this information. As RFID suppliers are now developing applications which utilize this type of data they need to be working closely with LMS suppliers and follow the latest RFID International Organization of Standardization (ISO) to standardize the set of data being exchanged [69].

The RFID technology provides a solution in managing, collecting, and distributing books effectively [70]. The RFID technology will bring significant savings in terms of staff costs, enhancing services, prevent book theft, provide a constant update of the library collections, and achieve real-time services [71, 72].

The RFID system offers a contactless identification, automatic retrieval of data, and wireless data storage. Data reading using RFID enhances performance and productivity by increasing the accuracy and speed of information communication. The tags, readers, and back-end servers are the three basic components needed in a basic library RFID system. The tag and the reader must work at the same specified frequency and conform to the same protocol to guarantee the compatibility of the communication system [55]. The tag and the reader antenna must be aligned to achieve the required polarization for the antenna to read the tag. The distance between the antenna and the tag must not exceed the maximum distance that a reader antenna can read [70].

The handheld reader antennas are connected to the librarian computers. The librarian computer is connected to the library network, which is connected to the backend server with a LAN network. The backend server uses middleware to filter and store all information for each specific RFID Tag, refer to Figure 2.1.

As previously explained in chapter 1, almost all libraries are equipped with high frequency (HF) RFID tags and readers operating internationally at a frequency of 13.56 MHz [71]. The main advantage of the 13.56 MHz operating system is the frequency is available in most countries. This frequency has been reserved for industrial, scientific and medical applications. However, the UHF RFID system surpasses the efficiency by increasing the reading range and providing multi-reading capacity compared to the HF RFID system in the library automation system [71] [65].

In a typical modern library system, according to ISO Standard 28560-1 [73], covers the exchanged information contains information set of data elements and general guidelines for implementation, to meet the needs for secure and efficient library system:

1. Circulation of library items.
2. Acquisition of library items.
3. Inter-library loan processes.
4. Data requirements of publishers, printers and other suppliers of library items.
5. Inventory and stock checking of items.

The importance of the ISO standard is to provide the framework to ensure interoperability between libraries in exchange of library items with RFID tags, the freedom of the library to acquire or renew equipment or library items from different software or vendors and interoperability of a single RFID application.

From this research, one issue was raised when the existing library systems were investigated, most of these systems do not comply with the ISO standard In fact; most of them just use the RFID tag as an identification only, storing all the other related information in the library system database, the system retrieves this information when the RFID tag is scanned. The design takes this case into consideration, as most of the existing RFID tags are not re-programmable, the system should keep track of these RFID tags in the database and exchange only the identification information.

The RFID tags that are found in the library of Auckland University of Technology are only used for anti-theft system; while the existing bar-code tag is used to identify the library object in general and track the borrow-return process.

4.1.1 RFID Tag Description

As previously covered in chapter 1, RFID tags can be passive, retrieving the necessary energy either from the interrogating wave or from a constant signal that some antennas transmit to power up the RFID Tag. The second type of tags is the active tag, which includes a battery as a power source, to enhance reading range. Passive tags are the cheapest to manufacture, do not require maintenance, more compact and lighter [16, 74]. World-assigned RFID frequency bands range from high frequency (HF) up to microwaves [23], see Table 4.1.

The allocated band at ultra high frequency ranges from 860 MHz to 930 MHz, divided into three sub bands corresponding to the three world regions. In general, UHF passive RFID tags present the lowest unit cost, which makes UHF RFID the preferred choice for mass applications, this type of RFID tags are the most commonly used in library systems. Table 4.1 shows the frequency bands and how the spectrum divided between them [68].

Table 4.1 Operating frequencies and performance characteristics

	Low frequency (LF)	High frequency (HF)	Ultra high frequency (UHF)	Microwave frequency (MF)
Frequency range	125-134kHz	13.56MHz	860-930MHz	2.45GHz
Read range(passive)	< 0.5 m	1.0m	3.0m	10m
Tag cost	High	Lower than LF tags	Lowest	High
Typical application	Tracking, Cardkey	Lower than LF tags Airline baggage handling, library book tracking, electronic article surveillance	Supply chain tracking, warehouse management	Electronic toll collection, Railroad monitoring

Frequency allocation is managed through regulations by governments. Which makes the frequency ranges specified in table 1 vulnerable for change between different countries. In Europe, the frequency range for UHF band is 902 MHz to 921 MHz, while the range for this band in New Zealand is narrower, from 915 MHz to 921 MHz, which limits the range for the allowed frequencies for RFID applications. The transmission frequency plays a sensitive role in the covered area that the RFID system can operate on. The majority of the RFID systems operate on one of four frequency bands: low frequency (LF), high frequency (HF), ultra high frequency (UHF), and microwave (MF) [3].

The data capacity for the RFID tag depends on the manufacturers' specifications. Some tags have capacity to store up to 2048 bits of information where most existing Library tags typically have space for 128-bits of information. This is adequate for current system demands, but trying to adhere with the minimum limit of 128-bit when designing a library system does not meet the ISO 28560-1 standard. This problems can be overcome by programming the tag to hold the Identification number only, the rest of the functionalities must be maintained in the middleware.

4.1.2 Middleware Architecture

The middleware is a repository software or device that connects RFID readers and the collected data with a unified data '*Repository*', providing the necessary APIs and integrates the library network components with each other on one side and with the repository on the other side [42].

The lowest layer in the hierarchy is the hardware layer, consists of the distributed physical RFID tags, hand-held reader antennas, fixed RFID reader antennas, librarian computers and all other library hardware. One layer above is the entity Read/Write (ERW) service, which defines a generic and unifying interface to the underlying physical passive RFID tag infrastructure, which uses RFID as the communication protocol. The Core Service Layer consists of generic services that operate with individual Smart Entities to achieve specific purposes they are designed for to serve the upper layers. The High-Level service layer is represented by a collection of specialized services that rely on the core services to provide service to the application layer on top in a robust, and fault-tolerant manner. Finally, the Application Layer contains the applications and services that are designed to serve the end-user and the library system in a consistent, interactive and fault-tolerant way. The application architecture is shown in Figure 4.2 [42].

The middleware architecture is extensible, facilitating the integration of additional services without affecting the existing components. For reliable operation, the redundancy resulting from the super-distribution of the RFID antennas is exploited by the middleware services for the realization of fault-tolerance mechanisms. Service upgrade or addition, and maintenance tasks are performed autonomously without affecting system performance and no downtime, thus reducing the need for manual intervention and servicing. For instance, the core middleware services support the integration of additional tags that are distributed at a later point in time without a noticeable service interruption [75]. Furthermore, the services provided by core service and high level service layers mask the complexities of applied fault-tolerance, self-organization, and self-calibration mechanisms as well as hardware-specific details from higher-level services and applications [19, 75].

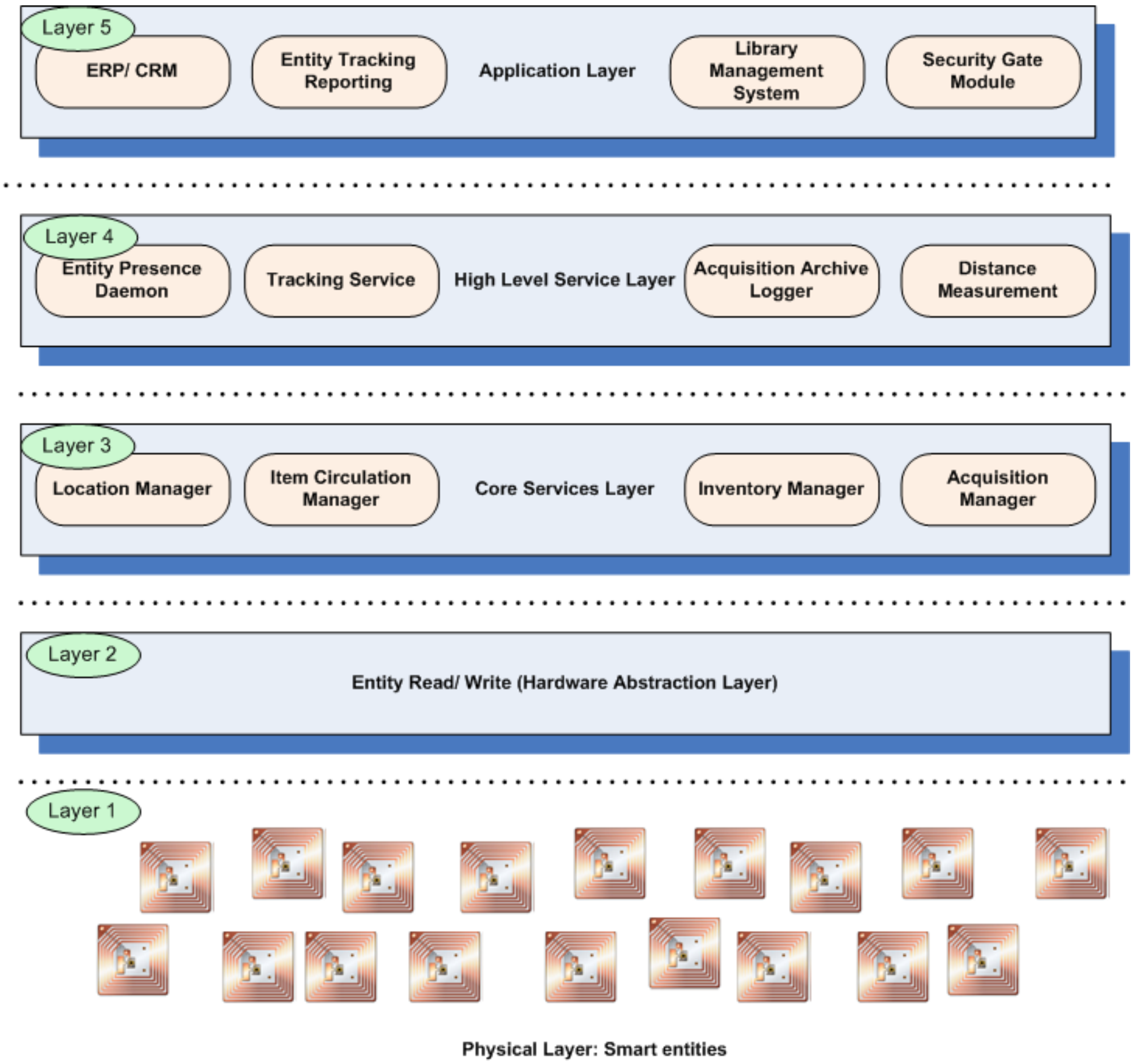


Figure 4.1 Library System Application Layer Architecture (Figure created by student)

4.2 Proposed Solution

The proposed library RFID system will be designed accurately to guarantee the coverage of the tags from at least three RFID antennas to guarantee that all tags are triangulated, for a simplified design for the aisles of a conceptual library. The solution consists of changing touching three layers:

1. ***Hardware Layer***, is the lowest layer, which contains the passive RFID tags, it was investigated to get the tag operating frequency. This is decided as to whether the RFID Tag infrastructure will need to be renewed if the operating frequency of the tags is in the 13.56 MHz range. As a result, this layer in our prototype solution consists of a distributed RFID tag infrastructure which

requires RFID antennas to be used to automate the information gathering and tracking.

2. ***Hardware Abstraction Layer*** is one layer above the hardware layer, which is where the communication between the entities takes place. Represented in the middleware, the engine that operates the application and integrates with the antennas processes signals from the tags.

3. ***Location Manager layer*** is responsible of locating the objects via a strong error-minimization function that interrogates all the passive RFID tags in the required area to obtain the required ID, and also interrogate this tag multiple times from multiple antennas in order to get the location, which introduces a new subject, '*Collision*' that will be discussed later.

All the layers above will be services and applications that rely on the layers below. In this layer, the LMS system will be built to track, locate and manage the library system. The library system will be an application that utilizes all the services in the layers below to track the library items and display the items' information to the librarian or the library user in graphical and useful format.

This research proposes a real-time locating and tracking technique based on the power loss value utilizing multiple spatially diverse reader antennas. Each antenna provides an RSSI value that depends on the distance that separates the tag from the antenna. Multiple antennas interrogating the same tag from different vantage points provide an RSSI dataset from which the tag location may be estimated [20]. Figure 4.3 shows the estimation procedure for sensing the location of the tags is based on spatial deployment of the reader antennas for maximum signal diversity [20]. The received signal strength detected by each reader antenna will be inserted into a mean-square error testing function. The error function is minimized at each test point in the interrogation zone to find the point with the minimum mean-square error [20]. The antennas must be distributed around the shelves depending on the maximum distance an antenna can cover, this depends on the antenna specifications. The proposed solution is using SkyRFID antenna model number SKYA902LPH106, which can cover an area of 20 to 21 metres. Taking into consideration, for this solution to work, each tag in the library must be covered by at least three antennas. According to the manufacturer's specifications, the number of simultaneous tags an antenna can read is 400 tags per second [63].

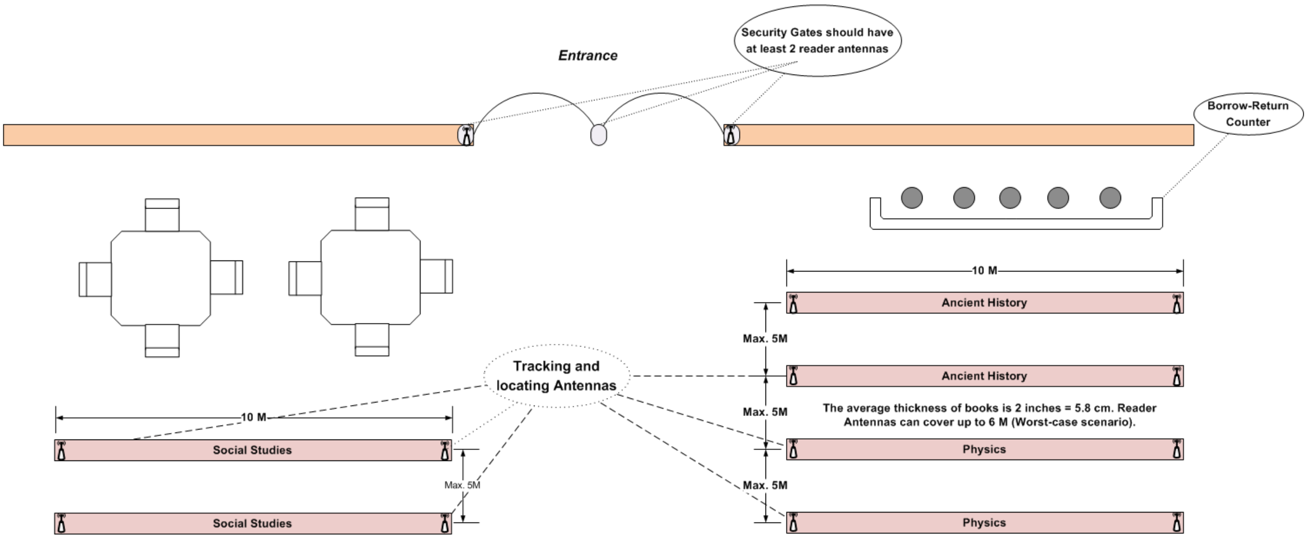


Figure 4.2 Library system design, antenna distribution (Figure created by student)

The average thickness of a library book is 2 inches, which is approximately equals to 5 cm [76]. Taking the worst-case scenario, each antenna can cover an area of 20 × 20 metres, 400 squared metres (m^2) in total. If this number is divided by 5cm, which is approximately the average thickness of a book, the result is 80 books, multiply this number with 5 the number of shelves in Auckland University of Technology (AUT) city campus library, the result is 400 books.

In order to guarantee that each book is covered by at least two antennas, the 20m distance must be divided into half, taking into consideration that the distance between two aisles is 5 metres.

Location estimating techniques can be classified as range-based and bearing-based. Range-based approaches trilaterate the transponder position using the estimated distance at reference points [28]. The existing tag locating techniques can be summarized into two main categories [20, 77]:

1. Received Signal Strength Indicator methods (RSSI-based) [20, 40, 78-80].
2. Phase-based methods [20, 40, 78-80].

RSSI-based localization has attracted considerable attention due to its simplicity and low cost [20]. RSSI utilizes the power loss value received from the passive RFID tag, as explained later on in this section, a single reader antenna may be able to determine the range of a tag based on the RSSI indicator, which is proven by the simulation model built in chapter 3. It requires multiple spatially diverse reader antennas to estimate the coordinates of a tag unambiguously and also to triangulate the tag location in the 3D world [75, 76].

Existing RSSI-based position techniques typically use radio maps of the environment of interest, or placement of reference tags, or both. The radio map method is a fingerprinting based positioning technique. RSSI values are measured at known locations to build a database; which is called a radio map. For tag locating, the measured RSSI value is compared against RSSI values in the database, and the tag position is then calculated according to the Mean Square Error Minimization algorithm. The reference tag method consists in having a large number of reference tags deployed at known locations [20].

The location of the target tag is estimated based on comparison with the RSSI of the reference tags. The work in [28] utilizes four adjacent reference tags for localizing a single target tag. There are also studies combining the radio map method with the reference tag method to enhance accuracy [20].

Both of these localization techniques require considerable effort to construct the radio map and/or record the positions of reference tags. This needs to be repeated whenever the environment changes. Another major problem is that the RSSI depends strongly on the type of tag and its orientation. Hence, the radio map and reference tag methods are suitable only if all the tags are the same and their alignment is known [20].

The newest generation of UHF RFID readers can extract phase information from the received signal by performing fully coherent demodulation. Tag phase information could then be utilized in different ways for localization applications. There are three main approaches based on the phase difference of signals received by adjacent antennas [20]:

1. Time domain phase difference of arrival (TD-PDOA).
2. Frequency domain PDOA (FD-PDOA).
3. Spatial domain PDOA (SD-PDOA) [81].

The modelled RSSI value from each antenna can be calculated at each testing point n using the Mean Square error value, which -based on RSSI value- can be calculated from every antenna at each test point n. The mean-square error (MSE) function is calculated using Equation (4.1) [20]:

$$MSE(n) = |P_1 - S_1(n)|^2 + |P_2 - S_2(n)|^2 + |P_3 - S_3(n)|^2 + |P_4 - S_4(n)|^2 \quad (4.1)$$

where P_1 to P_4 are the measured RSSI values from antennas 1 to 4. S_1 (n) to S_4 (n) are the modelled RSSI values for antennas 1 to 4 at point n. As mentioned earlier, the RSSI from each antenna is averaged over all frequencies as our basic signal model does not depend on frequency. The tag is estimated to be located at the point n_{mmse} with the minimum MSE (MMSE) [20].

The outcome of this research stage showed that the methodology of using spatial distribution of wide read range circular polarized antennas was able to locate the passive RFID Tags that are in the radio range of the antennas with an acceptable error percentage that does not exceed 5% of the total distance. The technique used to simulate RFID networks is a strong base method for further study in this research of developing wireless sensor networks.

4.3 RFID Hardware Manufacturers

This section covers all necessary RFID hardware part of the proposed solution in this research, which can be divided into:

1. RFID reader and antenna.
2. RFID Tags.

4.3.1 RFID Reader and Antenna

In this research, the handheld antennas are to be replaced with the fixed antennas that are going to be distributed around the library. The proposed readers and antennas are sold separately. The majority of the readers available commercially today have 4 ports, which will accommodate 4 antennas. See Figure 4.4:

Figure 4.3 Alien RFID Reader [82]

The RFID antennas vary in performance and the number of tags each antenna can detect, see Figure 4.6. The antenna properties and data sheets from several hardware providers have been investigated, they key providers are:

a) SkyRFID.
b) Atlas RFID Store.

Figure 4.4 Alien RFID Antenna [82]

Data sheets for few antenna models have been investigated; Figure 4.6 shows the data sheet for Atlas RFID antenna model number ALR-8696-C:

Parameter	Application
Antenna Part number	ALR-8696-C
Frequency Range	865 - 960 MHz
Gain	8.5 dBic
Cable loss	2.2dB (20ft)
Maximum VSWR	1.4:1
3 dB Beamwidth - Azimuth	65°
Front to Back Ratio	20 dB
Polarization	Circular Right-hand
Maximum Input Power	2 Watts
Input Impedence	50 ohms
Axial Ratio	1.2dB
Weight (Kg)	2.5 lbs (1.13)
Mechanical Size	10.2" x 10.2" x 1.32"
Antenna Connection	Coax Pigtail, Rev TnC Male
Radome	High Strength PC
Mount Style	100mm VESA mounting plate
Temperature operational	-25°C to +70°C
Humidity	MIL-STD-810G, Method 507.5 Procedure II Aggravated
Lightning Protection	DC Grounded
Environmental Rating	IP 54

Figure 4.5 Datasheet for antenna ALR-8696-C [82]

The antenna chosen for our research is SKYA902RHP9, which is pre-programmed to work on the frequency range 865-960 MHz. In New Zealand, the UHF frequency range from 902 to 915 MHz is reserved [82], which makes using any RFID instrument that utilizes any section in this frequency range requires authorization and approval from New Zealand Radio Spectrum Management.

4.3.2 Passive RFID Tag

Passive tags are only required when tagging a new library object. Existing tags can be utilized. The chosen tag for this research is SKYT960H3041, which works on the frequency range 860-960 MHz, see Figure 4.6.

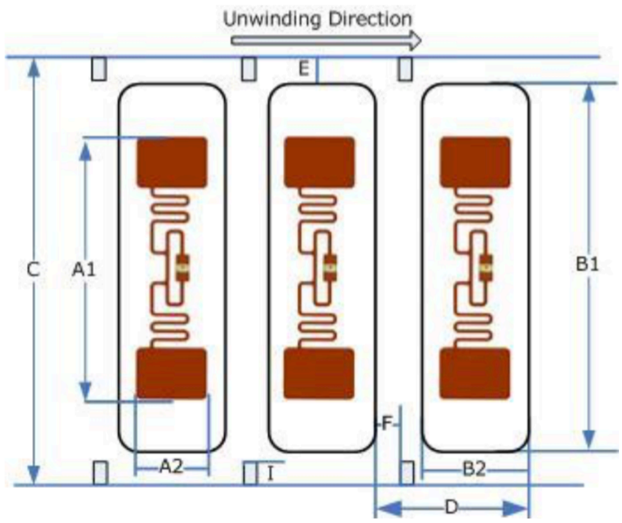


Figure 4.6 Picture for a sample passive SkyRFID tag

In this chapter, it is shown that automating a library system using conventional UHF RFID readers, antennas, and passive tags, is feasible. An estimation procedure for locating passive RFID tags was introduced, based on spatial deployment of the reader antennas for maximum signal diversity. The library system utilizes the received signal strength indicator calculated based on the detected tag signal. This indicator is incorporated into a mean square error minimization function in order to reduce the error percentage to the minimum. As the transmission frequency plays a sensitive role in the covered area that the RFID system can operate on, the frequency allocation was explored briefly in this chapter. Finally, the necessary hardware required to build the system was proposed

Chapter 5
Discussion and Conclusions

This research undertook many aspects, ranging from librarian interviews, contacting radio frequency hardware manufacturers, and software development. Every stage contributes strengths and flaws in its own specific parameters which may be carried through to the next stage.

At the beginning of the research, the librarians in Takapuna public library in auckland, North Shore, and Auckland University of Technology (AUT) university library, in New Zealand were interviewed to get a general understanding about the used library management system software and the nature of the business processes followed normally in the borrow/return, book shelving, book tracking, and anti-theft system.

Librarians in Takapuna public library confirmed that the library system is built to utilize the RFID system. All the library objects are tagged with pre-programmed passive RFID tags, which has a unique identifier and no other information is stored on the RFID tag, which makes the system non-ISO 28560-1 compliant. The librarians have their hand-held RFID scanners connected to their computers, which they use to scan books at the time of borrow and return. This process has some drawbacks such as, the distance between the tag scanner and the tag cannot exceed 10 centimetres in the best cases. The alignment of the tag and scanner to reach a specific polarization of the reader antenna can be time consuming to the librarian sometimes. There is also an auto restore shelve that the borrower can return books to which contains a fixed wide antenna that covers a wide area, which auto-detect books when they are placed on top of it.

The AUT library system is still utilizing both RFID, and the barcode system. The existing barcode system used to regulate the borrow/return process, and identify books in the library management software. The RFID system is used only in the anti-theft system.

The main areas covered in this research include the following points:

- Review the latest RFID commercially available technology for the antennas and tags that can achieve the desired read range/frequency.
- Software development which used the principles of java graphics to produce a simulation window to extract the behaviour of the RFID system components. The software employs the latest distance estimation algorithms to estimate the location of the RFID tag based on a randomly generated power loss value that is proportional to the distance between the tag and the antenna.
- Understanding Rifidi Edge server, Edge server builder, and configure both services to integrate via the available list of ports.
- Install and import Rifidi SDK platform in order to be imported into the middleware application.
- Design and implement error minimization function which is used in the library middleware application to collect the datasets that contain the tag power loss values in order to measure the distance, build an error function, and minimize the error until an accurate estimate is provided.

The software development stage involved two major stages. The first stage was to simulate the chosen hardware in the proposed solution using an existing software application called WSNSim and also building new RFID Simulation Software. This was the main part of this research simulating the solution proves its efficiency. The second stage, involved installing Rifidi Edge server, workbench, and Rifidi software development kit, which was required to build the integration applications to communicate with the RFID readers and antennas.

At the early stages of this research, all efforts have been made in learning C# programming language in order to enhance, and modify the code of the WSNSim software, for analysing and distance estimate. The development on the WSN software has stopped, as the code for the dependent libraries could not be retrieved for modification.

The second phase of software development, was to build a custom-made software simulator, that locates the RFID tags from fixed, and identified points in the simulation window, which holds the antennas. Data collection and acquisition were carried out alongside the completion of the programs.

RFID Simulator was built to provide a user-friendly interface to the user, and customizable simulation parameters that can be updated and deployed at runtime. Developing the RFIDSim from scratch has proven to be more versatile, easy to use application and more importantly the application achieved all the conditions and simulation requirements that the other simulations software could not achieve.

Programming errors were minimized as users did not need to return to the core design to make input changes. These manipulation capabilities within the RFIDSim programming give more control, flexibility and versatility to the users, minimizing errors in programming.

In summary, java programming was found to be powerful, flexible and efficient for developing any application that requires mathematical equations, graphics, or dynamically configuring runtime parameters. The reason behind this is the versatility of being able to control more from the user interface without going into the core application and alter the code for each deployment, hence minimizing programming error.

Regardless of the users background and programming experience, or which programs they are initially familiar with, or their readiness to use a particular software, this application was built to be fully customizable, it can be expanded to simulate any network.

In summary, this research investigated aspects involved in automating a library system using RFID technology as a communication protocol for the underlying RFID infrastructure. The library middleware application is a smart location-aware service that is used to identify and track library objects automatically and dynamically at runtime. As a beginning research area in the faculty, various aspects were developed more in an outlined manner rather than a strong focus in one area, using techniques used in the literature review. More focus work on a particular area can be done in future work of this research.

The simulation application has proved that this research produced satisfactory results, hence the methodology developed is potentially promising and can be used as a basis of practice for future work in implementing the library system automation solution.

The simulation results also show that the RSSI is proportional to $1/R^2$, which leads to the conclusion that the received power is proportional to the transmitted power and the gains of the reader and tag antennas. And inversely proportional to the distance-squared between the reader antenna and the tag. It is also a function of the polarization alignment between the reader and tag antennas.

Future work and recommendations required the following points in order to fully implement a RFID system into a library:

- RFID Antenna and Tags: more investigation can be carried out focusing on different hardware manufacturers, who are residing in New Zealand, and have the ability to provide hardware compliant with NZ Frequency spectrum.

- RFID Simulation Application include:
  - RFID Simulation can be used to simulate wireless networks. The code will need to be revised to be more dynamic, more features can be implemented to enhance the application.
  - The simulation software will need to be revised to build the signal model based on one directional distance relationship. The RSSI value is proportional to the inverse of the squared distance value.
  - Developing an extension program for integration with third party middleware applications to minimize or eliminate error when it is done manually, and provide better simulation results.
- Hardware Implementation include:
  - Build a prototype for a small implementation of a library system to prove the theoretical part of this research.
  - Conduct a heavy search for a New Zealand RFID hardware provider who can reprogram the required hardware to comply with the requirements specified by NZ Spectrum Management.

Appendix A

```
package com.util;
import com.datasets.Dataset;


/**
 *
 */
public final class MeanSquaredError implements MeasuredDistance<Dataset>
{
  private transient boolean init;
  private double threshold = 0.0;
  private double eta = 1.0;

  public MeanSquaredError()
  {
    init = false;
  }

  public double getThreshold()
  {
    return threshold;
  }

  public void setThreshold(double inThreshold)
  {
    threshold = inThreshold;
  }

  public double getEta()
  {
    return eta;
  }

  public void setEta(double inEta)
  {
    eta = inEta;
  }

  public void checkValues(Dataset data)
  {
    if(data == null)
      System.out.println(“Data set is empty”);
    int dataInt = data.getNumAttributes();

    if(dataInt == 0)
       System.out.println("No data attributes in the dataset");

    if(Double.isNaN(threshold))
      System.out.println ("Threshold is not a number");

    if(_threshold < 0.0)
      System.out.println ("Threshold < 0");

    if(Double.isNaN(_eta))
      System.out.println ("Eta is not a number");

    if(_eta < 1.0)
      System.out.println ("Eta < 1");

    init = true;
  }
```

```
  public double findMeanSquare(Dataset rData)
  {
    //Size of the cluster
    int  size = 0;
    double distanceEstimate = 0.0;
    if (!init)
      System.out.println ("Data error");

    if(data == null)
      System.out.println ("Dataset is empty");

    size = data.size();

    if (size == 0)
      return 0;

    //Find mean square error
    double mse = findMSE(rData);
    double inverseMSE = 1.0 / (mse + 1.0);

    if(inverseMSE > threshold)
      distanceEstimate = inverseMSE - threshold;

    if(_eta != 1.0)
      uFitness = Math.pow(distanceEstimate, eta);

    return uFitness;
  }

  /*
   * This method finds the mean value in a dataset.
   */
  private double[] findMeanSquareDS(Dataset data)
  {
    int dataAtt = data.getNumAttributes();
    double [] mean = new double[dataAtt];
    for(int i = 0 ; i < dataAtt ; i ++ )
    {
      mean[i] = findMeanValue(data, dataAtt);
    }

    return mean;
  }

  /*
   * Function input NumericDataset and attribute index parameters,returns
   * mean value of the attribute at an index
   */
  public static double findMeanValue(Dataset data, int dataAtt)
  {
    if(data == null)
      throw new RuntimeException("Null Dataset passed to calculate mean
value");

    if(dataAtt < 0 || dataAtt >= data.getNumAttributes())
      System.out.println("Incorrect attribute index for the dataset");

    double mean = 0.0;
    int size = data.size();

    for (int i : data.examples())
    {
       double dataAttDouble = (Double) data.getObject(i, dataAtt);
       if(Double.isNaN(dataAttDouble))
          System.out.println("Invalid arguments”);
        mean += dataAttDouble;
    }
    //Mean is zero for any attribute in empty dataset
    if(size == 0)
      mean = 0;
    else
```

```
      mean /= size;
    return mean;
  }

  /*
   *  This method calculates the Mean Square Error
   */
  public double findMSE(Dataset data)
  {
    int size = data.size();
    double mse = 0;
    int dataAttInt = data.getNumAttributes();
    //  Find mean of each data attribute
    double [] mean = findMeanOfDataSet(data);
    for (int i : data.examples())
    {
      double tempMSE = 0;
      for(int j = 0 ; j < nDataAtt ; j++ )
      {
        double attDouble = (Double) data.getObject(i, j);
        tempMSE += Math.pow(mean[j]- attDouble, 2) ;
      }
      mse += tempMSE;
    }

    if(size == 0)
      mse = 0;
    else
      mse /= size;

    return mse;
  }
}
```

References

1. Ching, S.H. and A. Tai, *HF RFID versus UHF RFID — Technology for Library Service Transformation at City University of Hong Kong.* The Journal of Academic Librarianship, 2009. **35**: p. 347-359.

2. Edwards, S. and M. Fortune, *A Guide to RFID in Libraries.* Book Industry Communication, 2008.

3. Aydın, K. and S. Yıldırım, *Case study about RFID System in Library Services.* International Journal of Synergy and Research 2012. **1**(2): p. 91–102.

4. Sari, K., *Selection of RFID solution provider: A fuzzy multi‑criteria decision model with Monte Carlo simulation.* Kybernetes, 2013. **42**(3): p. 448-465.

5. Ngai, E.W.T., et al., *RFID research: An academic literature review (1995–2005) and future research directions.* International Journal of Production Economics, 2008. **112**: p. 510-520.

6. Roh, J.J., A. Kunnathur, and M. Tarafdar, *Classification of RFID adoption: An expected benefits approach.* Information & Management, 2009. **46**(6): p. 357-363.

7. Bahri, S. and A. Ibrahim, *RFID in libraries: a case study on implementation.* Library Hi Tech News, 2013. **30**(5): p. 21-26.

8. Chelliah, J., S. Sood, and S. Scholfield, *Realising the strategic value of RFID in academic libraries: a case study of the University of Technology Sydney.* Australian Library Journal, 2015. **64**(2): p. 113-127.

9. Jain, P. and P. Sharma, *RFID Technology Implementation: A Case Study of Institute of Home Economics, University of Delhi.* 2014.

10. Chen, J.C., C.-H. Cheng, and P.B. Huang, *Supply chain management with lean production and RFID application: A case study.* Expert Systems with Applications, 2013. **40**(9): p. 3389-3397.

11. Dobkin, D.M., Weigand, S. M., & Iyer, N. , *SEGMENTED MAGNETIC ANTENNAS FOR NEAR-FIELD UHF RFID.* Microwave Journal, 2007: p. 50(6), 96-96,98,100,102.

12. Kapoor, K., et al., *RFID Integrated Systems in Libraries: Extending TAM Model for Empirically Examining the Use.* Journal of Enterprise Information Management, 2014. **27**(6): p. 1-23.

13. Anee, R. and N.C. Karmakar, *Chipless RFID Tag Localization.* Microwave Theory and Techniques, IEEE Transactions on, 2013. **61**(11): p. 4008-4017.

14. Preradovic, S. and N.C. Karmakar, *Chipless RFID: Bar Code of the Future.* Microwave Magazine, IEEE, 2010. **11**(7): p. 87-97.

15. Yanchao, Z., et al. *Fast identification of the missing tags in a large RFID system*. in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*. 2011.

16. Carla R. Medeiros, J.R.C., and Carlos A. Fernandes, *RFID Reader Antennas for Tag Detection in Self-Confined Volumes at UHF.* IEEE Antennas and Propagation Magazine, 2011. **Vol. 53**(No. 2).

17. Rundh, B., *Radio frequency identification (RFID).* Marketing Intelligence & Planning, 2008. **26**(1): p. 97-114.

18. Want, R., *An introduction to RFID technology.* Pervasive Computing, IEEE, 2006. **5**(1): p. 25-33.

19. Yongheng Wang, X.Z., *Internet of Things.* 2012. **312 2012**.

20. Shuai, S. and R.J. Burkholder, *Item-Level RFID Tag Location Sensing Utilizing Reader Antenna Spatial Diversity.* Sensors Journal, IEEE, 2013. **13**(10): p. 3767-3774.

21. Finkenzeller, K. and D. Müller, *RFID Handbook : Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, 2010, Wiley: Hoboken.

22. Chen, J.-L., et al., *Architecture design and performance evaluation of RFID object tracking systems.* Computer Communications, 2007. **30**(9): p. 2070-2086.

23. Chawla, V. and H. Dong Sam, *An overview of passive RFID.* Communications Magazine, IEEE, 2007. **45**(9): p. 11-17.

24. Xianming, Q., G. Chean Khan, and C. Zhi Ning, *A Broadband UHF Near-Field RFID Antenna.* Antennas and Propagation, IEEE Transactions on, 2010. **58**(12): p. 3829-3838.

25. Weilian, S., N. Alchazidis, and T.T. Ha, *Multiple RFID Tags Access Algorithm.* Mobile Computing, IEEE Transactions on, 2010. **9**(2): p. 174-187.

26. Caldwell-Stone, D., *RFID in Libraries.* Library Technology Reports, 2010. **46**(8): p. 38-46.

27. Landt, J., *The history of RFID.* Potentials, IEEE, 2005. **24**(4): p. 8-11.

28. Xin, X., et al. *An environmental-adaptive RSSI based indoor positioning approach using RFID*. in *Advanced Intelligence and Awarenss Internet (AIAI 2010), 2010 International Conference on*. 2010.

29. Chen, S.Y. and P. Hsu, *CPW-fed folded-slot antenna for 5.8 GHz RFID tags.* Electronics Letters, 2004. **40**(24): p. 1516-1517.

30. Xiao zheng, L., X. Zeming, and C. Xuanliang, *A Compact RFID Reader Antenna for UHF Near-Field and Far-Field Operations.* International Journal of Antennas and Propagation, 2013: p. 5.

31. Oliver, R.A., *Broken-loop RFID reader antenna for near field and far field UHF RFID tags*, 2008, Google Patents.

32. Xing, Z., et al., *Characteristics and Application of a Novel Loop Antenna to UHF RFID Receivers.* International Journal of Antennas and Propagation, 2011. **2011**: p. 7.

33. Padhi, S., et al., *A dual polarized aperture coupled circular patch antenna using a C-shaped coupling slot.* IEEE Transactions on antennas and propagation, 2003. **51**(12): p. 3295-3298.

34. Marrocco, G., *Gain-optimized self-resonant meander line antennas for RFID applications.* IEEE Antennas and Wireless propagation letters, 2003. **2**(1): p. 302-305.

35. Hirvonen, M., et al., *Planar inverted-F antenna for radio frequency identification.* Electronics Letters, 2004. **40**(14): p. 1.

36. Xianming, Q. and Y. Ning. *A folded dipole antenna for RFID*. in *Antennas and Propagation Society International Symposium, 2004. IEEE*. 2004.

37. Rao, K.V.S., P.V. Nikitin, and S.F. Lam, *Antenna design for UHF RFID tags: a review and a practical application.* Antennas and Propagation, IEEE Transactions on, 2005. **53**(12): p. 3870-3876.

38. Moncombu Ramakrishnan, K.N., *Performance benchmarks for passive UHF RFID tags.* ProQuest Dissertations & Theses Global, 2005(304990913)).

39. Mehdia Ajana El Khaddar, M.B., Hamid Harroud and Mohammed Elkoutbi, *RFID Middleware Design and Architecture.* InTech, 2011: p. 384

40. Ni, L.M., et al., *LANDMARC: Indoor Location Sensing Using Active RFID.* Wireless Networks, 2004. **10**(6): p. 701-710.

41. Dobkin, D.M., *The rf in RFID: uhf RFID in practice*. 2012: Newnes.

42. Bouhouche, T., et al. *A new middleware architecture for RFID systems*. in *Microwave Symposium (MMS), 2014 14th Mediterranean*. 2014.

43. Haifeng, W. and Z. Yu, *Passive RFID Tag Anticollision Algorithm for Capture Effect.* Sensors Journal, IEEE, 2015. **15**(1): p. 218-226.

44. Pavel V. Nikitin, K.V.S.R., *Antennas and Propagation in UHF RFID Systems.* IEEE RFID, 2008.

45. Khor, J., et al., *EPC Class-1 Generation-2 radio frequency Identification (RFID)-based Malaysian University Communities.* Scientific Research and Essays, 2012. **7**(8): p. 852-864.

46. Inc., S. *RFID Reader Antenna Tutorial - What you need to know*. 2007-2015; Available from: http://skyrfid.com/RFID_Antenna_Tutorial.php.

47. Yejun, H. and W. Xiaoye, *An ALOHA-based improved anti-collision algorithm for RFID systems.* Wireless Communications, IEEE, 2013. **20**(5): p. 152-158.

48. Wen-Tzu, C., *An Accurate Tag Estimate Method for Improving the Performance of an RFID Anticollision Algorithm Based on Dynamic Frame Length ALOHA.* Automation Science and Engineering, IEEE Transactions on, 2009. **6**(1): p. 9-15.

49. Jianwei, W., et al. *Fast Anti-Collision Algorithms in RFID Systems*. in *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2007. UBICOMM '07. International Conference on*. 2007.

50. Tao, C. and J. Li. *Analysis and Simulation of RFID Anti-collision Algorithms*. in *Advanced Communication Technology, The 9th International Conference on*. 2007.

51. Dongwook, L., et al. *Efficient dual bias Q-Algorithm and optimum weights for EPC Class 1 Generation 2 Protocol*. in *Wireless Conference, 2008. EW 2008. 14th European*. 2008.

52. Xu, H. and L. Son. *Efficient Dynamic Framed Slotted ALOHA for RFID Passive Tags*. in *Advanced Communication Technology, The 9th International Conference on*. 2007.

53. Zhang, D.-g., et al., *A new anti-collision algorithm for RFID tag.* International Journal of Communication Systems, 2014. **27**(11): p. 3312-3322.

54. Yuan-Cheng, L. and H. Ling-Yen, *General binary tree protocol for coping with the capture effect in RFID tag identification.* Communications Letters, IEEE, 2010. **14**(3): p. 208-210.

55. Bhattacharyya, R., C. Floerkemeier, and S. Sarma, *Low-Cost, Ubiquitous RFID-Tag-Antenna-Based Sensing.* Proceedings of the IEEE, 2010. **98**(9): p. 1593-1600.

56. Lai, Y.C. and L.Y. Hsiao, *General binary tree protocol for coping with the capture effect in RFID tag identification.* IEEE Communications Letters, 2010. **14**(3): p. 208-210.

57. Corporation, I., *IBM WebSphere RFID Premises Server V1.0 connects the edge to the enterprise.* 2014.

58. Server, R.E., *Rifidi Edge Server User's Guide.* 2015.

59. Langheinrich, M., *A survey of RFID privacy approaches.* Personal and Ubiquitous Computing, 2009. **13**(6): p. 413-421.

60. Walther, B.A. and J.L. Moore, *The concepts of bias, precision and accuracy, and their use in testing the performance of species richness estimators, with a literature review of estimator performance.* Ecography, 2005. **28**(6): p. 815-829.

61. ISO28560-1, *Information and documentation -- RFID in libraries -- Part 1: Data elements and general guidelines for implementation.* ISO, 2011.

62. Zhu, X., S.K. Mukhopadhyay, and H. Kurata, *A review of RFID technology and its managerial applications in different industries.* Journal of Engineering and Technology Management, 2012. **29**(1): p. 152-167.

63. Khor, J., et al., *UHF RFID proof of concept (POC) with open-source ILS at Universiti Sains Malaysia (USM) Libraries.* Program: Electronic Library & Information Systems, 2015. **49**(2): p. 135.

64. Singh, J., N. Brar, and C. Fong, *The state of RFID applications in libraries.* Information Technology & Libraries, 2006. **25**(1): p. 24-32 9p.

65. Molyneux, R.E., *An Open Source ILS Glossary: Version 2.* Public Library Quarterly, 2011. **30**(2): p. 165.

66. Tan, C.C., S. Bo, and L. Qun, *Secure and Serverless RFID Authentication and Search Protocols.* Wireless Communications, IEEE Transactions on, 2008. **7**(4): p. 1400-1407.

67. R-Moreno, M.D., et al., *Efficient Services Management in Libraries using AI and Wireless techniques.* Expert Systems with Applications, 2014. **41**(17): p. 7904-7913.

68. Tajima, M., *Strategic value of RFID in supply chain management.* Journal of Purchasing and Supply Management, 2007. **13**: p. 261-273.

69. Edwards, S. and M. Fortune, *A Guide to RFID in Libraries.* 2008.

70. C.-H. Loo, K.E., F. Yang, A. Z. Elsherbeni, D. Kajfez, A. A. Kishk, T. Elsherbeni, L. Ukkonen, L. Sydanheimo, M. Kivikoski, S. Merilampi, and P. Ruuskanen, *Chip Impedance Matching for UHF RFID Tag Antenna Design.* Progress In Electromagnetics Research, 2008: p. 359-370.

71. Daqiang, Z., et al., *TASA: Tag-Free Activity Sensing Using RFID Tag Arrays.* Parallel and Distributed Systems, IEEE Transactions on, 2011. **22**(4): p. 558-570.

72. Li, L., L. Yunhao, and L. Xiang-Yang. *Refresh: Weak Privacy Model for RFID Systems*. in *INFOCOM, 2010 Proceedings IEEE*. 2010.

73. ISO28560-1:2011, *Information and documentation -- RFID in libraries.* p. Part 1: Data elements and general guidelines for implementation.

74. Carla R. Medeiros, C.A.F., Jorge R. Costa, *UHF RFID READER ANTENNAS FOR SELF-CONFINED TAG DETECTION.*

75. Bohn, J., *Prototypical implementation of location-aware services based on a middleware architecture for super-distributed RFID tag infrastructures.* Personal & Ubiquitous Computing, 2008. **12**(2): p. 155.

76. Humenuk, S., *Automatic Shelving and Book Retrieval: A contribution Toward A Progressive Philosophy of Library Service for a Research Library.* 1966: p. 17.

77. Pereira, B.D.A., *A identificação por radiofrequência (RFID) como facilitadora das práticas Lean no contexto empresarial português-Um estudo exploratório.* 2014.

78. Gluhak, A., et al., *A survey on facilities for experimental internet of things research.* Communications Magazine, IEEE, 2011. **49**(11): p. 58-67.

79. Sánchez López, T., et al., *Adding sense to the Internet of Things.* Personal and Ubiquitous Computing, 2012. **16**(3): p. 291-308.

80. Daniel Giusto, A.I., Giacomo Morabito, Luigi Atzori, *The Internet of Things : 20th Tyrrhenian Workshop on Digital Communications*. 2010.

81. Nikitin, P.V., et al. *Phase based spatial identification of UHF RFID tags*. in *2010 IEEE International Conference on RFID (IEEE RFID 2010)*. 2010.

82. Business, T.G.L.o., *Regulatory status for using RFID in the EPC Gen2 band (860 to 960 MHz) of the UHF spectrum*. 2016.