Heterogeneous RFID framework design, analysis and evaluation

Oscar Botero Gomez Botero

Ecole Doctorale EDITE

Thèse présentée pour l'obtention du diplôme de Docteur de Télécom & Management SudParis

Doctorat conjoint Télécom & Management SudParis et Université Pierre et Marie Curie

Spécialité :

Systèmes Informatiques

Par

Oscar Botero

Conception, analyse et évaluation d'un Réseau RFID Hétérogène

Soutenue le 14 Mai 2012 devant le jury composé de :

Pr. Sébastien TIXEUIL. Président du jury. Professeur à Université Pierre et Marie Curie.
Dr. Nathalie MITTON, HDR. Rapporteur Chargée de Recherche, INRIA.
Dr. Yacine Ghamri DOUDANE, HDR. Rapporteur. Maitre de Conférence, ENSIIE EVRY.
Alexis Oliverau. Examinateur. Chercheur, CEA, Saclay.
Pr. Tijani CHAHED. Examinateur. Professeur à Telecom SudParis.
Pr. Hakima CHAOUCHI. Directeur. Professeur à Telecom SudParis

Thèse n° 2012TELE0011

Thèse de Doctorat

Télécom SudParis et

L'université Pierre & Marie Curie Université (Paris VI)

Spécialité

Systèmes Informatiques

Présentée par

M. Oscar BOTERO

Pour obtenir le grade de

DOCTEUR

Conjoint de Télécom SudParis et l'université Pierre et

Marie Curie

Conception, analyse et évaluation d'un Réseau RFID Hétérogène

Soutenue le 14 Mai 2012 devant le jury composé de

Pr. Sébastien TIXEUIL	**Président du jury**	**Professeur à Université Pierre et Marie Curie**
Dr. Nathalie MITTON, HDR	**Rapporteur**	**Chargée de Recherche, INRIA**
Dr. Yacine Ghamri DOUDANE, HDR	**Rapporteur**	**Maitre de Conférence, ENSIIE EVRY**
Alexis Oliverau	**Examinateur**	**Chercheur, CEA, Saclay**
Pr. Tijani CHAHED	**Examinateur**	**Professeur à Telecom SudParis**
Pr. Hakima CHAOUCHI	**Directeur**	**Professeur à Telecom SudParis**

Doctor of Science Thesis

Telecom SudParis and

Pierre & Marie Curie University (Paris VI)

Specialization

Computer Science

Presented by

M. Oscar BOTERO

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR of SCIENCE

Telecom SudParis and Pierre & Marie Curie University

Heterogeneous RFID Framework design, analysis and evaluation

Committee in charge:

Pr. Sébastien TIXEUIL	**President**	**Professor at Université Pierre et Marie Curie**
Dr. Nathalie MITTON, HDR	**Reviewer**	**Chargée de Recherche, INRIA**
Dr. Yacine Ghamri DOUDANE, HDR	**Reviewer**	**Maitre de Conférence, ENSIIE EVRY**
Alexis Oliverau	**Examiner**	**Researcher, CEA, Saclay**
Pr. Tijani CHAHED	**Examiner**	**Professor at Telecom SudParis**
Pr. Hakima CHAOUCHI	**Advisor**	**Professor at Telecom SudParis**

Résumé

Le paradigme de L'Internet des choses établit l'interaction et la communication avec une énorme quantité d'acteurs. Le concept n'est pas tout à fait nouveau; en fait, il combine un grand nombre de technologies et de protocoles et sûrement des adaptations des éléments préexistants pour offrir de nouveaux services et applications. Une des technologies clés de l'Internet des objets est l'identification par radiofréquence abrégée en anglais RFID («*Radio Frequency Identification*»). La RFID est une technologie bien connue qui est employée avec succès dans de nombreuses applications. Elle a été introduite dans les années 50, et présente des avantages par rapport à son prédécesseur, le code de barres, car elle nécessite de petits dispositifs électroniques pour transmettre un code d'identification (ID) et ne nécessite pas de ligne de vue pour fonctionner. Cette technologie propose un ensemble de solutions qui permettent le suivi et la traçabilité des personnes, des animaux et pratiquement n'importe quel objet en utilisant des liaisons sans fil. L'architecture RFID est composée d'étiquettes RFID dites Tags, qui fournissent un code d'identification, de lecteurs RFID, qui interrogent et obtiennent grâce aux ondes électromagnétiques ce code, et un logiciel médiateur («*Middleware*») qui fournit une plateforme pour interpréter et utiliser l'information recueillie.

En considérant le concept de l'Internet des choses, plusieurs technologies doivent être liées afin de fournir des interactions qui conduisent à la mise en œuvre de services et d'applications. Le défi est que ces technologies ne sont pas nécessairement compatibles et conçues pour fonctionner ensemble. De la même manière, une technologie peut avoir différentes variantes qui ajoutent plus de complexité à la conception du système. C'est le cas de la RFID, où nous pouvons trouver différentes fréquences d'opérations, de divers protocoles de communication et de normes. Dans un environnement multi dispositif et multi technologie, ces défis imposés doivent être abordés afin de fournir une plateforme d'information unifiée.

Dans ce contexte, l'objectif principal de cette thèse est de concevoir un « *framework* » hétérogène qui permettra l'interaction de divers dispositifs tels que la RFID, des capteurs (dispositifs qui obtiennent des mesures de variables de l'environnement) et des actionneurs (matériel qui peut effectuer des actions physiques telles que les moteurs ou relais électriques) afin de fournir de nouvelles applications et de services. Nous portons une attention particulière à l'intégration de dispositifs sans capacités RFID dans ce framework, leur permettant d'obtenir les informations fournies par les tags à l'aide d'une interface réseau soit filaire ou sans fil. À cet effet, dans ce travail, notre première contribution est la conception et l'analyse d'une architecture d'intégration pour les dispositifs hétérogènes (sans et avec la technologie RFID) définissant dans le détail l'ensemble de ses éléments constitutifs, les interactions et l'échange de messages. Nous avons évalué ses performances en utilisant des simulations afin de vérifier son applicabilité et faisabilité. Dans la seconde contribution, nous proposons un modèle d'évaluation de la topologie RFID et un outil d'optimisation qui aide dans le processus de planification de réseaux de cette technologie. Une fonction multi objective d'évaluation et des algorithmes génétiques ont été combinés dans une application logicielle développée et une description détaillée et des tests sont également fournis. Enfin, dans notre dernière contribution, nous avons implémenté une version simplifiée du *framework* en utilisant du matériel embarqué et indicateurs de performance sont fournis ainsi que la configuration détaillée de la plateforme de test.

Mots-clés:

RFID, hétérogénéité, algorithmes génétiques, Peer to Peer, Internet des objets, optimisation de la topologie de réseau RFID, matériel embarqué, Microcontrôleurs, WLAN, ZigBee.

Abstract

The Internet of Things paradigm establishes interaction and communication with a huge amount of actors. The concept is not a new-from-scratch one; actually, it combines a vast number of technologies and protocols and surely adaptations of pre-existing elements to offer new services and applications. One of the key technologies of the Internet of Things is the Radio Frequency Identification just abbreviated RFID. RFID is a well-known technology that has entered successfully in the realm of innumerable applications. RFID was introduced in the 50's, and it presents advantages over its predecessor the barcode. RFID requires small electronic devices to transmit an Identification code (ID) and non-line of sight is needed to operate. This technology proposes a set of solutions that allow tracking and tracing persons, animals and practically any item wirelessly. The RFID architecture is configured by Tags, which provide an Identification code, RFID Readers that, obtain wirelessly that code, and Middleware that provides a platform to interpret and utilize the collected information.

Considering the Internet of Things concept, multiple technologies need to be linked in order to provide interactions that lead to the implementation of services and applications. The challenge is that these technologies are not necessarily compatible and designed to work with other technologies. Additionally, one technology may have different variants that add more complexity to the design. That is the case of RFID, where we can find different frequency band operation, communication protocols and standards. In a multi-device, multi-technology environment these imposed challenges need to be tackled in order to provide a unified information platform.

Within this context, the main objective of this thesis is to design a heterogeneous framework that will permit the interaction of diverse devices such as RFID, sensors (devices that obtain measurements of environment variables) and actuators (hardware that can perform physical actions like motors or electric relays) in order to provide new applications and services. We pay especial attention to the integration of devices with non-RFID capabilities into this framework, allowing them to obtain the information provided by the tags using a network interface either wired or wireless. For this purpose in this work, our first contribution is the design and analysis of an integration architecture for heterogeneous devices (RFID and non-RFID enabled devices) defining in detail all its constitutive elements and message exchange interactions. We evaluated its performance by using simulations to verify its applicability and feasibility. In the second contribution, we propose an evaluation model for RFID topologies and an optimization tool that assists in the RFID network planning process. A Multi-objective fitness function and Genetic Algorithms were combined in a developed software application and a detailed description and tests are provided as well. Finally, in our last contribution, we implemented a simplified version of the framework by using embedded hardware and performance metrics are provided as well as the detailed configuration of the test platform.

Key Words:

RFID, Heterogeneity, Genetic Algorithms, Peer to Peer, Internet of Things, Network Topology Optimization, Embedded Hardware, Microcontrollers, WLAN, ZigBee.

Table of Contents

List of Figures

List of Tables

List of Acronyms

2D two-dimension

3D three-dimension

AES Advanced Encryption Standard

CDMA Code Division Multiple Access

CSMA Carrier Sense Multiple Access

DES Data Encryption Standard

DSB-ASK Double-sideband amplitude shift keying

EEPROM Erasable Programmable Read-Only Memory

EPC Electronic Product Code

EPC-Gen2 Electronic Product Code Generation 2

FAT File Allocation Table

FDMA Frequency Division Multiple Access

FHSS Frequency Hopping Spread Spectrum

FM0 Bi-phase encoding

GA Genetic Algorithms

GPRS General Packet Radio Service

GUI Graphical User Interface

HF High Frequency

HTML HyperText Markup Language

I2C Inter-Integrated Circuit

ID identification code

IEC International Electro technical Commission

ISO International Organization for Standardization

JXTA Set of P2P protocols developed by SUN (Juxtapose)

LBT Listen-Before-Talk

LF Low Frequency

MAN	Management and authenticator Node
MMN	Multi-Mode Node
MRN	Mobile Reader Node
NS2	Network Simulator 2
P2P	Peer to Peer
PIE	Pulse-Interval Encoding
PR-ASK	Phase-reversal amplitude shift keying
PSO	Particle Swarm Optimization
RF	Radio Frequency
RFID	Radio Frequency Identification
SD	Secure Digital
SDMA	Space-Division Multiple Access
SPI	Serial Peripheral Interface Bus
SRAM	Static random-access memory
SSB-ASK	Single-sideband amplitude shift keying
TARI	Reference time interval that comes from the standard ISO/IEC 18000-6 (Part A). Tari is the abbreviation for Type A reference interval.
TDMA	Time Division Multiple Access
TIS	Tag Identification Speed
UHF	Ultra High Frequency
USB	Universal Serial Bus
VRN	Virtual Reader Node
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
XML	eXtensible Markup Language

Chapter 1

Introduction

RFID technology has been used since the 50's in order to identify items wirelessly. However, nowadays a myriad and always expanding sort of applications has been proposed that take advantages of what it offers. Essentially, its architecture is rather simple. It is based on transponders (tags) that provide an Identification Code (ID), readers that obtain that code from the tags and finally a middleware that interprets processes and delivers the collected set of IDs serving specific purposes (Figure 1). This technology can be used to track and trace persons, animals, items and some applications include warehouse logistics, personal documents (passports, driving licenses), public transportation cards, building access control, and it is one of the main building blocks of the Internet of Things paradigm [**1**].

Figure 1. Basic RFID Architecture

In this work, we considered the problem of integrating heterogeneous devices into a single framework based on RFID. Heterogeneous in the context of this thesis is related to the diverse devices that can be connected and communicating into the framework i.e. devices with or without RFID capabilities and diverse network interfaces. It includes RFID readers working in various frequency bands, sensors and actuators. We also studied the RFID topology design and finally we implemented an embedded hardware platform that integrates RFID, network interfaces and sensors.

1.1 Thesis Objectives and Challenges

In a multi-technology, multi-device integration context like the one proposed by the Internet of Things, one of the main challenges is to allow the operation of diverse devices orchestrated into a well-defined purpose, i.e. providing information services. To achieve this one of the steps is to define and design an integration architecture that will take into account the diverse communication possibilities of each technology and will centralize certain functions such as management and supervision of the network.

The main objectives of this thesis are the following. First, is to design a single framework based on RFID and IP communication capabilities that will allow the integration of diverse devices (multiple RFID and non-RFID devices) in order to provide services and applications that require RFID information reading. We need to define and state the different protocols required for the framework to operate and to describe in detail its constitutive elements. Additionally, the use of multiple RFID readers can provoke the negative effect of data collision, thus this issue needs to be tackled as well. Second, since the framework will have several RFID readers, we need to deal with the problem of RFID network planning that requires an effective and efficient deployment of readers. This is needed in order to maximize the tag IDs reading and minimize the number of RFID readers to be installed among other optimization parameters such as coverage area and interference reduction. Finally, simulating the communication interactions and implementing the framework by using real

hardware can provide a better and complete vision of the whole architecture, thus, we proceeded with an embedded hardware implementation of some functionalities of the designed framework.

1.2 Thesis Contributions and workflow

In order to achieve the identified objectives we proceeded in the following way:

- First, we studied the state-of-the art related to RFID. It permitted to have a broad vision of the virtues and limitations of this technology. We contributed with the following book chapter:

  - Oscar Botero & Hakima Chaouchi, Chapter 5, *On RFID and research issues*. Chaouchi Hakima, the Internet of Things: Connecting Objects, Wiley-ISTE, 2010.

- Second, we designed a framework for heterogeneous devices based on RFID. It allows the integration of RFID readers (mobile and fixed), and devices with non-RFID capabilities in order to provide RFID-based services. The framework is conceived to work with several RFID readers deployed thus we also considered the interference problem of multiple readers working simultaneously. We took into account some security issues like network authentication. Our framework design is based in a P2P (Peer-to-Peer) network overlay architecture. The definition of the different protocols required is presented in order to provide RFID-based information services. Simulations were executed to obtain performance metrics as well. We contributed with the following research paper:

  - Oscar Botero & Hakima Chaouchi, *P2P Framework for RFID enabled and non-enabled users*. IoTs 2010, Hangzhou, China.

- Third, we studied the optimization problem of RFID network topology. We proposed an evaluation function or also called multi-objective fitness function based on six individual objectives (coverage area, tags detected, interference reduction, among others) that allows us to grade candidate solutions related to the deployment of multiple RFID readers in an effective and efficient way. The optimization method employed was Genetic Algorithms. We developed a software tool that helps on finding the right reader's layout to be deployed in a certain region. These efforts were published in the following research paper:

  - Oscar Botero & Hakima Chaouchi, *RFID network topology design tool based on Genetic Algorithms*. RFID-TA Barcelona, Spain 2011.

- Finally, we implemented a simplified version of the framework by using embedded hardware. Microcontrollers, RFID, Wi-Fi and ZigBee modules were combined to provide RFID tag IDs to users with no RFID capabilities but with the mentioned network interfaces instead. We performed measurements related to the performance of the test platform and we depict its design and implementation. We contributed with the following papers:

  - Oscar Botero & Hakima Chaouchi, *RFID for non-RFID users Embedded Hardware Implementation*. ANT Niagara Falls, Canada 2011.

  - Journal: Oscar Botero & Hakima Chaouchi, *Radio Frequency Identification framework for heterogeneous nodes.* Personal and Ubiquitous Computing. Springer 2012.

- A. Ait Wakrim, O. Botero, K. Raymond, H. Chaouchi “TRACK-IoT: Heterogeneous IoT Network”, Research Report, Telecom Sud Paris March 2012.

The rest of this thesis document is organized as follows. Chapter 2 proposes the RFID terminology and general concepts followed by Chapter 3 where the RFID heterogeneous Framework is described in detail. In Chapter 4, we introduce the Optimization basis focused mostly in the Genetic Algorithms method followed by the RFID network topology design, describing the model used to evaluate the solutions and a developed software tool as well as performance metrics obtained. In Chapter 5, we present the hardware implementation. Finally, we state the future research directions and provide the conclusions obtained.

RFID Framework for heterogeneous nodes

Chapter 2

RFID Concepts and Terminology

In this chapter, we provide a description of basic concepts on RFID technology. First, we describe the main components of the architecture. Then, we present research issues related to the technology. Finally, we describe different RFID applications and related technologies.

2.1 RFID

RFID stands for radio frequency identification and provides identification codes that can be associated to persons, animals and items for tracking and tracing purposes [**2**]. The main difference in relation to the barcode is that RFID requires no line of sight to transmit the ID code. This technology is used in innumerable applications like citizen's documents (passports, driving licenses, and social security cards), transportation, animal tracking, warehouse items management, building access control, public transportation among others [**3**].

Two of the most important entities related to research and development and to RFID standards are Auto-ID Labs [**4**] and EPC (Electronic Product Code) global Inc. [**5**]. The Auto-ID Labs is an independent network of currently seven academic research labs. These are: MIT (Cambridge US), Cambridge (UK), St. Gallen (Switzerland), Fudan (China), ICU (Korea), Adelaide(Australia), Keio (Japan). Its mission is to "*build a business driven, truly global, sustainable, robust, cost efficient, and future-proof EPC Network Infrastructure that is flexible enough to support future technologies, applications and industries*". EPC global is leading the development of industry-driven standards to support the use of RFID. The main activities of this group are assignment, maintenance and registration of EPC manager numbers, participation in development of EPC global standards, EPC global certification and accreditation program testing and training and education on implementing and using EPC technology and the EPC global network among others.

A RFID architecture is constituted of three elements (Figure 2) : transponders or tags that store an identifier (ID), readers or interrogators which obtain the IDs from the tags, and middleware that interprets the information provided by the readers by following specific purposes or applications. This information can be stored into servers to be shared and analyzed.

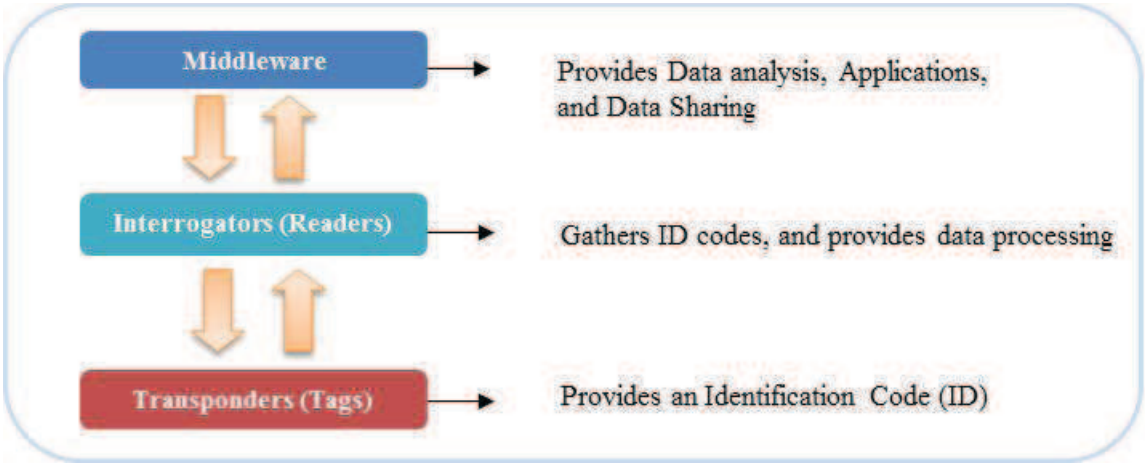


Figure 2. RFID Architecture: tags, readers and middleware

2.1.1 RFID Transponders or Tags

Tags or transponders carry an Identification code that can be retrieved by RFID readers. They are composed by the antenna part and a chip module (Figure 3). The antenna captures the energy from the reader and feeds it into the Chip in order to make it work. The Chip is the core of the tag that runs all the logic tasks related to the transmission of the ID and the execution of commands [**6**].

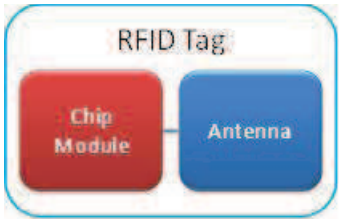


Figure 3. RFID Tag Modules.

The MIT Auto-ID Center established a tag classification system also used by EPC global Inc. Tags are grouped into passive, semi passive and active types as shown in Table 1.

Table 1. RFID tag classes

Tag Class	Type	Capabilities
Class 0	passive	read only
Class 1		read, write once
Class 2		read/write
Class 3	semi-passive	increased range
Class 4	active	tag communication
Class 5		reader capabilities

2.1.1.1 Passive tags

Passive tags are energized by the reader's electromagnetic field. They are low cost and size because no batteries are needed. Passive tags are divided into Class 0, 1 and 2.

- Class 0: This class refers to Read-only tags with a simple ID number. The ID is typically a manufacturer programmed 64 or 96 bit number, which can be the EPC and cannot be modified.
- Class 1: Read/write passive tags that can only be written once either for the manufacturer or the user.
- Class 2: Read/write passive tags that can be written several times. Additional functionalities like data logging and/or cryptography may be included.

2.1.1.2 Semi-passive tags

Semi-passive tags use energy generally from batteries to operate. The power provided by the batteries is used to feed the internal chip but not to broadcast radio frequency signals.

- Class 3: This class provides extended reading distance range (in contrast to classes 0, 1 and 2) as well as new functionalities like sensors and security.

2.1.1.3 Active tags

Active tags are provided with batteries allowing the tag to generate its own RF signals and provide additional features like sensors, encryption, data processing, among other functions.

- Class 4: Provide communication functionalities with other active tags and the same class 3 features.
- Class 5: It provides reader capabilities to communicate with tags and readers.

2.2 RFID Interrogators or Readers

As mentioned before, RFID readers are responsible for interrogating tags in order to obtain the ID they carry. They consist of three main components: a control module, a radio frequency module and an antenna part [**7**] (Figure 4).

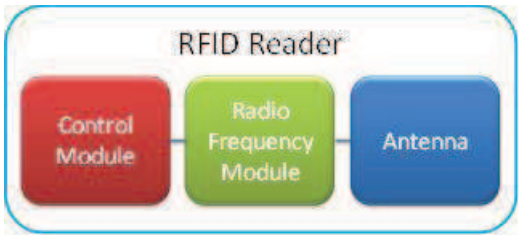


Figure 4. RFID Reader Modules.

The control module performs signal processing and runs the required communication protocols in order to communicate with tags and with the RFID middleware. The radio frequency module is responsible for the transmission and reception of information. This module is linked to the antenna part, which is the physical interface that will propagate and receive electromagnetic waves. Those waves carry the information that enables RFID communication.

RFID Readers can be classified regarding their mobility, the communication interface they provide, the power supply and the frequency band they operate [**7**] [**8**].

2.2.1 Mobility

Readers can be either fixed also called stationary or mobile. Fixed readers are mounted in portals or walls and generally, when the application does not require them to move, for example a security system in a library that detects tags attached to books. Mobile readers typically come in the form of a handheld device. Due to power considerations, handheld readers have less coverage range than stationary ones, but they provide mobility capabilities that are needed for certain applications.

2.2.2 Communication Interface

The RFID readers can be classified depending on the type of communication interface that they provide. They can be either serial or network readers. Serial readers use typical serial communication protocols to interact with applications (i.e. Universal Serial Bus). Conversely, network readers provide a variety of interfaces to communicate such as Ethernet, Wireless LAN, Bluetooth, and ZigBee.

2.2.3 Power Supply

The readers can be powered either by using AC/DC voltage adapters, connected to a power outlet or by using internal batteries. For mobile readers it is needed to have devices with batteries. On the other hand, fixed readers can be powered up by using wired power connections due to their stationary nature.

2.2.4 Frequency Band

RFID readers can operate in different frequency bands. The typically used are Low Frequency (LF) (125/134 KHz), High Frequency (HF) (13.56 MHz), Ultra High Frequency (UHF) (860-960 MHz, 2.4-2.45 GHz) and Super High Frequency (3 GHz and more) [**6**].

2.3 RFID Middleware

RFID Middleware is a software component that gathers and processes the data collected from the RFID readers. It also provides management and supervision functions of the RFID network [**9**]. The standard features of the RFID Middleware are data filtering, data aggregation, management capabilities, and Monitoring [**10**] (Figure 5).

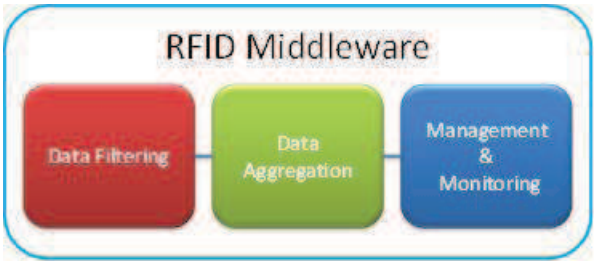


Figure 5. RFID Middleware functions.

Data filtering provides a way to minimize the amount of information treated. Data aggregation permits to build a repository of the information obtained for further access and/or processing. The Management features establish a configurable way to interact with the RFID network in order to control and optimize its operation. Finally, Monitoring allows keeping track of the activity, usage and performance of the network.

2.4 RFID Communications

Readers and tags communicate wirelessly. For passive and semi-passive tags, RFID uses Backscatter communication that is based on the properties of the objects to reflect electromagnetic waves. A tag has an antenna that allows it to obtain energy from the reader. This harvested energy will be used to feed the electronic circuitry embedded in the tag and that in turn will be used to communicate back with it [**2**]. A glance on the RFID communications basis is presented as follows by using the widely employed “EPC Generation 2” standard [**11**].

2.4.1 Physical Layer

2.4.1.1 Reader to Tag communications

RFID readers send information to tags by using a modulated RF (Radio Frequency) carrier. It can be modulated utilizing double-sideband amplitude shift keying (DSB-ASK), single-sideband amplitude shift keying (SSB-ASK) or phase-reversal amplitude shift keying (PR-ASK). ASK modulation is preferred over other types of modulation like OOK (On-off keying) or PSK (Phase-shift keying) because it requires a simple envelope detector and provides a constant power to the tag. The encoding is implemented by using Pulse-Interval Encoding (PIE) where a Tari (Type A Reference Interval) is defined as the width of the Data 0 symbol [**12**] (Figure 6). The modulated carrier also provides the required energy to the tag in order to operate.

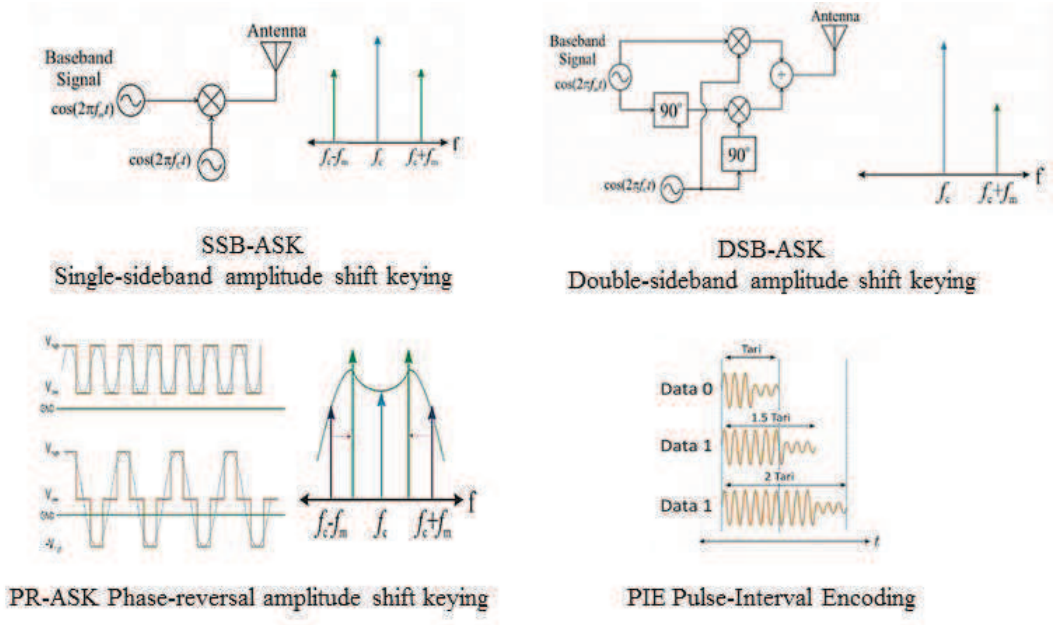


Figure 6. Reader to Tag Modulation Schemes and signal Encoding.

2.4.1.2 Tag to Reader communications

The tag transmits back to the reader by modulating the amplitude or phase of the RF carrier. In this case, the encoding format is either FM0 or Miller Modulation Coding. FM0 or bi-phase encoding inverts the phase of the signal at the beginning of each new symbol. During the period of a "1", symbol its value remains constant equals to "1". The value "0" has a single-phase change during its period [**11**] (Figure 7).

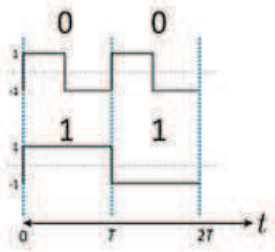


Figure 7. FM0 or bi-phase encoding.

Miller Modulation Coding provides a phase transition for the "1" value and "0" is constant over the symbol period. There is no phase transition in the symbols unless there are consecutive zeros. Finally, the baseband signal is multiplied by using a square wave [**12**] (Figure 8)

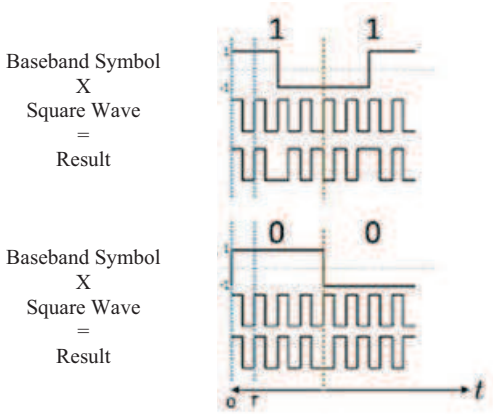


Figure 8. Miller Modulation Coding.

2.4.1.3 Tag-Identification Layer

In EPC-Gen 2, the readers talk first. The interrogators manage the tag IDs reading by using three basic operations [**11**]:

a) Select: This operation permits to choose a Tag set for inventory purposes from a group of tags. The analogy of this task is the selection of records from a database.
b) Inventory: This operation allows Tag identification and it is initiated by the Query command.
c) Access: this operation is used to communicate with Tags.

In order to avoid data collision in multi-tag environments EPC-Gen 2 uses an ALOHA-based protocol where the reader communicates the anti-collision parameters and then the tags randomly select a period to answer back. The anti-collision schemes are presented in more detail in the next section.

The example of a transaction between a reader and a tag is presented in Figure 9. First, the reader sends a Query command. The tag will reply with a 16-bit random number (RN16). Later, the reader sends an acknowledgment including the same RN16. This random number allows selecting only one tag in order to avoid data collision. The tag that matches this RN16 responds to the reader with its ID code. The reader then, can demand a transaction request (Req_RN) using the initial RN16. The tag will reply with a transaction handle that will be used for further commands transmission [**13**].

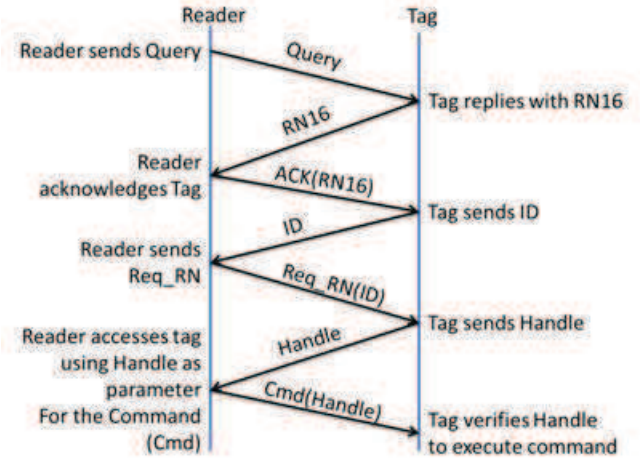


Figure 9. Reader-Tag communication messages.

2.5 Research issues on RFID.

The interest about RFID technologies increased the last decade especially for tracking and tracing objects and for a variety of applications that include the envisioned Internet of Things. Several research problems have been studied depending on technological limitations or specific application requirements. Here we present some of the research issues about RFID considering the Readers, Tags, and protocols.

2.5.1 Readers

The main function of RFID readers/interrogators is to obtain the identification code out of the tags. For passive and semi-passive tags, the reader powers them through electromagnetic waves thus sharing the same transmission channel and frequency. Consequently, the identification of the tags may be affected by interference and thus data collision. There are two types of collision: Reader collisions and Tag collisions [**14**]. Reader collision happens when near-located interrogators require the ID of the same tag. The reader signals interference does not allow the tag to be identified. On the other hand, Tag collision occurs when multiple tags reply at the same time to a Reader, therefore no Tag identification can be performed.

In order to avoid Reader collisions different methods have been proposed using access schemes like TDMA (Time Division Multiple Access), FDMA (Frequency Division Multiple Access) and CSMA (Carrier Sense Multiple Access). Colorwave [**15**] is a distributed TDMA based algorithm. In a network of readers, each of them will have a color to be identified with, thus the point is to have the smallest number of possible interrogators with the same color as neighbors. The readers in the network will randomly choose color ranges from [0, Maxcolors]. Maxcolor is an input to the algorithm and it remains constant during its functioning. This algorithm requires strict synchronization between the readers in order to operate. In FDMA, we can find the protocol Frequency Hopping Spread Spectrum (FHSS) where the operating frequencies change following a pseudorandom way [**16**]. In this way, readers can work using different non-interfering frequencies and avoid data collision. Both the reader and the tag must change to the same frequency at the same time what makes it a drawback for ordinary passive tags that have no frequency selectivity. In CSMA scheme, we find protocols like LBT (Listen-Before-Talk) and Pulse. LBT requires the reader to listen if there are existing ongoing transmissions in the channel before it accesses it. For dense reader deployments, this method is slow and can cause service denial if the saturation point is reached [**16**]. Pulse is a distributed protocol that separates the data and the control channels. This protocol transmits periodically a beacon signal in the control channel that indicates that a reader is reading tags. Any reader that wants to access the tags needs to check the control channel first. If no beacon is being detected, the reader can communicate with the tags. It will start transmitting a beacon in the control channel to avoid data collision [**17**].

2.5.2 Tags

Tags or transponders are composed by the antenna part and a chip module. The antenna captures the energy from the reader and feeds it into the Chip in order to make it work. The Chip is the core of the tag that runs all the logic tasks related to the transmission of the ID and the execution of commands [**6**].

2.5.2.1 Chip

The main objectives from research point of view are design optimization, new materials testing and performing protocols. This is done in order to obtain better performance and costs reduction [**18**]. New techniques as the Spread Spectrum is also considered [**19**]. Some other research go also towards the chip-less RFID systems [**20**] where the ID generation uses different alternatives such as SAW (Surface Acoustic Wave) [**21**], transmission lines and left-handed delay lines. This is useful especially for applications where electromagnetic waves are considered as a hazardous factor. This Chip-less

approach do not use silicon devices and a main advantage is that they can be printed directly on products.

2.5.2.2 Antenna

A considerable number of studies have been conducted on RFID tag design for metallic objects ([**22**], [**23**] and [**24**]) in recent years including antenna miniaturization [**25**].An interesting topic is the Chip-less RFID utilizing Inkjet-Printed Antennas. The field for this kind of devices is for electromagnetic sensitive applications [**26**], [**27**]. The implementation is based on low cost paper substrates and conductive ink, which consists of Nano-silver particles.

2.5.2.3 Data Collision

Multi-tag-parallel reading presents a drawback due to the interference generate for multiple responses (tags to reader) that leads to data collision. Tag data collisions are important to avoid, as every tag is important to be detected correctly. Anti-collision protocols are used to reduce this effect and thus minimizing the identification delay. The Anti-collision protocols preferred for RFID are those TDMA-based [**28**] (see Figure 10). Approaches like SDMA (Space-Division Multiple Access) and CDMA (Code-Division Multiple Access) are complex and expensive to be applied for commercial usage.

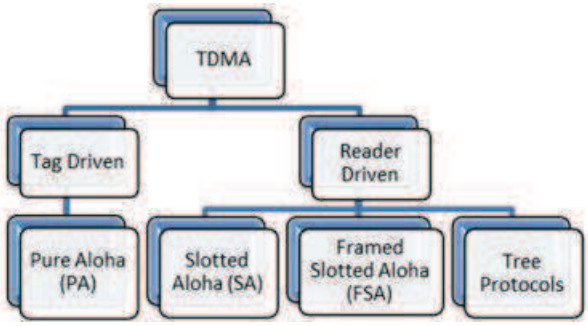


Figure 10.TDMA anti-collision tag-to-reader algorithms

In Pure Aloha (PA) algorithms, the tags transmit their IDs randomly after receiving power from the reader. The tags have a random counter that sets a delay and once the time is expired, they will try to send their IDs again if collision existed. PA variants (Figure 11) include slow down, muting, fast mode, and combinations of them. In slow down, tags are ordered to reduce their transmission rate and for instance reducing the collision probability. The muting variant mutes the successfully identified tags, reducing the reader's identification load. The Fast mode silences the tags that are not being identified.

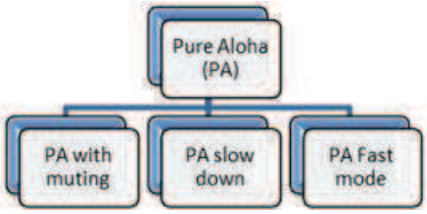


Figure 11.Pure Aloha variants

The Slotted Aloha (SA) works in synchronous mode, defining transmission periods or slots. There are four variants: muting and slow down, early end, early end and muting and slow down and early end (Figure 12). The muting and slow down work as mentioned before for PA algorithms, however synchronous slots are used to detect tags. The early-end feature allows to end a transmission slot and prevent other tags to collide with a successfully identification process in progress.

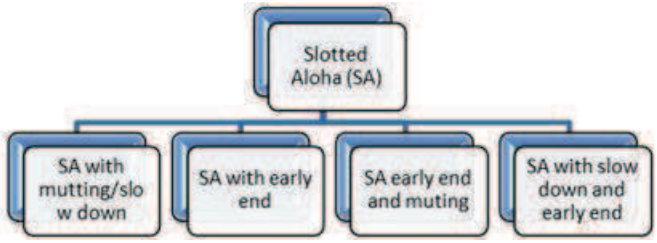


Figure 12.Slotted Aloha variants

In Frame Slotted Aloha (FSA) tags can transmit theirs IDs only once per frame. The basic and dynamic classification refers to the size of the frame defined from the reading process. For Basic FSA (BFSA), muting and no muting are possible. The early-end feature described before can also be applied generating two more variants (Figure 13).

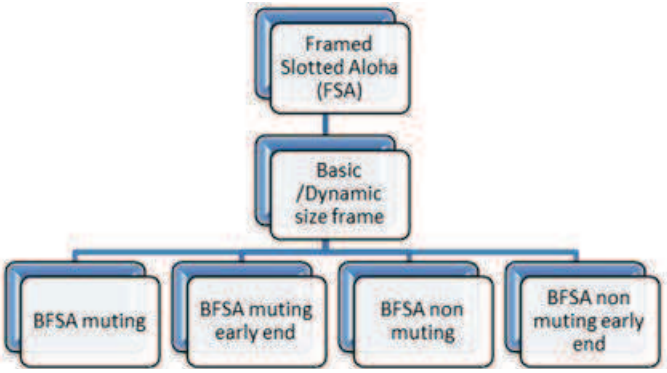


Figure 13.Frame Slotted Aloha variants

Tree protocols divide the tag space in order to perform the identification process. There are four categories: Tree Splitting (TS), Query Tree (QT), Binary Search (BS) and Bitwise arbitration (BTA). In Figure 14, the tree-based protocols are shown:

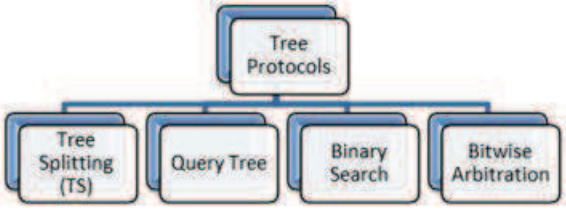


Figure 14.Tree Protocols

TS are divided in Basic (BTS) and Adaptive (ABTS), and enhanced (see Figure 15). TS divide the collided tags into n disjoint subsets. The BTS minimizes the subset until only one tag is present. The tags provide one counter to keep track of the tag position into the tree. ABTS reduces the idle timeslots obtaining a fast tags identification process. It requires two counters in the tags. Enhanced BTS keeps track of the ID bits transmission, indicating a colliding bit by a pointer. Tags will transmit later only the bits that start from the collided one marked with the pointer.

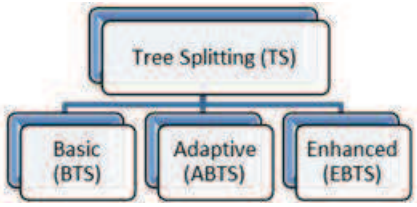


Figure 15.Tree splitting Algorithms

Query Tree (QT) algorithms relay on the processing power of the reader [**29**] (Figure 16). It will keep an appending queue registry, in order to identify tags that match with the binary sequence required. Some variants are Shortcutting (QTS) where the redundant queries are deleted, aggressive enhancement (QTAE) which requires appending multiple bits instead of one on the identification queue. Categorization implies prior knowledge of tag's IDs to classify the queries. Short-Long (SL) separates the queries into short (one bit appending) and long (whole ID). Adaptive (AQT) requires the reader to keep track of the past prefix required to be identified. The QT Improved version reduces the number of bits send by the tags back when collision exists. Randomized Hashing (RHQT) indicates to each tag the generation of random number from a predefined hash function. Intelligent QT exploits tag's prefix patterns.

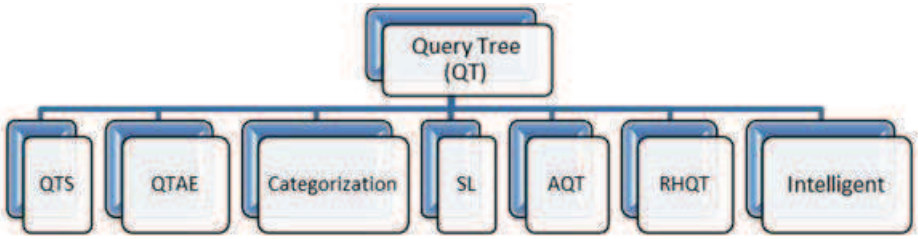


Figure 16. Main Query Tree protocols.

Binary Search algorithms (BSA) are based on the transmission of binary digits that tags will compare with their IDs enabling them to transmit them if the comparison results positive or if the digits are less than the IDs themselves. We can find two variants: Enhanced BS (EBSA) and Dynamic BS (DBSA) (Figure 17). On EBSA the reading process is not restarted after a successfully tag identification. DBSA does not require the whole ID to identify the tags and it can be divided to optimize tag's identification.

Figure 17. Binary Search Algorithms

Bitwise Arbitration algorithms (BSA) request tag's IDs in a bit-by-bit manner. The main variants are ID-Binary Tree Stack, Bit-by bit (BBT), Modified and Enhanced BBT, and Bit Query (Figure 18). ID-Binary Tree Stack splits bit by bit the ID creating a tree and the tags keep track on the bits in order to transmit when they identify their IDs and then go to sleep state. BBT uses separate channels to transmit the binary digits. In MBBT, multiple slots are not used to obtain the binary digits. EBBT require tags to send their entire IDs and then the reader observes the collided bits. Finally, BQ scheme transmit a binary query to tags, which respond based on a prefix.

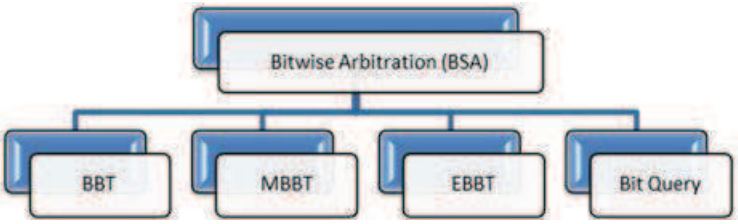


Figure 18.Bitwise Arbitration protocols

2.5.3 Middleware

There are several proposals of RFID Middleware architectures. One of the main targets is to tackle scalability issues in large RFID systems and to reduce data redundancy. In [**30**] a model based in a load balancing algorithm is presented in order to support large-scale enterprise RFID applications. In [**31**] researches present an architecture based on Distributed Hash table (DHT) and Peer-to-Peer in order to solve scalability issues in an Internet of Things context. A hierarchical structure of RFID middleware modeled by using UML (Unified Modeling Language) is presented in [**32**]. An application-oriented algorithm for data cleaning is proposed. The focus is to reduce the redundancy of the data collected from RFID readers [**33**].

2.5.4 Security

In order to protect the information exchange between tags and readers it is necessary to provide security mechanisms trying to avoid possible attacks, especially if sensitive information is managed. One of the most used standards on RFID technology the EPC global Gen2 faces a security issue because it transmits the EPC code as plain text. Many researchers have proposed hash-based protocols [**34**], [**35**], and [**36**] that consume less computational resources than cryptographic primitives hash functions such as DES (Data Encryption Standard) or AES (Advanced Encryption Standard). The asymmetric method is even more resource consuming and is mainly used for key management in RFID systems.

2.6 RFID Applications

There is a wide employ of RFID systems for tracking and tracing purposes, however, they are used for building access control, persons, animals or items tracing, location services, process control among other uses. Some real applications will be described for each of the different categories presented [**1**].

2.6.1 Logistics and Supply Chain

The most important and massive application of RFID technology is in logistics and supply chain. In distribution and logistics of many types of products, tracking and tracing, concerns a process of determining the current and past locations and related information of a unique item or property. Due to the non-line of sight characteristics of RFID, it makes this technology suitable to replace the predecessor barcode. The use of RFID for logistics on the supply chain allows the inventory and tracking of items more efficient and reliable. For price constraints, only passive tags are suitable for these applications. Some of the most important retail companies worldwide had implemented RFID system to provide a control and management platform. METRO Group in Germany had implemented RFID logistics control and storage management since 2004 by using passive tags [**37**]. Wal-Mart (US), Tesco (UK), AUCHAN (FR) and Proctor & Gamble (worldwide) are also using RFID technology for the same purpose.

2.6.2 Production, monitoring and maintenance

Another important application of RFID technology is on production lines. Typically open systems with no security features are employed. RFID gives control on assembly lines, indicating to the personal where and what parts are to be integrated, minimizing assembly errors and possible delays. BMW uses a system designed and installed by Siemens. The system places active RFID tags on finished vehicles as they leave the production line to help BMW workers instantly locate cars before they are shipped to vendors.

2.6.3 Product safety, quality and information

Most products can be in one way or another labeled with RFID tags. It provides a platform for tracking and tracing goods that serve for a variety of applications, e.g. fighting against counterfeiting; labeling and identifying original products allows them to be traced and tracked. In addition, the use of embedded sensors into RFID devices allows extending the basic identification capabilities to include environment sensing. A commercial example comes from an Italian company called CAEN [**38**]. It sells semi-passive tags, which include a temperature sensor with a capacity of 8000 samples, suitable for fresh/perishable food control. In hospital management RFID can help in tracking patients to reduce medical errors of prescriptions also for inventory management and medicines control.

2.6.4 Access control and tracking and tracing of individuals

RFID technology provides a useful approach to track and trace individuals. Non-line of sight properties, contactless cards, miniaturization, and Nano-technology are elements that combined with RFID systems provide interesting results. For example, the company “Destron Fearing” provides RFID solutions for cattle tracking on the frequency band of 134.2 KHz using a 15-digit identification number [**39**]. The company “Biomark” also proposes RFID technology to implant chips in animals with models that can measure about 12.50mm X 2.07mm in the frequency range of 134.2 kHz with a weight of 0.1020g [**40**]. VeriChip [**41**] is a tag looking as a rice grain that is inserted under the human skin for tracking purposes. Applications are currently targeting tracking emergencies in hospitals where injured persons cannot talk. MIFARE is a technology that has been selected for most contactless smart card projects. Its product portfolio includes products for applications such as loyalty, road tolling, access management and gaming. The cards support dynamic download of Java applications [**42**]. Metro systems as the Tokyo one, uses “Suica” cards for access control and vending machines [**43**]. In Paris and London the metro access, control works with contactless prepaid cards (Navigo Pass, Oyster Card). An application for inmates’ surveillance has been implemented and commercialized by the company “Tsiprism”, which is proposing a safe and reliable security control in prisons [**44**].

2.6.5 Loyalty, membership and payment

Contactless smart cards allow storing information that is used to identify clients for loyalty and/or membership affiliations. The main advantage is the fast and automated identification done by the readers. American food chain Dairy Queen deployed a mobile rewards loyalty program using RFID tags to send coupons and offers to consumers' handsets. Metro (Germany) also provides RFID cards for its membership/loyalty program. MasterCard (PayPass) provided a new contactless card to perform common transactions in stores. Passive tags are used for these applications.

2.6.6 Household

Controlled environments and Smart homes in home networking benefit of RFID technologies in order to identify, track objects, providing security solutions, inventory tracking, and location features. A commercial example is the RFID fridge (developed by Samsung in 2007) that allows inventory tracking of available products and communicating via internet or short message service with its users. All types of tags are suitable for these applications.

2.6.7 Other applications

Any application or system that requires mainly the identification, tracking and tracing of its elements can benefit from RFID technology. Some examples of these systems are public transportation, libraries, tree identification, and ecological related monitoring systems among others.

Other applications can use the identification information stored in the RFID and match it with some semantic specific to some application. For instance using RFID can improve some network functionalities such as location and mobility support of nodes. Combining Location based services with RFID system is also a very promising application assuming that the privacy issue is solved [**1**].

2.7 RFID Related technologies

The following technologies are related to RFID, presenting interesting variations that can be used for new applications.

2.7.1 NFC (Near Field Communication)

NFC is a communication technology employed for short-range high frequency wireless applications. It defines an interface and protocols build on top of RFID [**45**]. The effective range it is about 10 centimeters similar to proximity-card devices but with new features added like two-way communication (as well as one-way support), secured communications and simplification of the network configuration [**46**]. This technology is an extension of the standard ISO (International Organization for Standardization)/IEC (International Electro technical Commission) 14443 for smartcards and readers. Its frequency band is 13.56 MHz and supports varying data rates (106kbps, 212kbps, and 424kbps). Commercial products like the mobile phone Samsung Galaxy II include this feature and applications can cover contactless payment and ticketing [**47**]. “TouchaTag” [**48**] is a kit that can be used to develop applications exploiting the tag information to trigger actions.

2.7.2 Contactless Smart Cards

Smart cards communicate based on RFID technology but adding security features like encryption. They follow the ISO/IEC 14443 standard and require close proximity to an interrogator to perform any transaction. Basic applications cover mass transit systems, credit cards and access control. In contrast with NFC, they can only communicate in a one-way mode [**49**].

2.7.3 Nano-RFID

Nano-RFID is the miniaturization of RFID transponders based on nanotubes, which are carbon structures with special mechanical and electric properties [**50**]This technology offers an unlimited range of applications that include not only the traditional RFID identification features but also to measure and sensing external variables and possibly to have effect on biological functions when used on living beings including humans [**51**].

2.7.4 RFID sensors

RFID sensors extend the basic identification feature and provide extra capabilities. Applications include external sensing parameters, i.e. temperature, humidity, pressure among others [**50**].

2.8 Chapter Summary

In this chapter, we presented the basic concepts related to RFID. We described the different components that configure this technology, Readers, tags and middleware. RFID technology gained attention during this last decade especially in building the envisioned Internet of Things and its wide

use for tracking and tracing items. However, there are technical issues that are still under research in order to improve certain aspects of the technology. The collision problem and the devices heterogeneity (diverse frequency bands, protocols and usages) are key elements when considering multiple reader deployments. In addition, the interactions of RFID with other devices like sensors and actuators open new paths and will permit the development and implementation of more innovative applications.

Chapter 3

RFID Framework for heterogeneous nodes

This chapter presents the work related to our first thesis objective: provide a heterogeneous RFID-based platform to generalize RFID information based services. First, we will describe the architecture and then we will provide measurements obtained through simulations.

3.1 Research Problem

As previously stated, Radio Frequency Identification (RFID) technology is becoming very popular due to the reducing cost of their components [**52**] their versatility and the myriad of RFID applications that can be implemented [**1**] [**53**]. However, RFID hardware is not widely available in current personal mobile terminals limiting the provision of novel applications and services. Merging RFID networks and massively used portable devices (mobile phones, laptops and pads) for the provision of novel applications and services appears as an interesting issue from both commercial and technological perspectives. However, those devices are not yet widely RFID-enabled and few recent attempts are just limited to a short-range technology called NFC (Near Field Communication) [**45**] where the effective reading range is less than 10 centimeters. While we observe a lack of portable RFID-enabled devices, on the other hand, mobile phones with Wi-Fi capabilities have been invading the market over the last decade and recent studies forecast that 2.6 billion units will be available in 2015 [**54**]. Wi-Fi is based on IEEE 802.11 family of Standards that are the most popular short-distance wireless access protocols [**55**] enabling the implementation of affordable, flexible and scalable communication platforms.

In an Internet of Things scenario, different devices can be connected in order to provide information services. This is the case of sensors, actuators and both fixed and mobile RFID readers. In order to offer services to any device with or without RFID capability, we need a new architecture that combines these heterogeneous devices into a collaborative system that extends the RFID reading functionality and other information parameters to any connected device. We need also to consider the fact of having multiple RFID readers working at the same time. It might lead to interference that in turn can cause data collision and consequently affect the exploitation of the RFID information based application.

3.2 Research Objectives

Taking into account these premises, we propose the integration of RFID networks with a variety of devices including Wi-Fi enabled ones but also considering other protocols and interfaces (ZigBee, Ethernet) in order to provide RFID-based services. We need to provide the classical RFID Middleware functions, which are data filtering, data aggregation, monitoring and management, but also providing additional capabilities. Our objective is to build a unified access to the RFID information to devices with or without RFID interface, fixed or mobile devices connected to the network. More precisely, we propose a hybrid client-server-P2P (Peer-to-Peer) framework that targets the inclusion of heterogeneous devices (RFID mobile and fixed readers, portable Wi-Fi enabled devices, sensors and actuators) to provide information services and interaction among all the actors of the framework in a unified and efficient way. Peer-to-Peer (P2P) architectures are considered as flexible solutions when computational resources and data sharing are involved. In such approaches, nodes can act as clients or servers depending on the required tasks. The main advantage of P2P against classical client-server approaches lies in the distributed deployment of the services and the scalability, flexibility and fault-tolerance of the network. Hybrid architectures combine both the

centralized control and the resources sharing among the peers that are connected through a network Overlay We also considered the interference problem regarding the reader-to-reader interference in order to reduce the data collisions in the system. We propose a time-slot assignation algorithm that schedules the reading process of every RFID reader in the framework when required extending the basic RFID Middleware functions.

3.3 Related Work

P2P-RFID collaborative networks have been proposed in previous research work. In [**56**] to reduce reading collisions in multi-reader environments a P2P scheme is presented. In [**57**] a P2P Collaborative RFID Data Cleaning Model is suggested in order to identify and remove inaccurate reading of RFID data. A P2P data resolver is implemented in [**58**] that by using a P2P network identifies the object associated to a certain RFID tag ID. We observed also research on 802.11 and RFID integration in [**55**] for localization services. In [**59**] an architectural RFID system for localization is depicted. It presents a framework based on edge and intermediate nodes, in a distributed manner. We observe that the P2P approaches presented are not totally focus on integrating a variety of devices into a single information platform

In [**60**] a hybrid middleware for RFID-based parking management system is presented. This approach combines classical middleware with extended P2P events management. Researches in [**61**] propose a RFID Middleware platform called FlexRFID that targets the implementation of a device neutral interface to communicate with different hardware devices. It also provides an interface to access the hardware for management and monitoring. In [**62**] a middleware proposal links RFID technology with GPRS (General Packet Radio Service) and Bluetooth interfaces, in a Client-Server approach providing certain heterogeneity.

Regarding the RFID reader-to-reader interference different research work have been proposed ([**56**], [**63**], [**64**], and [**53**]). This problem has been tackled by using several techniques based in methods that share the same transmission medium to operate. As we have seen in Chapter 2, there are methods associated with TDMA, FDMA and CSMA schemes. In TDMA we find *Colorwave* [**15**] as a distributed algorithm. It operates in a synchronized way assigning colors to readers. They can operate if no similar colors are assigned to neighbor readers. In FDMA, we can find the protocol Frequency Hopping Spread Spectrum (FHSS). The readers will operate changing their frequency by following a pseudorandom way [**16**]. The use of different frequencies reduces the data collision. Both the reader and the tag must change to the same frequency at the same time what makes it a drawback for ordinary passive tags that have no frequency selectivity In CSMA, we find protocols like LBT (Listen-Before-Talk) and Pulse. LBT listens the channel to detect possible ongoing communications and when it is free, the reader can operate. For dense reader deployments, this method is slow and can cause service denial if the saturation point is reached [**16**]. Finally, Pulse is a distributed protocol that separates the data and the control channels. This protocol transmits periodically a beacon signal in the control channel that indicates that a reader is working. This beacon is checked periodically in order to avoid interference [**17**]. For our framework, the must-adapted scheme will be TDMA because we will work with heterogeneous devices that will operate in different frequency bands, with different protocols and from different vendors that might not be standard. We will orchestrate the scheduling at the P2P level.

The novelty of our contribution is the integration of non-RFID devices into the RFID network via the P2P framework to extend the use of applications and services. Additionally, we propose a high-level time-division-based scheduling mechanism allowing RFID readers to operate in a sequenced manner. The next section will describe the system's architecture depicting all the entities and features. Later the simulations regarding the service delay are presented. We used NS2 (Network Simulator 2) in order to model the data exchange process between nodes and measure the associated transmission delay making the following assumptions:

- Readers use the EPC global Generation-2(Gen2) protocol to obtain tag's identification numbers.
- Tags' antennae are never at 90° with respect to the reader thus tags can be always detected.
- All the nodes have a WLAN (Wi-Fi) interface and can run the P2P protocols required or they are attached to a device that can run them.

3.4 System Architecture

Our proposal has as primary goal the integration of devices with or without RFID hardware into a single framework. This framework will provide information services based on RFID and other devices like sensors and actuators. It offers also a TDMA-based data anti-collision mechanism in order to reduce the RFID reader-to-reader interference. Additional features include service authentication and AAA functions (authentication, authorization, and accounting). The framework consists of the following nodes (Figure 19).

- **Multi-Mode-Nodes (MMN):** This node is the interface with heterogeneous devices. It will provide the ports needed to connect RFID readers, sensors and actuators as well as network interfaces (WLAN, Ethernet, and ZigBee).
- **Management and Authenticator Nodes (MAN):** It performs AAA functions and resources management.
- **Users**: They are nodes that will request RFID information. They are grouped into
  - devices with non-RFID capabilities named Virtual Readers Nodes (VRN)
  - Mobile RFID Reader Nodes (MRN)
- Sensors, Actuators and RFID tags.

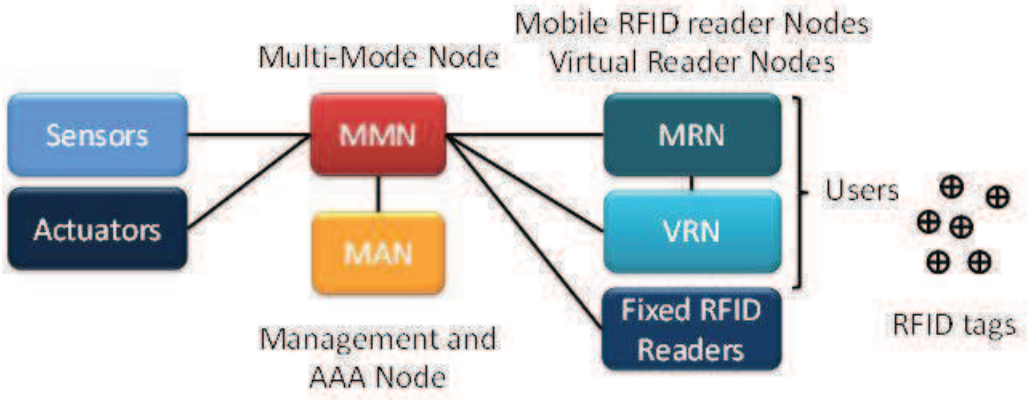


Figure 19. Heterogeneous RFID Framework

3.4.1 Peer to Peer Overlay

This work describes the architecture design of the system but no implementation is proposed. However, in order to provide the implementation of the hybrid P2P network overlay we can use JXTA, which is a Sun's set of standard protocols [**65**]. JXTA originates from the word *juxtapose* indicating that both approaches: Client-Server and P2P can coexist. With these protocols, developers can be focused only on the application layer to implement P2P solutions co-existing with Client-Server platforms. JXTA defines a set of six XLM-based protocols that allow the establishment and use of P2P networks. These protocols are independent of the underlying implementation. Functions include organize, discover and publish resources, communicate and monitor peer activity. With these protocols, a virtual network overlay can be established on top of physical networks in order to allow peers to interact no matter which network or connectivity type is used. The standard JXTA protocols are six: Endpoint Routing Protocol (ERP), Rendezvous Protocol (RVP), Peer Resolver Protocol (PRP), Peer Discovery Protocol (PDP), Peer Information Protocol (PIP), and Pipe Binding Protocol

(PBP). These protocols are independent from each other. A peer is not required to implement all of them to be a JXTA peer. It will implement only the protocols needed. JXTA provides a simple and generic P2P platform independent of the programming language used to implement it, the operating system, the network topology, and the security model used. Before describing the set of protocols, the following concepts are introduced:

3.4.1.1 XML

JXTA uses widely the standard eXtensible Markup Language (XML) to structure data in form of messages, protocols and advertisements. XML is language-neutral so any programming language able to parse text can parse XML messages. Its simplicity allows easy debugging and the message is self-describing because it is conformed of a series of tags that identifies each field. An example of an XML message is shown:

```
<?xml version="1.0"?>
<!DOCTYPE jxta:PipeAdvertisement>
<jxta:PipeAdvertisement>
<id>
jxta://596162616461626 14A757874614D50474168B1395E034DEA90F3BC8
CD7D
3618400000000000000000000000000000000000000000000000000000000000
0000401
</id>
</jxta:PipeAdvertisement>
```

3.4.1.2 Peers

It is the basic unit of a JXTA network. They can be any kind of networked device that implements the necessary JXTA protocols. They are identified by IDs. Each peer can publish one or more network addresses that define endpoints, used for communication between peers. Peers can be producers and services consumers at the same time.

3.4.1.3 Peer Groups

The way to organize peers is peer groups. They are conformed for peers that share common interests and/or services. Peers can join multiple groups. By default, the first group instantiated is the Network Peer Group.

3.4.1.4 Pipes

It is the basic element to perform communication between peers. It provides a virtual channel to send and receive messages connecting endpoints.

3.4.1.5 Advertisements

Advertisements are represented with XML documents that indicate the presence of resources (peers, peer groups, services, pipes, information).

3.4.1.6 Messages

A message is an XML document used to exchange information between peers in a JXTA network. When a message is created it only contains the basic structure and then the elements can be added. These elements can be of three different types:

- **ByteArrayMessageElement** The element consists of a byte array.
- **TextMessageElement** The element contains a string.
- **InputStreamMessageElement** Java InputStreams can be added directly into the element, which makes it ideal for file transfers.

3.4.1.7 Endpoint Routing Protocol (ERP)

The ERP establishes a mechanism where peers can discover a route (sequence of hops) in order to exchange messages with other peers. ERP is used to determine the route when network topologies change and previous routes are not available. The protocol defines the following messages:

- **Route Query Message:** It is sent by a peer when it wants to determine the set of ordered peers to use to send a message to a given Endpoint Address. The message format is shown below (Figure 20):

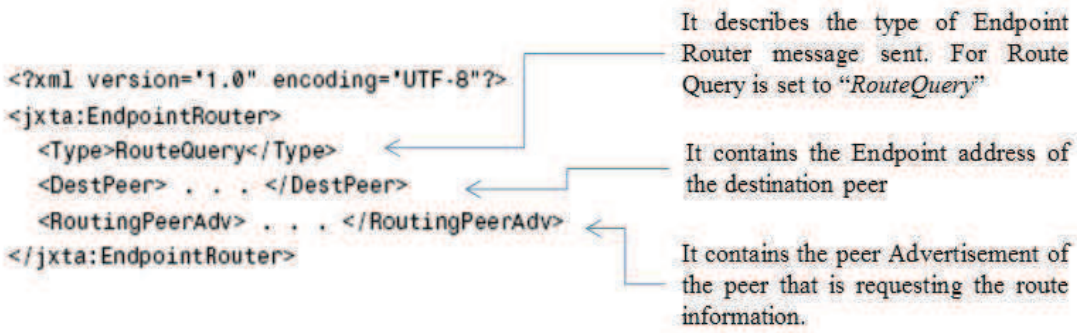


Figure 20. Route Query Message.

- **Route Response Message:** it is sent as a response to a Route Query Message. It describes a set of ordered Endpoint Addresses to use to send a message to a given destination peer. The message format is shown below:

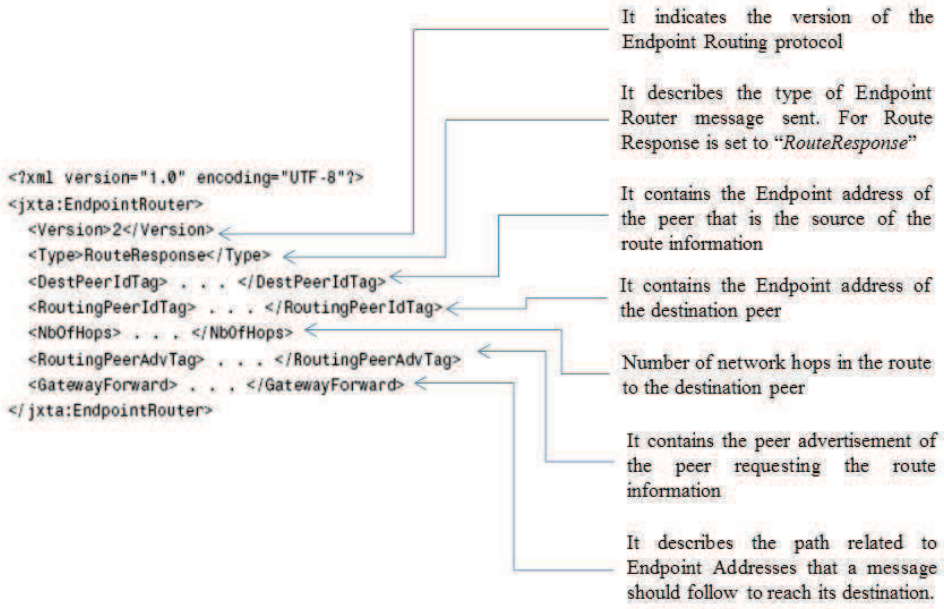


Figure 21. Route Response Message.

- **Endpoint Router Message:** It provides the route information among peers. The format of this message is shown in Figure 22:

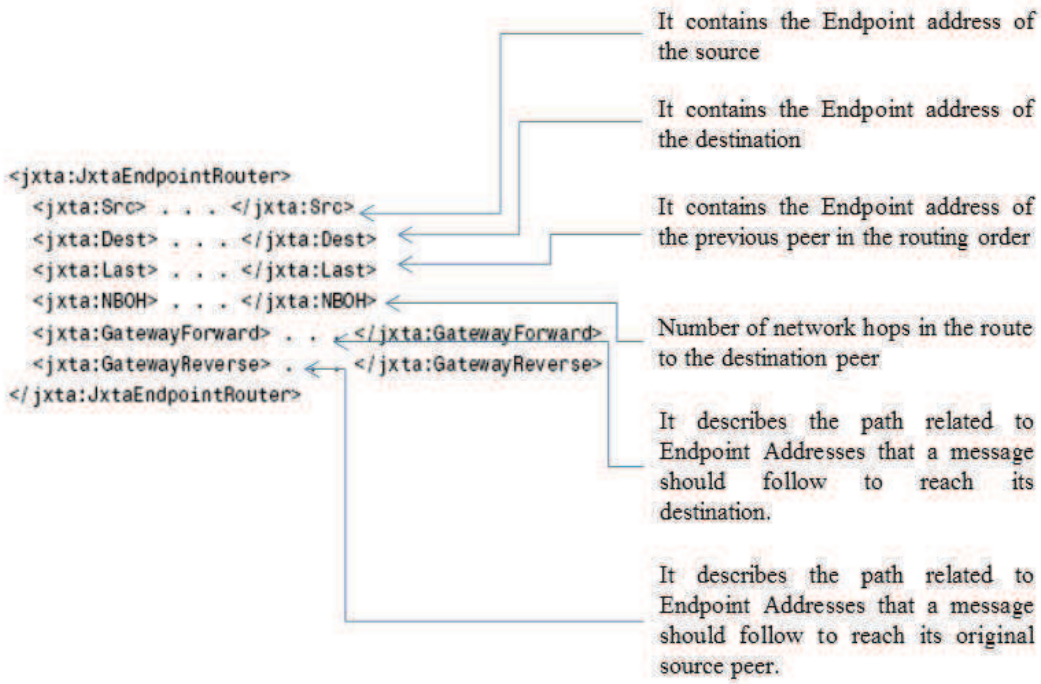


Figure 22. Endpoint Router Message.

3.4.1.8 Rendezvous Protocol (RVP)

RVP is used by peers to subscribe or be a subscriber to a propagation service. RVP allows sending messages to all the listening instances. This protocol is used by PRP and PBP to propagate messages. The RVP defines three message formats:

- **Lease Request Message** It is used by a peer to request a connection lease to the rendezvous peer. The message element is defined as follows:

Element Name	Element Content
jxta:Connect	The Peer Advertisement of the peer requesting a connection lease from the rendezvous peer.

- **Lease Granted Message** It is used by the rendezvous peer to approve a peer's Lease Request Message. It also provides the length of the lease. The message elements are defined as follows:

Element Name	Element Content
jxta:RdvAdvReply	(optional message element) It contains the Peer Advertisement of the rendezvous peer granting the lease.
jxta:ConnectedPeer	A required message element containing the Peer ID of the rendezvous peer granting the lease.
jxta:ConnectedLease	A required message element containing a string representation of the lease time, in milliseconds.

- **Lease Cancel Message** It is used by a peer to disconnect from the rendezvous peer. The message element is defined as follows:

Element Name	Element Content
jxta:Disconnect	The Peer Advertisement of the peer requesting

disconnection from the rendezvous peer set of connected peers.

These messages are not represented as XML formats but as message elements instead. They can be embedded as XML for other protocols (e.g. ERP).

3.4.1.9 Peer Resolver Protocol (PRP)

PRP establishes a mechanism for peers to send queries to one or several peers and receive one or multiple responses to those queries. The response message is matched by using a unique ID included in the message. The Resolver service needs two types of messages:

- **Resolver Query Message** it is used for sending queries. The message format is shown in Figure 23:

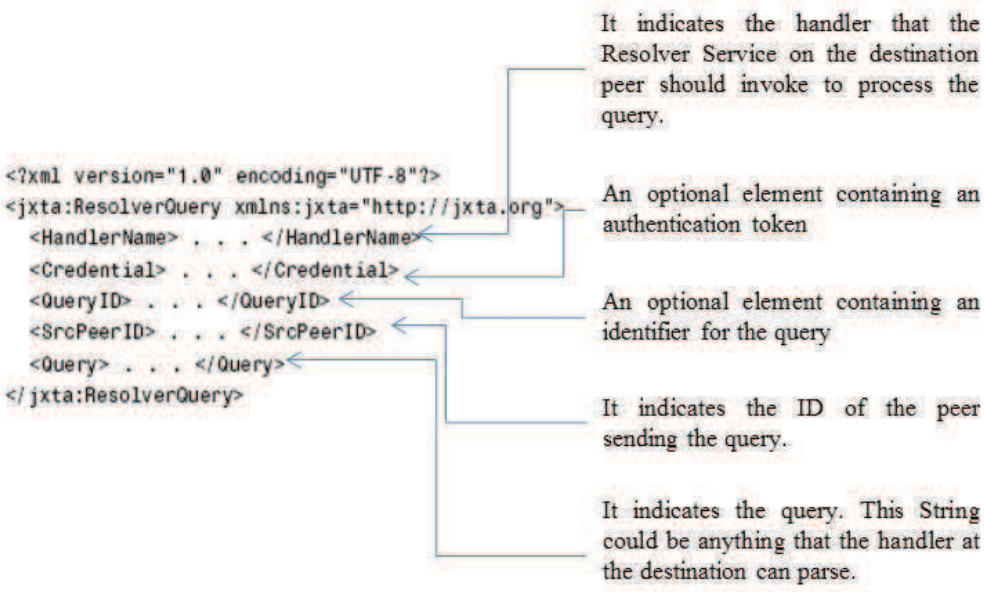


Figure 23. Resolver Query Message.

- **Resolver Response Message** it is used for sending responses to queries. The message format is shown in Figure 24 (the fields Handler, Credential and QueryID represent similar parameter as in the Resolver Query Message).

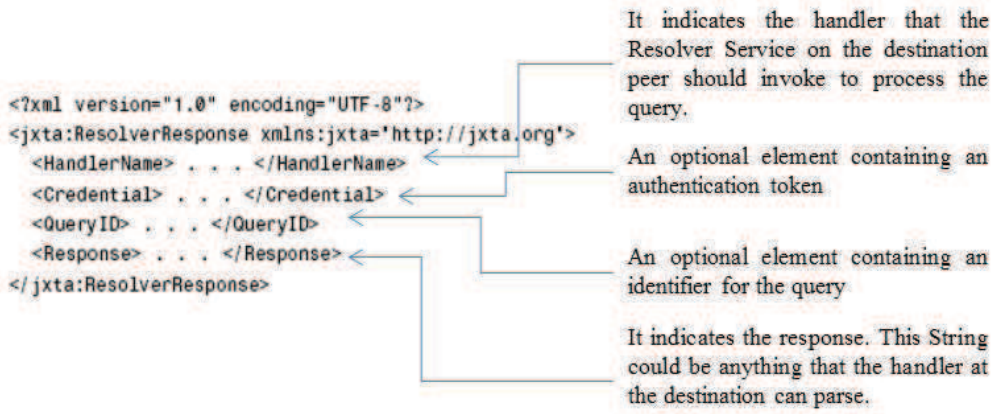


Figure 24. Resolver Response Message.

3.4.1.10 Peer Discovery Protocol (PDP)

This protocol allows peers to advertise and discover resources. Advertisements are programming language-neutral metadata structures that describe network resources and XML is used to represent them. This protocol consists of only two messages that define:

- A request format to use discover advertisements
- A response format for responding to a discovery request

The Discovery Query Message and the Discovery Response Message define all the elements required to perform the discovery process between peers. The Discovery Query Message format is shown in Figure 25:

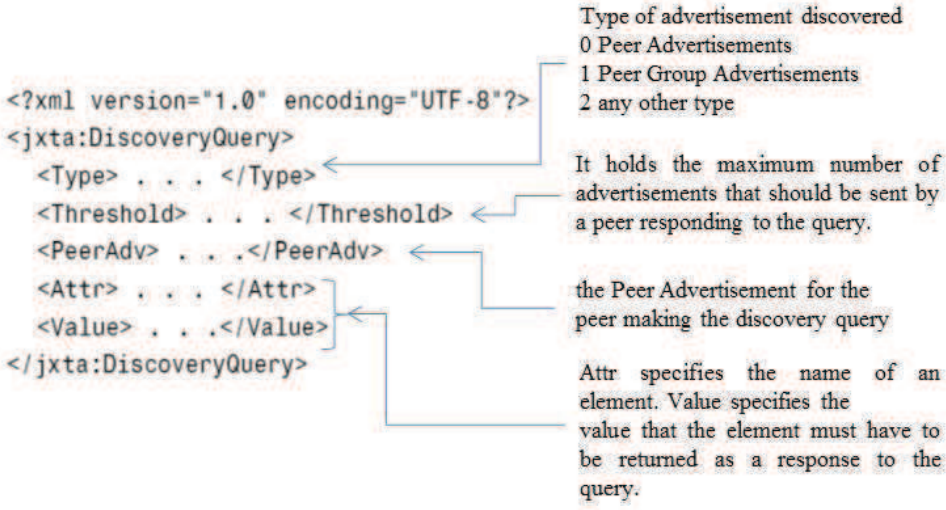


Figure 25. Discovery Query Message format.

The Discovery Response Message format is presented in Figure 26 :

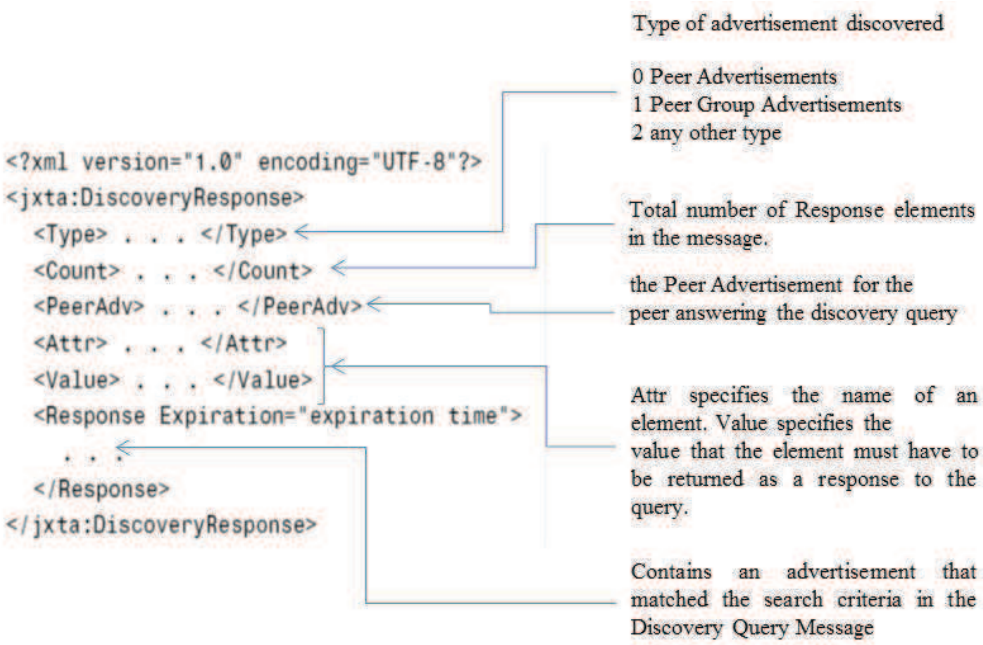


Figure 26. Discovery Response Message format.

3.4.1.11 Peer Information Protocol (PIP)

When information about peers (including state, uptime, traffic load, capabilities among others) is required the PIP is used. This protocol requires two types of messages:

- **Peer Info Query Message** It is used for querying a remote peer's status (Figure 27).

Figure 27. Peer Info Query Message.

- **Peer Info Response Message** It is used for providing a peer's status to other peers (Figure 28).

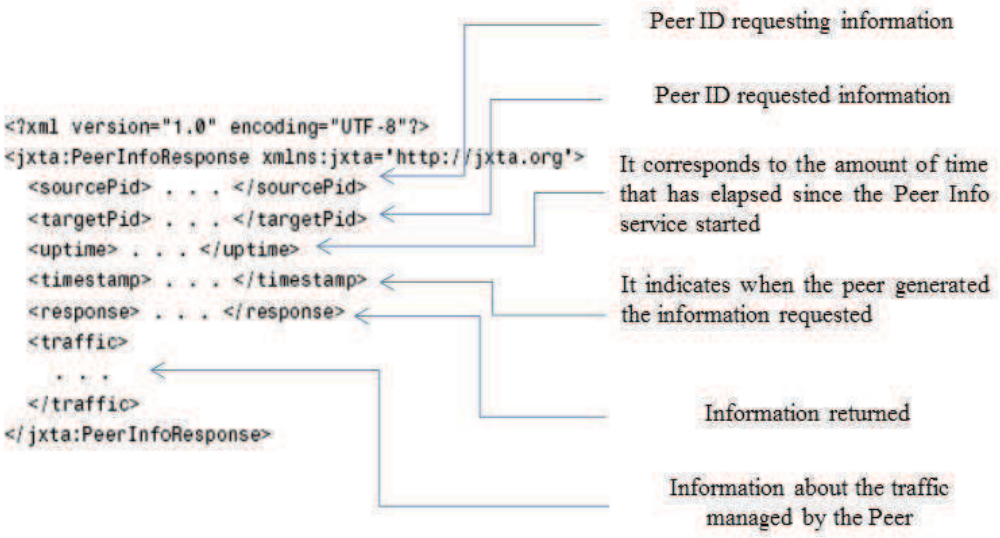


Figure 28. Peer Info Response Message.

3.4.1.12 Pipe Binding Protocol (PBP)

This protocol allows peers to establish a virtual communication channel or pipe between one or more peers. It is used to bind endpoints providing the basic communication mechanism in a JXTA network. Pipes are described using The Pipe advertisement, which is defined as follows (Figure 29):

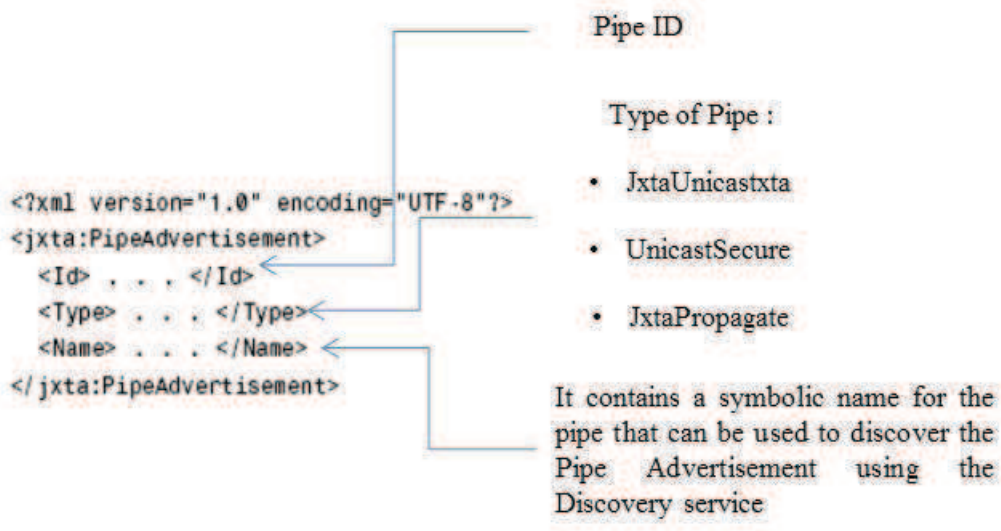


Figure 29. Pipe advertisement.

The PBP defines two messages to enable a peer to resolve a pipe:

- **The Pipe Binding Query Message** It is used for querying a remote peer if it has bound a pipe with a matching Pipe ID (Figure 30).

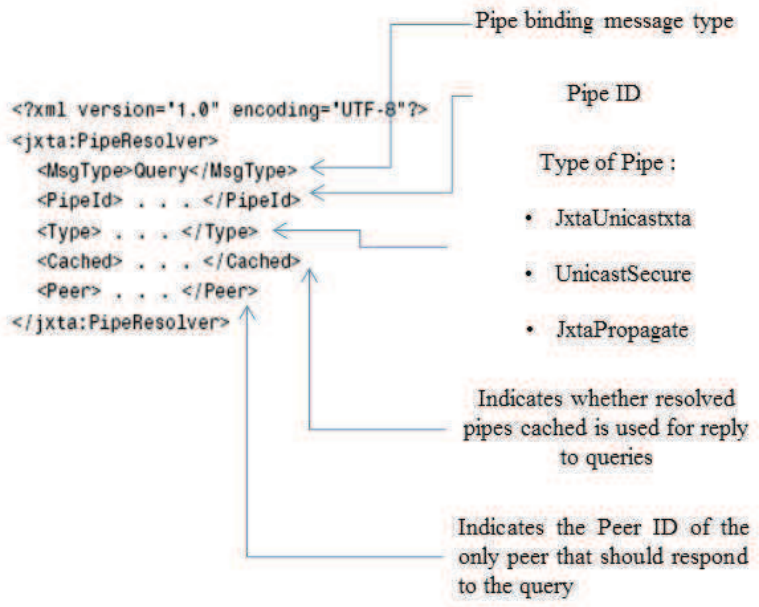


Figure 30. Pipe Binding Query Message.

- **The Pipe Binding Answer Message** It is used for sending responses to the query (Figure 31).

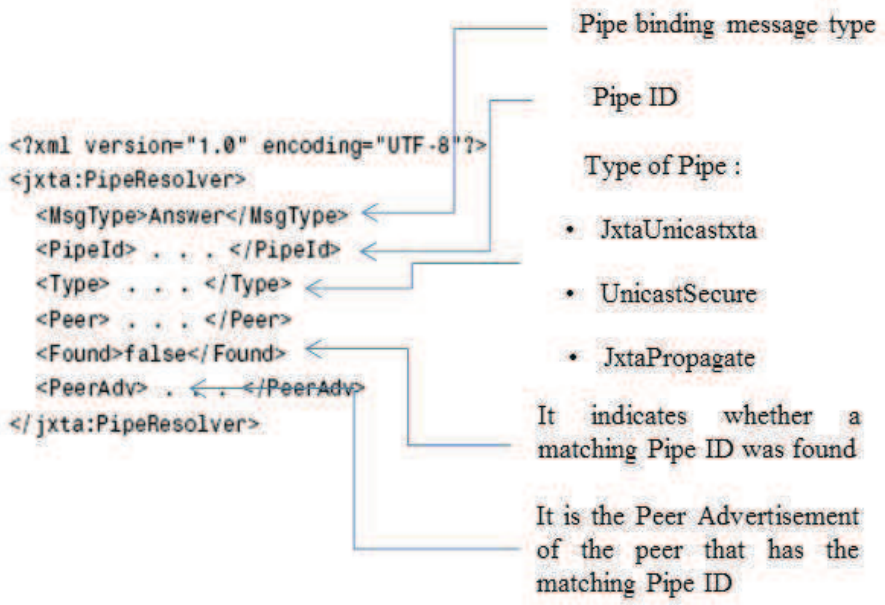


Figure 31. Pipe Binding Answer Message.

3.4.2 Multi-Mode Node (MMN)

3.4.2.1 Definition

A Multi-Mode Node is a device that will provide information services to users. It will also serve as control and supervision interface to manage diverse heterogeneous peripherals like fixed RFID readers, sensors and actuators. It provides wired and wireless connectivity to users. The MMN will also implement data filtering and data aggregation features. Data filtering is a mechanism that permits

performing specific queries for tag IDs or any other network information. Data aggregation permits building a repository of the information managed by the MMN keeping it available for access through the P2P Overlay. Based on JXTA the MMN will implement ERP for routing purposes, PRP to manage queries to resources and PDP in order to advertise and publish resources and PBP to establish the message exchange mechanism.

The MMN will perform the basic tasks required for RFID Middleware, which are data filtering, data aggregation, monitoring and management. Extended tasks include the management of sensors and actuators as well as the logic that links all the devices. The time slot assignment to reduce RFID data collision is another extended feature of the MMN (Figure 32).

Figure 32. MMN Middleware Features.

MMN operation is described as follows.

3.4.2.2 Operation

First, an initialization phase is executed. This phase begins by the MMN being authenticated by the MAN. The MAN provides configuration files and applications updates that the MMN will require to work. During the initialization phase, the MMN will build a list of the RFID tag IDs deployed by activating the available fixed RFID readers. In this way, Data Aggregation begins. All the information that is managed by the MMN will be updated based on events or in a regular basis building a repository accessible by using the P2P network. After the initialization phase, the MMN is ready to manage users' requests. The MMN can manage eight types of requests described as follows.

1. **User Authentication**: When a user (MRN/VRN) arrives to the network, it is authenticated in order to allow it to use the services provided. The MMN acts as an authenticator. It forwards the authentication requests to the MAN, which performs the security control. The user will also send to the MMN its profile information using a file in XML format (Figure 33). This profile will have information regarding the capabilities of the node, like interfaces available, power, data rate, etc. This file is configured on every node in an offline phase prior the use of the framework. Once the node is authenticated, it will obtain a list of available registered peers as well as the available services (for example, sensor values, interaction with actuators, etc.)

```
<profile>
    <user>
        <id>user's id</id>
    </user>
    <hardware>
        <power>device's power</power>
        <coverage>estimated coverage area</coverage>
        <interface>wireless interface</interface>
        <aditional>additional parameters</aditional>
    </hardware>
    <software>
        <app> applications required</app>
        <version>overlay version</version>
    </software>
    <services>
        <serv>services associated</serv>
    </services>
    <log>
        <last> last access to the system </last>
        <activity> user's activity </activity>
    </log>
</profile>
```

Figure 33. XML Profile information

2. **RFID Tag IDs request**: a user can request the available list of RFID tag IDs. In this case, the MAN will transmit the current list that has been previously collected. The tag list will be sent as a file that can be uncompressed or compressed to reduce the size. We can use gzip as compression utility [**66**]. The MMN will be capable of manage specific Tag IDs requests, for example specifying dates, ID numbers etc. This functionality is implemented as a Data Filtering Feature.
3. **Updated List of RFID Tag IDs**: The users can ask for an updated list of tag IDs. This request will indicate the MMN to read again the current RFID tag IDs deployed. Thus, the users can have a fresh view of the system. This request can be used for supervision, tracking and tracing applications. The updated list of tag IDs can be sent as a compressed or uncompressed file. Additionally, the MMN can be configured to update the list of tags in a regular basis or based on specific events previously defined, for example to update the list of tags if a movement sensor is triggered.
4. **Scanning Window (TDMA Anti-collision Mechanism)**: If the user node is a MRN and it requires reading the deployed RFID tag IDs, the MMN will offer the best scheduling to allow it to perform this task in concurrence with the existing readers. The MMN node will analyze the information regarding the profile of the MRN and of all connect and active readers. This is done in order to determine the technology of the rest of the readers currently connected and located in the concerned area of coverage. At this stage, several solutions may be proposed. If the entire nodes share the same technology, the MMN will either chose to rely on the anti-collision mechanisms already implemented for that kind of technology or assign a time slot. The user node will inform the MMN when it finishes reading the tags. If heterogeneous technologies are involved or if required, the MMN will also consider if there are simultaneous readers asking to read tags. If there is only one reader, the MMN will allow a scanning window and the reader will notify when it ends the reading. If parallel requests are solicited, a slot will be assigned to reduce collisions and minimize interference. The slots are assigned using first request first served, but a priority categorization can be implemented as well. These categories can be defined in the profile of each user inside the XML file obtained during the Profile registration step. The

flow diagram of the scanning window mechanism is depicted in the following figure (Figure 34)

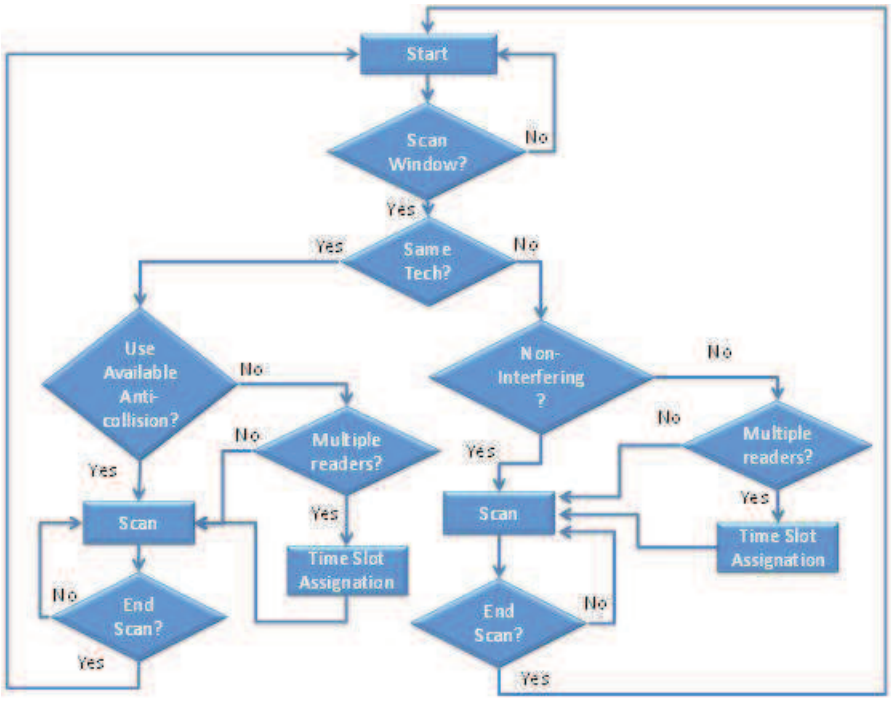


Figure 34. Scanning Window mechanism for RFID Readers.

5. **Get Sensor Value**: This request permits the user to obtain the value of a specific sensor that is available in the framework.
6. **Interact with Actuator**: This request allows users to interact with the available actuators.
7. **Resources Availability**: Since the network can provide different resources (Resource in this context is referred to peers, peer groups, services, pipes, and any relevant information) we create a general request that can embedded the query related to the resource searched.
8. **Network Disconnection**: A user can request a network disconnection. The MMN is informed so it can update the list of available peers.

The requests a MMN can manage are presented in the next figure (Figure 35).

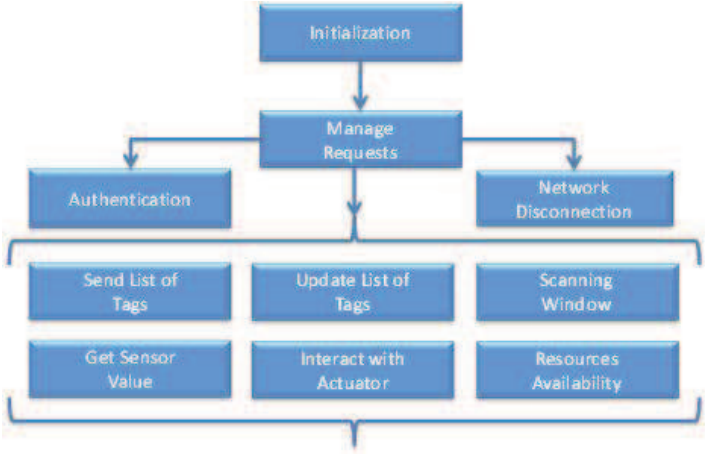


Figure 35. Multi-Mode Node Requests.

The message exchange of the MMN is presented in Figure 36.

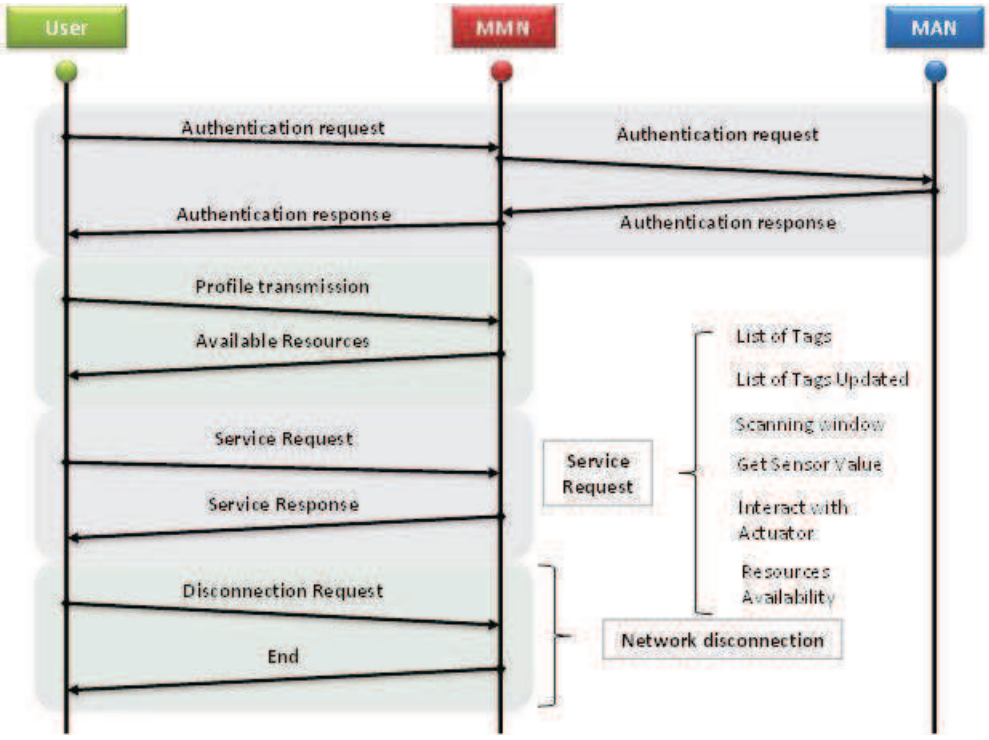


Figure 36. MMN Message exchange diagram

3.4.3 Management and Authenticator Node (MAN)

3.4.3.1 Definition

A Management and Authenticator Node is a device that will provide network authentication, and resource management to the framework. We decided to define this entity in order to reduce the load of the MMN. The MAN will be linked to an AAA server that can be located in the same node or in a distributed way [**67**]. It keeps statistics of the network as well as a list of the available users and services. MAN is based on a practical implementation that we designed and described in [**68**]. That solution was implemented in Java Standard Edition. It uses a set the JXTA P2P protocols [**69**]. Encryption and MAC (Message authentication code) are included in the message information exchange to secure the communications between the MMN and the MAN. Based on JXTA the MAN will implement ERP for routing purposes, PRP to manage queries to resources, PDP in order to advertise and publish resources, PBP to establish the message exchange mechanism and PIP in order to provide monitoring information. Its operation is described as follows.

3.4.3.2 Operation

First, an initialization phase is executed. At this step, the connection with the AAA server is established and after that, the MAN is ready to manage requests. At the same time, the MAN will run a process that will run through all its operation. This process will keep track of the resources available, users registered and log files related to the events of the framework. This information will be requested by the MMN when required. The MAN manages two types of requests:

1. **Authentication**: This request will permit MMNs and users to be authenticated and then be allowed to access to the services and devices provided. The MAN will forward the

authentication request to the AAA server that in turn will grant or deny the access to the platform.

2. **Resources Information:** The MAN will be constantly gathering information related to the resources available in the framework as well as network statistics like usage and nodes load. Every time a user is authenticated it keeps track of that event, thus it can provide at any time an updated list of nodes (MRN/VRN) connected.

In Figure 37, we observe the processes of the MAN.

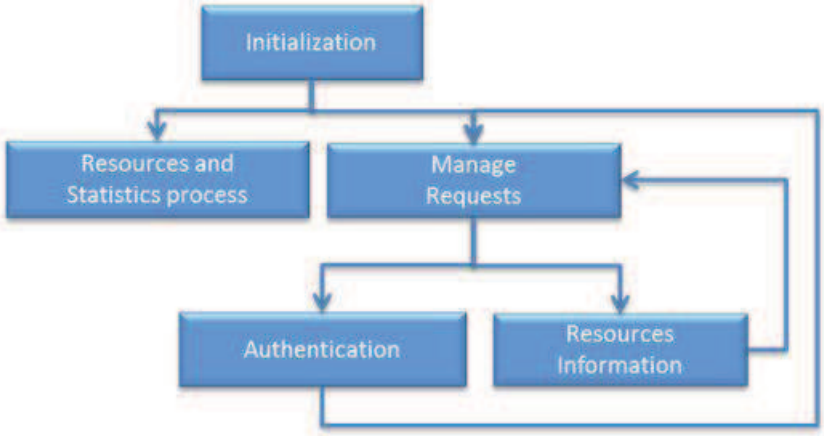


Figure 37. Management and Authenticator Node processes

In Figure 38, we present the message exchange diagram. The MMN will be authenticated first before it starts operating. Then the MAN will manage requests via the MMN in order to authenticate users as well as informing the MMN about resources availability and network statistics.

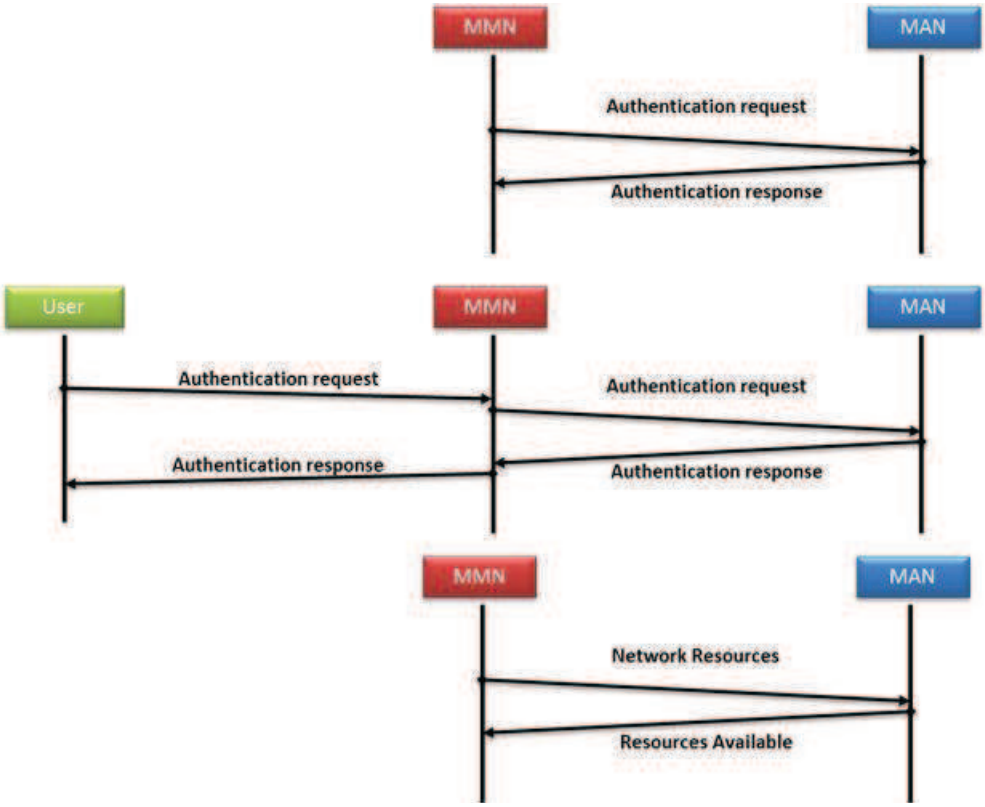


Figure 38. MAN Message exchange diagram

3.4.4 Users Nodes

3.4.4.1 Definition

User nodes are devices that will obtain the information provided by the framework for specific applications and purposes (inventory, tracking & tracing, sensor measurements, and interaction with actuators). As stated previously, there are two types of users Virtual Reader Nodes (VRN) and Mobile RFID Reader Nodes (MRN). VRN are portable devices (mobile phones, laptops, pads) with non-RFID capabilities but with one or several network interfaces. Based on our architecture, these nodes will obtain tags' IDs through a list forwarded from the MMN or other users (that in a P2P context can be also called peers) by using the P2P overlay. They are called "virtual" readers because even if they do not have available RFID hardware, through our framework they can obtain the list of the RFID tag IDs deployed. With this information, they can take advantages of the RFID system in order to use RFID based applications. MRNs are devices that have RFID reading capabilities as well as available network interfaces that allow them to register and use the platform. These devices can also benefit from the information gathered by the framework and reduce their RFID reading time and risk to interfere with other available readers in the area. Based on JXTA the users will implement ERP for routing purposes, PRP to manage queries to resources, PDP in order to advertise and publish resources and PBP to establish the message exchange mechanism.

3.4.4.2 Operation

A user works following the next processes. First, it performs a security control in order to be authenticated by the MAN as presented earlier. The MMN and the MAN are responsible for providing or denying the access to the P2P overlay. At this step, the user can also obtain the applications and settings required to operate. For example, an application that interprets the tag IDs for a specific

purpose like inventory in a warehouse, or a list of parameters to configure and provide information to the device like the list of available peers. Then the user can perform four types of requests:

1. **Get Tag IDs via user** (or also called peer in this context): Once the user is authenticated, it will obtain a list of available services and peers. The user can obtain the list of tag IDs available from the connected peers thus reducing the load at the MMN side.
2. **Get Tag IDs via MMN**: If the user requires the list of tag IDs directly from the MMN, it will use this request. Additionally, this request permits to obtain an Updated list of tag IDs as it is previously described in MMN section.
3. **Get Sensor Value**: Users can obtain variables measured by sensors connected into the platform. This information can be used for specific purposes that depend on the application implemented, for example, to trigger certain action when a temperature reaches certain threshold.
4. **Interact with actuator**: The users can interact with installed actuators in order to perform actions related to an application. For example to activate a motor that opens a door if certain tag ID is found in the list of tags.

The user will be able to run a Share process that will deal with other users or peers requesting for information. Finally, the user can disconnect from the network. The MRNs follows the same principle that VRNs, however depending on the interference they may cause, the MMN will assign them a time slot to work in, thus avoiding reader-to-reader collision, using a process called Scanning Windows and that was defined previously. The user node processes are shown in Figure 39.

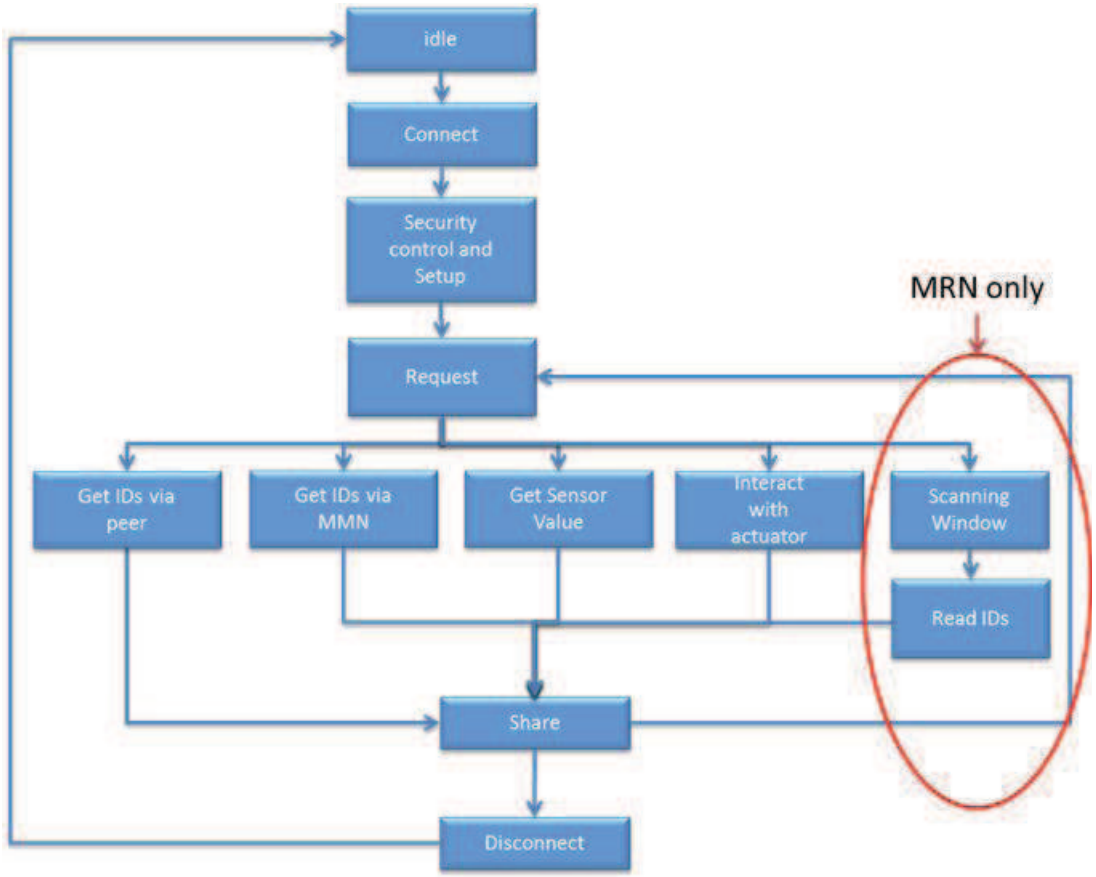


Figure 39. User (VRN and MRN) processes.

The message exchange regarding the User Requests is shown as follows (Figure 40).

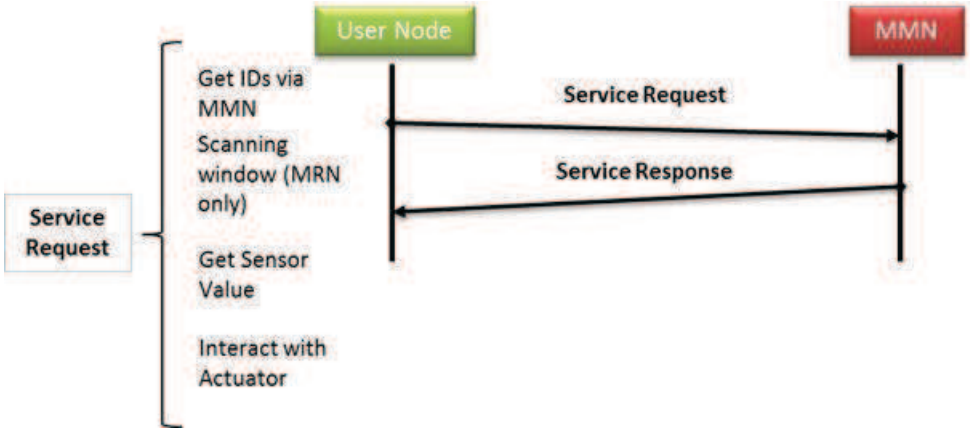


Figure 40. User Services Request

The Sharing process is presented in Figure 41.

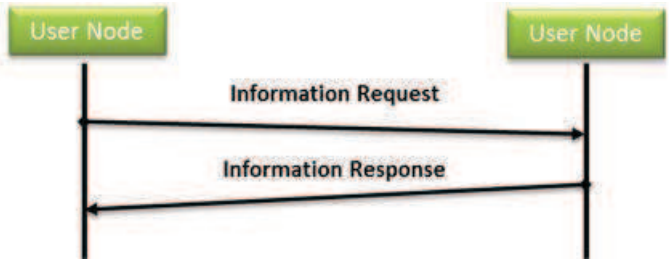


Figure 41. Sharing Information between nodes

Finally, nodes can proceed to disconnect from the overlay, informing the MMN of this event (Figure 42).

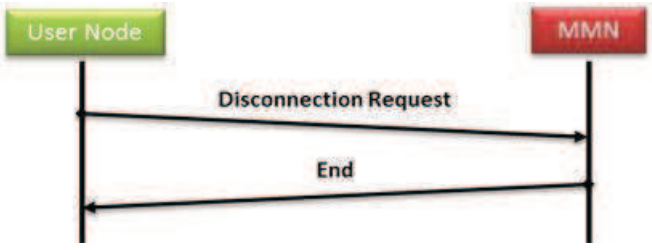


Figure 42 Network Disconnection Request from users to MMN.

3.5 Framework delay analysis

We used the NS2 simulator [**70**] in order to measure the transmission delay of RFID tag IDs between the MMN and users (VRNs or MRNs).

3.5.1 Simulation Setup

In a hybrid P2P overlay we can simulate the information transmission between a MMN, seen as a central node, and user nodes (MRN/VRN) seen as clients, by using a client-server approach that simplifies the P2P system. We set up a NS2 simulation based on 802.11g in order to evaluate the transmission delay of the MMNs towards the user nodes. The MMN was represented as a FTP service on top of TCP/IP and then linked to an 802.11g network interface. We used the two-ray ground reflection model with an operation frequency of 2.4 GHz on Direct Sequence Spread Spectrum (DSSS). The user node (MRN/VRN) was represented as an 802.11g interface linked to a TCP sink (Figure 43).The simulation environment is a 30x30 meters flat grid and we defined three scenarios:

1. A MMN in the middle of the grid (X=15m, Y=15m) and a single fixed user node located at Cartesian coordinate (0m, 0m).
2. A MMN in the middle of the grid (X=15m, Y=15m) and a single moving user node (MRN/VRN) performing random movements at a speed of 1 meter per second (1 m/s) inside the grid.
3. A MMN in the middle of the grid and 20 static user nodes (MRN/VRN) at random positions inside the grid.

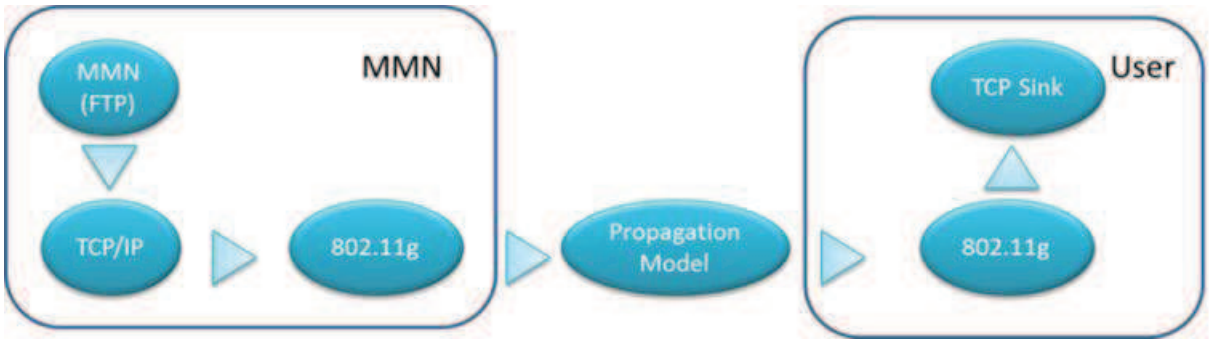


Figure 43. NS2 System Representation.

3.5.2 Transmission Delay

In order to measure the transmission delay we created a list of tag IDs to be sent from the MMN to the user. We obtained the file size associated to a specific number of tags. We computed random files (by using a bash shell script) with unique tags' IDs with 96 bits extension (Table 2) as the EPC standard stands **[**71**]**. These files were sent in a compressed (gzip) and uncompressed ways (raw file). These files were transmitted from a MMN to a user node and the data rate of the channel varied in these steps: 1, 6, 18, 36, and 54 Mbps.

Table 2. File-size representing tags' IDs

Number of Tags	Non-compressed file size (bytes)	Mean compressed file size (bytes)	Sample Variance Compressed files (bytes)	Confidence Interval 95%
1	192	71.633	2.68244	71.63 +/- 0.16
10	1920	274.444	3.88277	274.44 +/- 0.24
100	19200	2105.14	10.9783	2105.14 +/- 0.68
1000	192000	20335.9	33.7461	20335.9 +/- 2.09
2000	384000	40577.3	44.9006	40577.3 +/- 2.78
3000	576000	60850.1	49.6987	60850.1 +/- 3.08
4000	768000	81094.1	55.6748	81094.1 +/- 3.45
5000	960000	101333	58.3231	101333 +/- 3.61
6000	1152000	121610	59.6907	121610 +/- 3.69
7000	1344000	141850	59.442	141850 +/- 3.68
8000	1536000	162092	61.6346	162092 +/- 3.82
9000	1728000	182365	60.572	182365 +/- 3.75
10000	19200000	202606	63.5281	202606 +/- 3.93

The results for the first simulation scenario (A MMN in the middle of the grid and a single fixed user node at (X=0, Y=0)) are presented as follows. First, the uncompressed (raw) file is sent and the transmission delay measured (Figure 44). We observe that up to 100 tags the behavior is similar for the different data rates. After that value, the delay varies depending on the data rate used. For example, using a 1Mbps data rate transmitting 10K tag IDs, will lead to a delay of 21,186 seconds.

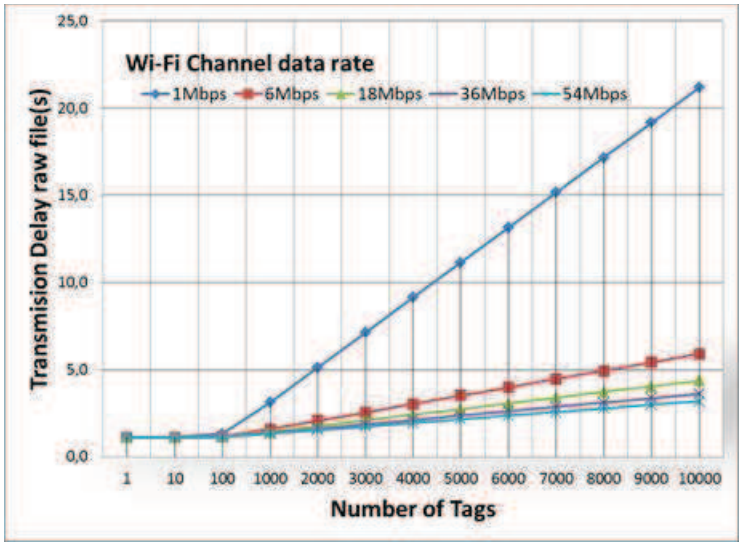


Figure 44. Transmission delay of raw list of tag IDs from MMN to fixed user node.

For compressed files, we obtain the delay that is presented in Figure 45. For the highest data rate (54 Mbps), the transmission delay for a file representing 10k tags (202,606 Kbytes) is about 1.33 seconds contrasted with 3.24 seconds for 1 Mbps data rate. From 0 to 100 tags the delay remains almost constant for all data rates (about 1.12 to 1. 32 seconds) and then increases linearly.

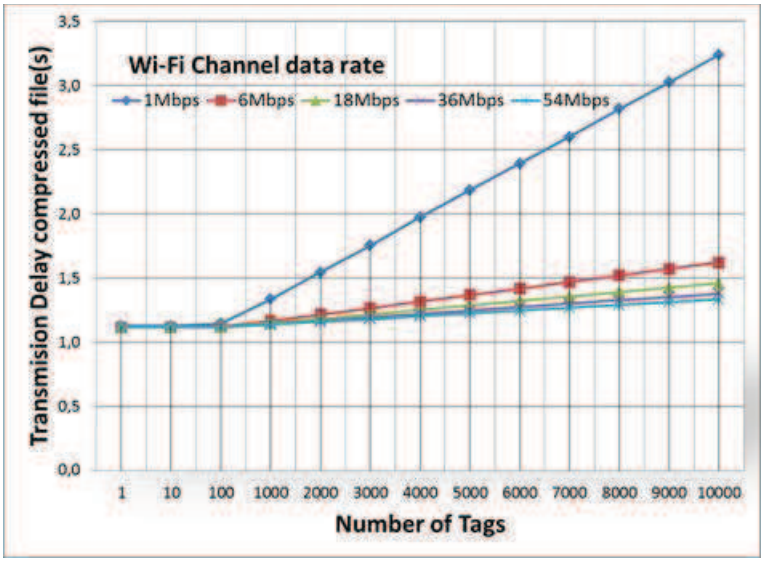


Figure 45. Transmission delay of compressed list of tag IDs from MMN to fixed user node.

In Table 3 we present the improvement that we obtain by using compressed files in contrast with the transmission of raw ones from a MMN to a fixed user node. The table presents the percentage ratio of the delay incurred. For example, if we transmit a file with 10k tag IDs at 54 MBps, we obtain about 58% less time required than if we just send the raw file.

Table 3. Compressed vs. Raw files delay improvement fixed node.

Tags	Compressed/Raw 1Mbps (%)	Compressed/Raw 54Mbps (%)
1	0,00	0,00
10	0,90	0,08
100	13,51	1,64
1000	57,29	14,04
2000	69,91	24,57
3000	75,45	32,24
4000	78,41	38,31
5000	80,39	43,18
6000	81,81	47,38
7000	82,86	50,73
8000	83,58	53,60
9000	84,21	56,04
10000	84,70	58,19

Considering the second simulation scenario (A MMN in the middle of the grid and a single mobile user node performing random movements at 1m/s) the plots are presented in Figure 46 and Figure 47. First, the raw files are sent. We observe that the delay increases slightly compared to the fixed node due to the node's movements that imply power variations due to the variation of the distance from the MMN. For a 54 Mbps data rate and 10 k tags, the delay is approximately 3.64 seconds. For 1 Mbps we obtained 21.56 seconds.

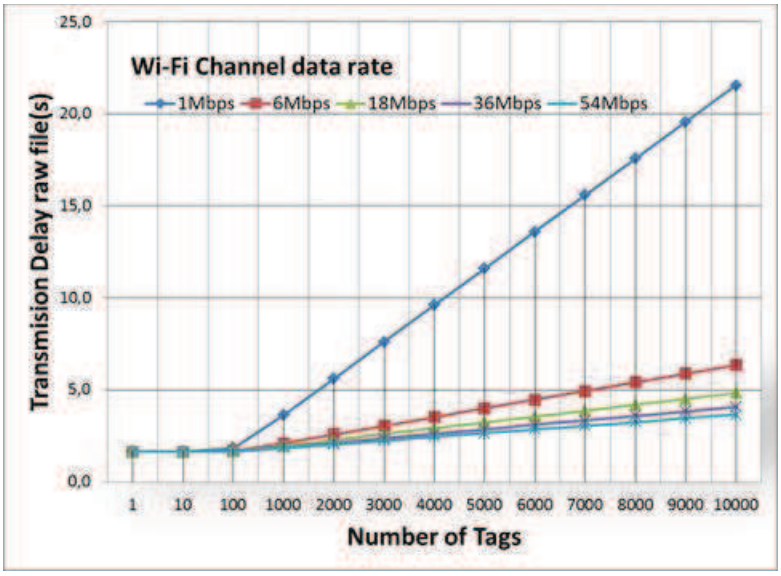


Figure 46. Transmission delay of raw list of tag IDs from MMN to mobile user node

In Figure 47, the delay obtained by transmitting the compressed files is shown. For a 54 Mbps data rate and 10 k tags, the delay is approximately 1.84 seconds. For 1 Mbps we obtained 3.74 seconds.

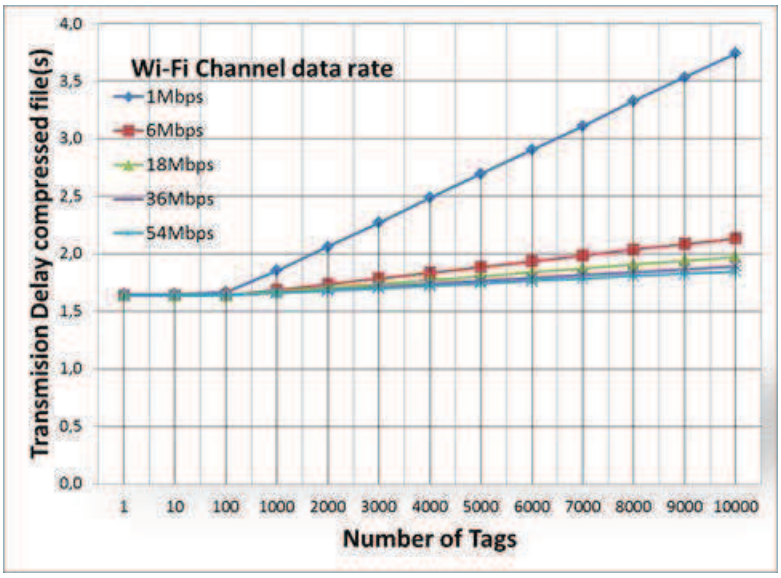


Figure 47. Transmission delay of compressed list of tag IDs from MMN to mobile user node.

In Table 4, we present the improvement that we obtain by using compressed files in contrast with the transmission of raw ones for a mobile node. For a file with 10k tag IDs at 54MBps, we obtain about 49% less time required than if we just send the raw file. For 1MBps, the improvement of the delay goes up to 82 % for 10k tags IDs transmitted.

Table 4. Compressed vs. Raw files delay improvement mobile node

Tags	Compressed/Raw 1Mbps (%)	Compressed/Raw 54Mbps (%)
1	0,00	0,00
10	0,63	0,07
100	9,54	1,06
1000	48,86	9,75
2000	63,30	17,65
3000	70,18	24,16
4000	74,09	29,51
5000	76,76	34,09
6000	78,64	38,05
7000	80,03	41,40
8000	81,06	44,35
9000	81,93	46,99
10000	82,64	49,34

We observed a substantial improvement in the transmission delay by using file compression (i.e. for 1Mbps and sending a file from a MMN to a mobile node we obtain about 82% less delay). Practical implementations of the platform must consider this feature.

For the third simulation scenario, 20 fixed user nodes were considered and we transmitted in parallel compressed files representing tag IDs. In Figure 48, we present the delay experienced. We noticed that from "0" to "100" tags, the transmission delay remains almost constant for all the different data rates evaluated (1.9 seconds). For a 10k tag list, we obtain a delay of 9.27 seconds for 54Mbps and 45.37 seconds at 1Mbps data rate. As expected, with higher data rates, the delay is reduced.

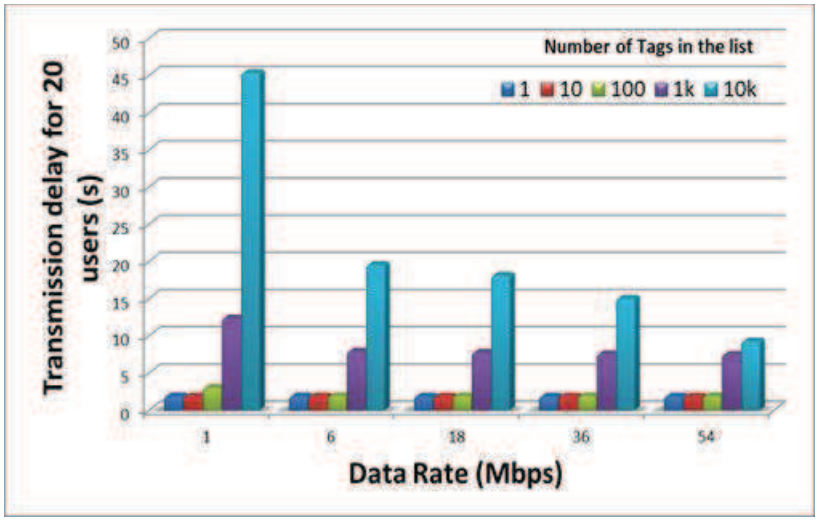


Figure 48. Compressed File transmission delay of Tag lists from a MMN to 20 fixed users

3.5.3 User Service Delay

With the previous results, now we can proceed to calculate the total service delay. However, before that we require the time a reader takes to read certain number of tags. Based on the results obtained in [**71**], [**72**], and [**36**] we considered the EPC Gen2 protocol to calculate the TIS (Tag Identification Speed) that refers to the time a reader takes in order to read the tag IDs of a number of tags. Assuming a data rate of 62.5 Kbytes between tags and readers, we obtain a constant rate of approximately 280 tags per second. In Figure 49 we observe the Tag Identification Speed required to read N tags where N ∈[0 10k].

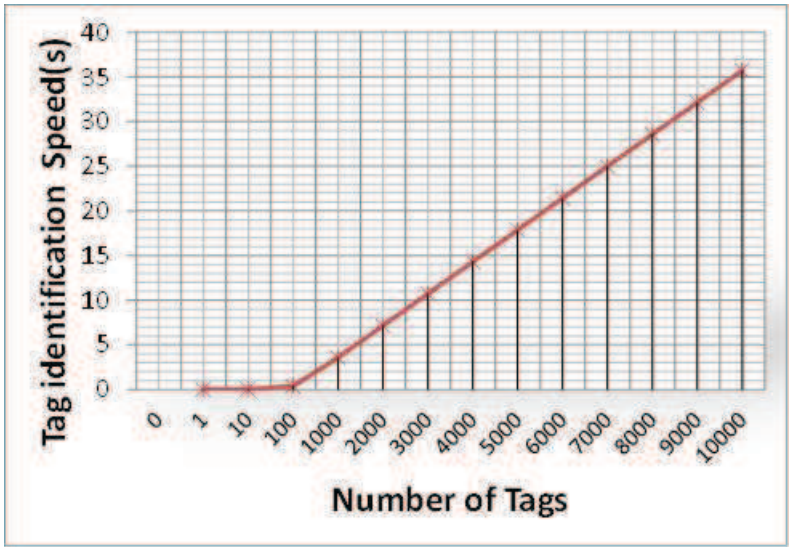


Figure 49. One Reader Tag identification Speed (EPC Gen2).

Now, we proceed to calculate the User Service Delay (USD). The first time a user is served, it experiences a service delay based on the following parameters (equation 1)

$$\text{User Service Delay (USD)} = T_{Auth} + T_{Setup} + T_{Req} + T_{readDelay} + T_{Tx} \quad (1)$$

Where:

- T_{Auth} = Delay of node's authorization process.
- T_{Setup} = Time to download required applications and configuration.
- T_{Req} = Time to send a request to the MMN.
- $T_{readDelay}$ = MMN tags' reading delay.
- T_{Tx} = Tags' IDs transmission delay.

For the subsequent requests (assuming no re-authentication), the Service Delay is reduced to equation (2).

$$USD = T_{Req} + T_{readDelay} + T_{Tx} \quad (2)$$

If the MMN had performed pre-reading of the deployed tags before a user performs a request, the USD will be defined by equation 3.

$$USD = T_{Req} + T_{Tx} \quad (3)$$

To estimate a rough value for USD we can assume:

- $T_{Setup} = T_{Tx} \approx 2$ seconds that represents the time to download a file of 200 k bytes at 54 MBps for a fixed node. This value represents the size of a settings file or an application needed for the users obtained at the setup step. This value will increase depending on the size of the application to be downloaded.
- T_{Req} and T_{Auth} are in the milliseconds order (60 Milliseconds), as obtained in our platform measurements in [**68**] thus we can neglect them (not significant compared with seconds magnitude order).
- $T_{readDelay}$ = 35 s (10K tags using Gen2 protocol as presented in Figure 49).

Thus, USD = 37 s for a single fixed node served at 54Mbps data rate obtaining a list of 10k tags. For 20 nodes, we can obtain USD = 35 s + 20 x 2 s = 75 s at 54 Mbps data rate obtaining a list of 10k tags. If we assume that, the MMN had already read the tags and has the list prepared by the time the request was made then: USD ≈ 2 s.

We conclude the simulation sections by stating that the tag list sent to the user will take the same time when the number of tags is less than a certain threshold (100 in our simulation). In addition, we note that the compression of the list of tags to be sent reduces considerably the transmission delay. We obtained an improvement when transmitting 10k tag IDs at 1 Mbps and fixed nodes of about 84% and 82% for mobile nodes. Additionally, if the MMN performs pre-reading of the deployed tags, the User Service Delay is reduced. This feature will be useful even for mobile readers. They can rely on the P2P overlay to obtain the tag IDs via the MMN or the peers in less time than it will take if they perform the reading independently.

3.6 Applications example

3.6.1 Museum

An application of our proposed framework is in a museum. Normally, visitors may use audio guides, which require typing an identification number in order to obtain the information associated to the piece of art they are interested. Alternatively, with our framework users can use their Wi-Fi enabled mobile devices to download the Museum's guide application through the P2P overlay (Figure 50). This application will automatically keep track of the works in exhibition by using RFID passive tags. The application can provide a plan of the museum indicating the location and information associated to the exhibition. Additionally MMNs can be activated to perform periodically surveillance tasks as well. Finally, the museum can publish exhibition's websites and keep an inventory database thanks to the RFID deployment.

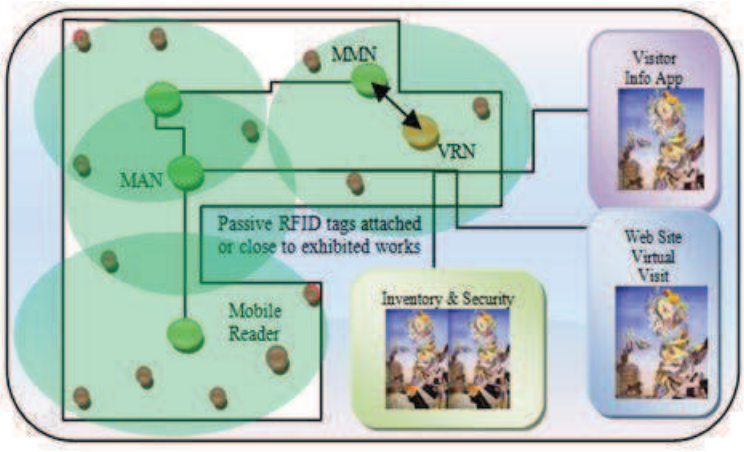


Figure 50. Museum Application

3.6.2 Library

The framework can be used to implement an automatic tracking and information system on a library. The system consists on smart shelves linked together by MMNs that will provide proxy functionality. Processes like inventory, book borrowing and security features can be also implemented. In the next figure (Figure 51), we present the block diagram of the solution.

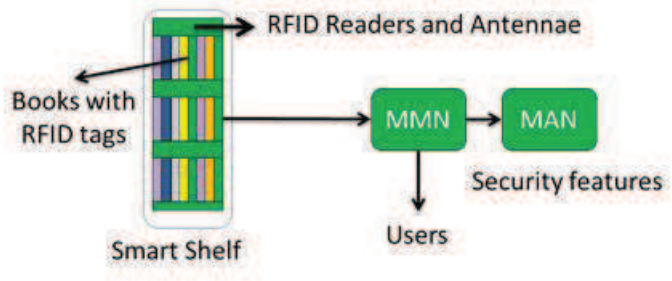


Figure 51. Smart Shelves in Library Application.

3.7 Chapter Summary

In this chapter, we propose a framework that enables RFID and non-RFID devices to benefit from a heterogeneous support by our designed framework to get the RFID information and access to related services. In this way, we can implement RFID-based services that can be extended to different users. These users can be nodes with non-RFID capabilities but network interfaces instead (WLAN, ZigBee) and RFID mobile readers. For mobile readers, we presented a scheduling mechanism based on TDMA. It targets to reduce the interference caused by multiple readers working simultaneously (reading RFID tags). We also consider into the architecture some security features like network authentication. A modular design was depicted defining its characteristics and message exchange protocols. The performance analysis shows the advantage of performing proactive RFID tag reading and sharing the list of tag IDs via the P2P overlay. This feature reduces load in the MMNs as authenticated users can share the information they obtain. Additionally, even mobile RFID reader benefit from this solution because they can obtain the updated lists of tags deployed directly from MMNs. However, if they want to read the tags by themselves, the platform provides a scheduling mechanism to reduce interference. Further work may include the implementation of the framework by using a test bed.

RFID Topology design

Chapter 4

RFID Topology design based on Genetic Algorithms

In this chapter, we present the work related to the RFID topology design. First, we give an overview of the optimization process describing its basic elements. We will focus on the Genetic Algorithm technique that is the method we used for the implementation of the RFID topology tool. We will depict the representation used to evaluate candidate solutions for the topology problem. Then, the rest of the chapter will describe the software implementation and modeling used to deal with the multi-RFID reader deployments.

4.1 Research Problem

As stated in previous chapters, RFID Technology is used for different applications especially for tracking and tracing items. Certain applications based on RFID such as warehouse items tracking, interactive museums, and tracing products in supermarkets require a network of RFID readers in order to operate. One of the issues as we previously discussed, is that the interference caused by multiple readers operating at the same time can provoke interference. This will be translated into data collisions thus the tag IDs will not be read or detected. A way to solve this problem is to deploy readers with their location and power level set to optimized values. This is achieved by finding an optimized topological design of the network [**73**]. This process is based on maximizing or minimizing different objectives such as coverage area, interference reduction, number of tags detected and low deployment cost. In order to optimally satisfying all these - often-opposed - objectives, an optimization non-linear problem with a large set of possible solutions need to be solved [**74**].

4.2 Research objectives

We focused on the topological design of an RFID network in order to obtain the optimal number, location and power level of the RFID readers that are to be deployed (Figure 52). Due to the nature of this problem, we observed that it is characterized by having a search space with a huge number of possible candidate solutions and that several parameters must be tuned at the same time ([**74**], [**75**]). Intuitively, and based on related work (see related work section) this problem suits for the application of a well-known search technique, the Genetic Algorithms (GA). GA is applied when dealing with huge solutions search spaces and multiple objectives are to be satisfied simultaneously. GA are used to obtain optimal solutions for a broad number of problems and are inspired in biological observations linked to the evolution of the organisms. Consequently, we propose a RFID network topology design based on GA to obtain RFID reader location and power levels relying on a multi-objective cost or fitness function that evaluates and rates the solutions provided. In this context, the multi-objective fitness function can be also called evaluation function, or simply fitness function.

4.3 Related work

A variety of RFID network issues have been proposed to be solved by using evolutionary techniques including GA. In [**76**] a new technique for RFID resources allocation based on GA was proposed to deal with the reader-to-reader interference problem. In [**77**], [**78**] and [**79**] GA and binary particle swarm optimization are used to solve multi RFID networks scheduling problem to optimize channels allocation in the system.

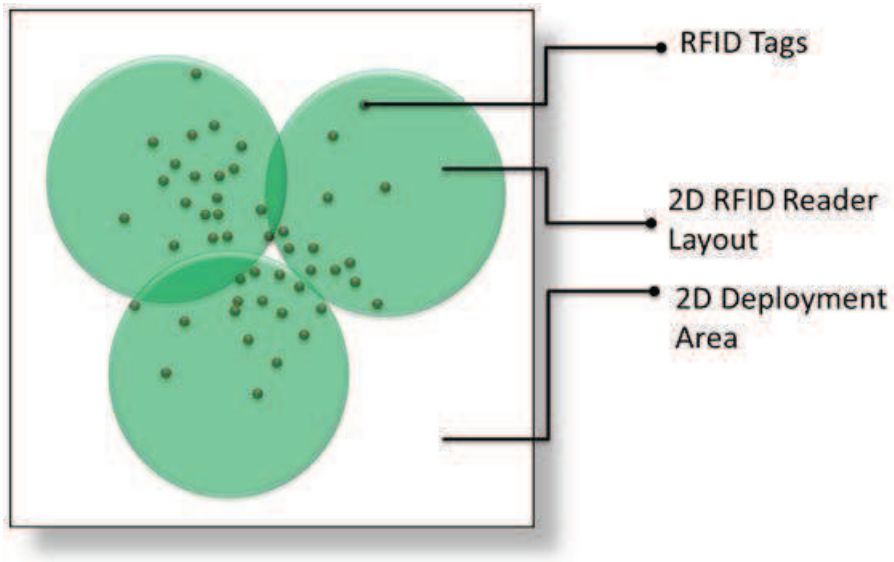


Figure 52. Multi-RFID reader deployment.

More related to our work are [**74**] and [**75**] where GA are used to propose RFID planning methods. The first proposal limits the area to a 10 x 10 m area, dividing it in equal spaced grids and locating the readers only inside those grids, no propagation models are indicated and the interference problem is not clearly reflected in the solution. The second approach considers the coverage and the interference as variables but do not guarantee that all tags will be detected. In our approach, the candidate solutions to the required topology design are evaluated by using a linear weighted multi-objective function. It considers six parameters: overlapping of the RFID reading area, number of useless readers deployed, number of tags covered, the number of readers deployed out of the area to be covered, number of redundant readers deployed and number of tags located inside overlapped reading areas. The readers can be placed at any position inside a rectangular area with a maximum size of 31.9x31.9 meters. We used two propagation models to calculate the coverage area of the RFID readers. Additionally, we developed a flexible software tool to assist in the topology design process that provides customizable GA parameters, functions to import and export scenarios and parameters settings, a scenario editor and visualization of the solving process and solutions obtained.

4.4 Basic Concepts

We present the basic concepts related to optimization and especially about Genetic Algorithms because it is the selected optimization method to be used in this work.

4.4.1 Optimization

Optimization is the process of obtaining the best solutions for a determined problem that fulfill certain design objectives [**80**] , [**81**]. Optimization techniques are procedures that permit finding those solutions without the need of testing all the possible combinations in the whole search space (called exhaustive search). It is noticeable that sometimes depending on the problem treated; suboptimal solutions can be accepted as valid ones, it means that even if they are not the global solution but a local one that satisfies the conditions imposed. Optimization can be classified depending on the nature of the function to be treated. It includes Linear, Nonlinear, Quadratic, and Convex. In relation to the search space and variables, the classification can be set as discrete (variables have discrete values), continuous (real values are assumed) and mixed (both real and discrete). Depending on the type of search algorithm used, optimization techniques can also be categorized as Deterministic and Stochastic. Deterministic algorithms permit to reproduce the same results following the steps taken

into account in order to obtain a solution. On the other hand, stochastic algorithms cannot reproduce the same results even with the same initial conditions applied because the process of obtaining solutions is associated with random variables. Some of the difficulties faced when optimizing problems is variables independency and when there is no formal definition of the function that will evaluate the solution. The following generic steps may be applied in order to propose an optimization process for a specific problem [**80**]:

1. Defining the problem boundaries: It is important to delimit the whole set of possible solutions or search space related to the proposed problem. This permits to target the efforts in finding solutions that will be applicable to the problem.
2. Performance measurement: This phase provides an indicator of how well a solution performs by solving the problem and establishing comparative metrics.
3. Variable Independence: This step allows to identify the relations between the different variables and to observe how they can be affected by parameters variations.
4. Modeling the problem: The interaction of the selected variables is translated into a mathematical model that represents the real problem considered. The model allows candidate solutions testing.

4.4.2 Genetic Algorithms (GA)

GA are a stochastic search technique for optimization problems that were proposed in the 60's by John Holland [**82**]. They were initially applied to study the natural adaptation of animal species and to represent it into computer models. They try to emulate as an abstraction what happens in nature regarding how organisms reproduce and survive based on the theories of evolution. Some of the GA terminology is taken from biology concepts. GA process populations constituted of several *individuals*. An individual is conformed of one or more *chromosomes.* The chromosome is the set of *genes* that encode particular properties of the individual. The genes can have different possible values called *alleles*. The genes are located in a particular *locus* or position inside the chromosome. In GA, the chromosome refers to a candidate solution to a problem. Generally, the chromosome is represented as a binary string that encodes it. The genes are bits which alleles can be either one or zero. However, there are different representations for chromosomes that include real numbers, alphabet characters, etc.

GA mimic what nature does regarding organisms' evolution. The main operators are Selection, Crossover and Mutation. Selection obtains the fittest individuals (best performing solution) in order to make them exchange their genetic information to produce a new offspring. Crossover performs the combination of the selected individuals to produce new solutions. Finally, Mutation flips alleles to add variety and reduce local optima convergence. GA search through a huge number of possible solutions where every chromosome represents one point in the space search. The solutions are appraised based on a fitness function that provides a measurement that indicates how well the individual solves the current problem. A simple GA workflow is as follows:

1. A random population is generated.
2. The population is evaluated by using a fitness function.
3. Through Selection, Crossover and Mutation we create a new population, called next generation, that is a next iteration.
4. Repeat from step 2 until a solution is found or other limitations are reached (execution time, memory capacity, number of generations, etc.).

Other search methods like Hill Climbing, Simulated Annealing and Tabu search as well as GA share the same principle: generate a set of candidate solutions, evaluate them according to a fitness function, discard or keep solutions based on fitness and produce variants of the solutions. What differentiates GA from the rest is that they use stochastic selection, crossover and mutation at the same time [**82**].

There is no rigorous answer to the question: why and when to use GA over other search methods [**83**]. Intuitively the applicability of GA relies if the search space is "large" and the problem does not require a global optimum to be obtained. It means that a good solution obtained is enough to satisfy the problem. The success of GA also depends on the way the problem is coded (problem coding), the genetic operators (selection, crossover and mutation) and the fitness function that evaluates the candidate solutions.

4.5 RFID Network Topology Modeling

4.5.1 Overview

As we stated previously we aim to obtain the optimal position and power level of the deployed RFID readers in relation with the current RFID tag scenario by following a certain criteria. In order to obtain these optimized values, we define a multi-objective fitness function that considers different factors that impact in the RFID network topology design [**74**], [**75**]. We assume a two-dimension squared area as our test area, assuming the RFID reader coverage as a circular area with the reader disposed at the center. We assume RFID passive tags. In the following section, the multi-objective fitness function used to evaluate the candidate solutions is described.

4.5.2 Multi-objective fitness function

In this work, we chose a linear weighted fitness function (f_T) to evaluate each possible solution (Equation 4). The linear representation allows us to calculate in a simple way the separate objectives of this function and to add them up in a weighted way. The weights serve as indicators of how important an objective is in relation to the others in the total calculation of the fitness value. The values of the weights are located in the interval [0, 1]. If the value of an individual weight is set to 0 the objective related to that weight is not considered in the calculation. The multi-objective fitness function provides a measureable indication on how well each evaluated individual performs at finding a RFID reader deployment satisfying the design objectives. This function is composed of the following six objectives: minimize the overlapping of the RFID reader area, minimize the number of useless readers, maximize the number of tags covered, minimize the number of readers located out of the deployment area, minimize the number of redundant readers and minimize the number of tags located in overlapped reading areas. Each individual objective is represented as (f_i), and the weights as (w_i).There will be six weights (one for each objective) and the sum of all of them should be 1.The maximum value that (f_T) can get is "1". The value of "1" represents a fully satisfaction of all the six design objectives of the multi-objective fitness function, in other words it means that we obtain the solution we expect.

$$f_T = \sum_{i=1}^{6} w_i * f_i \ , \qquad where: \sum_{i=1}^{6} w_i = 1 \tag{4}$$

4.5.2.1 First Objective: Minimize the Overlapping of the RFID reading area.

The tags located inside the reading area will be normally detected but if any other reader interferes, there might be collisions (Figure 53). This first objective is satisfied if the total overlapping area is below an acceptance threshold. Equations 5 to 8 show the definition of our first objective function (f_1).

$$f_1 = \frac{1}{1+\varepsilon^2} \tag{5}$$

$$\varepsilon = Target - Total\ OverlappingArea \tag{6}$$

$$Target = \sum_{all\ readers} CArea * CRatio \tag{7}$$

$$Total\ Overlapping\ Area = \sum_{all\ readers} Overlapping\ Area \tag{8}$$

The Coverage ratio (CRatio) specifies the amount of overlapping allowed on each reader. CArea indicates the total coverage area that a particular RFID reader deployment presents. The product (CArea X CRatio) is the reference threshold of overlapping allowed, thus indicating the target to be reached. Each reader is evaluated in relation with the rest of the deployed ones and the total overlapping area is calculated. The error is the difference of the target value and the total overlapping area. Then f_1 (Equation 5) will have a value close to 1 if the error tends to zero, that is what we expect in order to satisfy this objective. Experimentally, we observed that values from CRatio in the interval from 0.10 to 0.25 require less iteration in order to obtain suitable solutions. If CRatio is equals to zero, the GA will take more time to obtain a total no-overlapping deployment layout. It implies that the number of readers required will increase, as we need to pack inside a square perimeter enough readers to cover the area [**84**].

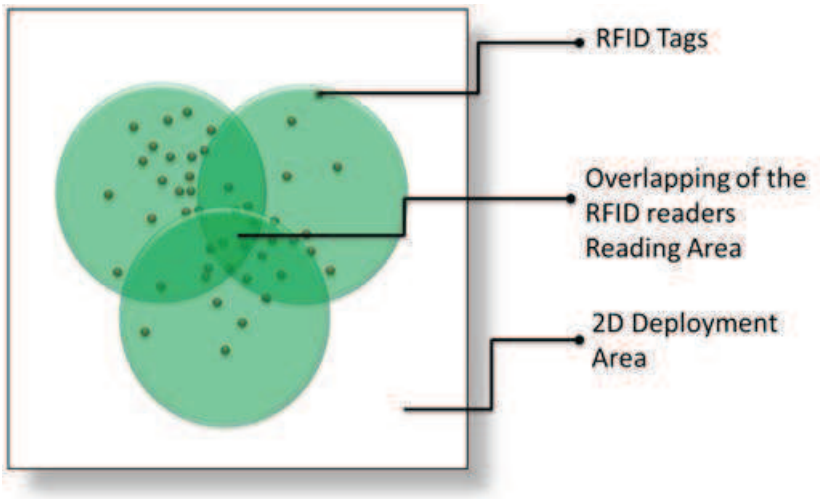


Figure 53. Overlapping of the reading area

In order to calculate the overlapping area we observe the three possible cases. To determine if there is overlapping or not we calculate the center-to-center distance of the two circles. In Figure 54, two circles are intersected and Equation 9 and 10 give their circles equations.

Equation of the Circle 1

$$x^2 + y^2 = r_1^2 \tag{9}$$

Equation of Circle 2

$$(x-d)^2 + y^2 = r_2^2 \tag{10}$$

The Center-to-Center distance is obtained by using Equation 11:

$$d = \sqrt{(x_1 - x_{2)}{}^2 + (y_1 - y_{2)}{}^2} \quad (11)$$

Then there are three cases, first if ($d > r_1 + r_2$) then there is no intersection. Thus

$$AIntersection = 0 \quad (12)$$

Second, if ($d < r_1 - r_2$) we have total overlap and three possibilities

- if ($r_1 < r_2$) then

$$AIntersection = r_1^2\pi \quad (13)$$

- if ($r_1 > r_2$) then

$$AIntersection = r_2^2\pi \quad (14)$$

- if ($d == 0$) then

$$AIntersection = r_2^2\pi = r_1^2\pi \quad (15)$$

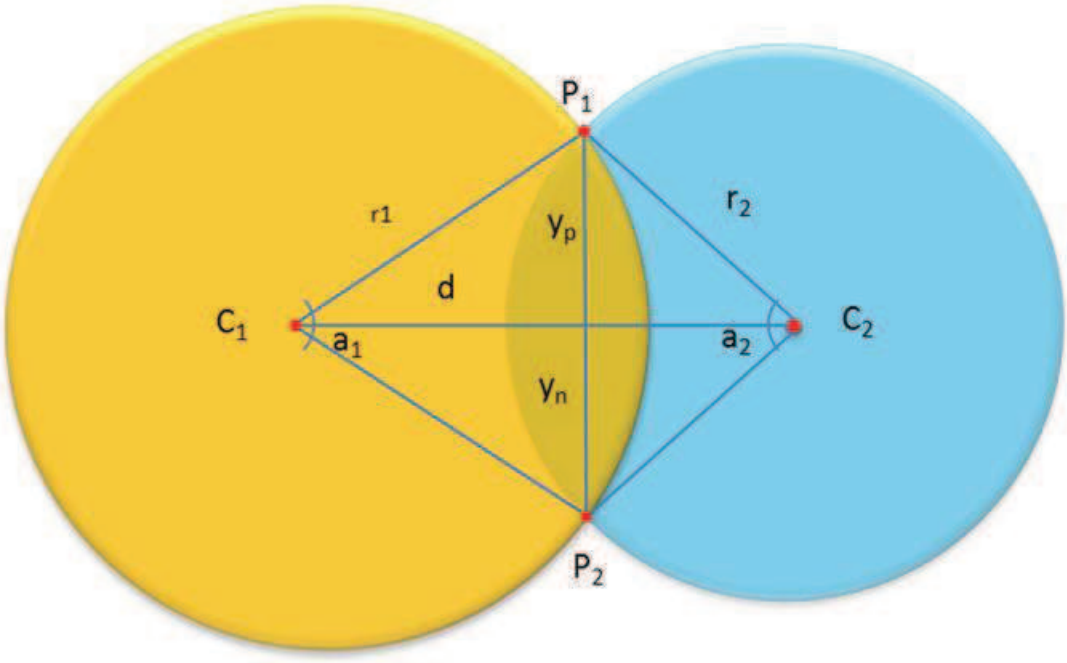


Figure 54. Overlapping Area of two circles.

For the last case if ($d < r_1 + r_2$) there is overlap and it is calculated as follows.

We calculate Y_p and Y_n using Equations 16 and 17.

$$Y_p = \sqrt{r_1^2 - x^2} \tag{16}$$

$$Y_n = -\sqrt{r_1^2 - x^2} \tag{17}$$

Where:

$$r_1^2 - x^2 = \frac{2r_1^2d^2 + 2r_2^2d^2 + 2r_1^2r_2^2 - r_1^4 - r_2^4 - d^4}{4d^2} \tag{18}$$

Now we can find the angles a_1 and a_2

$$a_1 = a_2 = 2\arcsin\left(\frac{lenght/2}{r}\right) \tag{19}$$

Where:

$$lenght = Y_p - Y_n \tag{20}$$

The intersected area is the area of the sector of circle minus the area of the triangle inside.

The area of the segment belonging to the first circle is given by:

$$Aseg1 = \frac{a_1\, r_1^2}{2} \tag{21}$$

Equation 22 provides the area of the triangle corresponding to this part:

$$Atriangle1 = x * Y_p \tag{22}$$

Thus, the first part of the intersection (right intersection side on Figure 54) is given by:

$$Asector1 = Aseg1 - Atriangle1 \tag{23}$$

Then the other three areas are respectively:

$$Aseg2 = \frac{a_2\, r_2^2}{2} \tag{24}$$

$$Atriangle2 = (d - x) * Y_p \tag{25}$$

$$Asector2 = Aseg2 - Atriangle2 \tag{26}$$

Finally, the intersection area is obtained by using the following equation

$$AIntersection = Asector1 - Asector2$$

The term: *"Overlapping Area"* in equation (8) is calculated by the sum of all the "*AIntersection"* calculations from each reader.

In order to calculate the coverage area of a RFID reader we used the following propagation models:

a) Outdoor

We used the Friis equation [**85**] in order to estimate the coverage area of each reader (Equation 27).

$$\Pr = Pt\frac{GtGr\lambda^2}{(4\pi r)^2} \qquad (27)$$

Where:

Pr: Received power, Pt: Transmitted power, Gr: Receiving antenna gain

Gt: Transmitting antenna gain, λ: wavelength of the frequency used

r: reading range

The reading range "r" is used in the first objective function in order to determine the radius of coverage of the RFID reader.

b) Indoor

In case of indoor scenarios, we use the ITU indoor propagation model [**86**]. The ITU indoor path loss model is expressed in equation 28.

$$L = 20\log f + N\log d + Pf(n) - 28 \qquad (28)$$

Where,

L = the total path loss (dB).

f = Frequency of transmission (MHz).

d = Distance (m).

N = the distance power loss coefficient.

n = Number of floors between the transmitter and receiver.

$P_f(n)$ = the floor loss penetration factor.

28 = random value, reflecting the attenuation (in decibel) caused by flat fading.

The distance-power loss coefficient, N expresses the loss of signal power due to the distance. This value is empirical. We can observe some values in Table 5.

Table 5. Empirical coefficient (addimentional) N (distance power loss)

Frequency Band	Residential Area	Office Area	Commercial Area
900 MHz	N/A	33	20
1.2 GHz	N/A	32	22
1.3 GHz	N/A	32	22

The floor-penetration loss factor is an empirical constant dependent on the number of floors the waves need to penetrate. Some typical values are shown in Table 6.

Table 6. Floor penetration factor

Frequency Band	Number of Floors	Residential Area	Office Area	Commercial Area
900 MHz	1	N/A	9	N/A
900 MHz	2	N/A	19	N/A
900 MHz	3	N/A	24	N/A
1.8 GHz	n	4n	15+4(n-1)	6 + 3(n-1)

4.5.2.2 Second Objective: Minimize the Number of useless readers.

Even if a layout complies with the overlapping parameter, it is possible to obtain readers that cover no tags and they need to be discarded from the optimal solutions (Figure 55). Thus, we measure and minimize the number of useless readers (Equations 29 to 31).

$$f_2 = \frac{1}{1+\varepsilon^2} \tag{29}$$

$$\varepsilon = Target - Total\ \ Useless\ Readers \tag{30}$$

$$Target = \ 0\ Useless\ Readers \tag{31}$$

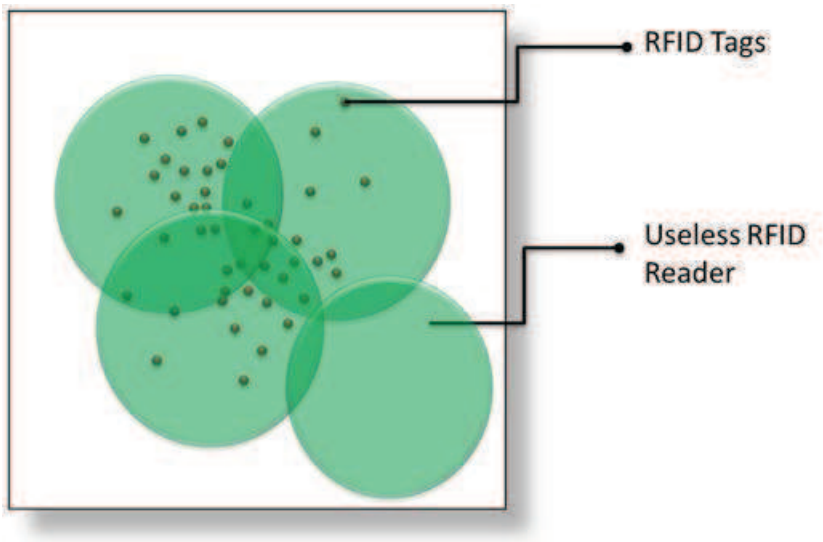


Figure 55. Useless Reader

4.5.2.3 Third Objective: Maximize the Number of tags covered.

One of the main objectives is to be able to read the totality of the IDs from the deployed tags. We fixed a target of 100% of tags IDs to be detected or read (Equations 32 to 34). We assume that if the tags are covered they will be detected but in practical applications it will not always hold, thus we provide an error reference value determined by defining three cases. In the first case, we associated a uniform probability distribution to the event of a tag being detected. The second and third cases define concentric circular areas inside the reading region with variable probabilities of detecting tags. For the two regions case we empirically defined 90% and 10% as the probability detection rates, whereas 90%, 70% and 10% for the 3 regions case (Figure 56). Finally, the parameter obtained is a reference value that indicates the number of tags that might not be detected.

$$f_3 = \frac{1}{1+\varepsilon^2} \tag{32}$$

$$\varepsilon = Target - Total\ Covered\ Tags \tag{33}$$

$$Target = \ 100\%\ of\ the\ deployed\ tags \tag{34}$$

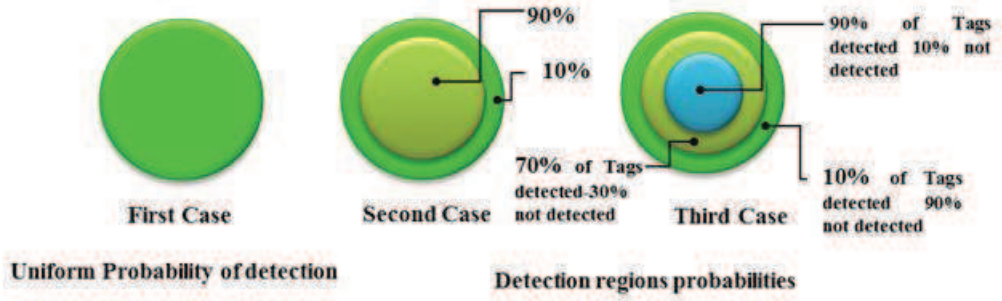


Figure 56. Zones with different tags detection probabilities.

4.5.2.4 Fourth Objective: Minimize the Number of readers located out of the deployment area.

Since we are allowing a huge number of solutions to be evaluated we need to restrict as valid ones all of those whose readers are only located inside the area to be covered. For that reason we measure the number of readers located out of the established bounds and we eliminate those solutions from the valid ones (Equations 35 to 37). This objective works as a filter for the solutions that has readers out of the deployment area.

$$f_4 = \frac{1}{1+\varepsilon^2} \tag{35}$$

$$\varepsilon = Target - Total\ \ Readers\ out\ of\ \ bounds \tag{36}$$

$$Target = \ 0\ Readers\ out\ of\ bounds \tag{37}$$

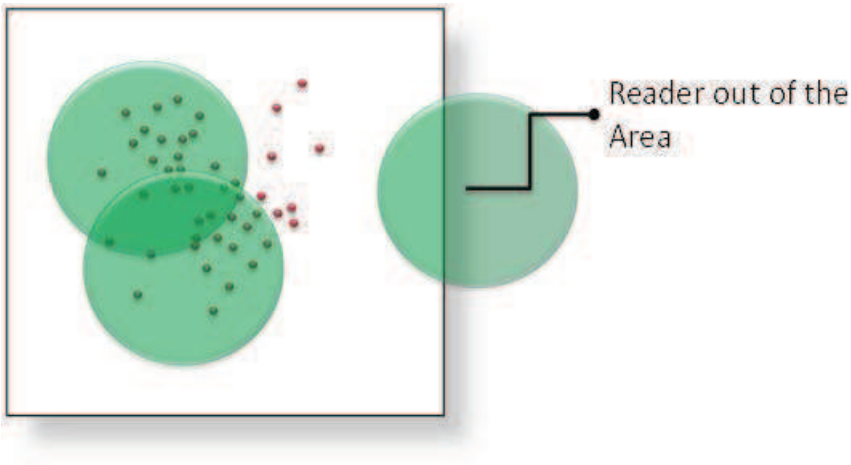


Figure 57.Minimize the Number of readers located out of the deployment area.

4.5.2.5 Fifth Objective: Minimize the Number of redundant readers.

Some solutions might include redundant readers (Figure 58), it means they cover the same or a subset of tags that are detected by another one. Those solutions must be discarded because they imply interference, no useful operation and additional cost. We counted the redundant readers by observing the set of tags that each reader detects. We compare this result with the rest of the deployed readers. If a reader contains a subset of tags already considered by another reader it is considered as redundant. The target is to have zero redundant readers (Equations 38 to 40).

$$f_5 = \frac{1}{1+\varepsilon^2} \tag{38}$$

$$\varepsilon = Target - Total\ \ Redundant\ Readers \tag{39}$$

$$Target = 0\ Redundant\ Readers \tag{40}$$

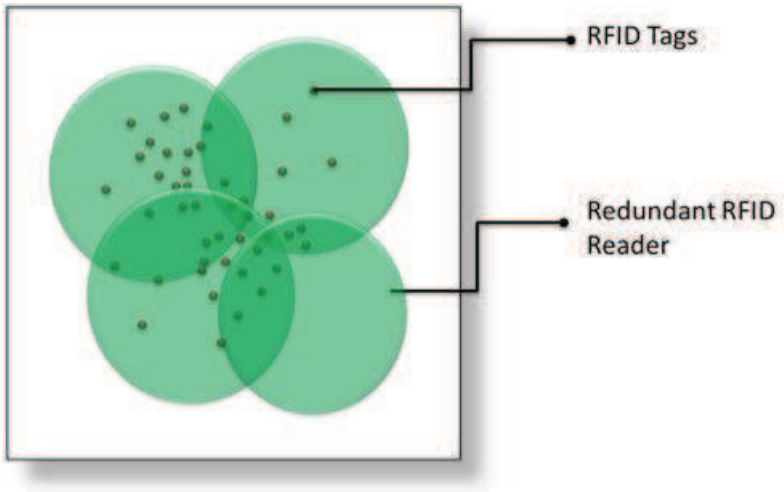


Figure 58. Redundant Reader

4.5.2.6 Sixth Objective: Minimize the Number of tags located in overlapped reading areas.

We observed that for geometrical considerations readers would tend to overlap in some areas. Additionally, it is possible to obtain solutions that comply with all the previous five objectives but that may present tags located inside overlapped areas (Figure 59). Consequently, reader-to-reader

collisions might appear because several readers can be performing a reading process at the same time. Thus, we defined our last objective function to take into account and prevent this problem (Equations 41 to 43). The target is to obtain zero tags in the overlapped areas.

$$f_6 = \frac{1}{1+\varepsilon^2} \tag{41}$$

$$\varepsilon = Target - Total\ \ Tags\ in\ overlapped\ area \tag{42}$$

$$Target = \ 0\ \ Tags\ overlapped \tag{43}$$

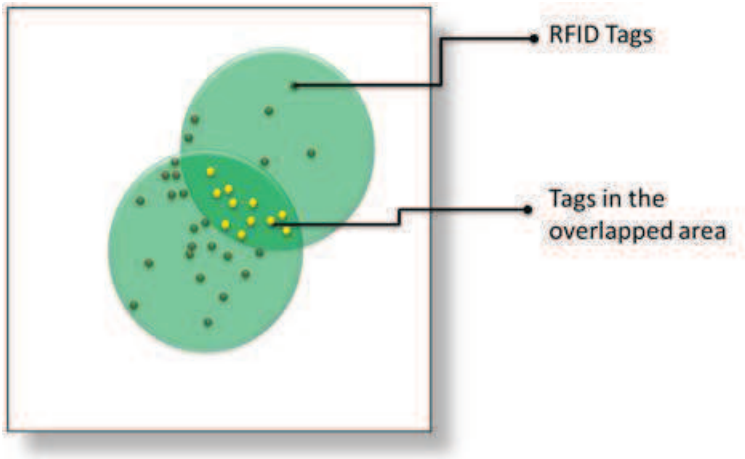


Figure 59. Affected tags

In Figure 60, we present the six objectives considered for the evaluation of our multi-objective fitness function.

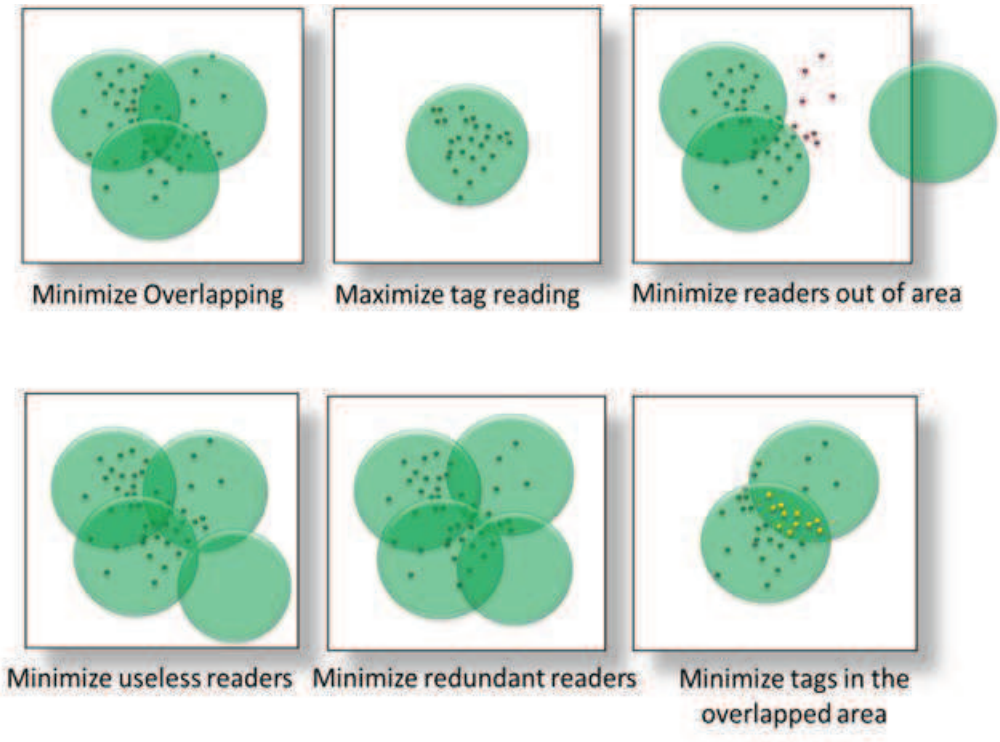


Figure 60. Problem modeled objectives.

4.6 RFID network topology design implementation with Genetic Algorithms.

4.6.1 Problem Coding

In order to code the topology design problem traditional binary coding is used. In our work, we chose to represent a reader with a two-dimension (2D) coordinate (X, Y) measured in meters (m) and a power level measured in Watts (W). We defined the Chromosome that represents a RFID reader as a binary string of 21 bits that holds 3 Genes, power, X and Y position of the reader. The first 3 bits provide seven possible values or alleles that are assigned to seven empiric-defined power steps. The following power steps are used: 0 (reader OFF), 0.1, 0.2, 0.4, 0.6, 0.8, 0.9 and 1W (maximum power). The binary string is divided as shown in Figure 61. The 2D position of the reader is defined by an X and Y Cartesian coordinates. Five bits define the integer part of the position, thus we can obtain 31 different values. The decimal part of the coordinates is defined by using 4 bits to code values from 0 to 9. We can place readers between the range of X [0, 31.9] and Y [0, 31.9] meters with a 10 cm resolution. For bigger areas, the string needs to be modified in order to allow more values.

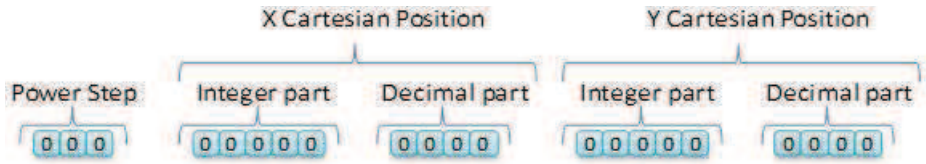


Figure 61. Problem Coding: Binary string of a RFID Reader.

If we define a maximum of 10 readers to be deployed, each candidate solution is composed by 10 x 21 = 210 bits. It gives us a huge search space of 2^{210} possibilities where GA performs successfully in order to obtain optimal solutions.

4.6.2 GA Operators

The GA method proposed has the following operators implemented:

4.6.2.1 Selection

We implemented the typical crossover operator called Roulette Wheel and the Tournament and Random Selection ones [**82**]. In the first method each individual is assigned a slice of a "roulette wheel" where the size of each division is related to the fitness they have. The roulette will spin as many times as the size of the new required population. The higher the fitness value, the higher the probabilities for an individual to be selected. In the Tournament method, we select k individuals from the population. The value "k" is known as the tournament size. The individual that has the highest fitness among the rest is the one to be selected. Finally, Random Selection chooses an individual among the total population randomly. We also implemented Elitism that keeps the best solution obtained through all the iterations performed.

4.6.2.2 Crossover

We implemented two types of crossover: the typical one point crossover with two-parent, and two-point crossover with three parents [**87**](Figure 62 and Figure 63). The idea of the crossover is to combine genes from the selected parents to obtain a new solution that in theory should perform better. The cutting points are selected randomly and then the genes are combined to form new individuals. Once the new individuals are generated, the algorithm chooses the best solution (highest fitness) among the new individuals as the new offspring.

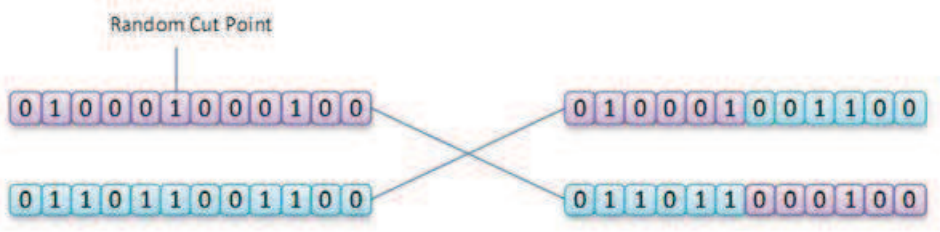


Figure 62. Two-parent crossover, one random cut point

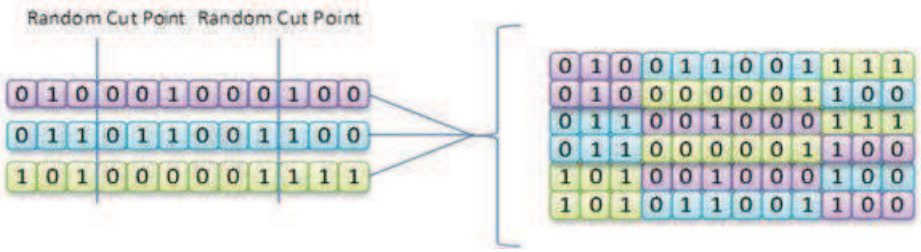


Figure 63. Three-parent crossover, two cut points

4.6.2.3 Mutation

The basic mutation operator flips one bit depending on a probability of mutation occurrence. In this implementation, we used that operator and two additional ones. The simplest one flips one randomly selected gen from "1" to "0" or vice versa. The other two operators are cataclysmic mutation and Migration [**88**]. The first performs intensive bits flips randomly in the whole population (except on the best solution kept if Elitism is used). The second method introduces randomly generated individuals. Mutation augments the diversity and reduces convergence to local optima.

4.7 RFID network topology Software tool design

The purpose of developing a software tool is to provide a flexible framework that permits experimenting with several parameter variations in order to obtain RFID topology design solutions. JAVA SE [**89**] was used to develop the application on the Netbeans 6.9 IDE [**90**]. The tool provides several customizable parameters that can be easily saved and recalled through a GUI (Graphical User Interface). The block diagram of the application is represented in the following figure (Figure 64).

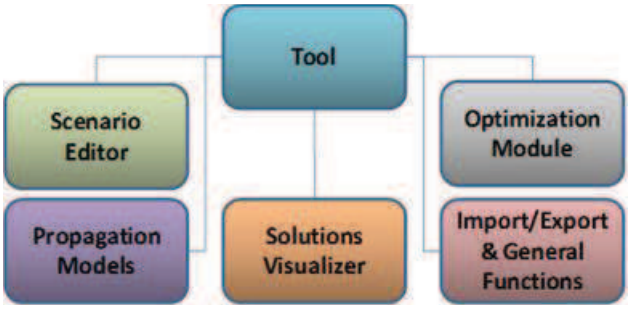


Figure 64. RFID Topology design Tool modular structure.

4.7.1 Scenario Editor

The scenario editor allows locating RFID tags in the deployment area in a graphical way (Figure 65). Once the tags deployed the layout can be saved and edited for future uses.

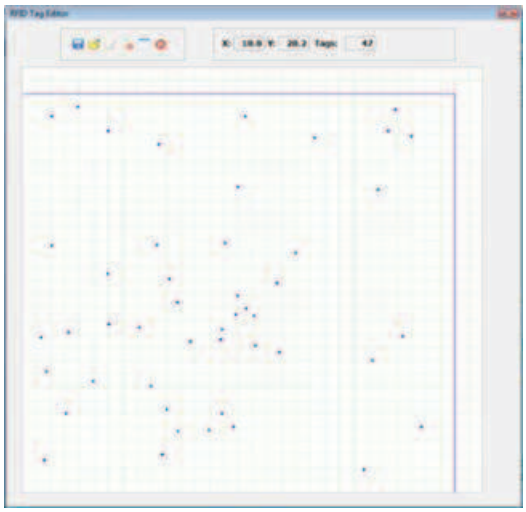


Figure 65. RFID planning Scenario Editor.

4.7.2 Propagation module

This module has the propagation models used to calculate the coverage area of the deployed RFID readers that will be used in the calculation of the overlapping reading area that leads to interference reduction. The models can be chosen on the GUI in the RFID parameter tab (Figure 66)

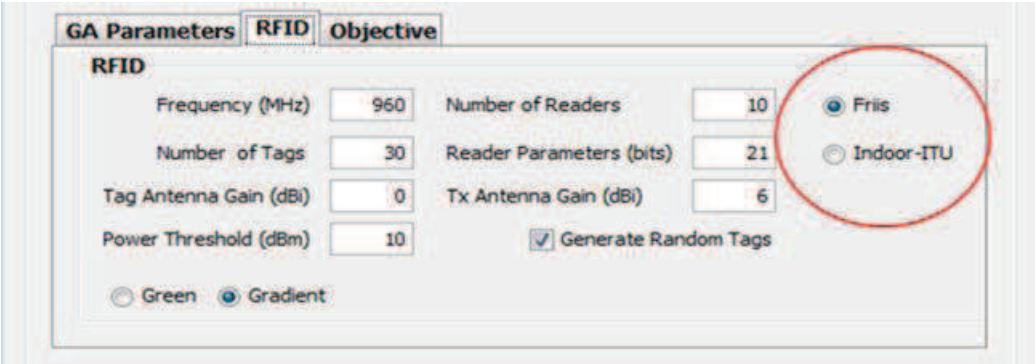


Figure 66. Propagation Model options.

4.7.3 Solutions Visualizer

The results and the execution of the GA can be seen in a graphical way. In Figure 67 we can observe the solutions visualizer screen. This information can be exported and saved for later analysis.

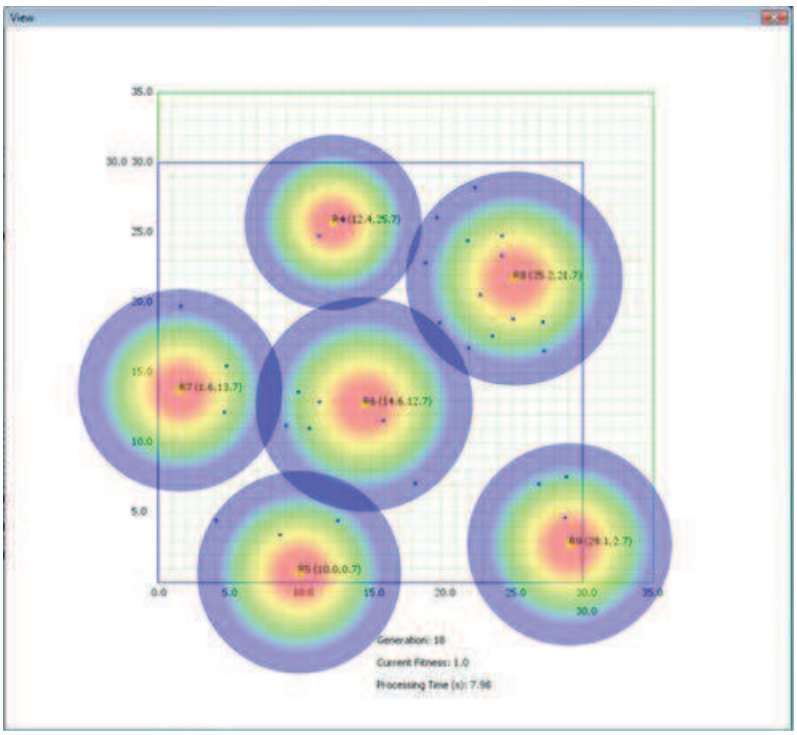


Figure 67. RFID topology Solutions Visualizer.

4.7.4 Optimization Module

This module provides all the functions and methods required for the tool in order to generate a solution that fulfills the different objectives definition. The parameters of the optimization and objective function can be modified on the GA parameters tab and Objective tab. For the GA we can define the following parameters: Population size, number of genes per individual, mutation and crossover rates, cataclysmic and migration counters, type of crossover, type of selection and Elitism (Figure 68).

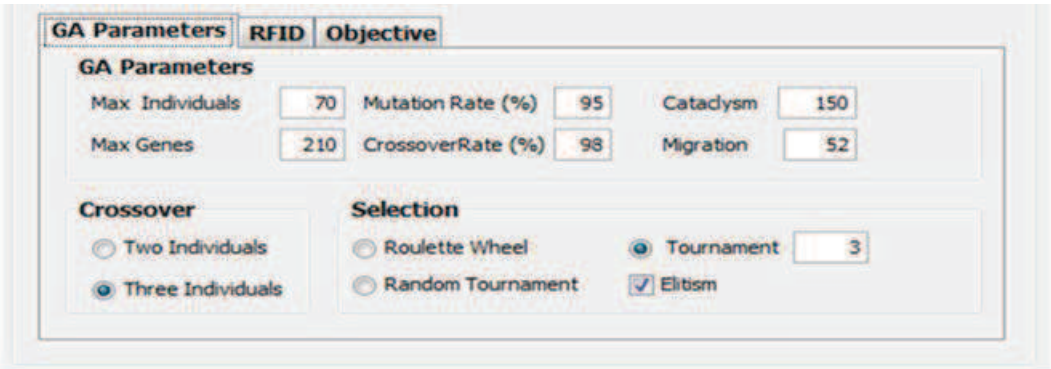


Figure 68. GA parameters.

We can also tune the set of weights of the multi-objective fitness function, the size of the area to be covered and the interference ratio, which is the parameter defined in the first objective function called CRatio that defines the maximum interference area allowed in the reader's deployment (Figure 69).

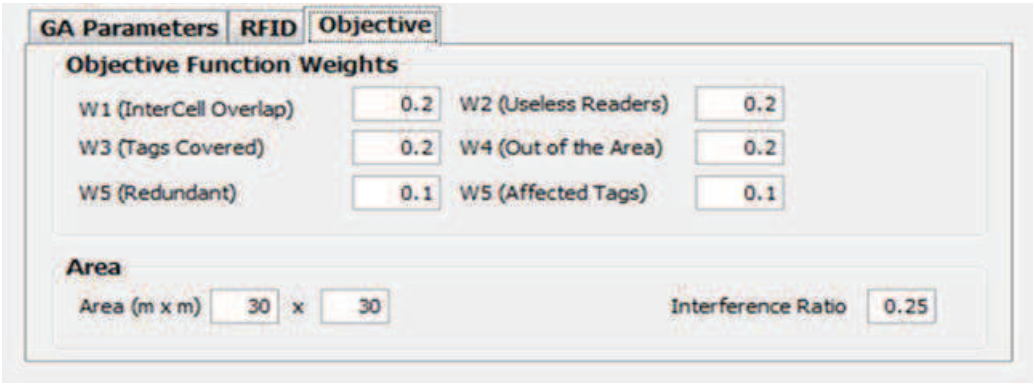


Figure 69. Multi-objective fitness function weights, area size, and Interference ratio (CRatio).

4.7.5 Import/Export & general functions Module

The tool provides saving and loading functionalities of all the parameters involved. Additionally, for testing purposes a function to generate, save and load random scenarios was included. Moreover, we can register the time and the number of generations used to find a solution into a text file, which can be utilized for statistics processing. We can also generate and export the solution report into pdf format. The button shortcuts of all the functions are shown in Figure 70.

Figure 70. Tool functions menu shortcuts.

Finally, the set of RFID parameters that can be modified are: Frequency operation, number of deployed tags, maximum number of readers, transmission antenna gain, tag antenna gain and the power threshold to detect tags (Figure 71). An option for testing that allows us to generate random deployed tags scenarios is also included.

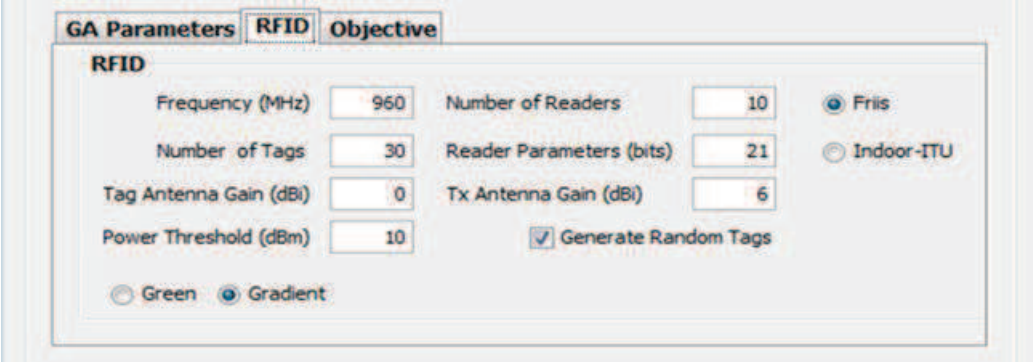


Figure 71. RFID parameters.

The GUI of the tool is shown in the next figure (Figure 72). We observe the menu shortcuts on top and below them the previously described tabs. The interface also presents a text area where the details about the running problem are presented. This information can also be exported into a single pdf file.

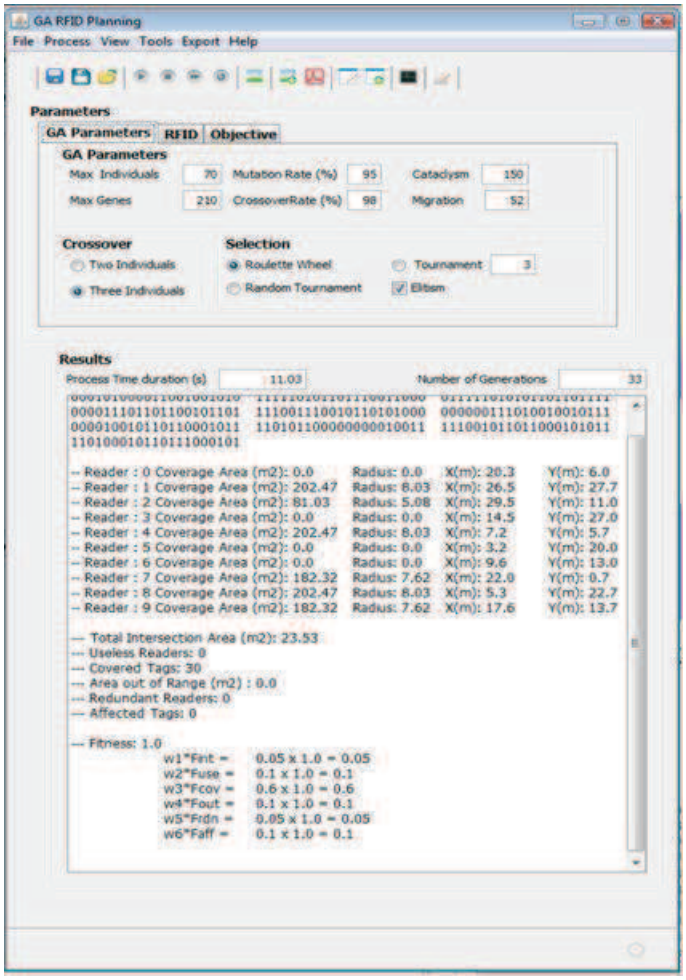


Figure 72. RFID Topology Tool GUI.

4.8 Experiment and Results

The objective of our experiment is to measure the time and the number of generations that were employed for different cases using our tool. We rely on the assumptions that the current or approximate position of the RFID tags can be obtained and RFID Tags' antennas are never at 90° with respect to the readers, thus they can always be detected. We measured the number of iterations and the time required to solve a scenario composed of 30 tags into a square area of 20 m x 20 m. The coefficients for the objective function were set empirically to 0.2 for w_1 to w_4 and to 0.1 for w_5 and w_6. We allowed an interference ratio of 0.25. The GA was set to run with a three individual crossover, roulette wheel selection and elitism activated. We ran 100 times each experiment. The complete list of parameters is presented in the Table 7.

Table 7. Simulation Parameters.

Parameters			
Number of Individuals	70	Cataclysmic counter	150
Num genes per individual	210	Number of Readers	10
Mutation percentage	0,09	Number of Tags	30
Croosover percentage	90	Readers parameters (bits)	21
Elitism	TRUE	Interference Ratio	0,25
X Area	20	Migration counter	52
Y Area	20	Operation Frequency (MHz)	960
W1 Intercell Overlap	0,2	Power Threshold (Rx Tags dBm)	10
W2 Useless Readers	0,2	Tx antena gain (dBi)	6
W3 Tags Covered	0,2	Tag antena gain (dBi)	0
W4 Out of Area	0,2	Three individual crossover	TRUE
W5 Redundant Readers	0,1	Roulette Selection	TRUE
W6 Tags in overlapped	0,1	Friis and ITU models	

The first measurement involved the comparison of Friis and ITU propagation model in terms of number of iterations and processing time. The results are expressed in Figure 73. We observe that the ITU model presents less coverage range than the Friis, consequently, it implies that more iterations and processing time are required for the ITU in order to obtain an optimal solution in contrast with the Friis model. Secondly, we compared the variation of the crossover choice in the GA. We used two and three individual methods for both ITU and Friis models. The results are presented in Figure 74 and Figure 75. We observe that three individual crossover allows the convergence of the solution in less number of iterations though more processing time per iteration is required. This is due to the complexity of the three-individual crossover function where more candidate solutions need to be evaluated in order to obtain the final individual. However, this increase in processing time is compensated by the smaller number of iterations required to obtain a solution with fitness equals to one.

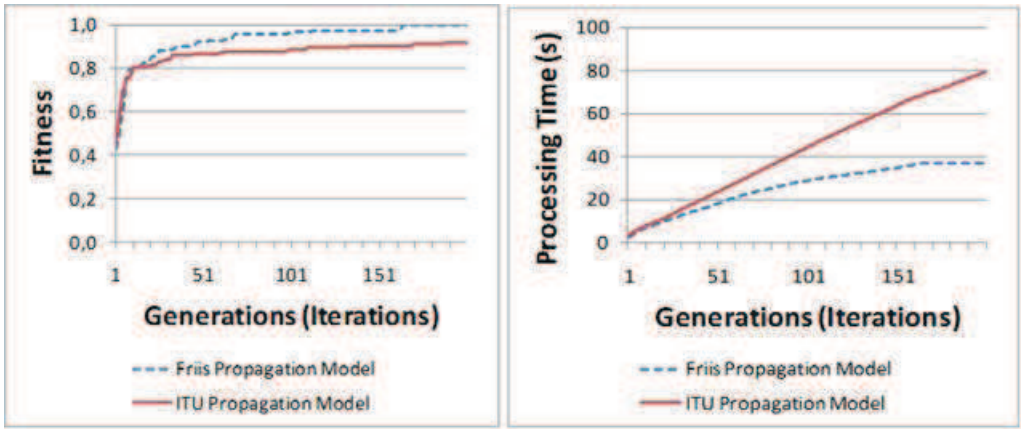


Figure 73. Generations vs. fitness and processing time for Friis and ITU models.

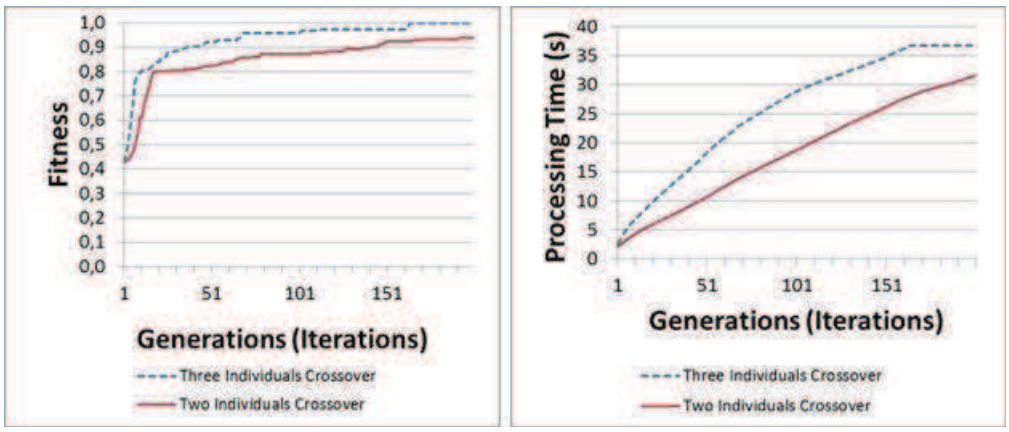


Figure 74. Generations vs. fitness and processing time for Friis model using "2" and "3" individual crossover.

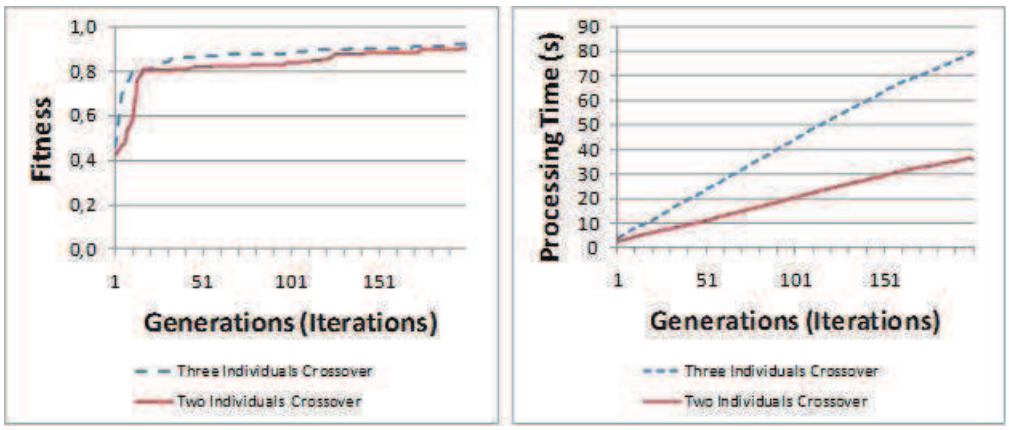


Figure 75. Generations vs. fitness and vs. processing time for ITU model using 2 and 3 individual crossover.

Finally, we noticed that the sixth objective of the multi-objective function (number of tags in the overlapped area) imposes certain load in the calculation of the optimal solution. We compared the evaluation of test problems considering the six objectives (as previously stated) and the evaluation of solving the same scenarios without taking into account the sixth objective. We observed that this condition increments in average the time of finding a solution in a range of about 26 %. It means that if we assume the readers will have an anti-collision protocol to deal with tags located in the interference area, we can reduce the load of the topology design calculation by setting the weight of the sixth objective to zero (see Table 8).

Table 8. Effect of the objective of minimizing the tags located in the overlapped area.

Without minimizing the tags in the overlapped area		
	Time (s)	Iterations
Average	9,56	39
Standard Dev.	3,52	17
Confidence Interval	0,75	3,77

Minimizing the tags in the overlapped area		
	Time (s)	Iterations
Average	13,07	63
Standard Dev.	9,59	51
Confidence Interval	2,04	11,03

4.9 Chapter Summary

In this chapter, we presented a RFID network topology design based on Genetic Algorithms. We proposed a multi-objective fitness function to evaluate the problem of deploying RFID readers into a square region. We considered six objectives such as minimize the overlapping of the RFID reader area, minimize the number of useless readers, maximize the number of tags covered, minimize the number of readers located out of the deployment area, minimize the number of redundant readers and minimize the number of tags located in overlapped reading areas. We also developed a customizable software tool to assist in the topology design and we performed measurements to compare its performance. We compared the process of finding solutions for different scenarios varying factors such as the propagation model and the genetic operators (crossover and selection). We observed that the use of Three-parent crossover operator perform better than the classical Two-parent approach independently of the propagation model used. Additionally, if we exclude the sixth objective (minimize the tags in the overlapping area) we obtain an average of 26% less time in order to find a satisfying solution. This was considered in case if the RFID network has a generic anti-collision system that will guarantee this objective. New additions to the tool like more propagation models as well as directional antenna pattern, electromagnetic propagation patterns and a three-dimensional (3D) deployment plan can improve the application for more complex scenarios.

Implementation

Chapter 5

RFID service for non-RFID enabled devices: Embedded hardware implementation.

In this Chapter we present an embedded hardware implementation of a platform that combines RFID with wireless network interfaces. This is done as explained in our contribution in chapter 3 in order to provide the RFID tag IDs to devices without RFID capabilities. We additionally measure the performance of the data transmission and propose practical applications.

5.1 Research Problem

RFID technology needs the use of specialized hardware in order to obtain the information that tags provide. However, this information can be also obtained by implementing platforms that take into account heterogeneous network interfaces as stated in Chapter 3. The challenge lies in how to integrate different devices into a single stand-alone platform that will permit the transmission of RFID tag IDs to other devices that will use that information.

5.2 Research Objectives

We require to design and implement a hardware solution that communicates the RFID tag IDs to any user that has a common network interface with the platform, bypassing the limitation of not having the RFID hardware available. We focused on the implementation of the solution by using embedded hardware due to the reduced equipment costs, flexibility and versatility to permit the development of different Internet of Things applications and services. We selected Wi-Fi and ZigBee as the common network interfaces due to the popularity of those protocols, the huge number of portable devices Wi-Fi enabled [**54**] the coverage range, and the network capabilities [**91**]. Conversely, due to the modularity and flexibility of the solution other network devices based on other technologies can be added such as Ethernet, Bluetooth, or GPRS.

5.3 Related Work

There are different uses and implementations of RFID systems by using embedded hardware to provide diverse services and applications. In [**92**] a verification and validation system is developed based on a microprocessor. It permits to reduce cost and provide easy maintaining of the system proposed. In [**93**] a Location Aware service is implemented based on an embedded RFID platform. In [**94**] a system to perform vehicular monitoring is implemented by combining RFID, GPS and GSM modules with a microcontroller as central unit. In this way, a tracking system can be put into practice for vehicular uses. The integration of RFID and Wireless Sensor Networks (WSN) by using embedded hardware can be seen in [**95**]. In that proposal a RFID and a RF transceiver are bind to a microcontroller in order to provide warehouse management services. In [**96**] an architecture and hardware implementation of an Access Control system is presented. It uses embedded hardware and shows a modular architecture approach. In our contribution, we propose the transmission of the RFID tag IDs to users that have no RFID hardware available but a WLAN (Wireless Local Area Network) interface and/or a ZigBee module. We developed a coordination layer to permit the control and information delivery by using a modular-built solution that provides RFID detection and wireless

connectivity. Additionally, we measured delay metrics and we provide application scenarios of the system developed.

5.4 System Architecture

5.4.1 Overview

The implementation provided can be defined as a proxy that transmits the RFID tag IDs to users connected by a common wireless interface. This overrides the necessity of RFID hardware available at the user side in order to obtain RFID tag IDs. This system is a simplified version of the Multi-Mode Node proposed in Chapter 3. In this node, no P2P protocols are implemented and a Client-Server approach is used. Regarding the implementation, we rely on a microcontroller as the core of the system and different hardware modules that perform specific tasks. In Figure 76 we present the basic modular structure of the system. Note that we also implemented an extended version of the platform to control heterogeneous devices of the Internet of Things (RFID, Sensors, and Actuators). It was performed in the context of the research project Track-IoT [**97**] where more devices and communication systems were used to offer a general RFID service based on heterogeneous readers (fixed and mobile), a control of the reader-to-reader collision, and a control of event sensing to trigger actions. The Track-IoT Platform is described in [**97**].

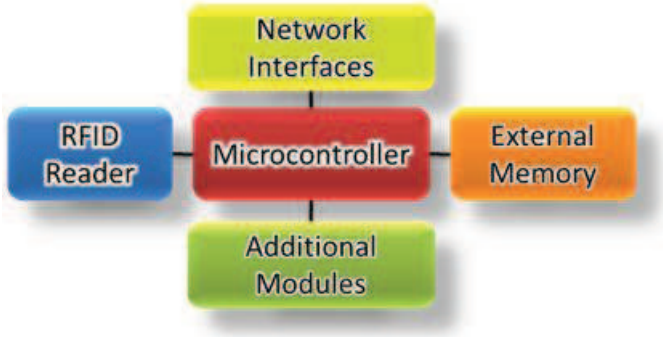


Figure 76. Block Diagram of the System.

5.4.2 Modules

There are five basic modules and that constitute the system. These are described as follows.

5.4.2.1 Microcontroller (MC)

The MC provides the control and coordination of the different interconnected modules. This device should provide enough memory space to hold the controlling algorithm as well as enough interface ports to branch the different modules. They will perform specific functions like RFID reading, external memory and the network interface among others specific to the application to be implemented.

5.4.2.2 RFID Reader

It performs the reading of the RFID tag IDs that are going to be transmitted to the users that request them via the network interface.

5.4.2.3 External Memory

Microcontrollers are limited by the memory they normally provide. For this reason, we require external storage space in order to record the tag IDs read by the RFID module.

5.4.2.4 Network Interfaces

This module is the common bridge to the users with non-RFID embedded hardware and the hardware that will provide the list of RFID detected tags. Some options can be Ethernet, WLAN, ZigBee, Bluetooth and GPRS.

5.4.2.5 Additional Modules

Extra components can be added to perform specific functions for example to provide time reference, visual information elements, user inputs, sensors, actuators, etc.

5.5 Implementation

Based on the structure previously depicted we implemented the platform as it is described in the following sections.

5.5.1 Overview

We implemented the proxy solution by using commercially available modules (Figure 77). A microcontroller is the core of the system and it provides enough ports, memory and processing power required to this implementation. We added four modules that perform the RFID detection, external memory space, WLAN interface and a ZigBee module respectively. The description of all these elements is provided as follows.

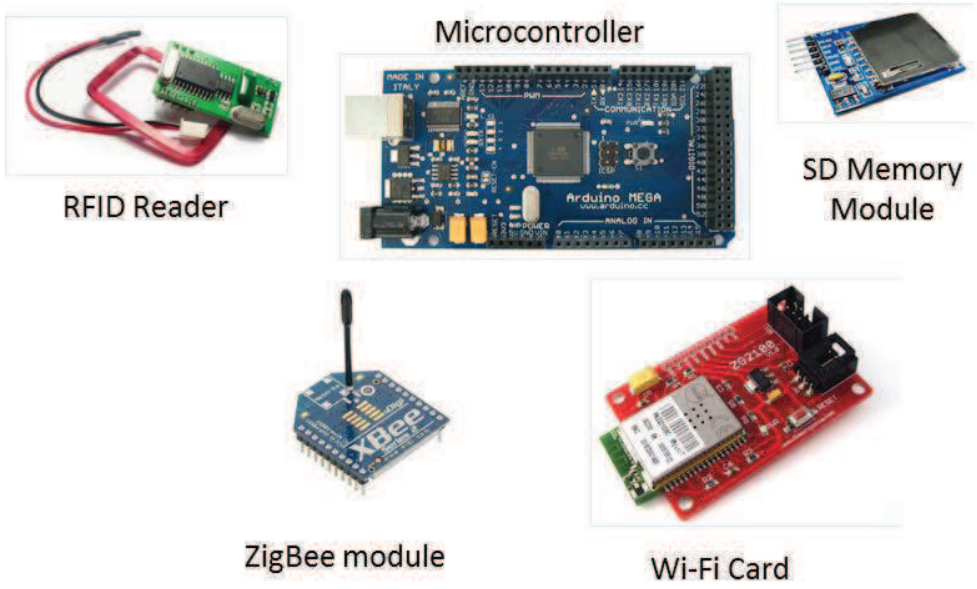


Figure 77. Hardware Modules used.

5.5.2 Modules

5.5.2.1 Microcontroller (MC)

We base our solution on the affordable and versatile Atmel 1280 microcontroller [**98**]. It provides an 8-bits-16MHz core with 128KBytes of flash memory, 8 Kb SRAM (Static random-access memory), 4 Kb EEPROM (Erasable Programmable Read-Only Memory), 54 digital ports, 16 analog ports, I2C(Inter-Integrated Circuit), SPI(Serial Peripheral Interface Bus), Serial ports and operates between 2.7 and 5.5 volts. A commercial board provided by Arduino [**99**] provides an USB (Universal Serial Bus) programming interface and several open source libraries in C-like code. We

remark that any other microcontroller or embedded board that provides enough resources to implement the system can be used (Annexes C and D).

5.5.2.2 RFID Reader

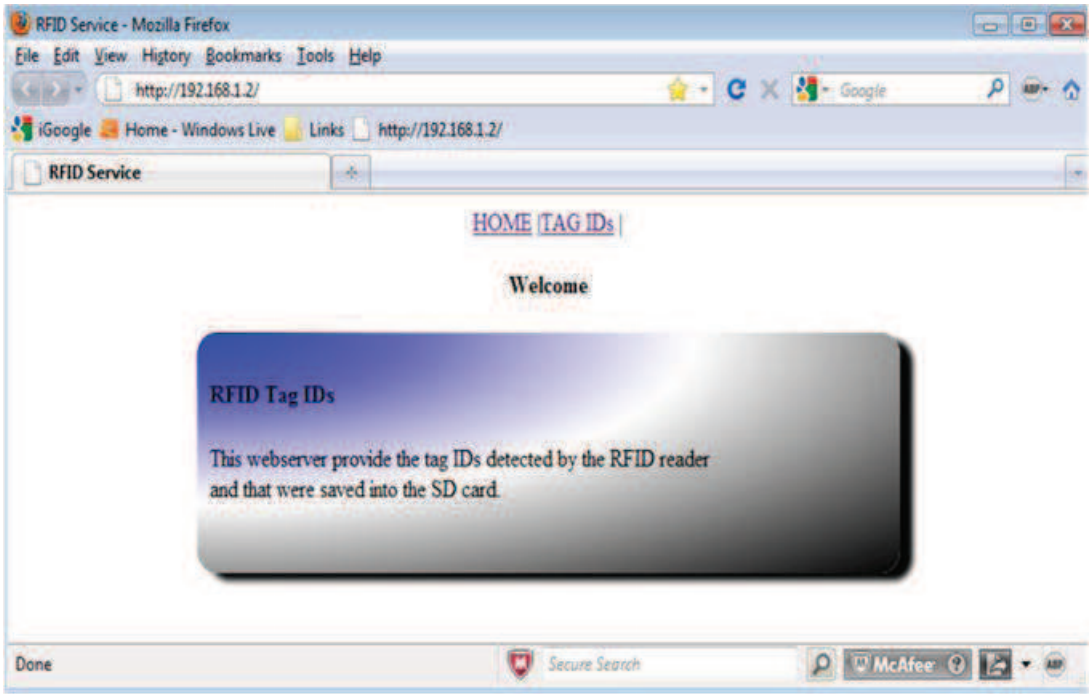
The RFID module is a Low Frequency (LF) Wiegand interface device [**100**]. Wiegand uses 26 bits format where 24 bits are user data (Tag ID) and 2 bit are used as parity. The decoding time is less than 100 Milliseconds with a maximum effective distance tag-reader up to 150 mm. It can read passive RFID tags that comply with the EM4100 standard. It uses two digital ports connection in the MC in order to transmit the detected tag ID. For this implementation the tag IDs are limited to 24 bits due to the limitations of the module however, other RFID readers can be used to implement different standards and have longer IDs.

5.5.2.3 External Memory

If the MC is powered off, the information not saved in the EEPROM will be lost. Due to the reduced amount of memory available (4Kbytes), we included an external storage module. A SD (Secure Digital) card unit allows us to store the tag IDs that had been obtained through the RFID reader module. We can provide up to 16GBytes of external storage by using a standard SD card. This module is connected to the MC by using a SPI interface. There are available FAT (File Allocation Table) libraries for the Arduino to read, write and edit the log files recorded [**101**].

5.5.2.4 Wireless Interfaces

We employed a 2.4 GHz, low power 802.11 a/b compliant module [**102**]. It provides WLAN connectivity up to 2Mbps. It can be setup in Infrastructure or Ad hoc modes and supports open, WEP (Wired Equivalent Privacy), WPA and WPA2 (Wi-Fi Protected Access II) methods. This unit also uses the SPI port of the MC. In order to provide the list of tags to the users we create a basic web server that provides via HTML the list of gathered tags by the RFID reader. In this way, users with a WLAN interface can connect to the network set up by the system and obtain the list of available tags (Figure 78).



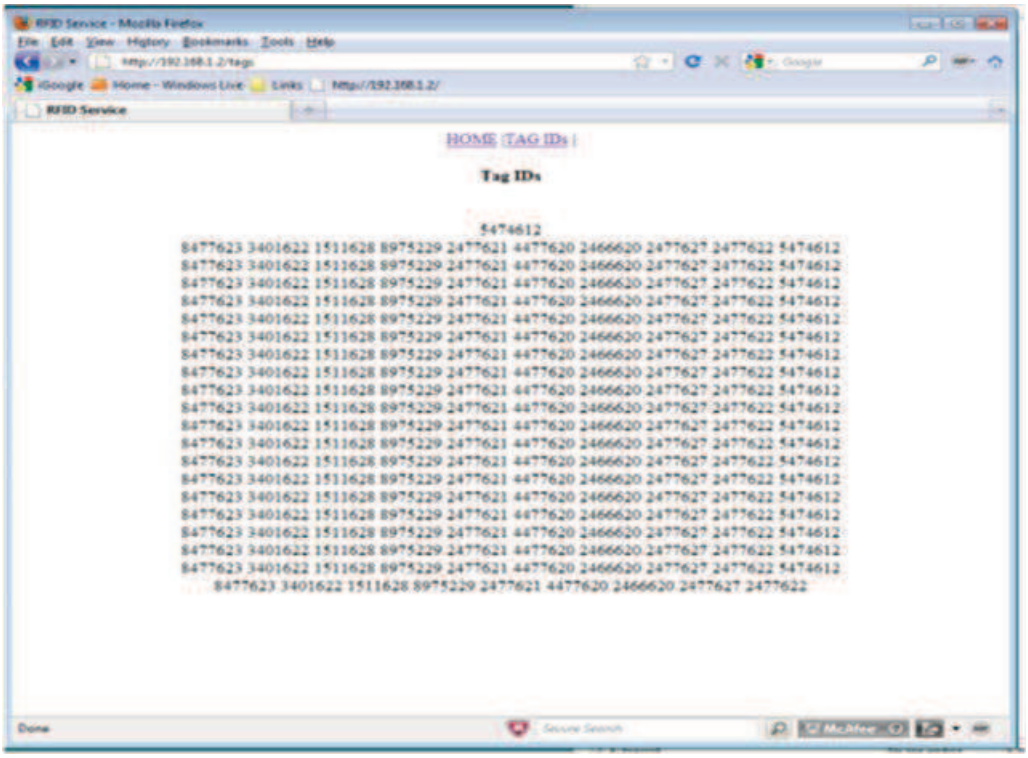


Figure 78. Web server, Tag IDs display.

We added a ZigBee module to the system in order to provide network connectivity for that set of protocols. The ZigBee module used is a commercial XBee Series 2 model [**103**] (Figure 79) (Annex G). It operates at 2.4 GHz using 802.15.4 protocol. The maximum power output is 2 Milliwatts with an Indoor coverage range of 40 m and 120 m for outdoor deployments.

Figure 79. XBee Series 2 802.15.4 ZigBee module.

5.5.2.5 RFID Tags

We used Standard EM4100 compliant RFID passive tags (Figure 80). The RFID reader we use for the experiment can detect 24 bits of the tag ID.

Figure 80. RFID passive tags.

5.5.3 Flow Diagram

The proposed system performs the following processes in order to provide the tag IDs to users connected to the network. At first, the system executes the Initialization phase where all the modules run certain routines to make their resources and functionalities available. The WLAN module is activated and the initial page is loaded thus, it is ready to be displayed when the requests arrive. The ZigBee module is also set to transmit and receive information. After that, the SD module is initialized providing access to the card in order to read and write information. Later, the routine that starts the RFID reader is executed. At this step, a port interruption is assigned in the MC in order to detect the pulses transitions that identify the bits in the RFID tag.

Once the Initialization process ends, the system is ready to deal with two types of events, either RFID tag detection or a user request to get the Tag list. If a tag is detected the MC sends the ID to the SD card and updates the information that will be published in the site provided by the web server. If a user connects to the server, it will send a HTML (HyperText Markup Language) that includes the list of detected tags that are stored into the SD card. The WLAN module is set to work in ad-hoc mode and different security methods can be implemented (open, WEP, WPA). The ZigBee module is configured as a ZigBee-Coordinator node and it will deliver the requested tag IDs to other compatible nodes when requested. The processes flow chart is presented in the following figure (Figure 81).

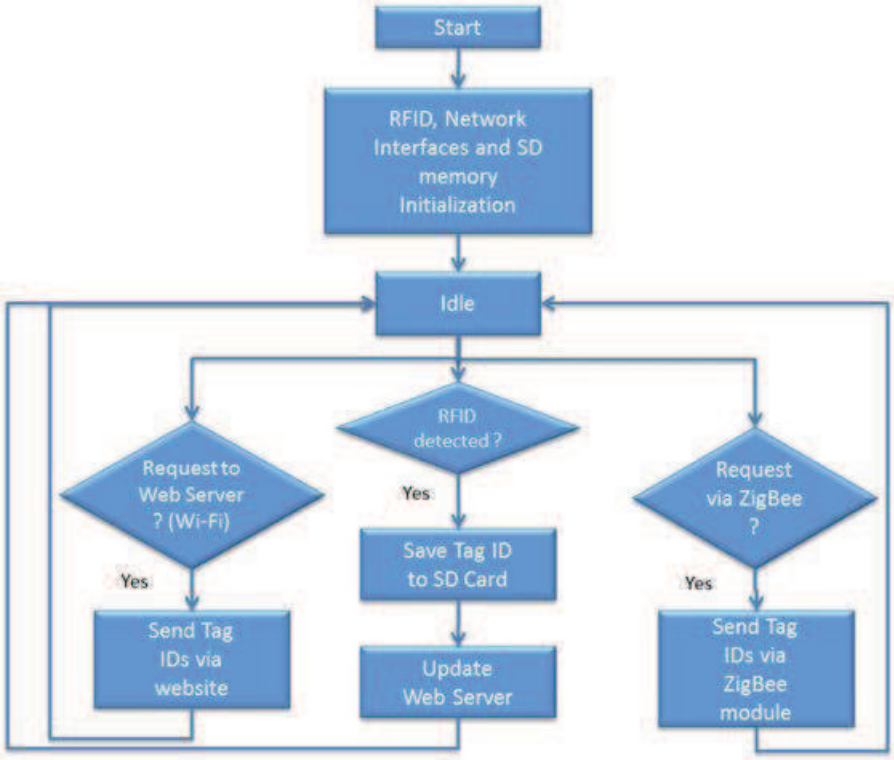


Figure 81. System Processes.

5.6 Experiment and Results

First, we measured the delay incurred while initializing the different modules. We obtain the total time required for the system to be ready to operate. Secondly, we measured the transmission delay of the tag IDs to a user connected via the web server. Additionally, we measured the Round Trip Delay for tag IDs transmission by using two ZigBee nodes. One was set as a Coordinator and the second one as Router and we compare this delay with a wired serial transmission as reference.

5.6.1 Initialization Delay

In this section, we present the initialization delay of the modules used.

5.6.1.1 WLAN Module

In Table 9, we show the average time required for the WLAN module to be fully functional.

Table 9. WLAN Initialization Delay.

Average(s)	**21.09**
Standard Deviation	0.72
Confidence Interval (95%)	0.15

5.6.1.2 SD Module

In Table 10 we present the average time required for the SD module in order to be ready to read and write files into the SD memory card.

Table 10. SD Initialization Delay.

Average(s)	**0.3476**
Standard Deviation	0.0005
Confidence Interval (95%)	0.0001

5.6.1.3 RFID Module

Finally, in Table 11 we show the average time required for the RFID module to be ready to detect tag IDs.

Table 11. RFID Initialization Delay.

Average(s)	**0.037**
Standard Deviation	0
Confidence Interval (95%)	0

The Total Initialization Delay (TID) can be calculated by using linear Equation (1).

$$TID = WLANDelay + SDDelay + RFIDDelay \quad (1)$$

The average TID is about 21.47 seconds where the WLAN module is the biggest contributor due to the complexity of the module and the setting up of different mechanisms that it requires in order to operate. The ZigBee module does not require significant time to be ready to operate so we discard this figure in the calculations.

5.6.2 RFID Tag IDs Transmission Delay

We conducted a test by using the software protocol analyzer Wireshark [**104**] in order to measure the time a user needs to wait in order to get the list of available tag IDs. A difference in relation with the description of the system in Chapter 3 is that in this case we have no delay associated with the Authentication or Initial configuration described previously. The results are presented in the next figure (Figure 82) and in the Table 12.

Table 12. Tag IDs transmission Delay.

Number of Tags	*1*	*10*	*100*	*200*
Average Time(s)	0.45	0.66	1.11	1.57
Standard Deviation	0.01	0.01	0.02	0.01
Confidence Interval	0.0012	0.0015	0.0035	0.029

We measured from 1 to 200 tag IDs and for the maximum value, we obtain a delay of approximately 1.6 seconds.

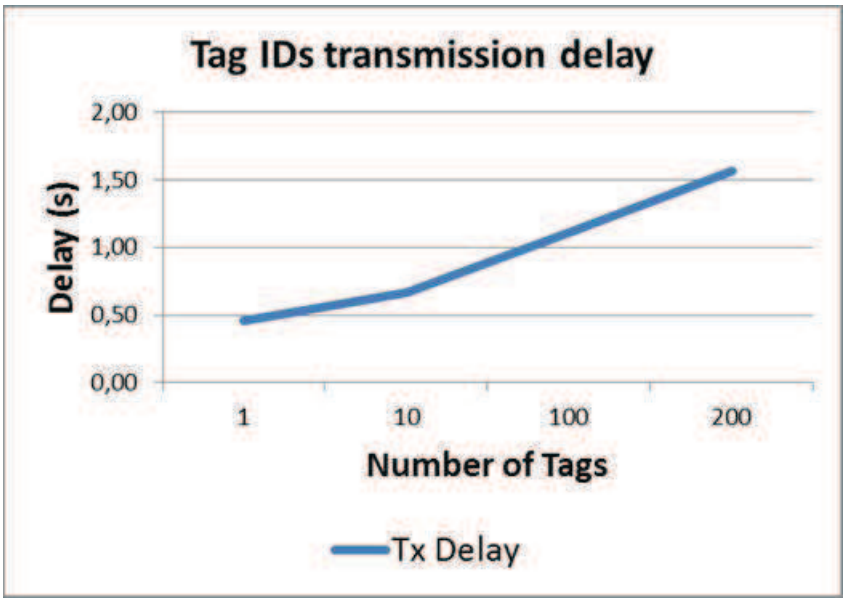


Figure 82. Tag IDs transmission Delay.

The next experiment was set up with the following configuration (Figure 83) in order to measure the delay required to transmit different number of IDs.

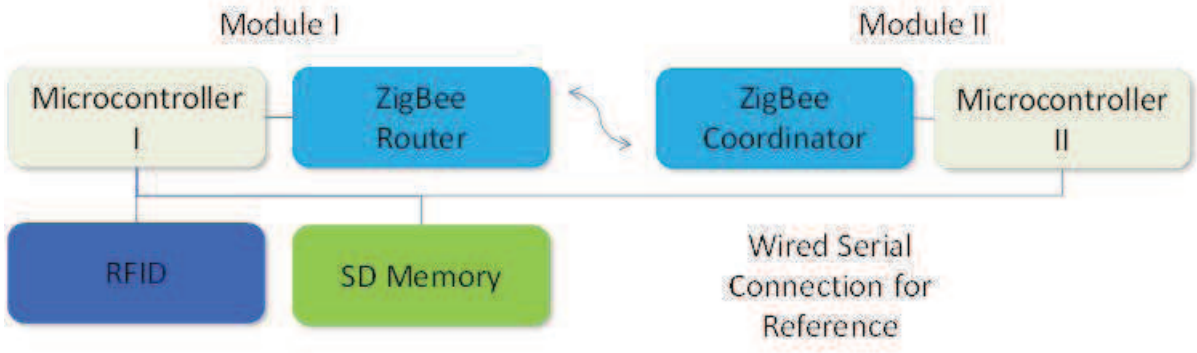


Figure 83. Modules to measure Tag IDs transmission Delay using ZigBee.

We used as for the WLAN experiment 1, 10,100 and 200 Tag IDs and we established a reference comparison metric by measuring the delay but instead of transmitting via the ZigBee modules, we used a wired serial transmission between the two microcontrollers. We set a baud rate of 9600 bps. The nodes were located at 1 m from each other and the XBee modules with line of sight. The results are provides in Table 13 and Figure 84.

Table 13. Transmission Delay wired and via ZigBee modules.

Number of Tags	*1*	*10*	*100*	*200*
Average Time(s) Wired	0.11	0.18	0.94	1.78
Average Time(s) ZigBee	0.11	0.18	0.98	1.98

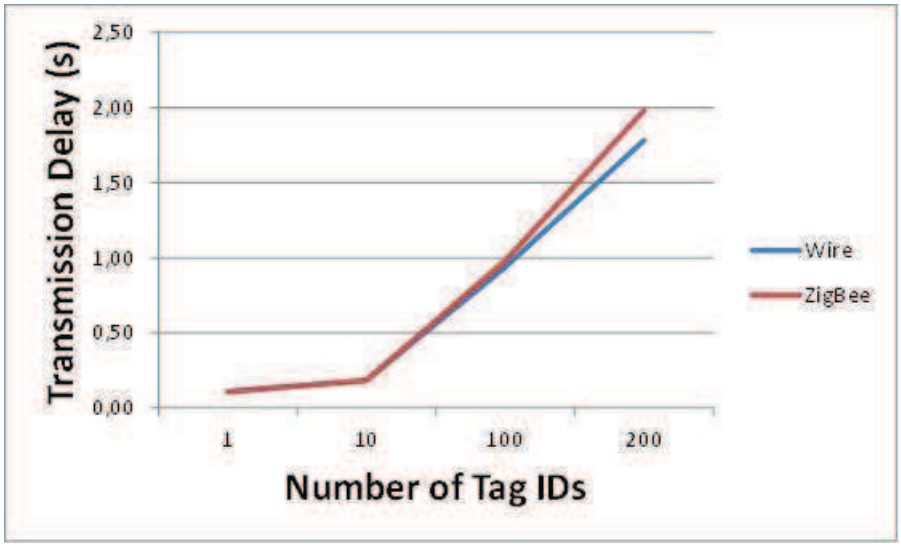


Figure 84. Round Trip Delay wired and via ZigBee modules.

We observe that there is a slight increase in the delay when the numbers of tags increase in relation to the same number of tags transmitted using the wired configuration. However, for the size of the information we transmit, the difference is not important no matter which interface we use.

Finally, we set up a different scenario with two nodes that will share some functionality. In this case, we have a node that will obtain the RFID Tag IDs and will transmit them by using the ZigBee module to a second node that will be in charge of enabling a WLAN interface for other uses with portable devices like mobile phones or pads (Figure 85).

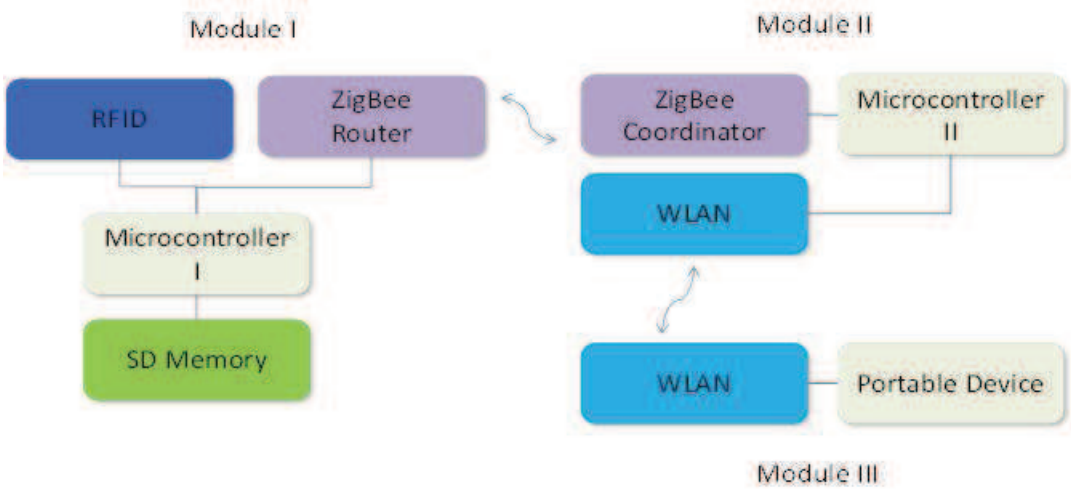


Figure 85. Nodes configuration.

The Service Delay is presented in Table 14.

Table 14. Service Delay.

Number of Tags	*1*	*10*	*100*	*200*
Delay(s)	0.56	0.84	2.05	3.55

5.7 Applications

In this section, we present two applications based on the implemented system.

5.7.1 Building Access Log

We can implement a Building Access Register by detecting the tag IDs of the persons who access the building. Additionally, we can add an external clock reference provided for example by the DS1307 from Maxim [**105**] in order to track the date and time of the access. We used an I2C module that allows us to obtain an external clock reference driven by a crystal and battery backup (Figure 86) (more details in Annex E). The log file is available via the WLAN connection.

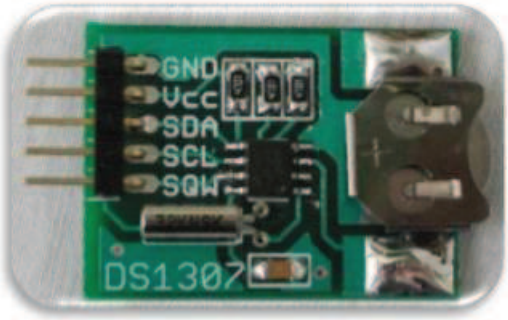


Figure 86. External Clock Reference Module.

The block diagram is shown in Figure 87. The way it works is similar to the basic configuration stated before but what changes is that every time the RFID reader registers a tag, it is saved into the SD card adding up the current date. The Web Server will be updated afterwards. The current tag list is available to the users that require that information.

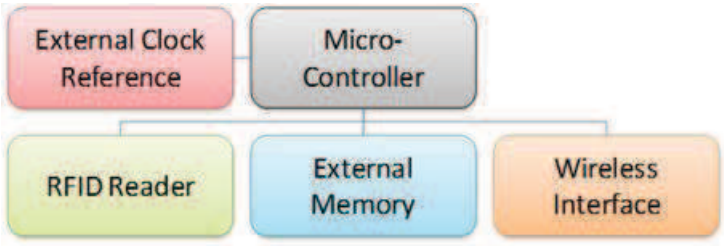


Figure 87. Block Diagram of the Building Access Log System.

5.7.2 RFID-WSN (Wireless Sensor Networks) and Actuators

The nodes can be expanded by adding other modules to provide connectivity for a WSN, actuators and other wireless interfaces as shown in Chapter 3. In this manner, data from sensors and RFID can be combined to provide an information platform and that does not require, at the user side, the use of specialized hardware such as RFID readers to obtain and exploit the data. In Figure 88 we present a general block diagram where data collected from sensors, GPS and other modules can be combined to provide an information and integration platform.

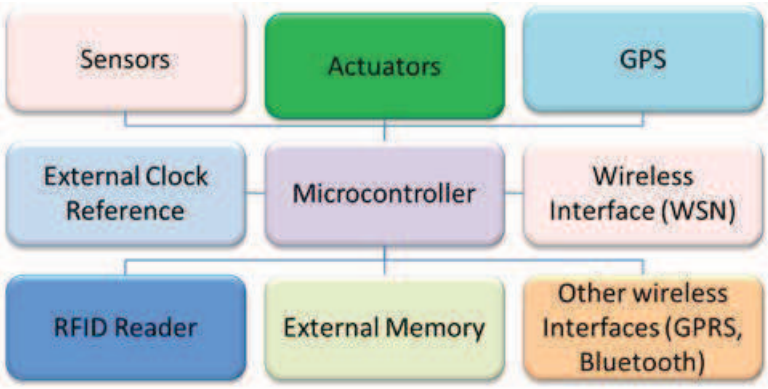


Figure 88. Wireless Sensor Networks and Actuators.

We implemented a practical application based on RFID and sensors. We have designed two boxes that contain different modules. The first one is configured with a microcontroller, a LF RFID reader and an external clock module an XBee module, a SD card module and three sensors (sound, light and temperature). The second module is simple an XBee module connected to an USB port in a Personal Computer (PC). The PC will run a Graphical User Interface (GUI) to interact with the devices. In the PC, a database collects all the information that comes from the Module I (Figure 89). Every time that a RFID tag is detected, it triggers the action of reading of the sensors. The values of the variables that the sensors obtain are stored into the SD card as well as in a database. We developed a GUI that shows the current and the stored measurements (Figure 90).

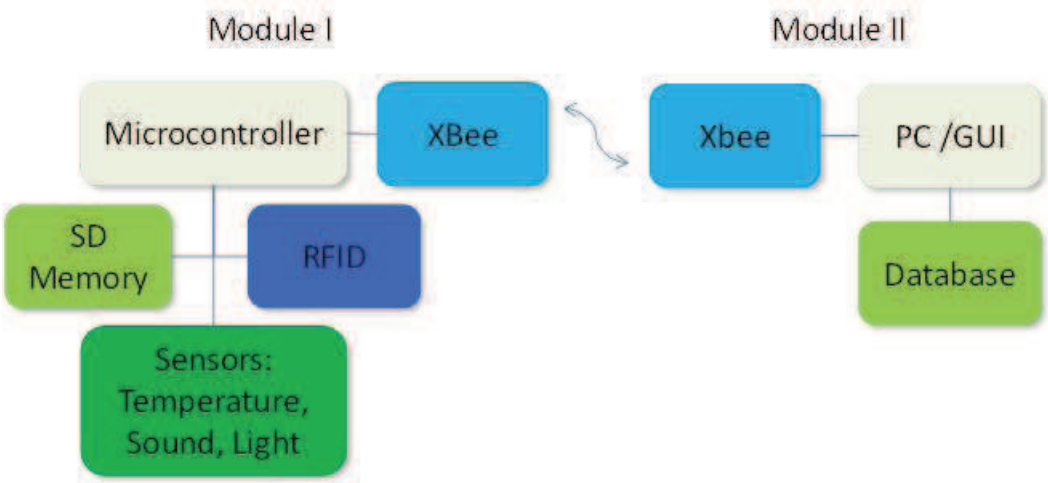


Figure 89. Modules for RFID and Sensors system.

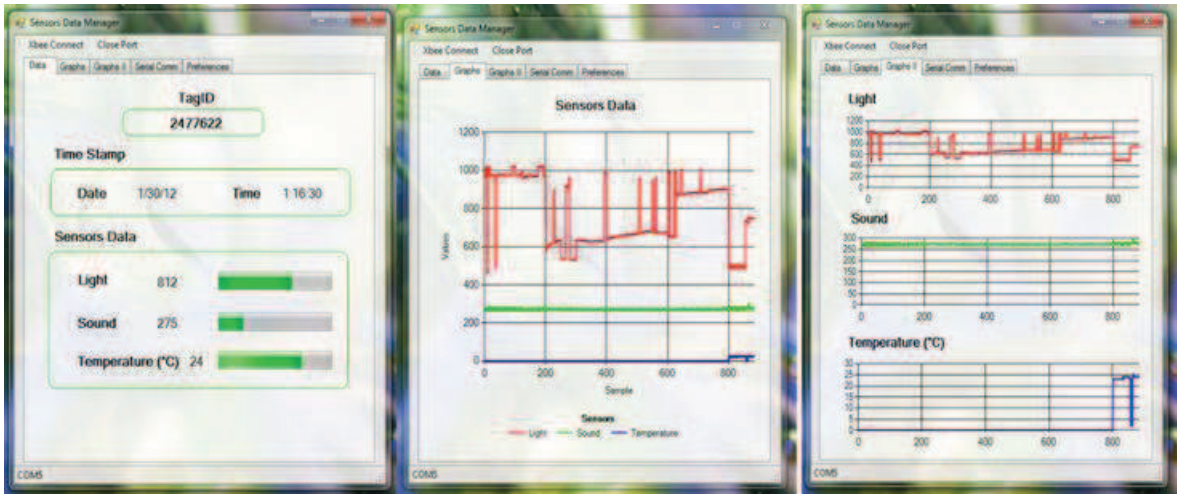


Figure 90. GUI for RFID-Sensors application.

In Figure 90, the sensor data collected is the light intensity that is mapped to a [0, 1024] interval, where 0 means total darkness and 1024 maximum light. The sound sensor follows the same principle where 0 means silence and 1024 means maximum level of sound. The temperature is directly expressed in centigrade. The date and tag ID used to trigger the sensors are presented in top of the GUI. We store every sample into the database and we present graphically the values. The X-axis represents the number of the sample and the Y-axis represents the value of the parameter measured.

5.8 Chapter Summary

In this Chapter, we tackled the issue of connecting modular devices in an embedded hardware platform to provide RFID-based applications. This platform will transmit the tag IDs read by a RFID reader and make them accessible through wireless network interfaces to users with non-RFID capability thus offering them to use RFID-based applications. The proposed system is flexible and modular allowing the extension and inclusion of additional features to the network like sensors, actuators and other components. Even the limitations imposed by the embedded hardware used, we could implement interesting features like a stand-alone web server that can publish RFID tag IDs as content, and easily get accessed by portable Wi-Fi-enabled devices. Regarding the delay measurements, we observe that they do not impose limitations for practical applications. The next step will be to implement a large-scale network with heterogeneous devices as we stated in Chapter 3.

Chapter 6

Conclusions

In this thesis, we presented our research work in the context of Internet of Things based on RFID in order to provide improvements to the RFID networks deployment and use. Although RFID is not a new technology, there are still research issues that need to be tackled in order to provide solutions to existing problems.

We presented our contributions related to the proposal of a RFID Framework for heterogeneous nodes as well as a network topology model to assist in the implementation and design of this platform. We used simulation tools to observe the behavior of the system regarding the transmission delay. The results indicate that for large amounts of IDs our approach is more efficient in relation to the TIS (Tag identification speed). An important key result is that the Framework allows nodes with no RFID hardware to take advantages of the information provided by the RFID tags deployed through a proxy node. It is also interesting to notice that devices with RFID readers can extend their reading range "virtually" by connecting to the P2P overlay and also avoiding reader to reader collision problems as the P2P overlays handles the scheduling of the RFID reading by all the participating readers (fixed or mobile).

We also designed a multi objective function to help in the optimization of the RFID network planning, and we implemented a RFID topology model and a software tool to assist us in the design of the network efficiently considering different parameters. It was developed by using Java SE and utilizing Genetic Algorithms as the optimization method. Different features are implemented including export, import, parameters variations and scenario editing.

In the last part of this work, we provided the design and implementation of a RFID information service by using embedded hardware. It offers tag IDs information to users connected via a common wireless interface (802.11/802.15.4).This system provides an integration platform into the RFID networks for devices that have no suitable hardware (RFID Reader). Due to the modularity of the system, more features and capabilities can be implemented to extend its use to diverse applications and requirements, i.e. Wireless Sensor Networks + RFID.

In the following sections, first, the main contributions of this work are summarized followed by the possible future directions in order to extend the research and experiments.

6.1 Contributions

Based in the Internet of Things paradigm and specifically in a building block of this concept; the RFID technology, we proposed as main contributions the following:

- First, we proposed a framework that enables RFID and non-RFID devices to benefit from RFID information and access to related services linked by using a P2P network Overlay. In this way, we can implement RFID-based services that can be extended to different users. These users can be nodes with non-RFID capabilities but network interfaces instead (WLAN, ZigBee) and RFID mobile readers. For mobile readers, we presented a scheduling mechanism based on TDMA. It targets to reduce the interference caused by multiple readers working in parallel. We also consider into the architecture some security features like network authentication.

- Second, we presented a RFID network topology design based on Genetic Algorithms by using a multi-objective fitness function to evaluate the problem of deploying RFID readers into a square region. We considered six objectives such as minimize the overlapping of the RFID reader area, minimize the number of useless readers, maximize the number of tags covered, minimize the number of readers located out of the deployment area, minimize the number of redundant readers and minimize the number of tags located in overlapped reading areas. We also developed a customizable software tool to assist in the topology design and we performed measurements to compare its performance. We compared the process of finding solutions for different scenarios varying factors such as the propagation model and the genetic operators (crossover and selection).
- Finally, we addressed the issue of connecting modular devices in an embedded hardware platform to provide RFID-based applications. This platform will transmit the tag IDs read by a RFID reader and make them accessible through wireless network interfaces to users with non-RFID capability thus offering them to use RFID-based applications. The proposed system is flexible and modular allowing the extension and inclusion of additional features to the network like sensors, actuators and other components.

6.2 Future Directions

In order to extend the work presented in this thesis some possible future paths may include:

- Study alternative anti-collision methods in order to tackle the data-collision problem and the devices heterogeneity (diverse frequency bands, protocols and usages). These elements are key point when considering large-scale RFID network deployments.
- Evaluate the interactions of RFID with other devices like sensors and actuators in order to open new scenarios for the development and implementation of innovative applications.
- New additions to the Topology Design tool like more propagation models as well as directional antenna patterns, electromagnetic propagation patterns, three-dimensional (3D) deployment and electromagnetic interfering objects modeling can improve the application for more complex scenarios.
- Implement a large-scale network with heterogeneous devices as we stated in Chapter 3 utilizing a P2P approach to provide connectivity and resource sharing into a unified framework.
- Finally, additional experiments with embedded hardware will allow us to create a cheap-to-implement, stand-alone and flexible platform. This platform can address the inclusion of multiple heterogeneous devices such as RFID readers, sensors and actuators in a modular approach.

Chapter 7

Thesis' French Version

The next pages correspond to a French résumé of this thesis.

Résumé

Le paradigme de L'Internet des choses établit l'interaction et la communication avec une énorme quantité d'acteurs. Le concept n'est pas tout à fait nouveau; en fait, il combine un grand nombre de technologies et de protocoles et sûrement des adaptations des éléments préexistants pour offrir de nouveaux services et applications. Une des technologies clés de l'Internet des objets est l'identification par radiofréquence abrégée en anglais RFID («*Radio Frequency Identification*»). La RFID est une technologie bien connue qui est employée avec succès dans de nombreuses applications. Elle a été introduite dans les années 50, et présente des avantages par rapport à son prédécesseur, le code de barres, car elle nécessite de petits dispositifs électroniques pour transmettre un code d'identification (ID) et ne nécessite pas de ligne de vue pour fonctionner. Cette technologie propose un ensemble de solutions qui permettent le suivi et la traçabilité des personnes, des animaux et pratiquement n'importe quel objet en utilisant des liaisons sans fil. L'architecture RFID est composée d'étiquettes RFID dites Tags, qui fournissent un code d'identification, de lecteurs RFID, qui interrogent et obtiennent grâce aux ondes électromagnétiques ce code, et un logiciel médiateur («*Middleware*») qui fournit une plateforme pour interpréter et utiliser l'information recueillie.

En considérant le concept de l'Internet des choses, plusieurs technologies doivent être liées afin de fournir des interactions qui conduisent à la mise en œuvre de services et d'applications. Le défi est que ces technologies ne sont pas nécessairement compatibles et conçues pour fonctionner ensemble. De la même manière, une technologie peut avoir différentes variantes qui ajoutent plus de complexité à la conception du système. C'est le cas de la RFID, où nous pouvons trouver différentes fréquences d'opérations, de divers protocoles de communication et de normes. Dans un environnement multi dispositif et multi technologie, ces défis imposés doivent être abordés afin de fournir une plateforme d'information unifiée.

Dans ce contexte, l'objectif principal de cette thèse est de concevoir un « *framework* » hétérogène qui permettra l'interaction de divers dispositifs tels que la RFID, des capteurs (dispositifs qui obtiennent des mesures de variables de l'environnement) et des actionneurs (matériel qui peut effectuer des actions physiques telles que les moteurs ou relais électriques) afin de fournir de nouvelles applications et de services. Nous portons une attention particulière à l'intégration de dispositifs sans capacités RFID dans ce *framework*, leur permettant d'obtenir les informations fournies par les tags à l'aide d'une interface réseau soit filaire ou sans fil. À cet effet, dans ce travail, notre première contribution est la conception et l'analyse d'une architecture d'intégration pour les dispositifs hétérogènes (sans et avec la technologie RFID) définissant dans le détail l'ensemble de ses éléments constitutifs, les interactions et l'échange de messages. Nous avons évalué ses performances en utilisant des simulations afin de vérifier son applicabilité et faisabilité. Dans la seconde contribution, nous proposons un modèle d'évaluation de la topologie RFID et un outil d'optimisation qui aide dans le processus de planification de réseaux de cette technologie. Une fonction multi objective d'évaluation et des algorithmes génétiques ont été combinés dans une application logicielle développée et une description détaillée et des tests sont également fournis. Enfin, dans notre dernière contribution, nous avons implémenté une version simplifiée du *framework* en utilisant du matériel embarqué et indicateurs de performance sont fournis ainsi que la configuration détaillée de la plateforme de test.

Mots-clés:

RFID, hétérogénéité, algorithmes génétiques, Peer to Peer, Internet des objets, optimisation de la topologie de réseau RFID, matériel embarqué, Microcontrôleurs, WLAN, ZigBee.

Chapitre 1

Introduction

La technologie RFID a été utilisée depuis les années 50 afin d'identifier des objets à l'aide des liaisons sans fil. Pourtant, de nos jours un grand et toujours croissant nombre d'applications a été proposé afin de profiter des avantages de ce qu'elle propose. Essentiellement, son architecture est assez simple. Elle est basée sur des transpondeurs (tags) qui fournissent un code d'identification (ID), de lecteurs qui obtiennent ce code à partir de tags et enfin un middleware qui interprète, analyse et fournit l'ensemble des identifiants recueillis servant à des fins spécifiques (Figure 1). Cette technologie peut être utilisée pour suivre et tracer des personnes, des animaux, et des objets. Certaines applications incluent la logistique d'entrepôts, les documents personnels (passeport, permis de conduire), les passes de transport public, le contrôle d'accès aux bâtiments, et elle est un des principaux piliers du paradigme de l'Internet des Objets [1].

Figure 1. Architecture de base RFID

Dans ce travail de thèse, nous avons examiné la problématique de l'intégration des dispositifs hétérogènes dans un framework unique basée sur les RFID. Hétérogénéité dans le contexte de cette thèse est liée aux différents dispositifs qui peuvent être connectés et communiquer dans le framework, par exemple de dispositifs avec ou sans la technologie RFID avec des interfaces réseau variées. Il comprend les lecteurs RFID qui travaillent dans différentes bandes de fréquences, des capteurs et des actionneurs. Nous avons également étudié la conception de la topologie des réseaux RFID et étudié l'optimisation de la topologie en utilisant la méthode des algorithmes génétiques. Similairement, nous avons développé un outil logiciel aidant à la découverte de la topologie optimale. Finalement, nous avons mis en œuvre une plateforme simple basée sur du matériel embarqué qui intègre la RFID, des interfaces de réseaux sans fil et de capteurs.

1.1 Les objectifs et les défis recherche

Dans un contexte d'intégration multi technologie, multi diapositive comme celui proposé par l'Internet des objets, l'un des principaux défis est de permettre le fonctionnement de différents dispositifs orchestrés dans un but bien défini, par exemple pour fournir des services d'information ou automatiser certaines tâches. Pour atteindre cet objectif, l'une des étapes consiste à définir et concevoir l'architecture d'intégration qui prendra en compte les multiples possibilités de communication de chaque technologie et qui va centraliser certaines fonctions telles que la gestion et la supervision du réseau.

Les principaux objectifs de cette thèse sont les suivants. Tout d'abord, de concevoir un framework unique basé sur la RFID et les capacités de communication IP. Cela va permettre l'intégration des différents dispositifs (multiples lecteurs RFID et appareils sans RFID) dans le but de fournir des

services et des applications qui nécessitent l'information qui RFID propose. Nous avons besoin de définir et de préciser les différents protocoles nécessaires pour le fonctionnement du framework et de décrire en détail ses éléments constitutifs. Il faut remarquer que l'utilisation de plusieurs lecteurs RFID peut provoquer l'effet négatif de la collision des données, donc ce problème doit être abordé ainsi. Deuxièmement, étant donné que le framework aura plusieurs lecteurs RFID, nous avons besoin de faire face au problème de la planification du réseau RFID, qui exige un déploiement efficace et efficient des lecteurs. Ceci est nécessaire afin de maximiser la lecture des ID qui provient des tags et de minimiser le nombre de lecteurs RFID à être installé entre autres paramètres d'optimisation comme maximiser la zone de couverture et de réduire des interférences. Finalement, les simulations du framework et l'utilisation du matériel réel peuvent fournir une meilleure vision et complète de toute l'architecture. Par conséquent, nous avons procédé à la mise en œuvre d'une version simplifiée du framework en utilisant du matériel embarqué pour implémenter certaines fonctionnalités du framework conçu.

1.2 Méthodologie et contributions

Afin d'atteindre les objectifs identifiés, nous avons procédé de la façon suivante:

- Tout d'abord, nous avons étudié l'état de l'art lié à la RFID. Elle a permis d'avoir une vision large des points forts et des limites de cette technologie. Nous avons contribué avec le chapitre de livre suivant:

  - Oscar Botero & Hakima Chaouchi, Chapter 5, *On RFID and research issues*. Chaouchi Hakima, the Internet of Things: Connecting Objects, Wiley-ISTE, 2010.

- Deuxièmement, nous avons conçu un framework pour les dispositifs hétérogènes basés sur la RFID. Il permet l'intégration de lecteurs RFID (fixe et mobile), et des dispositifs sans technologie RFID afin de fournir des services basés sur cette technologie. Le framework est conçu pour fonctionner avec plusieurs lecteurs RFID déployés. Nous avons considéré le problème d'interférence de plusieurs lecteurs qui travaillent simultanément. Nous avons pris en compte certains problèmes de sécurité comme l'authentification réseau. Notre conception du framework repose sur une couche P2P (Peer-to-Peer) pour faciliter la gestion et l'accès à l'information entre les différentes entités. La définition des différents protocoles nécessaires est présentée afin de fournir des services d'information basés sur le RFID. Des simulations ont été également exécutées pour obtenir des mesures de performance. Nous avons contribué avec le papier de recherche suivante:

  - Oscar Botero & Hakima Chaouchi, *P2P Framework for RFID enabled and non-enabled users*. IoTs 2010, Hangzhou China.

- Troisièmement, nous avons étudié le problème d'optimisation de la topologie du réseau RFID. Nous avons proposé une fonction d'évaluation ou également appelée la fonction de « fitness » multi objective, fondée sur six objectifs individuels (zone de couverture, le nombre de tags détectés, la réduction des interférences, entre autres) qui nous permet de trouver des solutions liées au déploiement de lecteurs RFID d'une façon efficace et efficiente. La méthode d'optimisation employée c'est les algorithmes génétiques. Nous avons développé un outil logiciel qui permet de trouver la disposition des lecteurs à être déployée dans une région donnée. Ces efforts ont été publiés dans l'article ci-dessous:

  - Oscar Botero & Hakima Chaouchi, *RFID network topology design tool based on Genetic Algorithms*. RFID-TA Barcelona, Spain 2011.

- Enfin, nous avons implémenté une version simplifiée du framework en utilisant du matériel embarqué. Des Microcontrôleurs, RFID, Wi-Fi et ZigBee ont été combinés pour fournir information basée sur la RFID pour les utilisateurs sans cette technologie, mais avec les interfaces réseau mentionnées. Nous avons effectué des mesures liées à la performance de la plateforme de test et nous montrons sa conception et sa mise en œuvre. Nous avons contribué avec les articles suivants:

  - Oscar Botero & Hakima Chaouchi, *RFID for non-RFID users Embedded Hardware Implementation*. ANT Niagara Falls, Canada 2011.

  - Journal: Oscar Botero & Hakima Chaouchi, *Radio Frequency Identification framework for heterogeneous nodes.* Personal and Ubiquitous Computing. Springer 2012.

  - A. Ait Wakrim, O. Botero, K. Raymond, H. Chaouchi "TRACK-IoT: Heterogeneous IoT Network", Research Report, Telecom Sud Paris March 2012.

Le manuscrit de thèse est organisé comme suit. Le chapitre 2 propose la terminologie et les concepts généraux RFID suivis par le chapitre 3, où le framework RFID hétérogène est décrit en détail. Dans le chapitre 4, nous introduisons les concepts de base sur l'optimisation axée principalement dans la méthode des Algorithmes génétiques. Ensuite, nous suivons avec la conception de la topologie des réseaux RFID, la description du modèle utilisé pour évaluer les solutions, l'outil développé ainsi que des indicateurs de performance obtenus. Dans le chapitre 5, nous présentons la mise en œuvre de la plateforme de test avec du matériel embarqué. Enfin, nous fournissons les conclusions obtenues et établissons les futures directions de recherche.

Résumé Chapitre 2 Concepts RFID et Terminologie

Dans ce chapitre, nous avons présenté les concepts de base liés à la RFID. Nous avons décrit les différentes composantes que configurent cette technologie, les lecteurs, les étiquettes dites tags et le middleware (Figure 2).

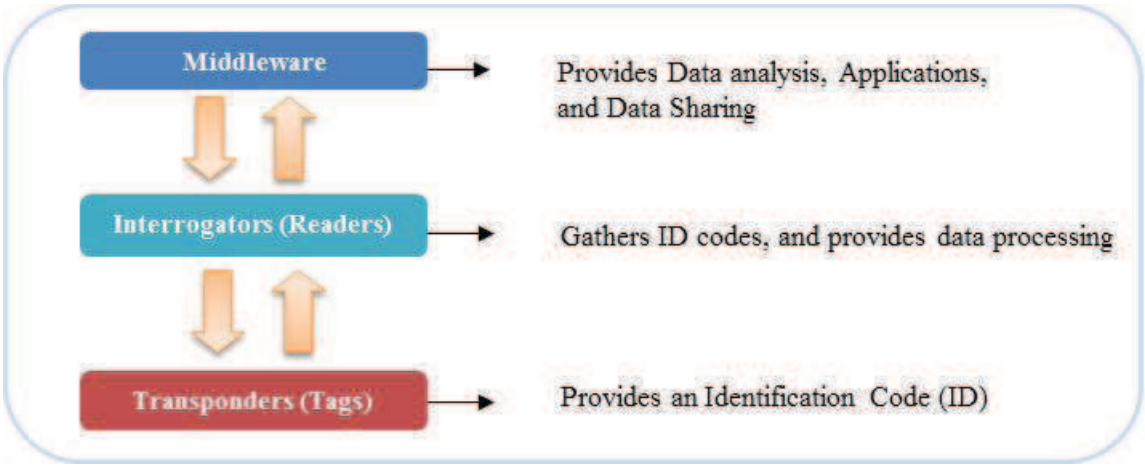


Figure 2. Architecture RFID: tags, lecteurs et middleware

Les étiquettes portent un code d'identification qui peut être récupéré par les lecteurs RFID. Elles sont composées par une d'antenne et un circuit intégré (Figure 3). L'antenne capte l'énergie du lecteur et la transmet à la puce électronique, afin de la faire fonctionner. La puce est en charge de toutes les tâches logiques liées à la transmission du code d'identification et l'exécution de commandes [6].

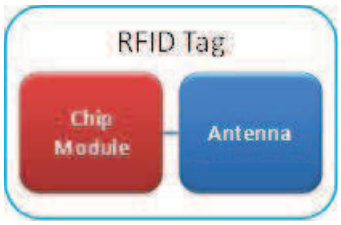


Figure 3. Tag RFID.

Comme a été mentionné précédemment, les lecteurs RFID sont responsables d'interroger les tags afin d'obtenir le code d'identification qu'elles portent. Ils sont constitués de trois composants principaux: un module de contrôle, un module de radio fréquence et un système d'antennes [7] (Figure 4).

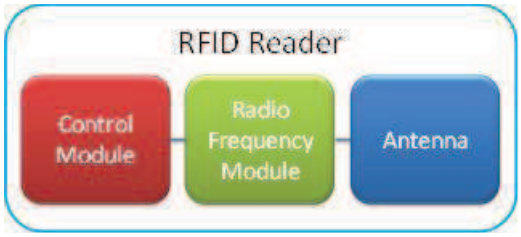


Figure 4. Lecteur RFID.

Le Middleware est un composant logiciel qui recueille et traite les données obtenues auprès des lecteurs RFID. Il fournit également des fonctions de gestion et de supervision du réseau RFID [9]. Les fonctionnalités standard de l'intergiciel RFID sont le filtrage des données, l'agrégation des données, des capacités de gestion et de suivi [10] (Figure 5).

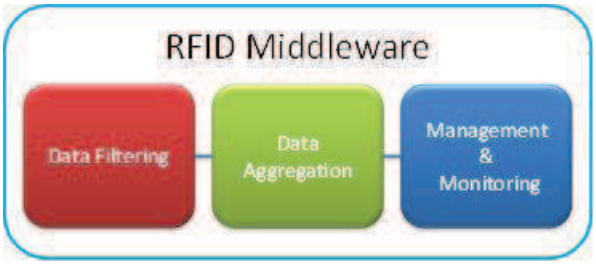


Figure 5. Middleware RFID.

Le filtrage des données fournit un moyen de réduire la quantité d'information traitée. L'agrégation des données permet de construire un référentiel de l'information obtenue qui sera traitée ou analysée. Les fonctionnalités de gestion mettent en place un moyen d'interagir avec le réseau RFID afin de contrôler et d'optimiser son fonctionnement. Enfin, le suivi permet de garder une trace de l'activité, l'utilisation et les performances du réseau.

La technologie RFID a attiré l'attention au cours de cette dernière décennie en particulier dans la construction de l'Internet des Objets et sa large utilisation pour le suivi et la traçabilité des objets. Cependant, il y a des questions techniques qui font encore l'objet de recherches afin d'améliorer certains aspects de cette technologie. Le problème de la collision et de l'hétérogénéité des dispositifs (diverses bandes de fréquences, des protocoles et des usages) sont des éléments clés lors de déploiements de multiples lecteurs RFID. Similairement, les interactions entre la RFID et d'autres dispositifs tels que les capteurs et les actionneurs ouvrent de nouvelles perspectives et permettront le développement et la mise en œuvre d'applications innovantes.

Résumé Chapitre 3 Framework RFID pour nœuds hétérogènes

Dans ce chapitre, nous proposons un framework qui permet aux dispositifs avec et sans RFID de se bénéficier d'une prise en charge par notre framework hétérogène visant à donner l'accès à des services d'information. Ce framework fournira des services d'information basés sur la RFID et d'autres dispositifs tels que des capteurs et des actionneurs. Également, Il offre un mécanisme d'anticollision de données basé sur TDMA afin de réduire l'interférence lecteur-lecteur. Les dispositifs additionnels incluent des fonctions de services d'authentification et AAA (authentification, autorisation et comptabilité (« *accounting* »). Le framework est constitué des nœuds suivants (Figure 6).

- **Multi-Mode-Node (MMN):** Ce nœud est l'interface avec les dispositifs hétérogènes. Il fournira les ports nécessaires pour connecter des lecteurs RFID, capteurs et actionneurs ainsi que des interfaces de réseau sans fil (WLAN, Ethernet et ZigBee).
- **Management and Authenticator Node (MAN):** Il exerce des fonctions AAA et la gestion de ressources.
- **Users**: Ils sont des nœuds qui demandent de l'information. Ils sont regroupés en
  - Les appareils sans capacités de lecture RFID nommés nœud virtuel (VRN)
  - Des Lecteurs Mobiles RFID (MRN)
- Des capteurs, des actionneurs et des tags RFID.

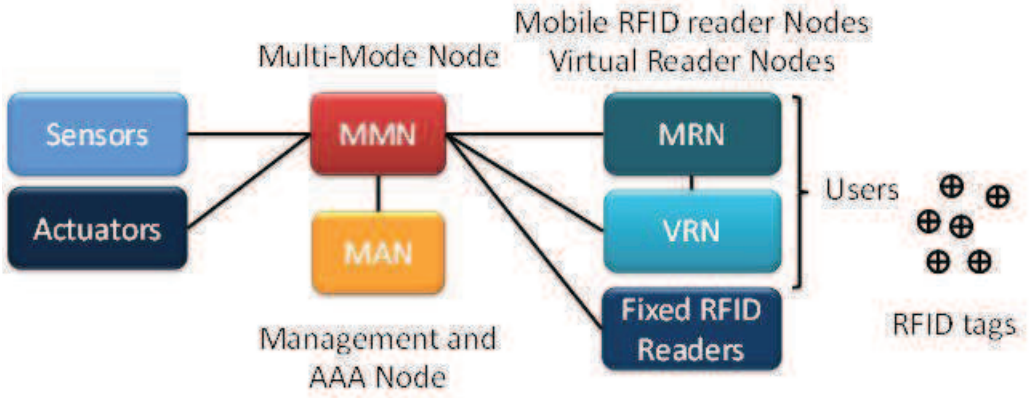


Figure 6. Framework Hétérogène RFID.

De cette façon, nous pouvons mettre en œuvre des services basés sur la RFID qui peuvent être étendus à différents utilisateurs. Ces utilisateurs peuvent être des nœuds avec et sans interfaces RFID, mais avec d'interfaces réseau (WLAN, ZigBee) et les lecteurs RFID mobiles. Pour les lecteurs mobiles, nous avons présenté un mécanisme pour réduire la collision de données basée sur TDMA. Il vise à diminuer l'interférence causée par plusieurs lecteurs qui travaillent simultanément (Figure 7).

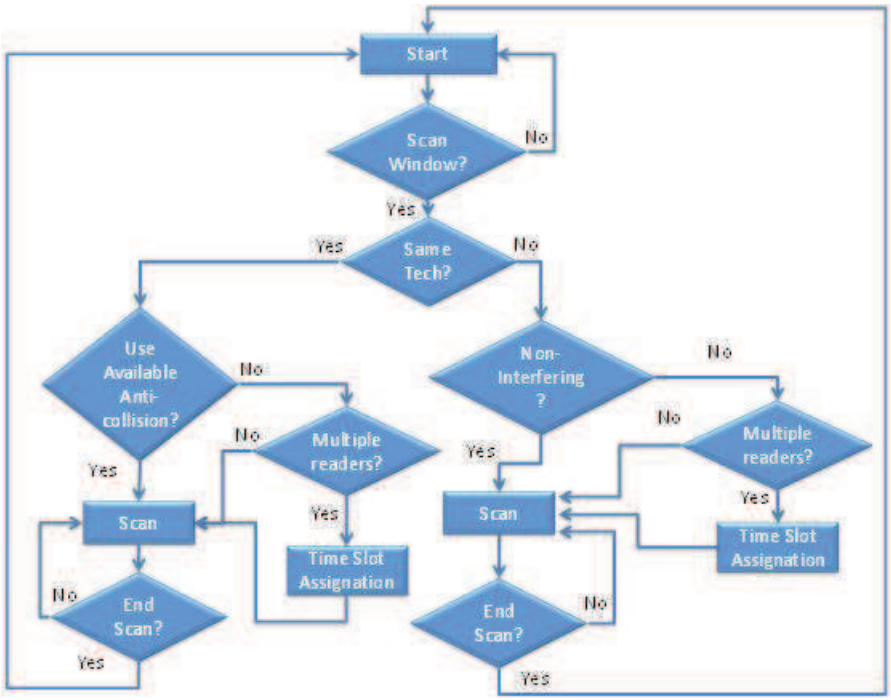


Figure 7. Mécanisme anticollision pour lecteurs RFID.

Nous considérons aussi dans l'architecture certains éléments de sécurité comme l'authentification réseau. Nous décrivons une conception modulaire, ses caractéristiques et ses protocoles d'échange de messages. L'analyse des performances montre l'avantage d'effectuer une lecture proactive des étiquettes RFID et le partage de la liste recueilli via la couche P2P. Cette caractéristique réduit la charge dans les MMN pour le fait que les utilisateurs authentifiés peuvent partager les informations qu'ils obtiennent avec d'autres utilisateurs. Identiquement, même les lecteurs RFID mobiles peuvent prendre avantage de cette solution, car ils peuvent obtenir les listes mises à jour de tags déployées directement à partir de MMN. Néanmoins, s'ils veulent lire les étiquettes par eux-mêmes, la plateforme fournit un mécanisme pour réduire les interférences mentionnées auparavant. D'autres travaux peuvent inclure la mise en œuvre du framework en utilisant des dispositifs réels.

Résumé Chapitre 4 Conception de la topologie RFID basée sur des algorithmes génétiques

Dans ce chapitre, nous avons présenté une conception de topologie des réseaux RFID basée sur des algorithmes génétiques. Nous avons proposé une fonction de « fitness » multi objective (équation 1) afin d'évaluer le problème du déploiement de lecteurs RFID dans une zone carrée (Figure 8).

$$f_T = \sum_{i=1}^{6} w_i * f_i \ , \qquad where: \sum_{i=1}^{6} w_i = 1 \tag{1}$$

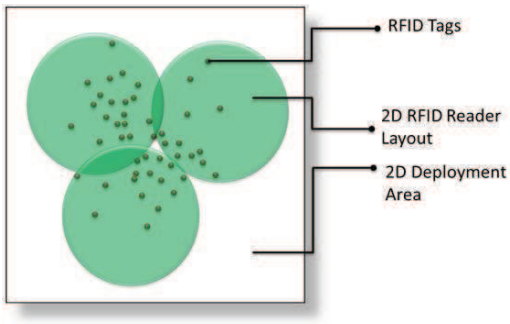


Figure 8. Déploiement de lecteurs RFID dans une zone carrée.

Nous avons considéré six objectifs pour minimiser la superposition de la zone de lecture RFID, de minimiser le nombre de lecteurs superflus, de maximiser le nombre de tags couverts, de minimiser le nombre de lecteurs situés hors de la zone de déploiement, de réduire le nombre de lecteurs redondants et de réduire le nombre de tags situés dans les zones de lecture superposées.

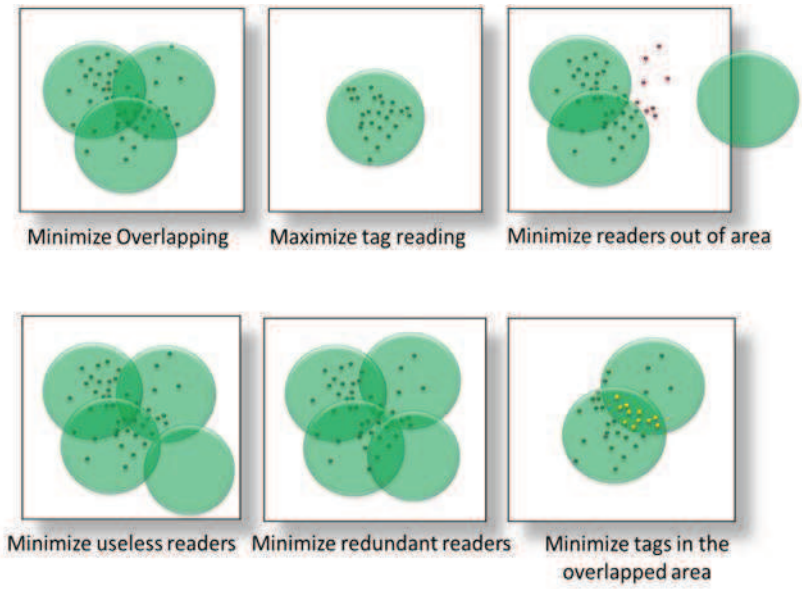


Figure 9. Multi objectives.

Nous avons également développé un outil logiciel (Figure 10, Figure 11) pour aider à la conception de la topologie et nous avons effectué des mesures pour comparer ses performances.

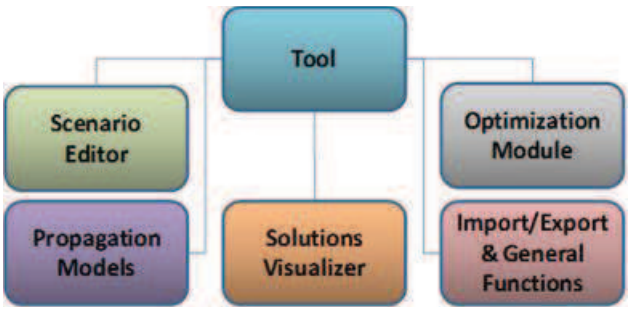


Figure 10. Structure modulaire de l'outil d'optimisation.

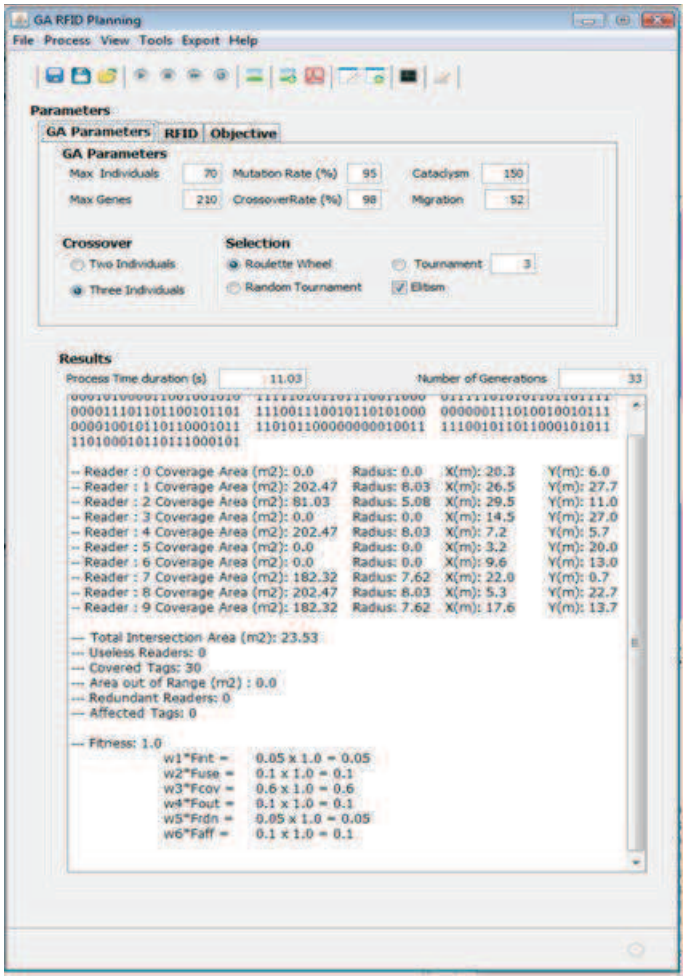


Figure 1191. Interface graphique de l'outil d'optimisation.

Nous avons comparé le processus de trouver des solutions pour différents scénarios, deux modèles de propagation et avec variations dans les opérateurs génétiques (croisement et sélection) (Figures 12, 13,14).

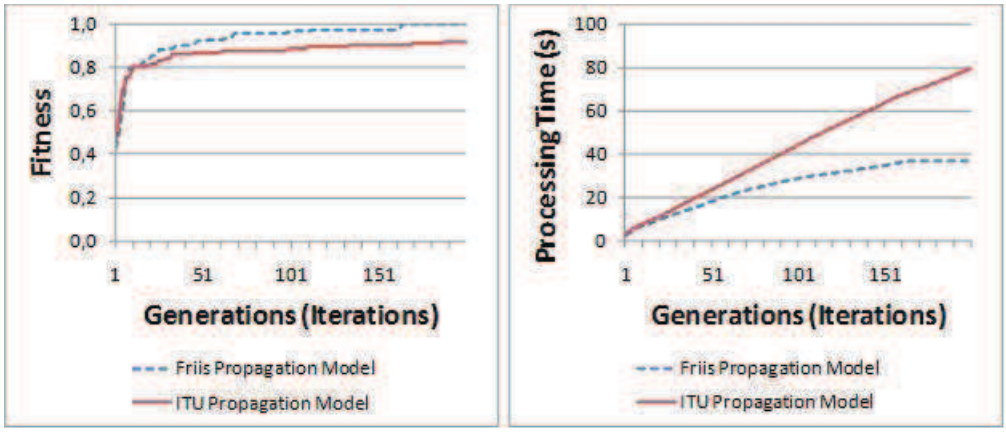


Figure 12. Générations vs. Fitness et Delay de calcul pour les modèles Friis et ITU.

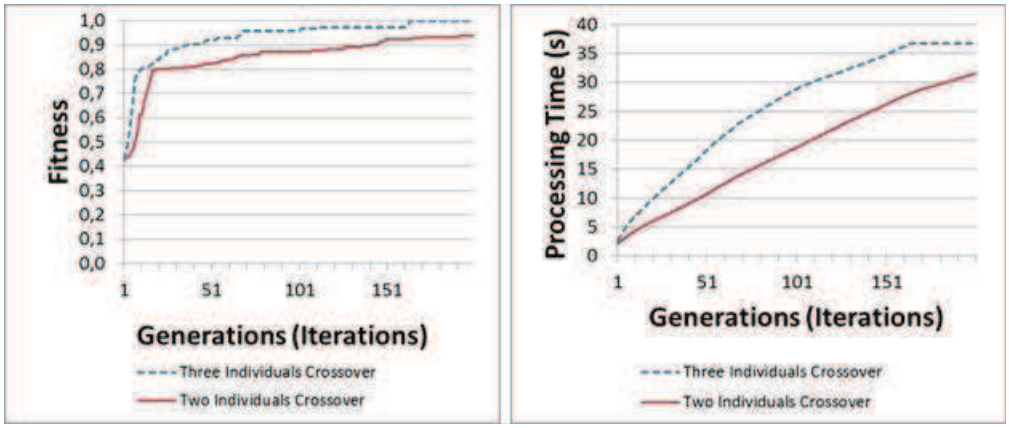


Figure 13. Générations vs. Fitness et Delay de calcul pour le modèle Friis avec "2" et "3" parents.

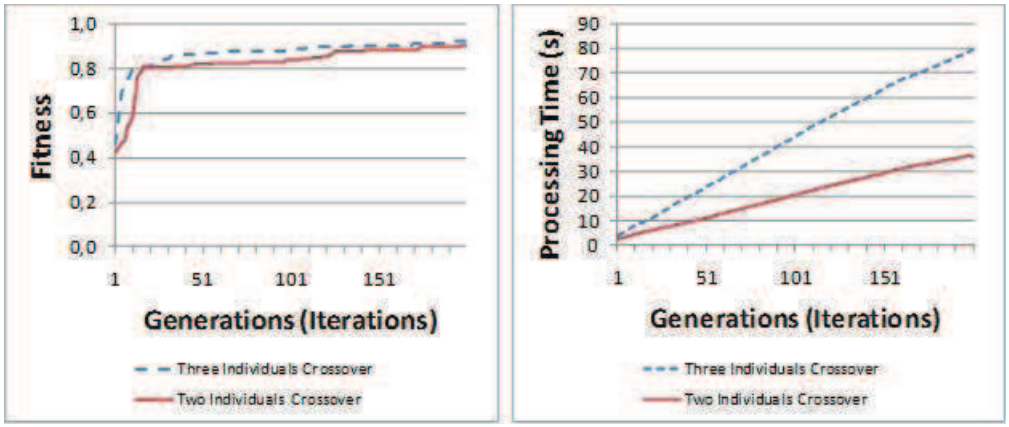


Figure 14. Générations vs. Fitness et Delay de calcul pour le modèle ITU avec "2" et "3" parents.

Nous n'avons observé que l'utilisation de l'opérateur de croisement de trois parents présente de meilleurs résultats que l'approche classique de deux parents, indépendamment du modèle de propagation utilisé. Aussi, si l'on exclut le sixième objectif (réduire au minimum de tags dans les zones de lecture superposées), nous obtenons en moyenne de temps 26% de moins dans le but de trouver une solution satisfaisante. Dans ce cas nous supposons que le réseau RFID a un système générique d'anticollision qui garantira cet objectif. Des nouvelles additions à l'outil comme plus des modèles de propagation ainsi que de modèles d'antenne directionnelles, un plan en trois dimensions (3D) du déploiement peuvent améliorer l'application pour la simulation des scénarios plus complexes.

Résumé Chapitre 5 Extension du service RFID pour les nœuds sans interface RFID: mise en œuvre sur matériel embarque.

Dans ce chapitre, nous avons abordé la connexion de dispositifs modulaires dans une plateforme du matériel embarqué pour fournir des applications basées sur la technologie RFID (Figure 15).

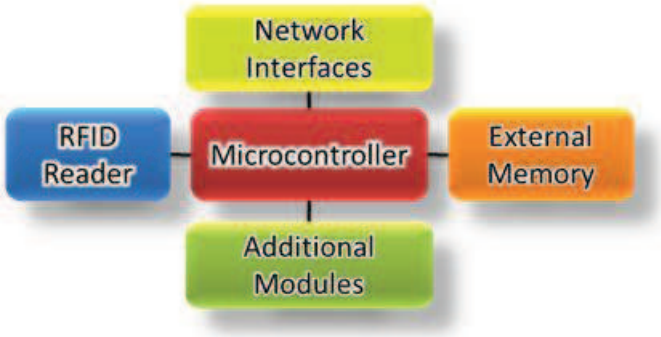


Figure 15. Modules du système embarqué.

Cette plateforme peut transmettre les ID des tags lus par un lecteur RFID et les rendre accessibles via des interfaces de réseau sans fil pour les utilisateurs qui n'ont pas de capacités RFID. Le système proposé est flexible et modulaire (Figure 16) permettant l'extension et l'inclusion de fonctionnalités supplémentaires comme des capteurs, des actionneurs et autres composants.

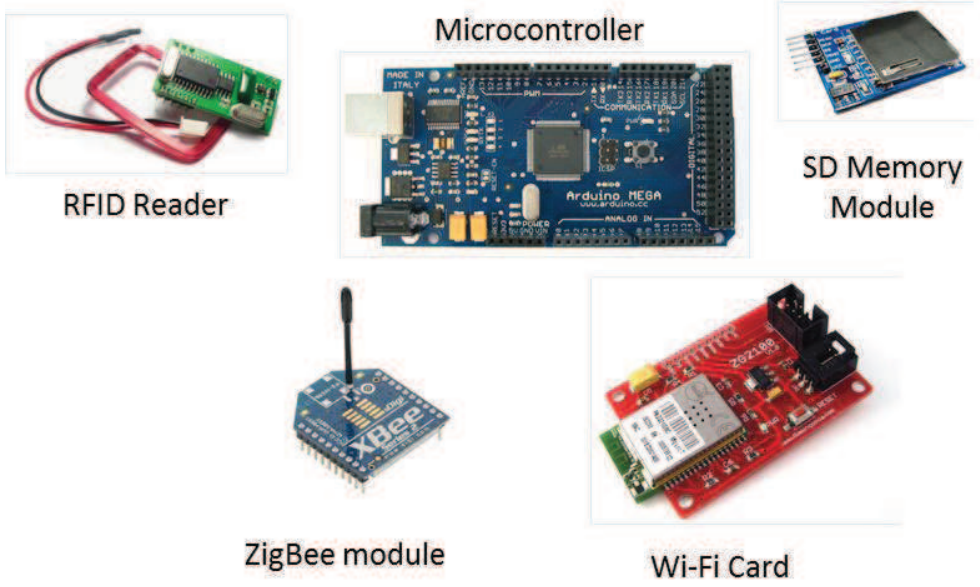


Figure 16. Modules réels utilisés.

Même avec les limitations imposées par le matériel embarqué utilisé (mémoire, processeur, vitesse), nous avons pu implémenter des fonctionnalités intéressantes comme un serveur web (Figure 17) autonome qui peut publier des ID de tags RFID en tant que contenu, pour être accessible pour de dispositifs portables Wi-Fi. Le diagramme fonctionnel est représenté dans la Figure 18.

Figure 17. Serveur web.

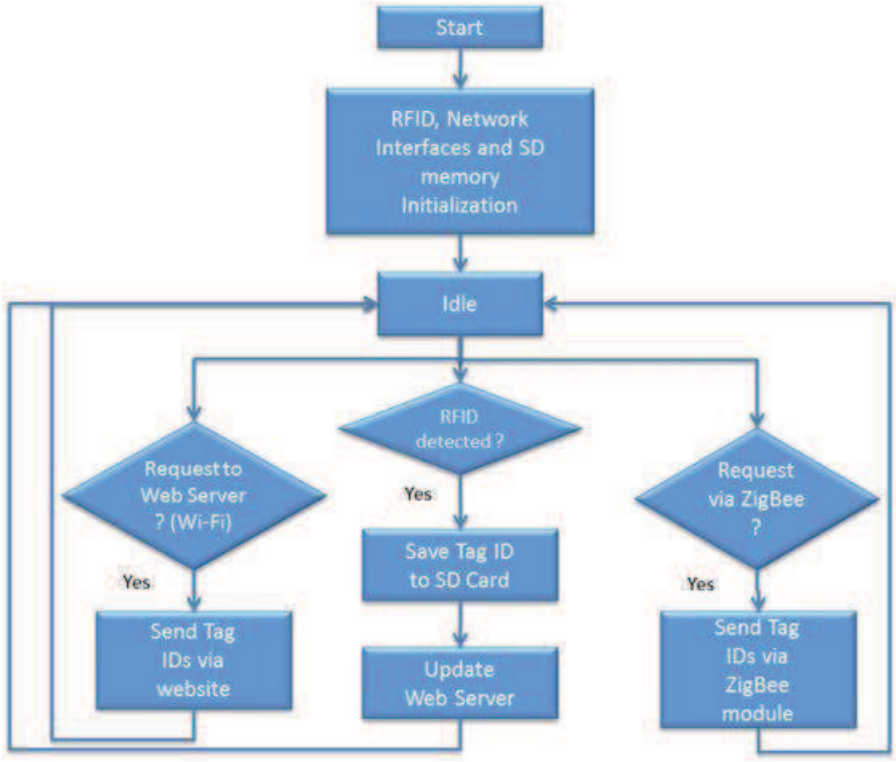


Figure 18. Diagramme fonctionnel du système.

En ce qui concerne les mesures de délai, nous observons qu'ils n'imposent pas de restrictions pour des applications pratiques. Nous avons effectué des tests en utilisant le logiciel analyseur de protocoles Wireshark [104], afin de mesurer le temps qu'un utilisateur doit attendre pour obtenir la liste des ID de tags disponibles. Une différence en relation avec la description du système dans le chapitre 3, c'est que dans ce cas nous n'avons pas de retard associé à la configuration initiale ou d'authentification décrite précédemment. Les résultats sont présentés dans la figure suivante (Figure 19) et dans le Tableau 1. Nous avons mesuré le retard pour 1, 10,100 et 200 IDs et pour la valeur maximale, on obtient un délai d'environ 1,6 secondes.

Tableau 1. Retard de transmission des IDs.

Number of Tags	*1*	*10*	*100*	*200*
Average Time(s)	0.45	0.66	1.11	1.57
Standard Deviation	0.01	0.01	0.02	0.01
Confidence Interval	0.0012	0.0015	0.0035	0.029

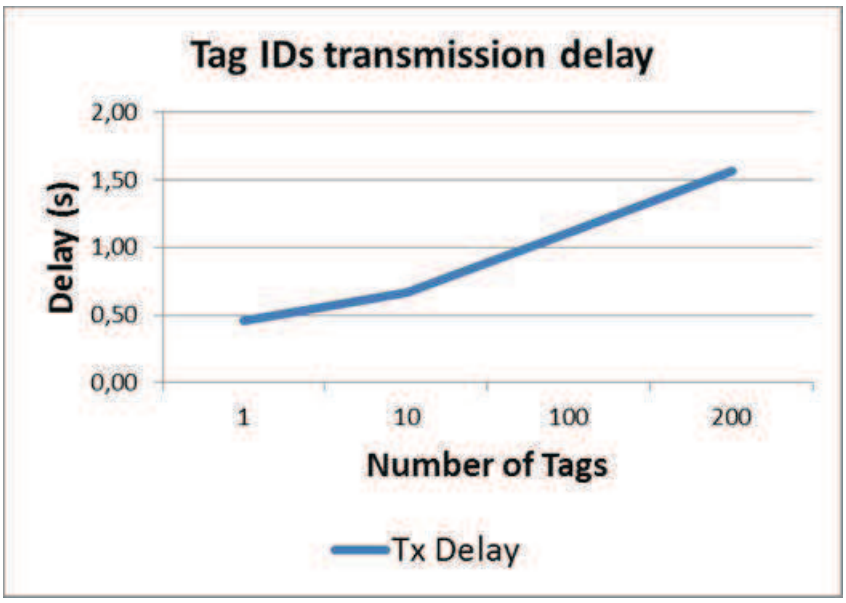


Figure 19. Retard de transmission des IDs.

L'expérience suivante a été mise en place avec la configuration montrée dans la Figure 20, afin de mesurer le délai nécessaire pour transmettre des IDs en utilisant des modules ZigBee.

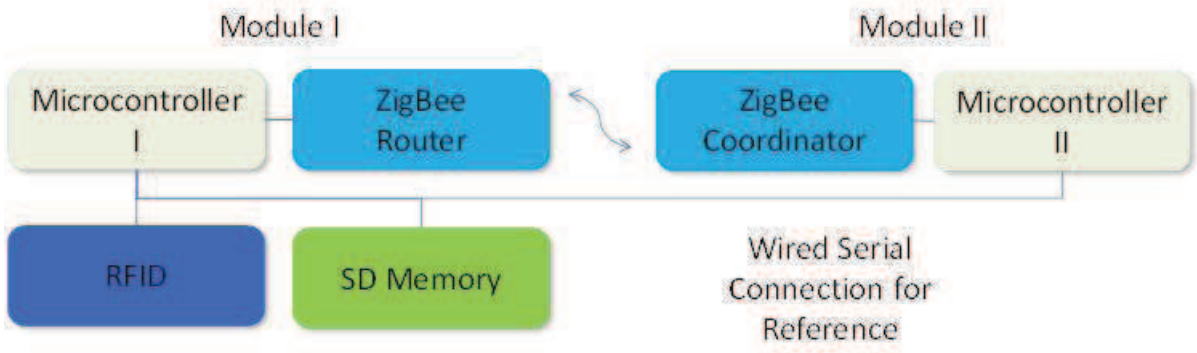


Figure 20. Configuration pour mesurer le retard en utilisant ZigBee.

Nous avons utilisé la même quantité d'IDs que pour les expériences précédentes : 1, 10.100 et 200 IDs. Nous avons établi une comparaison de référence en mesurant le retard, mais au lieu de transmettre par l'intermédiaire des modules ZigBee, nous avons utilisé une transmission filaire série entre les deux microcontrôleurs. Nous avons mis une vitesse de 9600 bps. Les nœuds sont situés à 1 m les uns des autres et avec ligne de vue. Les résultats sont présentés dans le Tableau 2 et la Figure 21.

Tableau 2. Retard de transmission connexion série et ZigBee.

Number of Tags	*1*	*10*	*100*	*200*
Average Time(s) Wired	0.11	0.18	0.94	1.78
Average Time(s) ZigBee	0.11	0.18	0.98	1.98

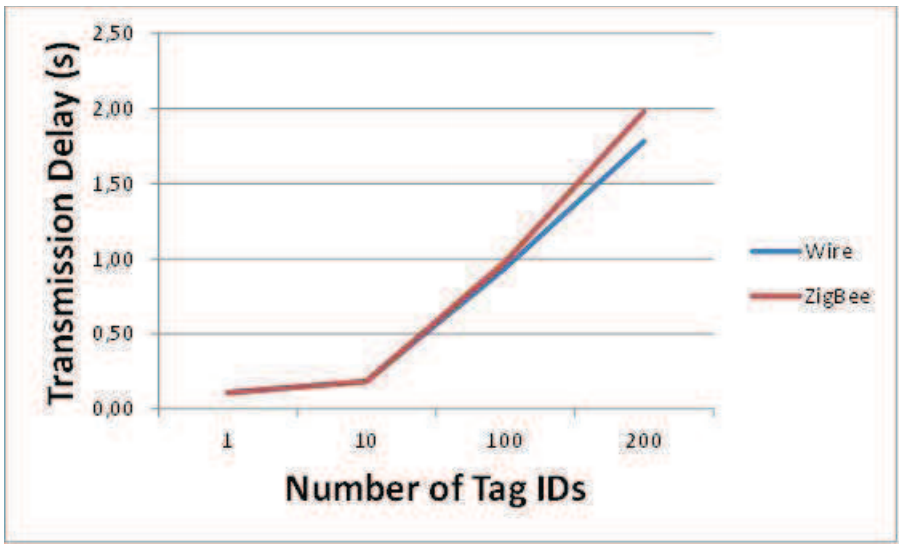


Figure 21. Retard de transmission connexion série et ZigBee.

Nous observons qu'il y a une légère augmentation (en utilisant le module ZigBee) dans le délai lorsque le nombre de tags augmente par rapport au même nombre de tags transmis en utilisant la configuration sérial. Toutefois, pour la taille de l'information que nous transmettons, la différence n'est pas assez importante, peu importe l'interface que nous utilisons. Enfin, nous avons créé un scénario avec deux nœuds qui partagent certaines fonctionnalités. Dans ce cas, nous avons un nœud qui va obtenir les identifiants RFID et les transmettre en utilisant le module ZigBee à un deuxième nœud qui veut être en charge d'une interface sans fil pour d'autres usagers avec des périphériques portables comme les téléphones mobiles ou des pads (Figure 22). Le délai de service est présenté dans le Tableau 3.

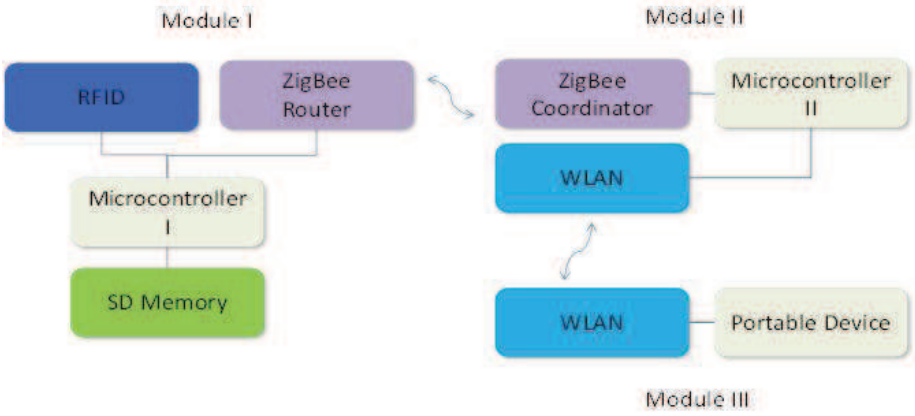


Figure 22. Configuration des nœuds.

Tableau 3. Délai de service.

Number of Tags	*1*	*10*	*100*	*200*
Delay(s)	0.56	0.84	2.05	3.55

La prochaine étape sera de mettre en place un réseau à grande échelle de dispositifs hétérogènes comme nous l'avons indiqué dans le chapitre 3.

Chapitre 6

Conclusions

Dans cette thèse, nous avons présenté nos travaux de recherche basés sur la technologie RFID en vue d'apporter des améliorations au déploiement et l'utilisation de réseaux de cette technologie. Bien que la RFID ne soit pas tout à fait nouvelle, il y a encore des problématiques de recherche qui ont besoin d'être abordées dans le but de fournir des solutions aux problèmes existants.

Nous avons présenté nos contributions relatives à la proposition d'un « *Framework »* RFID pour les nœuds hétérogènes, ainsi qu'un modèle de la topologie du réseau afin d'aider à la mise en œuvre et la conception de cette plateforme. Nous avons utilisé des outils de simulation pour observer le comportement du système concernant le délai de transmission. Les résultats indiquent que pour de grandes quantités d'identifiants (ID) notre approche est plus efficace par rapport à la TIS (vitesse d'identification Tag). Un résultat important c'est que le système permet aux nœuds sans matériel RFID de profiter des avantages de l'information fournie par les tags RFID déployés et détectés par un nœud proxy. Il est également intéressant de remarquer que les dispositifs avec les lecteurs RFID peuvent étendre leur zone de lecture "virtuellement" en se connectant à la couche P2P de la plateforme.

Également, nous avons mis en œuvre un modèle de topologie RFID et un logiciel pour nous aider dans la conception du réseau. Il a été développé en utilisant Java SE et en utilisant des algorithmes génétiques comme la méthode d'optimisation. Différentes fonctionnalités sont mises en œuvre, y compris l'exportation et l'importation de données, les variations des paramètres et l'édition des scénarios.

Dans la dernière partie de ce travail, nous avons fourni la conception et la mise en œuvre d'un service d'information RFID en utilisant du matériel embarqué. Il offre des identifiants de tags aux utilisateurs connectés via une interface sans fil commune (802.11/802.15.4). Ce système fournit une plateforme d'intégration dans les réseaux RFID pour les périphériques qui n'ont pas de matériel approprié (lecteur RFID). Grâce à la modularité du système, plus de fonctionnalités et des capacités peuvent être mises en œuvre pour diverses applications et exigences, par exemple les réseaux de capteurs sans fil + RFID.

Dans les sections suivantes, d'abord, les principales contributions de ce travail sont résumées et enfin, nous indiquons les directions futures possibles, afin d'étendre la recherche et l'expérimentation sur les sujets traités.

6.1 Contributions

Basées sur le paradigme de l'Internet des Objets, et plus précisément sur un bloc basique de ce concept, la technologie RFID, les principales contributions proposées dans cette thèse sont:

- Tout d'abord, nous avons proposé un « framewok » qui permet aux dispositifs RFID et non RFID pour bénéficier de l'information que la RFID apporte et l'accès à des services d'information à l'aide d'un réseau P2P. De cette façon, nous pouvons mettre en œuvre des services basés sur cette technologie qui peuvent être étendus à différents

utilisateurs. Ces utilisateurs peuvent être des nœuds avec et sans capacités RFID, mais avec d'interfaces réseau (WLAN, ZigBee) et les lecteurs RFID mobiles. Pour les lecteurs mobiles, nous avons présenté un mécanisme pour réduire la collision de données basée sur TDMA. Il vise à réduire l'interférence causée par les lecteurs en travaillant en parallèle. Nous considérons aussi dans notre architecture certains éléments de sécurité comme l'authentification réseau.

- Deuxièmement, nous avons présenté des recherches sur la conception de la topologie de réseau RFID basée sur des algorithmes génétiques. Nous avons utilisé une fonction de « fitness » multi objectif afin d'évaluer le problème du déploiement de lecteurs RFID dans une zone carrée. Nous avons considéré six objectifs pour minimiser la superposition de la zone de lecture RFID, de minimiser le nombre de lecteurs superflus, de maximiser le nombre de tags couverts, de minimiser le nombre de lecteurs situés hors de la zone de déploiement, de réduire le nombre de lecteurs redondants et de réduire le nombre de tags situés dans les zones de lecture superposées. Nous avons développé également un outil logiciel pour aider à la conception de topologie et nous avons effectué des mesures et comparé ses performances.
- Finalement, nous avons travaillé dans la mise en œuvre d'une plateforme embarquée simplifiée pour fournir des applications basées sur RFID. Cette plateforme peut communiquer les ID des tags enregistrés par un lecteur RFID et les rendre accessibles via des interfaces de réseau sans fil. Le système proposé est flexible et modulaire permettant l'extension et l'inclusion de fonctionnalités supplémentaires comme les capteurs, les actionneurs et autres composants.

6.2 Orientations futures

Afin d'étendre le travail présenté dans cette thèse, quelques possibles directions futures peuvent inclure:

- Étude d'autres méthodes anticollision afin de proposer de solutions au problème de collision et de l'hétérogénéité des dispositifs (bandes de fréquences différentes, des protocoles et usages divers). Ces éléments sont le point clé lors de la mise en œuvre à grande échelle de réseaux RFID.
- Évaluer les interactions de la RFID avec d'autres dispositifs tels que des capteurs et des actionneurs afin d'ouvrir de perspectives vers nouveaux scénarios pour le développement et la mise en œuvre d'applications innovantes.
- L'outil de conception de la topologie RFID peut être amélioré en ajoutant des modèles additionnels de propagation ainsi que l'inclusion de modelés d'antennes directionnelles, modélisation en trois dimensions (3D) du déploiement et de l'interférence cause par des objets. L'implémentation de ces points va permettre la représentation des scénarios plus complexes.
- Mettre en place un réseau à grande échelle avec de dispositifs hétérogènes comme nous l'avons défini au chapitre 3 en utilisant une approche P2P pour fournir la connectivité et partage des ressources nécessaires dans un « framework » unifié.
- Enfin, des tests supplémentaires avec du matériel embarqué vont nous permettre de créer une plateforme autonome à bas prix. Cette plateforme peut répondre à l'inclusion de multiples dispositifs hétérogènes comme de lecteurs RFID, capteurs et actionneurs dans une approche modulaire.

Thesis Publications

Journal:

- Oscar Botero & Hakima Chaouchi, *Radio Frequency Identification framework for heterogeneous nodes.* Personal and Ubiquitous Computing. Springer 2012.

Book Chapter:

- Oscar Botero & Hakima Chaouchi, Chapter 5, *On RFID and research issues*. Chaouchi Hakima, the Internet of Things: Connecting Objects, Wiley-ISTE, 2010.

International Conferences:

- Oscar Botero & Hakima Chaouchi, *P2P Framework for RFID enabled and non-enabled users*. IoTs 2010, Hangzhou China.

- Oscar Botero & Hakima Chaouchi, *RFID network topology design tool based on Genetic Algorithms*. RFID-TA Barcelona, Spain 2011.

- Oscar Botero & Hakima Chaouchi, *RFID for non-RFID users Embedded Hardware Implementation*. ANT Niagara Falls, Canada 2011.

Research Report:

- A. Ait Wakrim, O. Botero, K. Raymond, H. Chaouchi "TRACK-IoT: Heterogeneous IoT Network", Research Report, Telecom Sud Paris March 2012.

References

[1] Gerhard Metz,Miia Korpela,Mikko Nikkanen,Katariina Penttilä Leif Wiebking. (2012, January) CERFID. [Online]. http://www.rfid-in-action.eu/public/results/roadmap

[2] Leonid Mats, and Peter J. Hawrylak Marlin H. Mickle, "Physics and Geometry of RFID," in RFID Handbook, Applications, Technology, Security, and Privacy.: CRC, 2008, ch. 1, pp. 3-15.

[3] S. Sarma, J. Williams S. Miles, RFID Technology and Applications, Massachusetts Institute of Technology, Ed.: Cambridge University Press, 2008, pp. 23-30.

[4] (2012, January) AUTO-ID LABS. [Online]. http://www.autoidlabs.org/

[5] (2012, January) EPCGLOBAL. [Online]. http://www.gs1.org/epcglobal

[6] D. Dobkin, The RF in RFID: Passive UHF RFID in Practice.: Newnes, 2008, pp. 8-24.

[7] S. Preradovic and N.C. Karmakar, "RFID Readers - A Review," in Electrical and Computer Engineering, 2006. ICECE '06. International Conference on, 2006, pp. 100-103.

[8] Sanjay Sarma, "RFID technology and its applications," in RFID Technology and Applications.: Cambridge University Press, 2008, p. 19.

[9] Gi Oug Oh, Doo Yeon Kim, Sang Il Kim, and Sung Yul Rhew, "A Quality Evaluation Technique of RFID Middleware in Ubiquitous Computing," in Hybrid Information Technology, 2006. ICHIT '06. International Conference on, 2006, pp. vol.2, no., pp.730-735.

[10] S.Z. Mohd Hashim, Mardiyono, N. Anuar, and W.M.N. Wan Kadir, "Comparative analysis on adaptive features for RFID middleware," in Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on, 2008, pp. 989-993.

[11] EPC Global. (2004-2008) EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 Mhz - 960 Mhz Version 1.2.0. [Online]. http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2

[12] D. Dobkin, The RF in RFID: Passive UHF RFID in Practice. Burlington, MA: Newnes, 2008.

[13] S. Tung, R. Hoare, J. Cain S. Dontharaju, "Design Automation for RFID Tags and Systems," in RFID Handbook, Applications, Technology, Security, and Privacy.: CRC, 2008, ch. 3, pp. 48-49.

[14] W. Lee, T. Shih J. Myung, "Adaptive Tag Anticollision Protocols for RFID Passive Tags," in RFID Handbook, Applications, Technology, Security, and Privacy, CRC, Ed., 2008, ch. 8, p. 140.

[15] D. Engels, S. Sarma J. Waldrop, "Colorwave: an anticollision algorithm for the reader collision problem," in ICC '03. IEEE International Conference on Communications, 11-15 May 2003, pp. 1206-1210 vol.2.

[16] D. Hanny, A. Pachano, G. Thompson J. Banks,.: Wiley, 2007, p. 77.

[17] Y. Chen D.i Yang and B. Wang, "RFID anti-collision algorithm in logistics service system," in IEEE International Conference on Service Operations and Logistics, and Informatics, 2008, pp. vol.2, no., pp.1891-1896.

[18] G. Choi, J. Chae, B. Kim C. Park, "A Design for Passive RFID System on a Chip," in 11th International Conference on Advanced Communication Technology ICACT 2009, Feb. 15-18, 2009, pp. vol.01, no., pp.836-839.

[19] Zhu Qiuling et al., "A robust radio frequency identification system enhanced with spread spectrum technique," in IEEE International Symposium on Circuits and Systems ISCAS 2009, 24-27 May 2009, pp. vol., no., pp.37-40.

[20] S. Shrestha, M. Balachandran, M. Agarwal, V.V. Phoha, and K Varahramyan, "A Chipless RFID Sensor System for Cyber Centric Monitoring Applications," in IEEE Transactions on Microwave Theory and Techniques, May 2009, pp. vol.57, no.5, pp.1303-1309.

[21] P.R Hartmann, "A passive SAW based RFID system for use on ordnance," in IEEE International Conference on RFID, 2009 , 27-28 April 2009, pp. vol., no., pp.291-297.

[22] H.-W. Son, G.-Y. Choi, and C.-S. Pyo, "Design of wideband RFID tag antenna for metallic surfaces," in Electronics Letters, 2 March 2006, pp. vol.42, no.5, pp. 263- 265.

[23] K.-H. Kim, J.-G. Song, D.-H. Kim, H.-S. Hu, and J.-H. Park, "Fork-shaped RFID tag antenna mountable on metallic surfaces," Electronics Letters, vol. vol.43, no.25, pp. pp.1400-1402, December 2007.

[24] H. Kwon and B. Lee, "Compact slotted planar inverted-F RFID tag mountable on metallic objects," Electronics Letters, vol. vol.41, no.24, pp. pp. 1308- 1310, November 2005.

[25] Sung-Lin Chen, "A Miniature RFID Tag Antenna Design for Metallic Objects Application," Antennas and Wireless Propagation Letters, IEEE, vol. vol.8, pp. pp.1043-1045, 2009.

[26] Li Yang, Rongwei Zhang, D. Staiculescu, C.P. Wong, and M.M. Tentzeris, "A Novel Conformal RFID-Enabled Module Utilizing Inkjet-Printed Antennas and Carbon Nanotubes for Gas-Detection Applications," Antennas and Wireless Propagation Letters, IEEE, vol. vol 8, pp. pp.653-656, 2009, 2009.

[27] A. Rida, Li Yang, R. Vyas, and M.M. Tentzeris, "Conductive Inkjet-Printed Antennas on Flexible Low-Cost Paper-Based Substrates for RFID and WSN Applications," Antennas and Propagation Magazine, IEEE, vol. vol.51, no.3, pp. pp.13-23, June 2009.

[28] D.K. Klair, Kwan-Wu Chin, and R. Raad, "A Survey and Tutorial of RFID Anti-Collision Protocols," Communications Surveys & Tutorials, IEEE, vol. vol.12, no.3, pp. pp.400-421, 2010.

[29] Lee K., Siu K Law C., "Efficient memory-less protocol for tag identification," in 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Toronto, CA, 2000, pp. pp. 75–84.

[30] Hailong Jia and Wang Yun, "Research on agent-based load balancing model for RFID middleware," in Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International

Conference on, 2011, pp. vol.1, no., pp.275-278.

[31] L. Schmidt, N. Mitton, D. Simplot-Ryl, R. Dagher, and R. Quilez, "DHT-based distributed ALE engine in RFID middleware," in RFID-Technologies and Applications (RFID-TA), 2011 IEEE International Conference on, 2011, pp. 319-326.

[32] Guo Fu-liang, Liang Ying-jie, Xiao Xue-fu, and Liu Jin-biao, "Research and Design of RFID Middleware Based on UML," in Control, Automation and Systems Engineering (CASE), 2011 International Conference on, 2011, pp. 1-4.

[33] Chunkai Zhang, Yuan Li, and Yan Chen, "Application oriented data cleaning For RFID middleware," in Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on, 2011, pp. 1544-1549.

[34] S. Weis, "Security and Privacy in Radio-Frequency Identification Devices," Mass. Inst. of Technology (MIT), Master's thesis May 2003.

[35] M.Y.-W. Chia et al., "Electronic Beam-Steering IC for Multimode and Multiband RFID," Microwave Theory and Techniques, IEEE Transactions on, vol. vol.57, no.5, pp. pp.1310-1319, May 2009.

[36] Chonggang Wang, M. Daneshmand, K. Sohraby, and Bo Li, "Performance analysis of RFID Generation-2 protocol," Wireless Communications, IEEE Transactions on, vol. vol.8, no.5, pp. pp.2592-2601, May 2009.

[37] (2012, January) METRO. [Online]. http://www.future-store.org/fsi-internet/html/en/375/index.html

[38] (2012, January) CAEN, RFID semi-passive with temperature sensor. [Online]. http://www.caen.it

[39] (2012, January) DESTRONFEARING,RFID for animal tracking. [Online]. http://www.destronfearing.com/cattle.php

[40] (2012, January) BIOMARK, RFID for animals. [Online]. http://www.biomark.com

[41] (2012, January) Verichip. [Online]. http://www.positiveidcorp.com/

[42] (2012, January) MIFARE. [Online]. http://mifare.net/

[43] (2012, January) SUICA. [Online]. http://en.wikipedia.org/wiki/Suica

[44] (2012, January) TSIPRISM. [Online]. http://www.tsiprism.com/

[45] (2012, January) NFC Forum. [Online]. http://www.nfc-forum.org

[46] "REF NFC 1 , Near Field Communication: White Paper,. Ecma/TC32-TG19/2004/1.," Ecma, White Paper ECMA 2004.

[47] (2012, January) Samsung. [Online]. http://www.samsung.com/global/microsite/galaxys2/html/feature.htm

[48] (2012, January) TouchATag. [Online]. http://www.touchatag.com

[49] (2012, January) Smart contactless cards. [Online]. http://www.smartcardalliance.org

[50] J. Minhun et al., "All-Printed and Roll-to-Roll-Printable 13.56-MHz-Operated 1-bit RF Tag on Plastic Foils," Electron Devices, IEEE Transactions on, vol. vol.57, no.3, pp. pp.571-580, March 2010.

[51] P. Burke. (2012, January) Towards a single-chip, implantable RFID system: is a single-cell radio possible? pdf. [Online]. http://nano.ece.uci.edu/papers/BurkeRFID.pdf

[52] (2009) ODIN Technologies - RFID tag pricing guide. May 2009. [Online]. http://www.odintechnologies.com

[53] K. Finkenzeller, , Second, Ed.: Wiley, 2003.

[54] Steve Smith, "Wi-Fi enabled mobile phone handsets in the US, 2010-2015," Coda research Consultancy, 2010.

[55] L. Chen, D. Chen, H. Yuan B. Ding, "Application of RTLS in Warehouse Management Based on RFID and Wi-Fi," in Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08, 2008.

[56] Z. Luo, E. Wong, C.J. Tan, J. Luo S. Zhou, "Interconnected RFID Reader Collision Model and its Application in Reader Anti-collision," in IEEE International Conference on RFID, 2007.

[57] Z. Ji, Z. Luo, E. C. Wong, C. J. Tan X. Peng, "A P2P Collaborative RFID Data Cleaning Model," in Hong Kong The 3rd International Conference on Grid and Pervasive Computing - Workshops2008 IEEE, 2008.

[58] M. Leuchtner, M. Beigl C. Decker, "A Peer-To-Peer Approach for Resolving RFIDs. TecO," in Poster presented at Ubicomp, 2003.

[59] Kalyan B. and Sajal K. D. Pradip D., "Ubiquitous Architectural Framework and Protocol for Object Tracking using RFID Tags," in MOBIQUITOUS, 2004.

[60] L.F. Cervantes, Young-Seok Lee, Hyunho Yang, and Jaewan Lee, "A Hybrid Middleware for RFID-based Parking Management System Using Group Communication in Overlay Networks," in Intelligent Pervasive Computing, 2007. IPC. The 2007 International Conference on, 2007, pp. 521-526.

[61] M.E. Ajana, H. Harroud, M. Boulmalf, and H. Hamam, "FlexRFID: A flexible middleware for RFID applications development," in Wireless and Optical Communications Networks, 2009. WOCN '09. IFIP International Conference on, 2009, pp. 1-5.

[62] F. Esposito, F. Chiti, R. Fantacci, S. Hosio, and Junzhao Sun, "Agent Based Adaptive Management of Non-Homogeneous Connectivity Resources," in Communications, 2006. ICC '06. IEEE International Conference on, 2006, pp. 1754-1759.

[63] Engels Daniel W. The reader collision problem. AUTO-ID Center White paper. [Online]. http://autoid.mit.edu/whitepapers/MIT-AUTOID-WH-007.PDF

[64] D.W., S.E. Sarma Engels, "The reader collision problem," in IEEE International Conference on

Systems, Man and Cybernetics, Hammamet, Tunisia, 2002.

[65] JXTA. [Online]. https://jxta.dev.java.net/

[66] (2012, January) gzip. [Online]. http://www.gzip.org/

[67] (2012, January) Jradius. [Online]. http://coova.org/JRadius/FreeRADIUS

[68] H. Chaouchi O. Botero, "Platform and experimentation of secure service location with P2P/Client-Server over ad hoc networks," in IFIP Wireless Days, 2009.

[69] Li Gong, "JXTA: a network programming environment," in Internet Computing IEEE, 2001, pp. 88 – 95.

[70] (January, 2012) The Network Simulator - ns-2. [Online]. http://www.isi.edu/nsnam/ns/

[71] "EPCglobal Tag Data Standards Version 1.4," EPCglobal, Standards 1.4, June 11, 2008.

[72] W. Alsalih, K. Ali, and H Hassanein, "Optimal distance-based clustering for tag anti-collision in RFID systems," in Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on, 2008, pp. 266-273.

[73] Penttinen A, "Chapter 10 – Network Planning and Dimensioning, - Introduction to Teletraffic Theory," Helsinki University of Technology, Lecture Notes: S-38.145 Fall 1999.

[74] Yahui Yang, Yujie Wu, Min Xia, and Zhijing Qin, "A RFID Network Planning Method Based on Genetic Algorithm," in Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on, April 2009, pp. 534-537.

[75] Hanning Chen, Yunlong Zhu, and Kunyuan Hu, "RFID networks planning using a multi-swarm optimizer," in Control and Decision Conference, 2009. CCDC '09, 2009, pp. 3548-3552.

[76] Hyunsik Seo and Chaewoo Lee, "A New GA-Based Resource Allocation Scheme for a Reader-to-Reader Interference Problem in RFID Systems," in Communications (ICC), 2010 IEEE International Conference on, 2010, pp. 1-5.

[77] Qiang Guan, Yu Liu, Yiping Yang, and Wenshneg Yu, "Genetic Approach for Network Planning in the RFID Systems," in Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on, 2006, pp. 567-572.

[78] Chiu Chui-Yu, Ke Cheng-Hsin, and K.Y. Chen, "Optimal RFID networks scheduling using genetic algorithm and swarm intelligence," in Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, 2009, pp. 1201-1208.

[79] Ben Niu, E.C. Wong, Yujuan Chai, and Li Li, "RFID Network Planning Based on MCPSO Alogorithm," in Information Science and Engineering (ISISE), 2009 Second International Symposium on, 2009, pp. 8-12.

[80] K. M. Ragsdell G. V. Reklaitis A. Ravindran, Engineering Optimization: Methods and Applications, Second Edition ed.: Wiley, 2006.

[81] Konstantinos E. Parsopoulos and Michael N.Vrahatis, Particle swarm optimization and intelligence : advances and applications.: IGI Global, 2010.

[82] Melanie Mitchell, An introduction to genetic algorithms.: Massachusetts Institute of Technology, 1996, pp. 11-12.

[83] Melanie Mitchell, An introduction to genetic algorithms.: Massachusetts Institute of Technology, 196, pp. 116-117.

[84] Graham R Lubachevsky B., "Minimum perimeter rectangles that enclose congruent non-overlapping circles," Journal Discrete Mathematics 309 (2009) Elsevier, 2009.

[85] Syed Ahson and Mohammad Ilyas, RFID handbook : applications, technology, security, and privacy.: CRC , p. 76.

[86] J. Seybold, Introduction to RF Propagation.: Wiley, 2005.

[87] A. and Raué, P. and Ruttkay, Zs. Eiben, "Parallel Problem Solving from Nature PPSN III," Springer, vol. vol. 866, pp. pages 78-8, 1994.

[88] Darrell Whitley, "A Genetic Algorithm Tutorial," Statistics and Computing Journal, vol. 4, pp. 65-85, 1994.

[89] (2012, January) Java SE. [Online]. http://www.oracle.com/technetwork/java/javase/overview/index.html

[90] (2012, January) NetBeans. [Online]. http://netbeans.org/

[91] ZigBee Alliance, "ZigBee Specification Document 053474r17: ZIGBEE SPECIFICATION," 2007.

[92] M.T. Lockman and A. Selamat, "Verification and validation communication layer of embedded Smart Card system," in Electronic Design, 2008. ICED 2008. International Conference on, 2008, pp. 1-5.

[93] Ming-Shen Jian and Shu Hui Hsu, "Location Aware Public/Personal Diversity of Information Services based on embedded RFID Platform," in Advanced Communication Technology, 2009. ICACT 2009, 2009, pp. 1145-1150.

[94] Huiping Li, Yi Zhou, Lu Tian, and Chunlin Wan, "Design of a Hybrid RFID/GPS-Based Terminal System in Vehicular Communications," in Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on, 2010, pp. 1-4.

[95] Zhou Xiaoguang and Long Wei, "The research of network architecture in warehouse management system based on RFID and WSN integration," in Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on, 2008, pp. 2556-2560.

[96] I. Fonseca and F. Lopes, "An embedded system and an architecture for access control and access management: An application to the educational environment," in Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on, 2010, pp. 1-5.

[97] O. Botero, K. Raymond, H. Chaouchi A. Ait Wakrim, "TRACK-IoT: Heterogeneous IoT Network," Telecom Sud Paris, Paris , Research Report March 2012.

[98] (2012, January) ATMEL. [Online]. http://www.atmel.com/dyn/products/product_card.asp?part_id=3633

[99] (2012, January) Arduino. [Online]. http://arduino.cc

[100] (2012, January) Wiegan Interface. [Online]. http://en.wikipedia.org/wiki/Wiegand_interface

[101] FAT16 library. (2012, February) Arduino FAT16 library. [Online]. http://code.google.com/p/fat16lib/

[102] (2012, January) Wi-Fi module. [Online]. http://ww1.microchip.com/downloads/en/DeviceDoc/70624A.pdf

[103] (2012, January) XBee. [Online]. http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module.jsp#overview

[104] (2012, January) Wireshark. [Online]. http://www.wireshark.org/

[105] (2012, January) Maxim. [Online]. http://www.maxim-ic.com/datasheet/index.mvp/id/2688

[106] Finkenzeller K, RFID handbook , Second, Ed.: Wiley, 2003, pp. pp.195–219.

[107] M. Daneshmand, K. Sohraby, and B. Li C. Wang, "Performance Analysis of RFID Generation-2 Protocol," in IEEE Transactions on Wireless Communications, 2009.

Annexes

Annex A: Topology Tool source code excerpts

Java function to calculate RFID readers overlapping area

```
private double CalculateIntersections (double[][] IndValues) {
      double IArea = 0, d = 0;
      double X1 = 0, X2 = 0, Y1 = 0, Y2 = 0;
      double R1 = 0, R2 = 0, X = 0, Y = 0;
      double Yp = 0, Yn = 0, Angle1 = 0, Angle2 = 0;
      double Lenght = 0, Aseg1 = 0, Aseg2 = 0, Atrian1 = 0, Atrian2 = 0;
      double Asect1 = 0, Asect2 = 0, Aintersection = 0;

      for (int i = 0; i < this.Readers; i++) {
         R1 = getRadius(IndValues[i][0]);
         X1 = IndValues[i][1];
         Y1 = IndValues[i][2];
         for (int j = i + 1; j < this.Readers; j++) {
            R2 = getRadius(IndValues[j][0]);
            X2 = IndValues[j][1];
            Y2 = IndValues[j][2];

            d = Math.sqrt(
                  Math.pow((X1 - X2), 2) + Math.pow((Y1 - Y2), 2));

            if (d > R1 + R2) {
               Aintersection = 0;
               IArea += Aintersection;
            } else if (d < Math.abs(R1 - R2)) {
               if (R1 < R2) {
                  Aintersection = 3.14 * R1 * R1;
                  IArea += Aintersection;
               } else if (R1 > R2) {
                  Aintersection = 3.14 * R2 * R2;
                  IArea += Aintersection;
               }
            } else if (d == 0) {
               Aintersection = 3.14 * R1 * R2;
               IArea += Aintersection;
            } else if (d < R1 + R2) {
               //Overlap
               X = ((R1 * R1) + (d * d) - (R2 * R2)) / 2 * d;
               Y = ((2 * R1 * R1 * d * d) + (2 * R2 * R2 * d * d) + (2 * R1 * R1 * R2 * R2)
                     - (R1 * R1 * R1 * R1) - (R2 * R2 * R2 * R2) - (d * d * d * d))
                     / (4 * d * d);
               Yp = Math.sqrt(Y);
               Yn = -1 * Math.sqrt(Y);
               Lenght = Yp - Yn;
               Angle1 = 2 * Math.asin((Lenght / 2) / R1);
               Angle2 = 2 * Math.asin((Lenght / 2) / R2);
               Aseg1 = R1 * R1 * Angle1 / 2;
               Aseg2 = R2 * R2 * Angle2 / 2;
               Atrian1 = X * Yp;
               Atrian2 = (d - X) * Yp;
               Asect1 = Aseg1 - Atrian1;
               Asect2 = Aseg2 - Atrian2;
               Aintersection = Asect1 + Asect2;
               IArea += Aintersection;
            } } }       return round(2, IArea);    }
```

Genetic Algorithm two individuals Crossover method

```
private Individual crossover(Individual Indv1, Individual Indv2) {

    Random randomGenerator = new Random();
    int Cross = randomGenerator.nextInt(100);
    Individual NewIndv = new Individual(this.Max_Genes);

    if (Cross <= CrossoverRate) {
       int cut_point = 0;

       Individual NewIndv1 = new Individual(this.Max_Genes);
       Individual NewIndv2 = new Individual(this.Max_Genes);

       cut_point = cut_point();
       //Crossover
       for (int i = 0; i < cut_point; i++) {
          NewIndv1.set_Genes(i, Indv1.get_Genes(i));
       }
       for (int i = cut_point; i < this.Max_Genes; i++) {
          NewIndv1.set_Genes(i, Indv2.get_Genes(i));
       }

       for (int i = 0; i < cut_point; i++) {
          NewIndv2.set_Genes(i, Indv2.get_Genes(i));
       }
       for (int i = cut_point; i < this.Max_Genes; i++) {
          NewIndv2.set_Genes(i, Indv1.get_Genes(i));
       }
       //evaluate fitness to select best breed
       NewIndv1.set_fitness(get_fitness(NewIndv1));
       NewIndv2.set_fitness(get_fitness(NewIndv2));

       if (NewIndv1.get_fitness() >= NewIndv2.get_fitness()) {
          NewIndv = NewIndv1;
       } else {
          NewIndv = NewIndv2;
       }
       //Mutate
       NewIndv = mutation(NewIndv);
       return NewIndv;
    } else {
       //No crossover
       if (Indv1.get_fitness() >= Indv2.get_fitness()) {
          return Indv1;
       } else {
          return Indv2;
       }       }    }
```

Genetic Algorithm Mutation function

```
private Individual mutation(Individual Indv) {
    //probability of 1/Mutator Rate to mutate a gen
    Random randomGenerator = new Random();
    int randomMutator = randomGenerator.nextInt(100);
    //to select the bit to be mutated
    int randomInt = randomGenerator.nextInt(this.Max_Genes);
```

```
    if (randomMutator <= MutatorRate) {

        int gen_value = Indv.get_Genes(randomInt);
        int inverse_gen = 0;
        if (gen_value == 1) {
            inverse_gen = 0;
        } else {
            inverse_gen = 1;
        }
        Indv.set_Genes(randomInt, inverse_gen);
    } else {
    }

    return Indv;
}
```

Genetic Algorithm Main Execution Code

```
    public Task startGA() {
        return new StartGA(getApplication());
    }

    private class StartGA
            extends org.jdesktop.application.Task<Object, Void> {

        StartGA(org.jdesktop.application.Application app) {
            super(app);
            Generation = 0;
            StateZeroPaint = false; //to show the final results
            start = System.nanoTime();//to calculate the execution time
            clear();

        }

        @Override
        protected Object doInBackground() {
            // This method runs
            // on a background thread
            double best = 0;
            Individual[] first_generation = null;

            int cataclysm_counter = 0, migration_counter = 0;

            GenI = Init();

            first_generation = GenI.generate_random_pop();
            next_generation = GenI.Next_Generation(first_generation);
            succeeded(Generation);

            do {
                next_generation = GenI.Next_Generation(next_generation);
                best = Double.parseDouble(GenI.get_best_fitness(next_generation));
                Generation++;

                cataclysm_counter++;
                if (cataclysm_counter
                        == Integer.parseInt(Text_Cataclysm.getText())) {
                    next_generation =
```

```
                GenI.CATACLYSM(GenI.get_best_Individual(next_generation));
            cataclysm_counter = 0;
        }

        migration_counter++;
        if (migration_counter
              == Integer.parseInt(Text_Migration.getText())) {
            next_generation =
                GenI.MIGRATION(GenI.get_best_Individual(next_generation),
                next_generation);
            migration_counter = 0;
        }

        CurrentFitness = best;
        succeeded(Generation);

        //To stop the execution
        if (stopit) {
            return 0;
        }

        //*****function to log the iterations, fitness and time
        EventsLog(CurrentFitness, Generation, duration);

    } while (best != 1.0);

    //error tags probabilistic approach**********************************
    GenI.getTagsProbabilistic(GenI.get_best_Individual(next_generation));

    return 0;  }

@Override
protected void succeeded(Object result) {
    // Update the GUI based on
    // the result computed by doInBackground().

    Text_Max_Generations.setText(((Integer) result).toString());
    print_Results(GenI.get_best_Individual(next_generation));

    //Gets the type of rendering: Gradient or Green
    if (RadioButtonGreen.isSelected()) {
        renderingOption = 0;
    } else {
        renderingOption = 1;
    }

    if (ViewGA != null) {
        //to update area
        double W = Double.parseDouble(XTarget.getText());
        double H = Double.parseDouble(YTarget.getText());
        //calls the routine to paint
        ViewGA.SetParameters(duration, Generation, NumTags, Readers,
              CurrentFitness, Tags, solutiondec, W, H, renderingOption);
    }

}

private void EventsLog(double Fitness, int Gen, float dur) {
    //Records the number of iterations; fitness and time to generate
    //statistics and comparison charts
```

```
        //create a file
        File StatsFile = new File("Stats/LogFileComparison");
        FileWriter fout;
        //open the file  , append and close

        try {
            fout = new FileWriter(StatsFile, true);
            BufferedWriter writer = new BufferedWriter(fout);
            writer.write(Gen + "\t" + Fitness + "\t" + dur + "\t" + "\n");
            writer.close();

        } catch (IOException e) {
            System.err.println("Unable to write to file");
            System.exit(-1);
        }
    }
}
```

Microcontroller

RFID LF Reader with external time stamp (Maxim DS1307) code

```
#include "Wire.h"
#define DS1307_ADDRESS 0x68

byte RFIDcardNum[4];
byte evenBit = 0;
byte oddBit = 0;
byte isData0Low = 0;
byte isData1Low = 0;
int recvBitCount = 0;
byte isCardReadOver = 0;

void setup()
{
  Wire.begin();
  Serial.begin(9600);
  Serial.println("Starting...");
  attachInterrupt(0, ISRreceiveData0, FALLING );  //data0/rx is connected
to pin 2, which results in INT 0
  attachInterrupt(1, ISRreceiveData1, FALLING );  //data1/tx is connected
to pin 3, which results in INT 1

}

void loop(){
  //read card number bit
  if(isData0Low||isData1Low){
    if(1 == recvBitCount){//even bit
      evenBit = (1-isData0Low)&isData1Low;
    }
    else if( recvBitCount >= 26){//odd bit
      oddBit = (1-isData0Low)&isData1Low;
      isCardReadOver = 1;
      delay(10);
    }
    else{
      //only if isData1Low = 1, card bit could be 1
```

```
      RFIDcardNum[2-(recvBitCount-2)/8] |= (isData1Low << (7-(recvBitCount-
2)%8));
    }
    //reset data0 and data1
    isData0Low = 0;
    isData1Low = 0;
  }
  //print the card id number
  if(isCardReadOver){
    if(checkParity()){

      Serial.print(*((long *)RFIDcardNum));
      Serial.print('\t');
      printDate();
    }
    resetData();
  }

}
byte checkParity(){
  int i = 0;
  int evenCount = 0;
  int oddCount = 0;
  for(i = 0; i < 8; i++){
    if(RFIDcardNum[2]&(0x80>>i)){
      evenCount++;
    }
  }
  for(i = 0; i < 4; i++){
    if(RFIDcardNum[1]&(0x80>>i)){
      evenCount++;
    }
  }
  for(i = 4; i < 8; i++){
    if(RFIDcardNum[1]&(0x80>>i)){
      oddCount++;
    }
  }
  for(i = 0; i < 8; i++){
    if(RFIDcardNum[0]&(0x80>>i)){
      oddCount++;
    }
  }
  if(evenCount%2 == evenBit && oddCount%2 != oddBit){
    return 1;
  }
  else{
    return 0;
  }
}
void resetData(){
  RFIDcardNum[0] = 0;
  RFIDcardNum[1] = 0;
  RFIDcardNum[2] = 0;
  RFIDcardNum[3] = 0;
  evenBit = 0;
  oddBit = 0;
  recvBitCount = 0;
  isData0Low = 0;
  isData1Low = 0;
  isCardReadOver = 0;
```

```
}
// handle interrupt0
void ISRreceiveData0(){
  recvBitCount++;
  isData0Low = 1;
}

// handle interrupt1
void ISRreceiveData1(){
  recvBitCount++;
  isData1Low = 1;
}

byte bcdToDec(byte val)  {
// Convert binary coded decimal to normal decimal numbers
  return ( (val/16*10) + (val%16) );
}

void printDate(){
  // Reset the register pointer
  Wire.beginTransmission(DS1307_ADDRESS);
  Wire.send(0);
  Wire.endTransmission();
  Wire.requestFrom(DS1307_ADDRESS, 7);

  int second = bcdToDec(Wire.receive());
  int minute = bcdToDec(Wire.receive());
  int hour = bcdToDec(Wire.receive() & 0b111111); //24 hour time
  int weekDay = bcdToDec(Wire.receive()); //0-6 -> sunday - Saturday
  int monthDay = bcdToDec(Wire.receive());
  int month = bcdToDec(Wire.receive());
  int year = bcdToDec(Wire.receive());

 print the date EG   1/1/11 11:11:11
  Serial.print(month);
  Serial.print("/");
  Serial.print(monthDay);
  Serial.print("/");
  Serial.print(year);
  Serial.print(" ");
  Serial.print(hour);
  Serial.print(":");
  Serial.print(minute);
  Serial.print(":");
  Serial.println(second);

}
```

Annex B: Topology Tool Report Example

RFID Reader's Deployment Solution

Binary Solution :
1010010000010011111100 0001010010010001010010 0110010111010001100000 0000100101100101011000
1111010000111011101000 0001101111101101110011 1111110110011111110010 0001111101001111111110
1000111011001011001111 1100011101110101100010

-- Reader : 0 Coverage Area (m2): 161.87 Radius: 7.18 X(m): 4.1 Y(m): 2.7
-- Reader : 1 Coverage Area (m2): 0.0 Radius: 0.0 X(m): 20.9 Y(m): 18.0
-- Reader : 2 Coverage Area (m2): 81.03 Radius: 5.08 X(m): 5.5 Y(m): 26.0
-- Reader : 3 Coverage Area (m2): 0.0 Radius: 0.0 X(m): 9.6 Y(m): 12.0
-- Reader : 4 Coverage Area (m2): 202.47 Radius: 8.03 X(m): 20.3 Y(m): 7.7
-- Reader : 5 Coverage Area (m2): 0.0 Radius: 0.0 X(m): 27.5 Y(m): 27.0
-- Reader : 6 Coverage Area (m2): 202.47 Radius: 8.03 X(m): 29.9 Y(m): 19.7
-- Reader : 7 Coverage Area (m2): 0.0 Radius: 0.0 X(m): 31.4 Y(m): 9.0
-- Reader : 8 Coverage Area (m2): 121.48 Radius: 6.22 X(m): 14.4 Y(m): 25.7
-- Reader : 9 Coverage Area (m2): 182.32 Radius: 7.62 X(m): 7.7 Y(m): 14.7

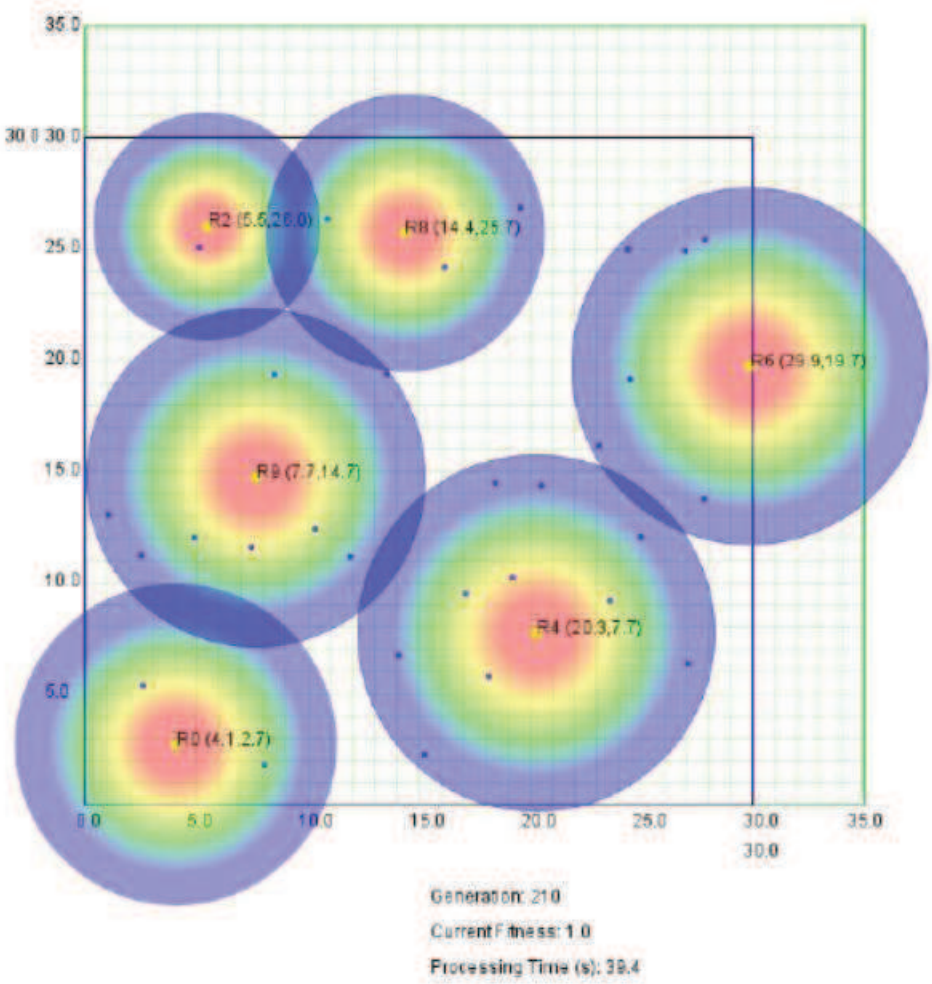
--- Total Intersection Area (m2): 38.07
--- Useless Readers: 0
--- Covered Tags: 30
--- Area out of Range (m2) : 0.0
--- Redundant Readers: 0
--- Affected Tags: 0

--- Fitness: 1.0
w1*Fint = 0.2 x 1.0 = 0.2
w2*Fuse = 0.1 x 1.0 = 0.1
w3*Fcov = 0.2 x 1.0 = 0.2
w4*Fout = 0.1 x 1.0 = 0.1
w5*Frdn = 0.2 x 1.0 = 0.2
w6*Faff = 0.2 x 1.0 = 0.2

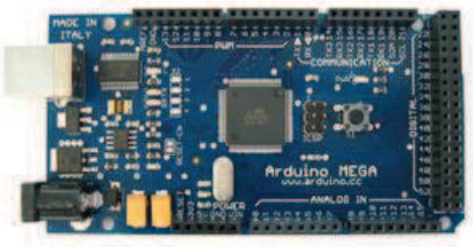
Generation: 210

Processing Time (s): 39.4

RFID Readers' Location



Annex C: Arduino Mega



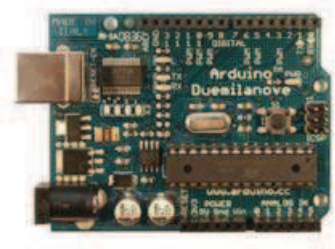
Overview

The Arduino Mega is a microcontroller board based on the ATmega1280. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Summary

Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Annex D: Arduino Duemilanove



Overview

The Arduino Duemilanove ("2009") is a microcontroller board based on the ATmega168 or ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

"Duemilanove" means 2009 in Italian and is named after the year of its release. The Duemilanove is the latest in a series of USB Arduino boards.

Summary

Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz

Annex E: Maxim DS1307

MAXIM

DS1307
64 x 8, Serial, I2C Real-Time Clock

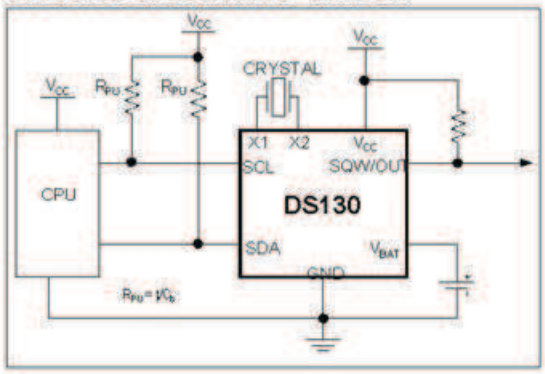
GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I2C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

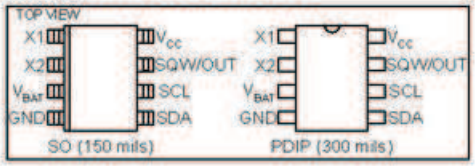
FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- I2C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

TYPICAL OPERATING CIRCUIT



PIN CONFIGURATIONS



ORDERING INFORMATION

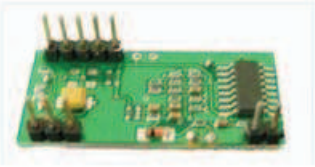
PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

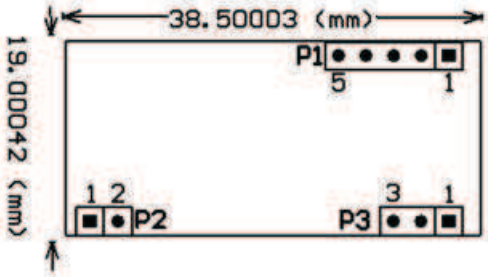
+Denotes a lead-free/RoHS-compliant package.

**A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device.*

Annex F: LF RFID module

RDM630 Specification





1. Pin Definition (WEIGAND):

P1:

PIN1	DATA0
PIN2	DATA1
PIN3	
PIN4	GND
PIN5	+5V(DC)

P2:

PIN1	ANT1
PIN2	ANT2

P3:

PIN1	LED
PIN2	+5V(DC)
PIN3	GND

2. Pin definition (TTL interface RS232 data format):

P1:

PIN1	TX
PIN2	RX
PIN3	
PIN4	GND
PIN5	+5V(DC)

P2:

PIN1	ANT1
PIN2	ANT2

P3:

PIN1	LED
PIN2	+5V(DC)
PIN3	GND

Specification and Parameter:

Frequency	125KHz
Baud Rate	9600 (TTL Electricity Level RS232 format)
interface	Weigang26 Or TTL Electricity Level RS232 format
Power supply	DC 5V (±5%)
Current	<50Ma
Operating range	>50mm (Depend on Card/Tag shape, manufacturer)
Expand I/O port	N/A
Indication light	N/A
Working temperature	-10℃ ~+70℃
Storage temperature	-20℃ ~+80℃
Max. humidity	Relative humidity 0 ~ 95%
Size	38.5mm×19mm×9mm

Annex G: XBee (ZigBee module)

1. XBee Series 2 OEM RF Modules

The XBee Series 2 OEM RF Modules were engineered to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices.

The modules operate within the ISM 2.4 GHz frequency band.

1.1. Key Features

High Performance, Low Cost

- Indoor/Urban: up to 133' (40 m)
- Outdoor line-of-sight: up to 400' (120 m)
- Transmit Power: 2 mW (+3 dBm)
- Receiver Sensitivity: -95 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

Retries and Acknowledgements

DSSS (Direct Sequence Spread Spectrum)

Each direct sequence channel has over 65,000 unique network addresses available

Point-to-point, point-to-multipoint and peer-to-peer topologies supported

Self-routing, self-healing and fault-tolerant mesh networking

Low Power

XBee Series 2

- TX Current: 40 mA (@3.3 V)
- RX Current: 40 mA (@3.3 V)
- Power-down Current: < 1 µA @ 25°C

Easy-to-Use

No configuration necessary for out-of box RF communications

AT and API Command Modes for configuring module parameters

Small form factor

Extensive command set

Free X-CTU Software (Testing and configuration software)

Free & Unlimited Technical Support

1.2. Specifications

Table 1-01. Specifications of the XBee Series 2 OEM RF Module (PRELIMINARY)

Specification	XBee Series 2
Performance	
Indoor/Urban Range	up to 133 ft. (40 m)
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)
Transmit Power Output (software selectable)	2mW (+3dBm)
RF Data Rate	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 230400 bps (non-standard baud rates also supported)
Receiver Sensitivity	-95 dBm (1% packet error rate)
Power Requirements	
Supply Voltage	2.8 – 3.4 V
Operating Current (Transmit)	40mA (@3.3 V)
Operating Current (Receive)	40mA (@3.3 V)
Power-down Current	< 1 uA @ 25° C
General	
Operating Frequency Band	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)
Operating Temperature	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip, RPSMA, or U.FL Connector
Networking & Security	
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer & Mesh
Number of Channels (software selectable)	16 Direct Sequence Channels
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
Agency Approvals	
United States (FCC Part 15.247)	Pending
Industry Canada (IC)	Pending
Europe (CE)	Pending

Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors. For more information, refer to the "XBee Series 2 Antenna" application note located on MaxStream's web site http://www.maxstream.net/support/knowledgebase/article.php?kb=153

Annex H: WLAN module

zero G
CONNECTING THE INTERNET OF THINGS

PRELIMINARY
ZG2100M/ZG2101M Wi-Fi Modules Datasheet

Description

The ZG2100M & ZG2101M modules are low-power 802.11b implementations. All RF components, the baseband and the entirety of the 802.11 MAC reside on-module, creating a simple and cost-effective means to add Wi-Fi connectivity for embedded devices. The module(s) implement a high-level API, simplifying design implementation and allowing the ZG2100M or ZG2101M to be integrated with 8- and 16-bit host microcontrollers. Hardware accelerators support the latest Wi-Fi security standards.

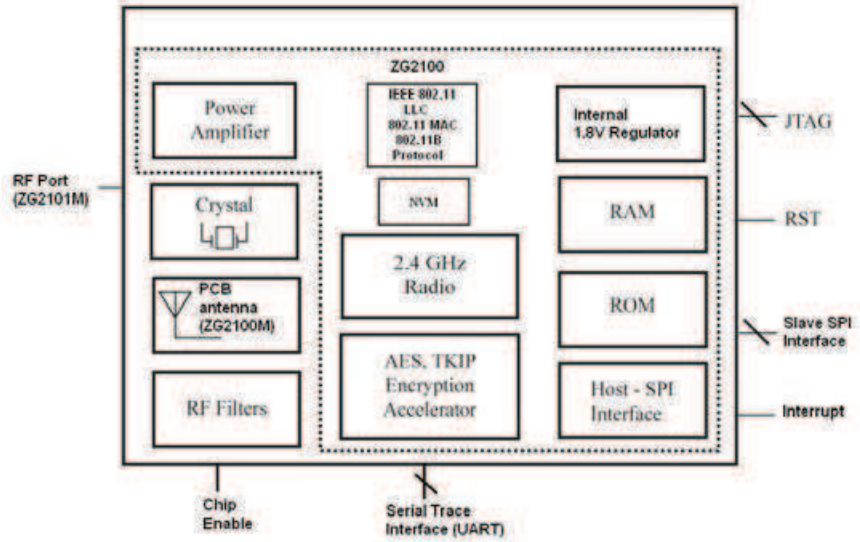


Figure 1 ZG2100M/ZG2101M Modules: Functional Block Diagram

1. Key Features

Ease of Software Development

- Simple API suited for embedded market
- Targeted for low resource host processors
- Entire MAC integrated on-chip
- Serialized MAC address, each device comes with an unique MAC address in range 001EC0xxxxxx
- Wireless driver library provides all required control of device
- Simple usage model, no requirement for OS

Low Power Operation

- Low power, 250uA sleep mode with fast wake up, 0.1uA hibernate,
- Sleep power state managed by ZG2100, enabling low average power while maintaining AP association without host control

RF

- Integrated PA
- Support for external PA for high RF output power applications
- Power output +10dBm typical at antenna
- Power output programmable from +0dBm to meet varying application needs
- Min RX sens.of-91dBm @ 1MB/Sec. at antenna
- Integrated PCB antenna (ZG2100M)
- Support for external antenna available (ZG2101M)

Low External Component Count

- Fully integrated RF frequency synthesizer
- Single external crystal is needed, with no external caps , as a source for reference clock
- Single 3.3V supply with internal built in 1.8V regulator

Wi-Fi & Regulatory

- Supports 1Mbps & 2Mbps and module-based solutions are "Wi-Fi certified" for 802.11b
- Hardware support for AES, and RC4 based ciphers (WEP, WPA, WPA2 security)
- FCC Certified (USA, FCC ID: W7O-ZG2100-ZG2101), IC Certified (IC: 8248A-G21ZEROG), Wi-Fi Certified, RoHS and CE compliant, and fully compliant with European Market and meet the R&TTE Directive for Radio Spectrum