UNIVERSITY OF DEUSTO

RFID ANTI-COLLISION PROTOCOL BASED ON THE WINDOW METHODOLOGY TO REDUCE THE ENERGY CONSUMPTION AND THE IDENTIFICATION TIME REGARDLESS OF THE TAG ID DISTRIBUTION

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy by Nikola Cmiljanic

within the PhD Program in Engineering for the Information Society and Sustainable Development

Directed by Dr. Hugo Landaluce and Dr. Asier Perallos

UNIVERSITY OF DEUSTO

RFID ANTI-COLLISION PROTOCOL BASED ON THE WINDOW METHODOLOGY TO REDUCE THE ENERGY CONSUMPTION AND THE IDENTIFICATION TIME REGARDLESS OF THE TAG ID DISTRIBUTION

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy by Nikola Cmiljanic

within the PhD Program in Engineering for the Information Society and Sustainable Development

Directed by Dr. Hugo Landaluce and Dr. Asier Perallos

The doctoral student Director Director

Bilbao, May 2018

RFID ANTI-COLLISION PROTOCOL BASED ON THE WINDOW METHODOLOGY TO REDUCE THE ENERGY CONSUMPTION AND THE IDENTIFICATION TIME REGARDLESS OF THE TAG ID DISTRIBUTION

Author: Nikola Cmiljanic
Directors: Hugo Landaluce and Asier Perallos

Text printed in Bilbao
First edition, May 2018

To my Family

Acknowledgements

Writing acknowledgments is usually an especially pleasant task. It unavoidably evokes memories, remembering the people you have known and the places where you have been. I should acknowledge not only the people I have met and who helped me in the last three years in Bilbao, but I also need to name those who contributed in their own direct or indirect way to the completion of this dissertation.

Before I start to present this thesis, I would like to thank all those people who helped me to obtain it. This is a very important goal in my life. During my entire stay in Bilbao, I have met new friends and contacts who have contributed great experiences in my life.

I would like to express my heartfelt gratitude to my supervisors Hugo Landaluce and Asier Perallos. Their experience has been very important to address confusing aspects. Hugo was a constant help with my work, making clarifying suggestions and providing support during the whole project. He always welcomed me as a friend and helped to develop the ideas in this thesis - without his help this thesis would not have been what it is. I would like to thank all the members and friends in the Mobility Department of the University of Deusto for their support, which was much needed during my studies. I appreciate my dear friends Jen and Paddy. They were my support every day and were valuable for my language skills.

My greatest gratitude is dedicated to my family, my parents, and my sister, who have been with me through good and bad times, as well as the rest of my family and friends, because without all of you things would not be the same today. I would like to give my deepest gratitude to Nina, for her enormous help and support. All of you have motivated me to do things that I would otherwise have given up on. For that, I hope to forever be a blessing for you.

Thank you very much!

Nikola Cmiljanic

May 2018

Abstract

Radio Frequency Identification (RFID) is a technology that uses radio frequency signals to identify objects. This technology enables the communication between the main devices used in RFID, the reader and the tags. The tags are attached to the different objects to be uniquely identified, and the reader receives an identification code (ID) stored on every tag. This produces great advances in applications such as manufacturing, shipping, and distribution environments. Using this technology for tracing or locating objects in the supply chain increases its efficiency and reduces the number of errors.

The tags share a communication channel. Therefore, if several tags try to send information at the same time, the reader will be unable to distinguish these signals. This is called the tag collision problem. The consequences of this increase the time needed for identification as well as the energy consumption of the system.

To minimize tag collisions, RFID readers must use an anti-collision protocol. Different kinds of anti-collision protocols have been proposed in the recent literature. One of the most popular strategies to solve this problem are the query tree based protocols, where the current response of each tag only depends on the current reader command (the query) but not on the past history of the reader's queries. Under these types of protocols, the window methodology was proposed to decrease the number of bits transmitted by the tags, which also decreases the energy consumed by the RFID system. This methodology, however, assumes the ability of the reader to differentiate the query from the window size in the reader command. The first contribution proposed in this thesis is a protocol that is able to locate the bit string representing the

window size in the received command by standardizing the number of bits used for that purpose, called the Standardized Query window Tree (SQwT) protocol. Apart from solving this issue, this protocol manages to decrease the number of bits transmitted by the reader and therefore the total number of bits transmitted in the identification process with respect to the windowed protocols.

The adoption of tag IDs with the Electronic Product Code (EPC) standard provides an improvement of RFID, allowing it to access global networks. The EPC provides every tag with a unique ID that details the company, the type of product, the type of transport and a serial number. This organization causes tags stored in the same area to share some of their ID fields, such as the company or the type of product. Usually, tree based protocols have been successfully tested on tag populations with randomly generated IDs that possess a uniform distribution of all the IDs. That, however, is not the case when the EPC standard is used since the reader may face not only uniform, but also non-uniform distributions. The behaviour of tree based protocols under non-uniform distributions is severely affected, decreasing their performance. The second proposal in this thesis is a novel anti-collision protocol called the Flexible Query window Tree (FQwT) protocol, that is proposed with the aim of estimating the tag ID distribution, taking into consideration the partial tags' responses of the window methodology to decrease the time needed to identify all the tags in the interrogation area with low energy consumption regardless of the type of tag ID distribution.

Abstrakt

Radio Frequency Identification (RFID) je tehnologija koja koristi radio frekvenciju za identifikaciju objekata. Čitač (engl. reader) preuzima identifikacioni kod (ID) koji je zapisan na svakom tagu, pozicioniranom na različitim predmetima, koji moraju biti jedinstveno locirani. Navedene prednosti omogućavaju značajan napredak u oblastima kao što su: proizvodnja, transport i distribucija. Upotreba ove tehnologije u sektoru snadbijevanja povećava efikasnost, smanjuje greške i značajno poboljšava kvalitet.

RFID tehnologija omogućava komunikaciju između glavnih uređaja za zasnivanje konekcije: čitača i tagova. Navedeni uređaji dijele komunikacioni kanal omogućavajući većem broju tagova da šalju informacije u istom trenutku. Čitač, koji preuzima njihove signale, je onemogućen da uspješno identifikuje svaki tag. Navedeni problem je poznat kao problem kolizije tagova (engl. tag collision problem). Posljedice ovog događaja rezultiraju povećanju identifikacionog vremena kao i potrošnji energije u cijelom sistemu.

Da bi se ublažio problem kolizije tagova, RFID čitač je prinuđen da koristi anti-kolizione protokole (engl. anti-collision protocols). Različite vrste protokola su predstavljeni u skorijoj literaturi. Jedna od trenutno najpopularnijih strategija koja rješava navedeni problem su query tree based protokoli, zasnovani na upitu od strane čitača. Kod ovih protokola trenutni odgovor od strane taga jedino zavisi od čitačeve komande (upita). To znači da odgovor tagova nije u vezi sa istorijom čitačevih komandi. U okviru query tree based protokola je predstavljena window methodologija u cilju smanjenja ukupnog broja bitova u komunikaciji, što direktno utiče na smanjenje energije korištene u

RFID sistemu. Ova metodologija omogućava čitaču da razlikuje upit od veličine prozora u čitačevoj komandi. Prvi predstavljeni protokol u disertaciji je protokol koji omogućava identifikaciju dijela bitova koji označava veličinu prozora u primljenoj poruci od dijela komande predviđenog za trenutni upit. To je omogućeno implementiranjem standardizacije prozora u čitačevoj komandi na način da čitač koristi konstantan broj bitova za veličinu prozora. Protokol koji to obezbjeđuje naziva se Standardized Query window Tree (SQwT) protocol. Pored toga što SQwT rješava navedeni problem, ovaj protokol značajno smanjuje broj bitova koje transmituje čitač i na ovaj način se smanjuje ukupan broj bitova neophodan za identifikacioni proces.

Uvođenjem EPC standarda u RFID tehnologiju je obezbijeđen značajan napredak. Od tada je dozvoljen pristup globalnoj mreži. EPC standard omogućava da svaki tag ima jedinstven ID sa detaljima kompanije, vrste proizvoda, oznake transporta i serijskog broja. Takođe, organizacija koja upravlja EPC standardom je primorala sinhronizovanu upotrebu RFID tagova, na način da se u određenoj zoni podaci dijele, kao npr. ime kompanije ili vrsta proizvoda. Obično, tree based protokoli su testirani u okruženju populacije tagova, gdje su ID vrijednosti generisane proizvoljno. Ovaj tip distribucije se naziva homogena distribucija (engl. uniform distribution - UD). To nije slučaj sa nepravilnom distribucijom (engl. non-uniform distribution - non–UD) koja je takođe zastupljena u RFID sistemima. Ponašanje tree based protokola je značajno degradirano u okruženju sa nepravilnom ID distribucijom. Drugi predstavljeni protokol u disertaciji je Flexible Query window Tree (FQwT) protokol. Ovaj protokol je kreiran u cilju da se identifikuje vrsta distribucije prije samog procesa identifikacije. Ova metoda je omogućena uzimajući u obzir parcijalne odgovore tagova. FQwT protokol značajno smanjuje vrijeme identifikacije svih tagova iz dostupne zone kao i energiju neophodnu za komunikaciju nebitno od distribucije u kojoj tagovi funkcionišu.

Contents

List of Figures

List of Tables

Whoever is careless with the truth in small matters cannot be trusted with important matters.

Albert Einstein

1

Introduction and motivation

In recent years, Automatic Identification has become very popular and useful, with the aim of simplifying many activities in service industries, purchasing and distribution logistics, manufacturing companies, and so on. This technology was created to provide information about goods, objects, and people. The most commonly used form of electronic data-carrying devices in use in everyday life is the smart card, based upon a contact field. The mechanical contact that the smart card uses is very impractical. The ideal and flexible solution of transferring data between the reader and the data-carrying device is contactless communication. Technology that uses procedures with the aim of transferring power and data wirelessly is called Radio Frequency Identification (RFID).

Nowadays, RFID has become very popular and its growth is increasing every day. The main feature of this technology is that RFID tags do not require close handling, and no line of sight is required between the reader and the object that needs to be identified. In 2015, over 10.1 billion RFID tags were produced, and this figure will rise to 19 billion by 2026. The numerous companies using RFID systems and the global sales of RFID systems were approximately 900 million $US in 2000. It is estimated that in 2025, the RFID technology market will be worth 40.5 billion $US [IDTechEx, 2017].

One of its most important features is its ability to identify objects wirelessly without being in their line-of-sight, as opposed to bar code technology. RFID has many advantages in comparison to other identification systems: a faster identifying speed, greater longevity, its data encryption, and, most importantly, no line of sight is necessary. This proves to be beneficial in many industries, such as, health care, retail, inventory management, supply-chain management, and wireless sensor networks. The main aim of RFID is to reduce logistical overhead, cost, and minimize product losses. It provides a process with higher productivity. RFID originated from radar technology, and it has greatly evolved since then. Currently, this technology goes beyond identification purposes, and it is being used in localization and sensing applications. Furthermore, there is a growing class of battery-free computational and sensing tags which go beyond simple barcode replacement functionality, referred to as Computational RFID (CRFID) and RFID sensor tags.
This section will present the main Automatic Identification technologies; these include RFID, which is the baseline of this thesis. Later, the main problems addressed here will be explained.

1.1 Automatic Identification Systems

Automatic Identification Systems (AIS) are widely used with the aim of identifying a person or object with a higher percentage of success. The technologies most used for AIS are: barcode, optical character recognition, and biometric procedures [Floerkemeier 07].

1.1.1 Barcode

A barcode is an optical translation organized in the form of alternating vertical bars and spaces. Also, it consists of a digital or alphabetic code. This is used to identify the object that carries the barcode. This system became commercially successful when it started to be used in supermarket checkout systems. According to the statistics, barcode systems in Western Europe at the beginning of the 1990s had a turnover volume of around 3 billion.

The barcode most used in the global market is the European Article Number (EAN). It was created in 1976 to fulfil the requirements of the grocery industry. It is defined for all marketed products. This code comprises 13 numbers, divided into four components: the number system, the manufacturer code, the product code, and the check digit. The first three digits of EAN-13 (GS1 prefix) are used to identify the GS1 Member Organization which the producer has joined. The following 5 digits correspond to the manufacturer code assigned to each manufacturer by the indicated GS1 prefix. One company uses the same manufacturer code. The 5-digit number limits all the manufacturers that there can be to 99,999. The next code, the product code, is assigned by the manufacturer. This code needs to follow the manufacturer code. The length of this code can be between 9 or 10 digits, depending on the length of the country code (2 or 3 digits). Figure 1.1 shows the EAN-13 number.

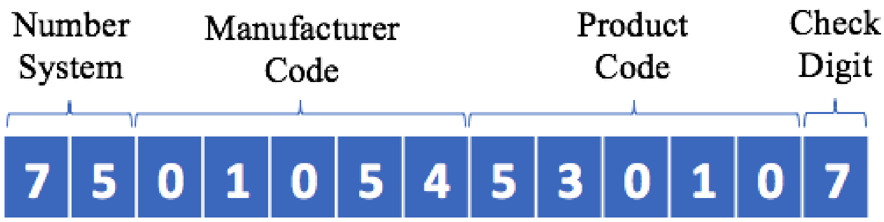

Figure 1.1: Example of the structure of a barcode in EAN coding.

1.1.2 Optical Character Recognition

Optical Character Recognition (OCR) is a system to recognize printed or written text characters using a computer. This technology was used for the first time in the 1960s. It uses a photoscanning of the text, character by character. After that, the system analyses the scanned image and translates the character images into ASCII character code. Nowadays, OCR is used in production, service, and in banks for the registration of cheques. It is also used to check processes, and sort credit cards by the received mails.

One of the advantages of OCR is the high density of its information and it is able to read data visually in the emergency. An example of OCR is shown in Figure 1.2.

This technology is not acceptable for universal use because of its high price and the complex reader systems required.

Figure 1.2: Example of the use of OCR technology [World 15].

1.1.3 Biometric Procedures

Biometric procedures measure particular human characteristics. They are used to identify people by comparing their unique physical characteristics. In practice, the most used systems are fingerprinting procedures, voice identification, and retina (or iris) identification.

Voice Identification is the identification of a person from the characteristics of their voice. This technology uses a microphone that collects voice signals and sends them to a linked computer.

Fingerprinting procedures recognize the lines on human hands organized in the fingertips. This technology has been used since the 1920s in order to identify criminals. Usually, the fingerprint readers are placed at an entrance to a building in order to verify the people who want to get in. Some modern systems require less than a second to recognize a fingerprint. An example of the fingerprinting procedure is shown in Figure 1.3.

Figure 1.3: Example of the use of biometric technology [Web 15].

1.1.4 Smart card

A smart card is a type of electronic data storage system. It has an integrated computing capacity and a microprocessor placed in a plastic card with a size similar to a credit card. The first smart cards were produced in the 1980s in the form of prepaid smart cards. This identification model needs to be inserted into a reader that makes a connection to the contact surfaces of the smart card. Using the embedded microcontroller, smart cards have the ability to store large amounts of data. Smart cards use high encryption model and mutual authentication. This technology has been very popular since the 1990s. The smart card market represents one of the fastest growing technologies in the field of AIS. The most popular forms of smart cards are memory cards and microprocessor cards. Memory cards operate using a sequential logic and can also incorporate security algorithms. They are very efficient and have a low price. Another form of smart card is the microprocessor card, which contains a microprocessor connected to some memory. These flexible cards are used in security-sensitive applications. A comparison of these AI systems is shown in Table 1.1.

Technology	**Barcode**	**OCR**	**Biometric IS**	**Smart card**	**RFID**
Line of sight	Yes	Yes	Yes	Yes	No
Degradation	Limited	Limited	/	Contact	No influence
Data density	Low	Low	High	Very high	Very high
Machine readability	Good	Good	Expensive	Good	Good
Readability by people	Limited	Simple	Difficult	Impossible	Impossible
Integration complexity	Low	Medium	Very high	Low	Medium
Reading speed	$\sim$4 s	$\sim$3 s	$\sim$5–10 s	$\sim$4 s	$\sim$0.5 s
Distance from the reader	0–50 cm	$<$1 cm	Contact	Contact	0–10 m
Typical data quantity	1–100 B	1–100 B	/	16–64 kB	16–64 kB
System cost	Low	Medium	Expensive	Low	Medium

Table 1.1: Comparison of different AI systems showing their advantages and disadvantages [TSL 17].

1.2 RFID technology

RFID technology belongs to the field of AIS. It is relatively similar to the smart card technology, where the data is stored in an electronic data device, the transponder [Finkenzeller 10]. Unlike a smart card, RFID does not use galvanic contact to exchange the data between the reader and transponder. It uses magnetic or electromagnetic fields. In RFID technology, the information is carried by radio waves. This technology uses low power in order to transmit and receive information between the reader and transponders attached to a person or an object. RFID represents one of the most pervasive computing technologies due to its low price and broad usability. These features have brought this technology into mainstream applications. Nowadays, RFID can be found in areas such as retail chains, warehouses, industry, logistics centres, etc. One of the main advantages, unlike the traditional bar code technology, is that it enables identification at a distance. However, RFID can provide much more.
In the following, a brief review of the history of RFID, the system components that participate in its operation, its features, coupling, and frequencies will be given.

1.2.1 A brief history of the RFID technology

The first traces of RFID date back to World War II, when some countries began to use the radar technology discovered in 1935 by the Scottish physicist Sir Robert Alexander Watson–Watt in order to warn land forces of approaching planes while they were still far away. Later, during the 1970s, RFID technology was used by the military to control access to some important areas, such as nuclear plants and weapon stockrooms. The first implementation of reflected power (backscattering) RFID tags was by Stev Depp, Alfred Koelle, and Robert Frayman at the Los Alamos National Lab in 1973. This technology used only passive and semi-passive tags. It contributed to significantly reduce the price of the tags and maintenance of the whole system. The first patent associated with RFID was granted in 1983. Standardization for the interoperability of RFID began in the 1990s. The new century has contributed to improve this technology, which has led to its miniaturization, while the cost of the RFID has constantly continued to fall. In 2004, the MIT Auto-ID centre started to promote the EPC standard [EPCglobal 8]. In the most recent period, from 2005 on, RFID technology has been widely used in every industrial and private sector with a high percentage of usability [Landt 05]. Nowadays, RFID technology is used widely by customers and companies.

1.2.2 The Global RFID Industry

The IT industry had a boom in the previous century and in the present one. There has been a similar situation with RFID: it has grown rapidly, exceeding 10 billion $US in 2015. Analysts estimate this will surpass 17 billion $US in 2020. Nowadays, this global industry is concentrated in European and American markets. In 2015, these areas represented nearly 70% of the mature applications of the global market of RFID. Development of RFID depends of the whole industry chain. A current issue is that the production of RFID chips is monopolized by companies such as Alien Technology, NXP, and Freescale. In terms of RFID transponders, China is in the leading position and made the largest production base, over 70% of the world's total. The increasing need for RFID technology is omnipresent in the healthcare sector. This technology will improve supply chain efficiency and ensure patient safety. Also, human errors in healthcare processes will be drastically reduced. The use of RFID has increased in larger companies, such as Airbus, Boeing, and the U.S. Department of Defence. Aerospace companies use RFID in order to improve the supply chain and enable efficient logistics and manufacturing operations. Inevitable integration of RFID is omnipresent in the bigger markets, like Wal-Mart. Wal-Mart replaced barcode technology with RFID and enhanced its efficiency.

1.2.3 Components of an RFID System

RFID systems consist of the following components: the reader, a back-end database, and the tags [Floerkemeier 07]. The reader is a control unit with one or more antennas. This device interrogates the tags to transmit their data. It consists of a radio frequency (RF) module, a control unit, and a coupling element to wirelessly communicate with tags at short distances. Readers are able to communicate with an application subsystem using an interface in order to transfer data between the application software and the tags. Some examples of readers in real scenarios are

presented in Figure 1.4. The back-end database is intended to store and further process the data about the identified tags, in a database.
Every RFID system contains one or more transponders (tags). Every tag has a unique identification code (ID) and includes an integrated chip and an antenna, and can be attached to objects. They are made up of an integrated circuit (IC), an antenna, and a substrate. Using a coiled antenna, tags can receive and transmit RF queries using an RFID transceiver, basic modulation circuitry, and non-volatile memory that stores the unique data.

Figure 1.4: Types of commercial RFID readers [ElectroSome 17].

In accordance with their power supply, tags can be categorized into three groups: active, passive, and semi-passive.

- Active tags contain an integrated battery and can initiate communication. These tags can be identified from a greater distance than passive tags. This type of tag has more features than passive tags, e.g., in terms of data storage and sensor capabilities. An active tag has a limited lifetime because of the integrated battery: when it is exhausted, the tag is unusable. For this reason, active tags are not efficient for disposable consumer products.

◇ Passive tags do not contain a battery: their power is supplied by the reader. They are energized by harvesting energy from the reader. Passive tags are used when low cost and long lifetime are sought, and provide lower coverage. Thus, the tags' hardware is simpler. This process will be explained in the following subsection. Some shapes of tags are shown in Figure 1.5.

◇ Semi-passive tags have an integrated power supply but it is used only for the internal control circuitry. They use power from the reader in order to transmit data. A comparison of active and passive tags is shown in Table 1.2.

Figure 1.5: Different construction formats of RFID tags [ElectroSome 17]

Criterion	Active tags	Passive tags	Semi-passive tags
Power source	Built in reader	Provided by the reader	Battery on tag for chip operations
Availability	Continuous	Within the field of the reader	Within the field of the reader
Signal strength (Reader to tag)	Very low	Very high	Low
Signal strength (Tag to Reader)	High	Very low	Low
Operating range	$>$100m	$<$7m	$<$10m
Price	10–50 $US	0.05 US$	10–50 $US
Communication Principle	Neither backscatter nor inductive coupling. Tag generates electromagnetic waves on their own.	Either inductive coupling or backscatter (Near or Far Field)	Backscatter (Far Field)

Table 1.2: Comparison of active and passive tags [TSL 17].

1.2.4 Frequency, Range and Coupling

The most important features of an RFID system are the operating frequency of the reader, its coupling methods, and the range of the system. RFID systems operate in low frequency (LF), high frequency (HF), and ultra-high frequency (UHF) bands. Every RFID system that uses different frequency range has advantages and disadvantages.

RFID systems that operates in LF (30–300 KHz) have the slowest data rate (DR) but have very good capabilities for operating near metal or liquid surface. Typically, LF RFID systems operate at 125 kHz. Using this range, tags can be identified up to 10 cm and the system has a slower reading speed.

The HF band ranges from 3 to 30 MHz. Most HF RFID systems operate at 13.56 MHz with a read range from 10 cm to 1 m. Systems that operate under this frequency are used for ticketing, payment, and data transfer applications.

The UHF frequency band covers the range from 300 MHz to 3 GHz [Barthel 06]. The read range with the use of these frequencies is up to 12 m and has a faster DR than systems which operate at LF and HF. Also, UHF systems have a very important advantage: the price of the tag is lower than 1 $US. A comparison by frequency range is shown in Table 1.3.

RFID systems use two coupling methods: inductive coupling and electromagnetic backscattering or far-field propagation. These coupling methods are shown in Figure 1.6. Using inductive coupling, the reader forms a magnetic field between itself and tags. It supplies the tags with power from the magnetic field.

Passive and semi-passive RFID do not use radio transmission as used by active tags, instead they use a modulation of the reflected power from the reader antenna that emits electromagnetic energy. All the tags reflect back some of the power transmitted by the reader, but change some of its properties, and in this way send back data to the reader. This is called backscattering. The microchip uses the energy to change the load on the antenna and reflect back an altered signal. In far-field propagation, the reader transmits a signal to the tags and they backscatter a response back to the reader. This propagation enables the reader to identify tags at longer distances than with inductive coupling. Passive and semi-passive tags modulate some of the energy being transmitted by the reader in a backscattering process. They use a Continuous Wave (CW) signal from the reader and change the loading of the antenna from absorptive to reflective.

Frequency	Key applications	Standard
125 kHz (LF)	Low price of passive RFID tags for animals	ISO 18000-2
13.56 MHz (HF)	Low price of passive RFID tags for objects (library book, clothes, etc.)	ISO 14443
400 MHz	For remote control for vehicle centre (locking systems)	ISO 18000-7
868 MHz, 915 MHz, 922 MHz (UHF)	For logistics - active and passive RFID tags (Europe, U.S. and Australia)	Auto-ID Class 0 Auto-ID Class 1 ISO 18000-6
2.45 GHz (microwave)	An ISM band -active and passive RFID tags (with temperature sensors, GPS)	ISO 18000-4
5.8 GHz (microwave)	Used for long read range for active and passive RFID tags (vehicle identification, highway toll collection)	ISO 18000-5

Table 1.3: Frequencies used in RFID systems [TSL 17].

1.2.5 Security and Privacy Issues

It can be said that RFID systems are subject to many attacks, from attacks on the physical layer to attacks on the application layer. Physical attacks can be very simple, such as interfering with the identifying tags using aluminium foil. In this case, the reader will be unable to read a tag attached to some objects (usually in supermarkets). More sophisticated physical attacks are jamming attacks, which can permanently damage radio devices. However, hackers have one main focus on breaking the identification/authentication schemes using weaknesses of implemented algorithms. Also, they are able to clone or modify information integrated in the tag. The overriding attack on RFID systems is the spoofing or impersonation attack. In this attack, the hacker is able to clone information from a tag without physical replication. A lot of attacks have been put an end to, but there still exist open issues when security is considered.

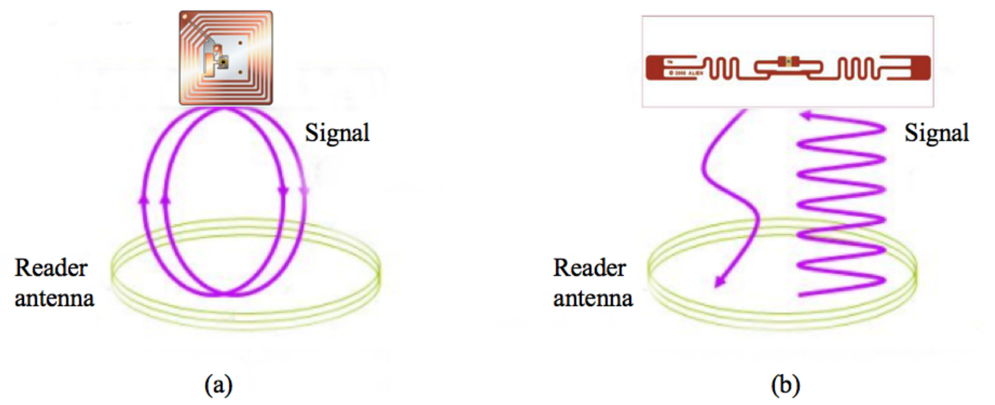

Figure 1.6: The coupling methods: (a) inductive coupling, (b) electromagnetic backscattering.

The main privacy concerns in RFID technology are the tracking of people and their locations, and the tracking of customers and their habits by retail companies. One of the main issues in RFID is providing a sufficient level of privacy. This is similar to the tracking of animals and articles, the real threat of abuse of this technology is also referred to people. Tags are small enough to be inserted under human skin or to be integrated into the clothing. RFID technology becomes more and more advanced every year, and people suffer from higher possibilities of being tracked. For instance, Wal-Mart integrated RFID chips into every product and after purchase the customer can be tracked. These issues maybe can be eliminated by finding a chip in the clothes and destroying it. Greater problems will occur with passports. The U.S. government made a decision to integrate an RFID in every passport and simplify the process of identification at the border. This passport will be protected and no one can do anything with the chip. It enhances the possibility of tracking the person. This feature can also help terrorists and thieves to determine the traveller's identity. This privacy issue can be solved by the following approaches: encryption, all information will be encrypted; access control, the key to decrypt the data will be encoded in the passport and could be obtained by scanning with an optical reader; the passport covers will be covered by a metallic mesh that creates a Faraday cage and this method will neutralize identification when the passport covers are closed. It can be concluded that every person can control the effect of this technology on their privacy.

1.3 The tag collision problem

In most RFID systems, several tags can be found within the range of the antenna. These tags may transmit information to the reader at the same time using the same communication channel. This causes the reader to be unable to decode their responses, forcing them to retransmit their messages until they are received correctly. This leads to an increase in the time needed for any identification, as well as the energy consumed in the process, and is called the tag collision problem [Leong 05], [Klair 10], [Abraham 2]. Figure 1.7 shows an example of the tag collision problem, where the reader tries to receive the data from all the tags in the interrogation area.

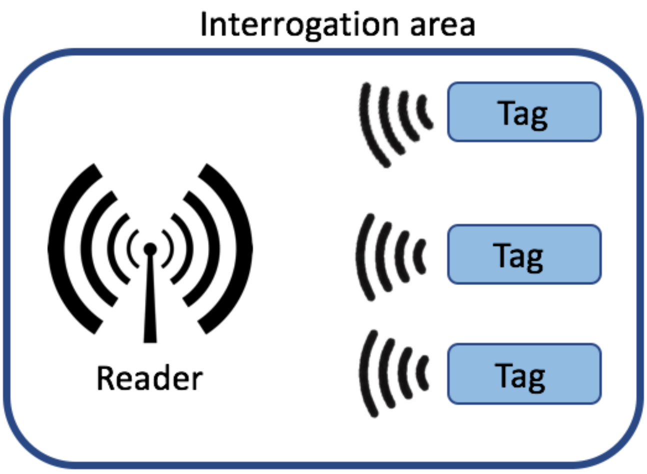

Figure 1.7: The tag collision problem.

In order to solve this issue, a multi-access method needs to be adopted. However, RFID possesses several properties that make this problem unique:

- ◇ Tags are usually simple devices, do not have a power source, and operate with low computational capacity.
- ◇ The number of tags is initially unknown to the reader.
- ◇ Tags cannot communicate with each other since their responses are so weak that only the reader is able to decode them. Therefore, the reader manages the whole process.

◇ Due to the limited resources of the tags, the protocol needs to be computationally simple at the tag side.

All these features demand the creation of protocols, called anti-collision protocols, to solve this problem. These protocols try to resolve the collisions in a fast and energy efficient manner. One of the most common types of anti-collision protocol found in the literature are the tree-based protocols. These protocols use queries, which are bit-strings generated by the reader, that the tags compare and respond if they match them. There are two types of tags: those with and those without memory (memoryless). The main difference is that the response of a memoryless tag depends only on the query immediately transmitted by the reader, and on the ID of the tag, whereas memoryless tags use internal counters saved from one response to the other. This feature causes the reinitialisation of the identification procedure if the tag loses power. Memoryless protocols are, therefore, query based protocols, and the research presented in this thesis is based on this type of protocol. These protocols will be explained in Chapter 2.

Tag 1	0 0 1 0 1 0 0 1
Tag 2	0 1 1 1 1 1 0 0
Tag 3	1 0 0 0 1 0 0 1
Tag 4	1 1 1 0 0 1 1 1

Figure 1.8: Example of four tags in the interrogation area.

1.4 The tag ID and its associated problems

Every tag has a unique ID, which is stored in an RFID electronic chip with any additional information. The tag ID is a bit string of 0's and 1's that uniquely defines the tag. All the tags in the range of a reader's antenna form a group of tag IDs, collections of 0's and 1's, that together become a distribution of bit strings. This distribution can be homogeneous, i.e., the likelihood of finding a 0 or 1 in the same position of all the IDs is equal, or heterogeneous if there are ID parts repeated among the different IDs. The main issue in some anti-collision protocols, particularly in tree- and query-based protocols, is that their behaviour directly depends on the type of the distribution and on the length of the ID. Figure 1.8 shows four 8-bit tags with different forms of organization of their IDs.

1.4.1 The Electronic Product Code

The Electronic Product Code (EPC) is designed to provide a unique identity for every physical object in the world [GS1 11]. This standard is used in the information systems that need to track or perform someother operation that refers to physical objects. A huge number of applications use the EPC standard and are based on RFID tags as the data carrier [Technology 15]. The EPC can differentiate between two identical products, and a single EPC can provide particular product information (date of production, origin, batch number, etc.). The basic format is shown in Figure 1.9. The header presents the main information (length, type, structure, version, and generation of the EPC). The EPC Manager Number is the entity that is responsible for maintaining the subsequent partitions. The Object Class identifies the class of objects, and the serial number identifies the instance of that class. This way of organizing the tag IDs can cause tags stored in similar places to share a great part of their IDs, e.g. being differentiated by only one bit. The organization of the EPC produces heterogeneous distributions.

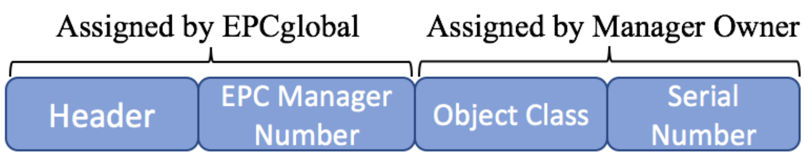

Figure 1.9: The basic format of EPC.

The protocols presented in the recent literature have usually been tested under homogeneous distributions. Most of the tree-based protocols, and particularly those based on queries, are severely affected by any heterogenity of the distributions. This means they are highly dependent on the distribution of the IDs of the tags and the protocol features are highly dependent on the parameters of such a distribution. In this thesis, an anti-collision protocol to deal with these problems is proposed.

1.5 Motivation, hypothesis, and objectives

The main idea for writing this thesis comes from the need to continue the work started in the thesis of Landaluce [Landaluce 4], where the window methodology was presented. That methodology was created with the aim of managing the number of bits transmitted by the tags. It makes effective use of an adaptive window, controlled by the reader, which restricts the total number of bits that are transmitted by the tags and decreases the total number of transmitted bits as well as the identification time. This methodology, however, has some drawbacks:

- ◇ The number of reader bits transmitted is considerably increased due to the increase of the number of slots needed to identify a tag.
- ◇ The protocol assumes that the tags are able to differentiate among the query and the window size on every reader command. This is, however, not possible since the length of the query and the window size are dynamic variables that would need additional information to indicate the sizes of the fields.
- ◇ The proposed solution helps to improve the performance of the growing number of current RFID applications, such as sensing and asset management and will not increase the complexity of the tags to be able to work in passive manner.

1.5.1 Hypothesis

Once the problem has been presented, the following hypothesis is proposed with the aim of solving it:

> ”It is possible to develop an RFID anti-collision protocol, using memoryless passive tags in order to reduce the energy consumption and decrease the identification time regardless the type of the tag ID distribution.”

In this thesis a tree-based anti-collision protocol will be presented. It is a memoryless protocol: the tags do not require any counter or memory. The tags are passive and use coupled power originating from the reader. The main focus will be on designing and implementing a tag ID distribution estimator that will be incorporated in a tree-based protocol in order to work efficiently with different tag ID distributions. This estimator reduce the identification time and energy consumption of a passive RFID system.

1.5.2 The objectives

This thesis will propose two approaches to alleviate the influence of the tag ID distribution and resolve other issues that affect the functioning of the protocol. Both approaches will operate with memoryless tags based on queries. The first approach proposes the standardization of the window size used by the protocol. This will solve the problem of differentiating the window size from the query in a reader command and also will decrease the total number of bits transmitted in the identification process.
The second approach can estimate the tag ID distribution. The estimation results will be used in order to calculate the ideal number of bits the tags must use when responding to the queries during the interrogation round. Furthermore, it will show the flexibility of its features, providing good results for different tag ID distributions. The secondary objectives that arise from the hypothesis and are needed for the fulfilment of the main objectives are:

- ◇ Analysing the existing anti-collision protocols and the window methodology to determine which processes can be improved.
- ◇ Development of a technique to standardize the window sizes in order to solve the problems encountered in the window methodology. It will also need to decrease the total transmitted bits in the identification process.
- ◇ Design and development of a tag ID distribution estimator. An updating strategy based on the window methodology will allow the protocol to aggressively advance through the common parts of the IDs during the identification process.

◇ Development of an anti-collision protocol using the tag ID estimator to work under different tag ID distribution conditions.

◇ A testing environment needs to be configured that will be used for the comparison of the previous solutions with the one proposed in this thesis.

◇ After performing the simulations, the results will be evaluated in various aspects. The main protocol criteria for the evaluation will be the reduction of the identification time and the energy consumed by the reader. Therefore, the bits transmitted by the tag and the reader will be analysed as well as their influence on the previous measurements.

1.6 Methodology of the investigation

RFID technology is progressing every day, with accelerated growth [Idtechex 15]. The methodology of the investigation needs to be appropriate to this progress. Here, the methodology followed in this research is presented.

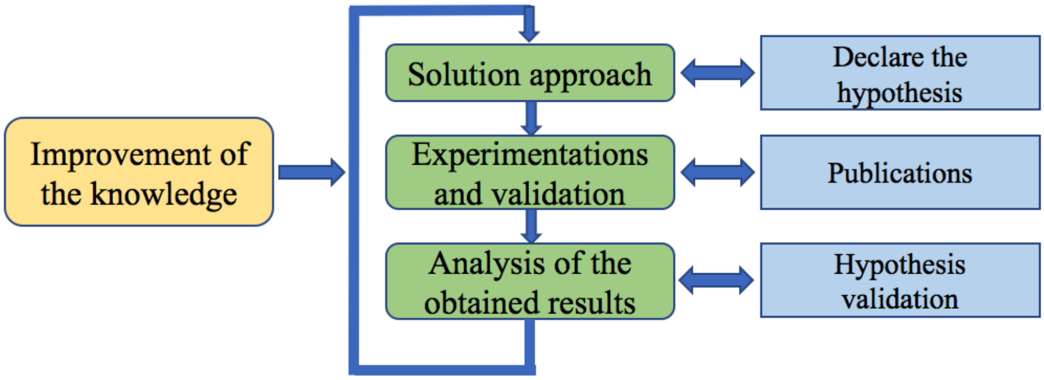

Figure 1.10: Methodology used in this thesis.

The methodology presented in Figure 1.10 can be summarized in the following points:

◇ The knowledge in this field can be improved by using the related literature and all the publications of the scientific community. This allows determining the limitations of the already existing protocols.

◇ Design a solution that will solve the problems defined in this thesis. The hypothesis and objectives need to be defined that make connections between the problems and the potential solutions.

◇ Implementation of the designed procedures.

◇ Experimentation and validation need to be performed in order to compare the obtained results.

◇ Analysis of the obtained results. This will justify the hypothesis presented.

◇ Validation in scientific publications in order to present the investigations to the scientific community.

◇ Spreading of the knowledge obtained during the whole investigation process, contributing to the scientific community.

1.7 Structure of this thesis

This subsection presents the structure of this thesis. The thesis consists of the following chapters:

◇ Chapter 1. This chapter introduces the thesis and provides a basic understanding of RFID technology, presenting the main components of every RFID system, its operation, types of tags, methods of transferring data, etc. The explanations in this chapter help to better understand the benefits of this technology as well as its disadvantages. RFID technology is introduced as the basis for presenting the tag collision problem that is addressed in this thesis. Additionally, the main hypothesis and the objectives are presented.

◇ Chapter 2. This chapter provides explanations of some of the most important anti-collision protocols in RFID systems. It gives a basic classification of these protocols in order to better understand the different existing strategies used to solve collision problems. In this chapter, the Aloha-based protocols, the tree-based protocols, and hybrid protocols will be reviewed.

◇ Chapter 3. This chapter presents one of the main procedures of this thesis, window standardization. This window standardization has been created with the aim of differentiating the query from the length of the window. The proposed Standardized Query window Tree (SQwT) has the ability to locate the bit string representing the window size inside the reader command.

◇ Chapter 4. This chapter provides the explanations of the main contribution in this thesis, the Flexible Query window Tree (FQwT). It is a flexible anti-collision protocol with the aim of estimating the tag ID distribution and therefore providing improved results in terms of identification time and energy consumed by the RFID system, regardless of the ID distribution.

◇ Chapter 5. Here the hypothesis presented at the beginning is validated. The simulations provided compare the protocols from the literature and the proposed protocols in this thesis. This chapter also includes a discussion of the results.

◇ Chapter 6. Finally, this chapter ends the thesis and collects all the conclusions of the previous chapters. Also, this chapter proposes the next steps that research in this field may take.

The happiness of your life depends upon the quality of your thoughts, therefore guard accordingly.

Marcus Aurelius

2

State of the art

RFID is a technology that uses radio frequency in order to identify tags, which do not need to be positioned accurately relative to the reader [Finkenzeller 10], [Azambuja 08]. The operation of an RFID system involves the situation where numerous tags are sharing information in the interrogation zone of a single reader at the same time. RFID systems can differentiate two main types of communication. The type of the communication where the reader transmits data to many tags is called broadcast, and is shown in Figure 2.1. The second type, depicted in Figure 2.2, forces all tags in the interrogation round to transmit their data to the reader. This type is called multi-access [Finkenzeller 10], [Shih 06].
The reader sends out electromagnetic waves which the tag antenna receives and backscatters with its ID, converting the waves to digital data. In every interrogation round, the reader attempts to read a single tag from among a population of many tags. When more than one tag transmits to a reader, this leads to the cancellation of their respective information, and the resulting message is illegible. This tag state is called *collision* [Leong 05], [Klair 10], [Abraham 2]. This is a widely known topic in RFID research and it is called the 'tag collision problem'. Collision results in a loss of identification time and an increase of power consumption. To reduce the impact of the tag collision problem, RFID readers use an anti-collision protocol.

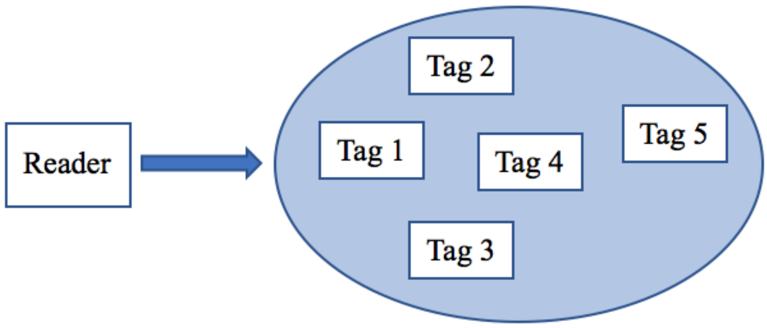

Figure 2.1: Broadcast mode: The reader sends data to all tags in the interrogation round.

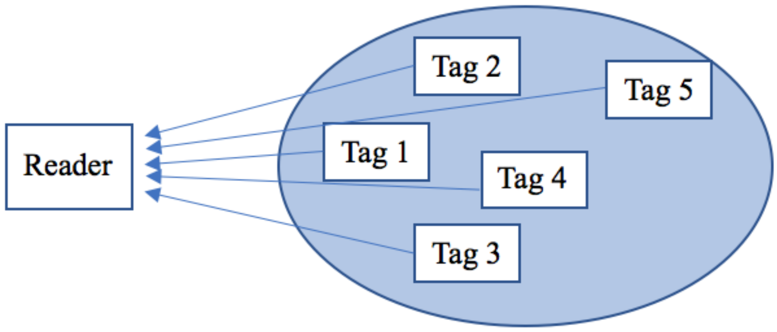

Figure 2.2: Multi-access form: All tags transmit data to the reader simultaneously.

Firstly, some basic terms will be given in order to properly understand the most closely related anti-collision protocols:

- Slot: The period of time that divides the tags' responses is called a slot. It includes a reader command and a tag response. During the identification process, and depending on the number of tag responses received by the reader, three types of slot can occur: collision, idle, and success. A collision occurs when more than one tag answers the reader's command in the same slot. When no tag responds to the reader's command, then an idle slot happens. A success occurs when just one tag is correctly read by the reader and, therefore, identified.
- Query: A broadcast command transmitted by a reader to require the tags to

respond. In case a tag's ID does not match the query, the reader command will be rejected.

◇ Identification process: This is the time period that includes a certain number of time slots or rounds that the reader needs to identify all tags in the range of its antenna.

◇ Latency: This is the time needed to complete an identification process.

◇ Efficiency: This measures the exploitation of the tags' responses and the influence of a collision on a tag response. It is calculated using the expression $\frac{n}{sl_t} \times 100$, where n is the number of tags, and sl_t denotes the total number of consumed slots.

Communication between a single reader and several tags is based on sending and receiving information. Every communication channel has a predefined capacity, which is specified by its maximum data rate and the range of its antenna. The available channel capacity must be divided between all tags in the interrogation zone in such a way that all the data can be transmitted from the several tags to a single reader without mutual interference. In early radio technology the problem of multi-access was noticed. Many participants tried to access a single satellite or base station. For this reason, many procedures have been developed with the aim of separating the individual participants' signals from one another.

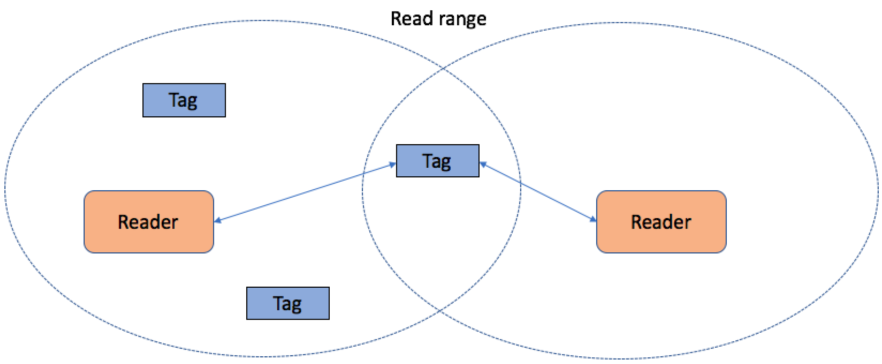

Figure 2.3: Reader collision.

2.1 The collision problem

The general problem of collision in RFID can be classified into two different situations:

- ◇ A reader collision occurs when the reader tries to make communication with tags that are in the coverage area of another reader. This type of collision is shown in Figure 2.3. This collision causes two different problems:
  - ○ Signal interference occurs when the fields of two or more readers overlap and interfere. This problem can be solved by programming all readers to read at fractionally different times.
  - ○ Multiple reads of the same tag occur when the same tag is read once by every overlapping reader.
- ◇ A tag collision occurs when more than one tag attempts to transmit its ID at the same time: the reader will receive a mixture of the tags' signals and cannot undersand it. This type of collision is shown in Figure 2.4.

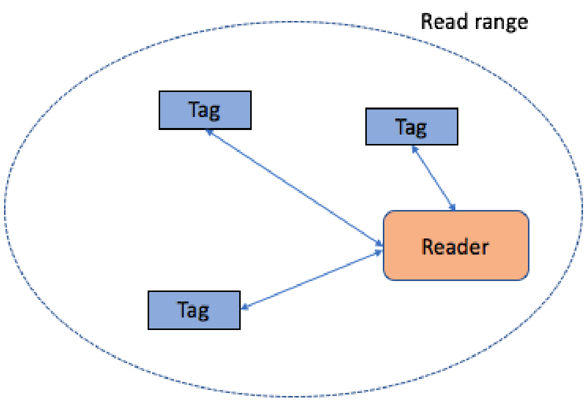

Figure 2.4: Tag collision.

Simultaneous responses from numerous tags prevent the reader from translating the signal correctly, which decreases throughput. No tag is aware of the activity of any other tag, and so they cannot avoid transmitting at the same time as other tags. The transmissions of three tags, shown in Figure 2.3, are not synchronised, but in many cases the reader is synchronised with at least one tag in the interrogation zone. In the presented illustration, the reader is prevented from decoding the entirety of the transmission and it has experienced a collision. In order to solve the mentioned problem, anti-collision protocols have an important influence.

2.2 Multi-access methods

Each anti-collision protocol uses some multi-access methods for identification in order to separate physically the transmitters' signals. Accordingly, they can be categorized into four different types: Space Division Multiple Access (SDMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA), and Time Division Multiple Access (TDMA) [Finkenzeller 10], [Burdet 04], [Klair 10]. Figure 2.5 shows various multiple access and anti-collision procedures.

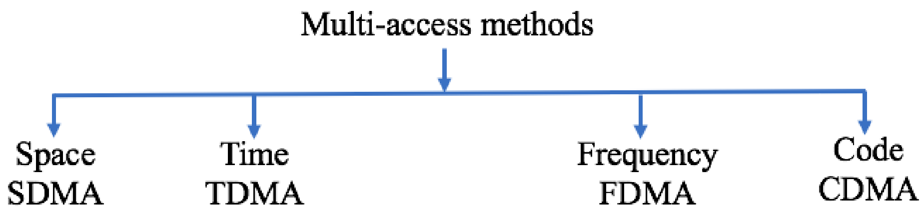

Figure 2.5: Multi-access methods.

SDMA – The term space division multiple access relates to dividing the channel capacity into separated areas. Protocols based on this method can point the beam at different areas in order to identify tags. The channel is spatially separated using complex directional antennas. Another way of achieving this is by using multiple readers. As a result, the channel capacity of adjoining readers is enhanced. A huge number of tags can be read simultaneously as a result of the spatial distribution over the entire layout. This method is pretty expensive and requires complex antenna design. The use of this type of method is restricted to a few specialized applications [Yu 08], [Sayeed 09], [Banks 05].

FDMA – Tags transmitting in one of several different frequency channels requiring a complex receiver at the reader. Consequently, different frequency ranges can be used for communication from and to the tags: from the reader to the tags, 135 kHz, and from the tags to the reader, in the range 433–435 MHz. However, this technique is expensive and is only intended for some particular applications [Liu 09].

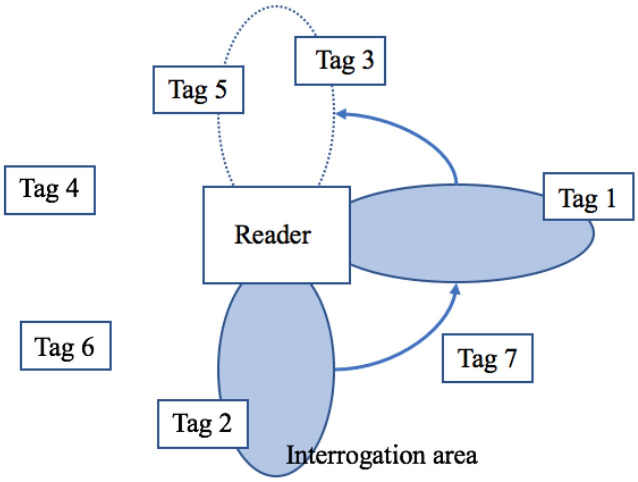

Figure 2.6: SDMA procedure.

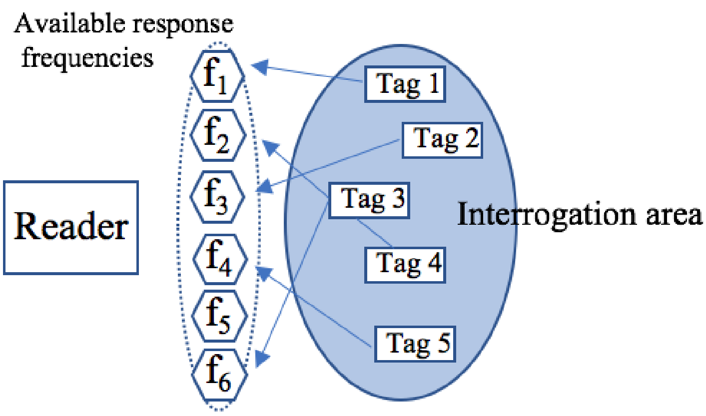

Figure 2.7: FDMA procedure.

CDMA – Requires tags to multiply their ID by a pseudo-random sequence (PN) before transmission. CDMA is very good in many ways, such as the security of the communications between the RFID tags and the reader, and multiple tag identification. It adds a lot of complexity and is expensive for RFID tags. Furthermore,

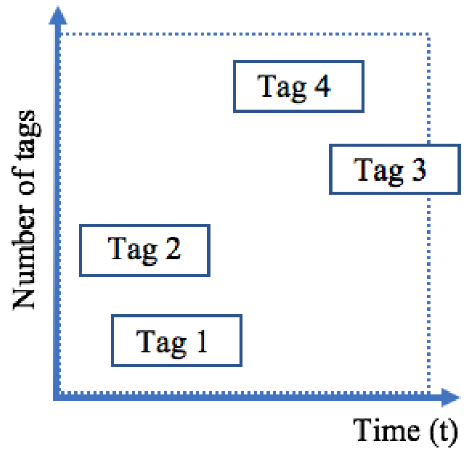

Figure 2.8: CDMA procedure.

this method consumes a lot of power and can be classified as a group with elevated demands [Loeffler 10].

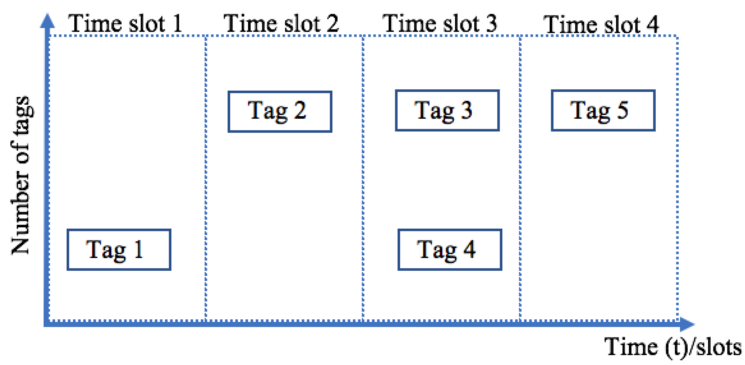

Figure 2.9: TDMA procedure.

TDMA – As it is less expensive, this method is the most widely used. This method involves the largest group of anti-collision algorithms. The transmission channel is divided between the participants and ensures that the reader can identify a tag at different times in order to avoid interfering with another one. The space-distributing characteristic of tags is not considered. The number of tags in the interrogation zone is reduced after every successful response. Another option involves the ability to mute all tags except the transmitting tag. After that, the tags are activated one by one [Tang 07], [Wang 09b].

In an RFID environment, anti-collision protocols typically use the TDMA method. Protocols that use this method first select an individual tag from a large group using a certain algorithm and then the communication takes place between the selected tag and the reader. Significant increases in the number of collisions in the identification process decreases the throughput and increases the number of transmitted bits. These protocols can be divided into three categories: Aloha-based protocols, tree-based protocols, and hybrid protocols (which use a combination of the first two methods). In the following subsections there will be presented some Aloha, tree-based, and hybrid protocols.

2.3 Aloha protocols

Aloha-based protocols use a random-access strategy in order to successfully identify the number of tags in an interrogation area [Wang 09a], [Schoute 10], [Vogt 02a], [Zhu 10b], [Maguire 09], [Zhu 10a], [Eom 08], [Chen 09], [Lin 10], [Li 11], [Vales-Alonso 11]. They belong to the group of probabilistic protocols because the tags transmit their own ID in randomly selected slots in a frame in order to reduce the possibility of a collision. However, it is not guaranteed that all the tags will be identified in the process of interrogation. These protocols suffer from the well-known tag starvation problem, in the sense that a tag may not be correctly read during a reading cycle due to an excessive number of collisions with that same tag. Every frame consists of a certain number of slots, and the tags can only respond once per frame [Schoute 10].
The main Aloha-based protocols can be divided into four subgroups: Pure Aloha (PA), Slotted Aloha (SA), Frame Slotted Aloha (FSA), and Dynamic Frame Slotted Aloha (DFSA) protocols.

2.3.1 Pure Aloha

Pure Aloha (PA) is one of the simplest anti-collision protocols. It is based on TDMA [Abramson 70], [Roberts 75]. Whenever tags enter the interrogation zone, they randomly choose a frequency on which to transmit their data. A collision will occur if several tags transmit data at the same time, and so there can be complete or incomplete collisions. A complete collision is when the messages of two tags collide entirely; an incomplete collision, however, is when only part of the tag message collides with another tag message. This procedure is shown in Figure 2.10 and will be repeated until all tags are successfully identified.

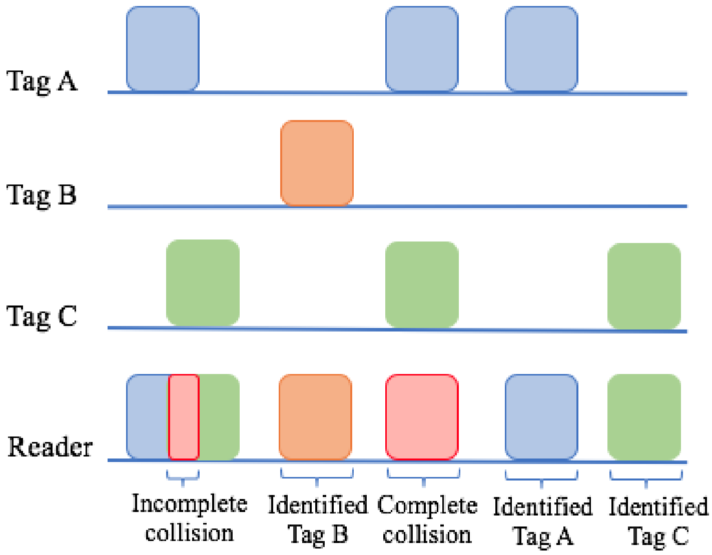

Figure 2.10: An example of PA.

PA has several variants:

- ◇ PA with muting. When this type of Aloha protocol is used, the total number of tags will be reduced after each successful tag response. As can be seen in Figure 2.10, tag B will be silenced by the reader, first going into "sleeping" mode and will not respond again [Burdet 04], [Klair 07].

- ◇ PA with Slow Down. This version of the Aloha protocol uses a "slow down" command instead "muting" command. The aim is to reduce the data communication rate. After a tag identification, the reader slows down the identified tag's frequency, resulting in the tag's adapting its random back-off counter to a decreased data rate [Burdet 04], [Klair 07].

- ◇ PA with Fast Mode. Every time the reader detects a tag at the start of transmission, it will send a "silence" command in order to stop the other tags from transmitting. Tags will resume transmitting only when the reader has sent the ACK command or until their waiting time has expired [Burdet 04], [Klair 07].

- ◇ Other variants involve combinations of these: e.g. PA with fast mode and muting, and PA with fast mode and slow down [Burdet 04], [Klair 07].

2.3.2 Slotted Aloha

In order to avoid incomplete collisions, Slotted Aloha (SA) has been created. In SA, the time is divided into several slots and each tag must randomly select a slot in which it will transmit its data [Abramson 70], [Burdet 04], [Klair 07], [Rohatgi 07], [Namboodir 12]. The communication between the reader and the tag is now synchronous. An example of communication with this protocol is presented in Figure 2.11.

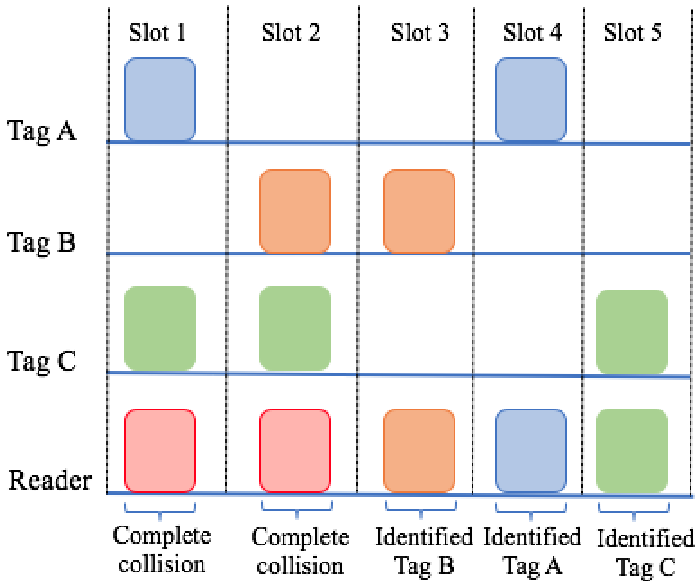

Figure 2.11: An example of SA.

Also, SA has some more variants:

- ◇ SA with muting/slow down – The identification process is similar to that of PA with muting/slow down, but in a slot manner.

◇ SA with early end. If the reader detects no communication at the beginning of the slot, the reader will close the slot early. The reader uses two commands: start of frame (SOF) and end of frame (EOF). The first command is used to start the identification process, and the second is used in order to close an empty slot earlier.

◇ SA with early end and muting. When the reader successfully identifies a tag, it will send a mute command, thereby reducing the number of transmitting tags. If the reader detects no response in a short time, it will also close the slot earlier using the EOF command.

◇ SA with slow down and early end. This variation combines the decrease in the data rate of the identified tags with the early end of a slot if no tag responds there.

2.3.3 **Framed Slotted Aloha and Dynamic Framed Slotted Aloha**

In Framed Slotted Aloha (FSA), the time is divided into a variable number of frames, and each frame consists of several slots [Bueno-Delgado 09], [Schoute 10], [Vogt 02a], [Vogt 02b], [Cha 05]. All tags need to transmit data into a fixed length frame, but each tag must choose only one slot in a frame to transmit data within. This protocol significantly reduces the probability of collision since tags can only respond once in a frame. Before the beginning of communication, the reader generates a random time less than the fixed frame size, to select just one slot in frame. If a collision occurs, the involved tags will again choose a slot in which to respond in the next frame. The main inconvenience of FSA is the slot wastage when the number of tags is small and the size of the frame is significantly bigger [T. W. Hwang 06] [Wieselthier 89].

In order to ameliorate this disadvantage, the Dynamic Slotted Aloha (DFSA) protocol has been developed [Yan 08], [Lin 10], [Schoute 10]. DFSA is capable of changing the frame size according to an estimate of the number of tags. At the beginning of each frame, the reader informs the tags of the frame length. Every tag selects a random number $[0, F-1]$, where F denotes the frame size and all tags respond within the number of slots. At the end of the frame, the reader estimates the number of colliding tags, then adjusts F accordingly. Tag estimation can involve some disadvantages, such as: increased computational costs in the identification process and errors that degrade the protocol's efficiency. Examples of these two protocols are shown in Figure 2.12.

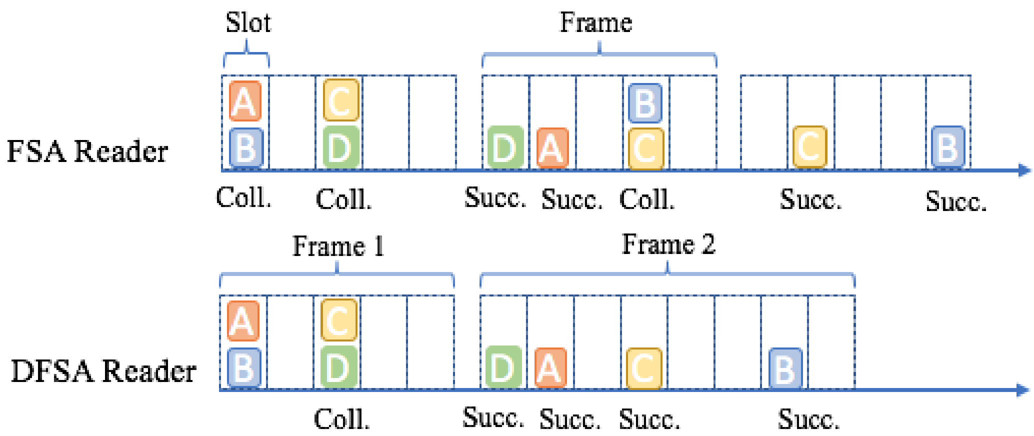

Figure 2.12: Example of FSA and DFSA.

2.3.3.1 **Q protocol**

The Q protocol is used in the EPCglobal Generation-2 (Gen-2) standard [EPCglobal 8], [Maguire 09]. The Q protocol is a DFSA type protocol that modifies the frame size using the feedback from the last frame accomplished [Wang 09a]. The Q algorithm can jump into the following frame without finishing the current one [Wang 09a]. The Q algorithm operates with two basic parameters: Q, and a constant c that can be modified depending on the situation [Namboodir 12] [Bueno-Delgado 09]. The variable Q is an integer ranging from 0 to 15. This protocol works with three types of commands:

- Query command, transmitted by the reader to all tags in the interrogation area in order to force all tags to choose a slot number (SN) from $[0, 2^Q - 1]$. This command initiates the identification process by providing a new value of Q.

- QueryAdjust is the command used to instruct all tags to increase, decrease, or keep unchanged the value of Q and reselect their SN. Q_{new} denotes the last calculated Q. Accordingly, Q could be increased by c, decreased by c, or left unchanged, according to the algorithm.

◇ QueryRep is used in order to notify all tags to decrease their SN by 1.

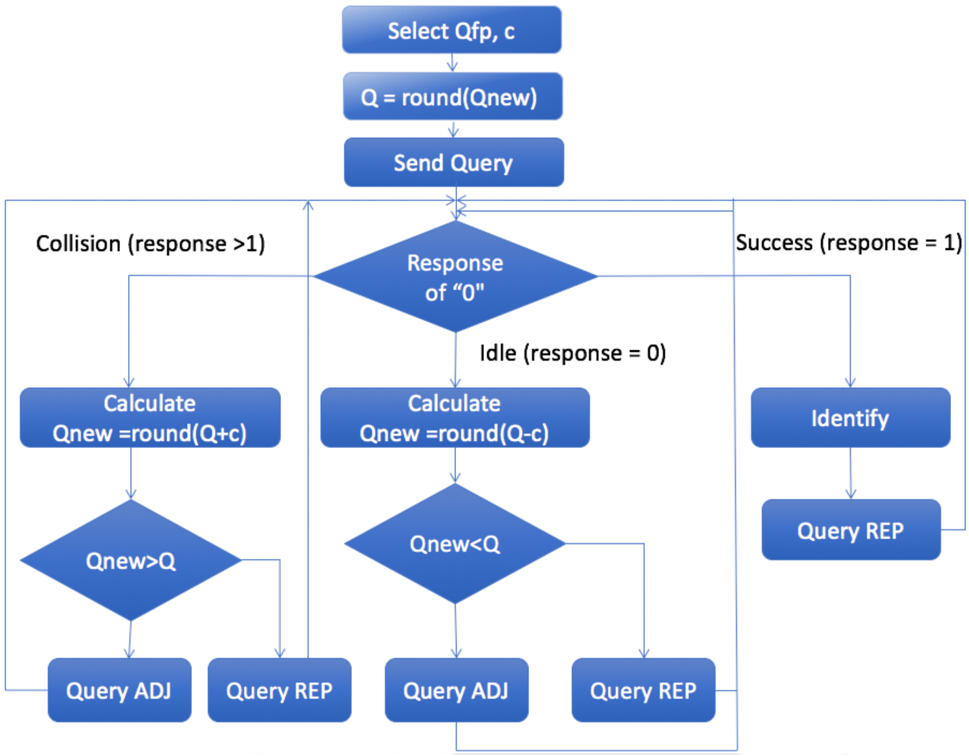

Figure 2.13: Flow chart of the Q protocol.

The procedure of the Q protocol is given in the flow chart in Figure 2.13. If it is time to start a new inventory round, the reader will transmit a Query command. If the tags receive *Query* or *QueryAdjust*, they need to choose SN from $[0, 2^Q - 1]$. If they receive a *QueryRep*, all unidentified tags decrease their SN counter by 1. Only tags with SN=0 will generate a 16-bit random number (RN16) and respond with an RN16 to the reader. There are three posibilities, depending on the tags' response:

◇ Successful reply. If only one tag responds and the reader successfully received RN16. Subsequently, the reader sends an acknowledgment command (ACK) and only the tag that successfully responded recognises the ACK and reports its EPC to the reader.

- ◇ Collided reply. In case more than one tag transmits RN16, a collision occurs. Then, the tags will increase Q_{new} by the constant c. Typical values for c are $0.1<c<0.5$. The value of c is adjusted according to the type of application. Higher values of c will provide more aggressive frame adjustments.

- ◇ No reply. When no tags respond in the slot, the reader decreases Q_{new} by c.

A similar protocol solution provides different updating steps for no-reply and collided reply by calculating the probabilities of idle slots and collided slots, without considering the parameters in Gen-2 [Lee 07], [Cha 05].
Table 2.1 summarizes the observations regarding the Aloha protocols.
rotating,booktabs,tabularx,ragged2e

2.4 Tree-based protocols

One of the main features of tree-based protocols is that they are deterministic, since they are ideally designed to identify the whole set of tags in the interrogation area [Law 00], [Finkenzeller 10], [Peeters 10], [Yeh 11], [Myung 06], [Gou 10], [Zein 10], [Lai 09], [Lai 12], [Lai 10], [Yang 11], [Hush 98]. These protocols have simple design tags and work very well with a uniform set of tags.
Tree-based protocols usually work with a muting capability since they need to have identified tags stay quiet after their identification. First, the most popular tree-based protocols are presented here. Then, a selected group of protocols will be presented with the common feature of the use of Manchester coding. The first group includes: Query Tree (QT), Query Window Tree (QwT), and Smart Trend Transversal (STT). Another group consists of tree-based protocols that use Manchester coding: Binary Search (BS), Collision Tree (CT), Optimal Query Tracking Tree (OQTT), and Collision Window Tree (CwT).

2.4.1 Query Tree protocol

The query tree protocol (QT) is one of the most representative memoryless protocols, where the reader must provide the tags with a query and the matching tags must respond with their full ID [Law 00]. Tag response directly depends on the current

Protocol name	PA	SA	FSA	DFSA	Q
Protocol feature	Tags transmit after random time to the reader. In the case of a collision tags will retransmit after a random delay.	Tags transmit their ID in synchronous time slots. In case of a collision, tags retransmit after a random delay.	Each tag responds only once per frame.	Tags transmit once per frame. The reader uses a tag estimation function to vary the frame size.	The reader dynamically adjust critical parameter (Q) based on the type of replies from tags.
Disadvantages	In a dense tag environment the the number of collision increases significantly.	In a dense tag environment the the number of collision increases significantly. The reader requires synchronization with tags.	It uses a fixed frame size and does not change the size during the identification process.	It cannot move into the next frame at any time based on the situation of collision without finishing the current frame.	This protocol may encounter some problems (lower throughput) on adjusting Q, especially when the frame size is larger than the number of tags.
RTF/TTF	TTF	RTF	RTF	RTF	RTF
Throughput	18.4%	36.8%	42.6%	42.6%	
System cost	Very low	Low	Expensive	Very expensive	Very expensive
Complexity	Very simple	Simple	High	Very high	Very high

Table 2.1: A comparison of Aloha protocols

query, ignoring the past history of the communication. QT tags involve only simple hardware requirements because they only compare the reader query with their own ID and respond if it coincides. The identification process consists of more rounds in which the reader sends a query, and tags whose ID prefix match the current query respond with their whole ID binary value. In the case of a collision, the reader forms two new queries by appending q with a binary 0 or 1. New queries will be placed in a Last Input First Output stack (LIFO). If there is no answer to a query, the reader knows that there is no tag with the required prefix, and the query is rejected. This kind of slot is called idle. If just one tag responds to the reader query, that tag will be identified. By extending the query prefixes until only one tag's ID matches, the algorithm can identify the rest of the tags. The identification procedure is completed when the LIFO stack is empty.

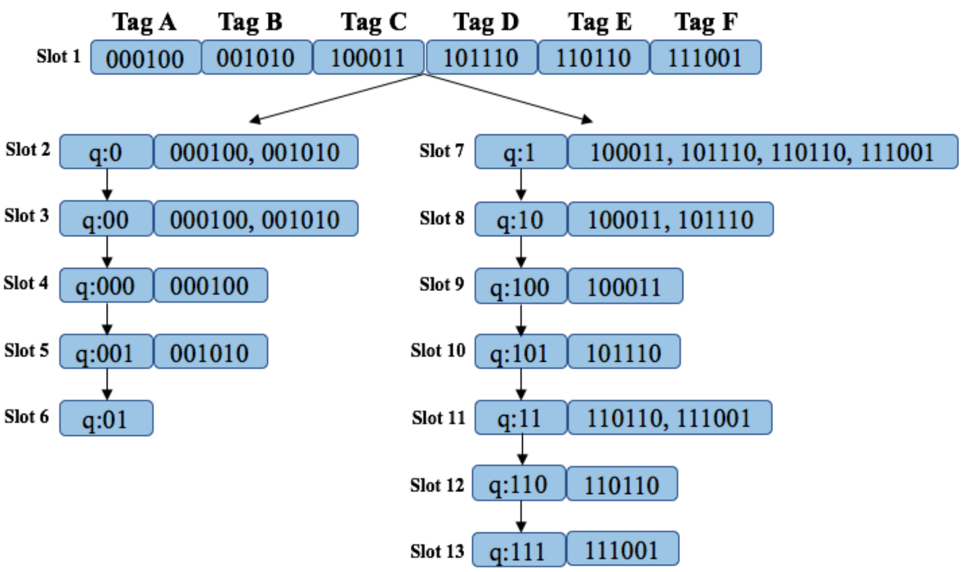

Figure 2.14: Example of the QT protocol.

Figure 2.14 shows the QT protocol being used to read 6 tags (Tag A – Tag F). Each tag uses an ID length of $k = 6$ bits. At the beginning, the LIFO stack is empty, and the reader starts with the null string. After a collision occurs, the reader pushes queries 0 and 1 into LIFO stack. During the second round, the reader pops from the stack and transmits query 0. In the example in Figure 2.14, tags 000100 and 001010 match the required prefix, which causes both to transmit and collide. The reader is unable to understand the messages from the tags. The reader then pushes into the stack queries 01 and 00. In the next round, the reader transmits query 00. Again, both protocols respond with their ID and a new collision occurs. In the stack, the following new queries are added, 001 and 000. The reader transmits query 000 and just one tag responds (000100). This tag is successfully identified and will not answer any of the following reader requests. The reader then transmits query 001 in slot 4, which matches tag 001010. In the next round, the reader pops and transmits query 01. For this query, there will be no response since there are no tags with that prefix. In round 7 the reader transmits the query 1 and the tag from the right side of the tree responds. Four tags will receive this query and a new collision occurs. The reader experiences a collision, since tags 100011, 101110, 110110 and 111001 responded to the query 1. As a result, queries 11 and 10 are pushed onto the stack. The identification process is repeated until round 13, in which the reader transmits the last query (111) from the stack. Overall, the reader uses 13 rounds to read 6 tags. In the literature there have also been proposed numerous extensions to the QT protocol [Choi 06], [Myung 06], [Zhou 04], [Choi 07], [Bhandari 06].

2.4.2 Smart Trend Traversal protocol

The Smart Trend Traversal protocol (STT) is a deterministic and memoryless protocol, created with the aim of reducing the number of collisions in the QT protocol [Pan 11].

This protocol has the ability to dynamically issue queries according to an online learned tag density and distribution. It proposes a combination of the QT protocol and the shortcutting method in order to skip a query which results in collision. When the protocol detects the potential possibility of a collision, it will avoid it and move to the bottom level of the binary query tree. STT provides trend recognition.

The reader keeps track of the tag density and distribution in order to issue the subsequent queries, and consequently, minimizes the number of empty slots and collision slots.

In this protocol, it is not necessary to have any prior knowledge of the network, and it outperforms the existing protocols. The ideal number of queries can be the total number of single nodes. The ideal queries group, referred to as the query traversal path (QTP), is denoted by $Q = q_1, q_2, q_3, \ldots, q_n$, where q_n is the last query used in the identification process [Yan 12].

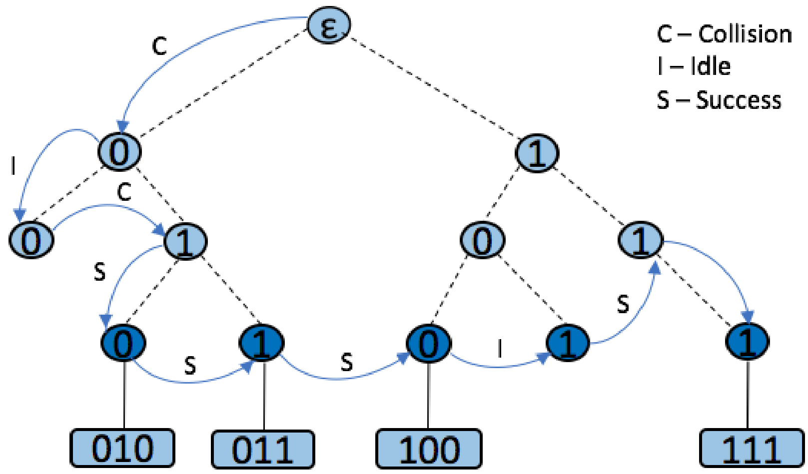

Figure 2.15: An example of STT protocol.

It is difficult to achieve, but it is desirable to get close to its value. The reader can calculate the subsequent queries depending on the tag response, which can be classified into three types:

- ◇ A collision occurs when QTP is at too high a level and should be move down by adding a longer prefix to the query. Consequently, the reader appends t bits of 0's to the last query, where $t = s + n_{col} - 1$. Let s denote the minimum increase, and n_{col} be the number of consecutive colliding slots.
- ◇ An idle slot occurs when no tag responds to a reader query. QTP needs to traverse up just one level, which can lead to a new collision. This rule will be applied only to the right side. If the empty response comes from the left side of the tree, QTP must move horizontally to the right. The reader will decrease the query length by m bits, where $m = s + n_{emp} - 1$ and n_{emp} is the number of consecutive idle slots.
- ◇ Upon a successful response, a single node is visited, indicating that the tag has been identified successfully by the reader. Then QTP moves to the symmetric

node if the query finishes with 0, but it returns one level if the query finishes with 1.

The identification process of the STT protocol, which was explained above, is depicted in Figure 2.15 with 4 tags.
In conclusion, STT significantly reduces the number of collisions, the identification time, and the energy consumption, compared to the existing Aloha-based and tree-based protocols.

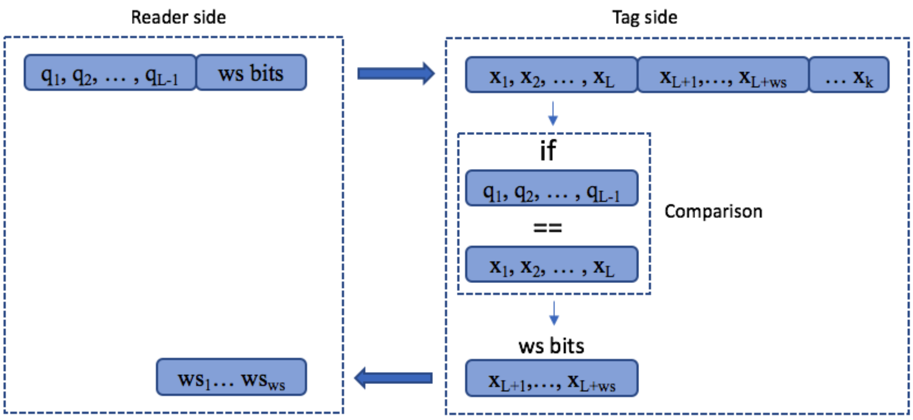

Figure 2.16: Example of a communication slot between the reader and one tag.

2.4.3 Window based protocols

In the majority of tree-based protocols, tags respond with their full ID or with the bits from the last query, when the query sent by the reader matches the tag ID prefix. In Figure 2.16 an example of a communication slot between the reader and the tag is presented. In order to reduce the number of bits transmitted by the tag, a window method has been proposed [Landaluce 13], [Landaluce 14], [Landaluce 16]. In the identification process, a lot of slots end up colliding, and this contributes to a huge waste of bits. Protocols that use the window method reduce the number of bits transmitted by the tags. The window is defined as a bit-string of length *ws* bits transmitted by a tag in a slot. This bit-string is computed at the reader side, respecting the condition $0<ws<k$. It is shown in Figure 2.17. Most tree based protocols use a fixed tag response during the identification process, but some

protocols use different methods based on an operational process with a dynamic response that is based on window synchronization.

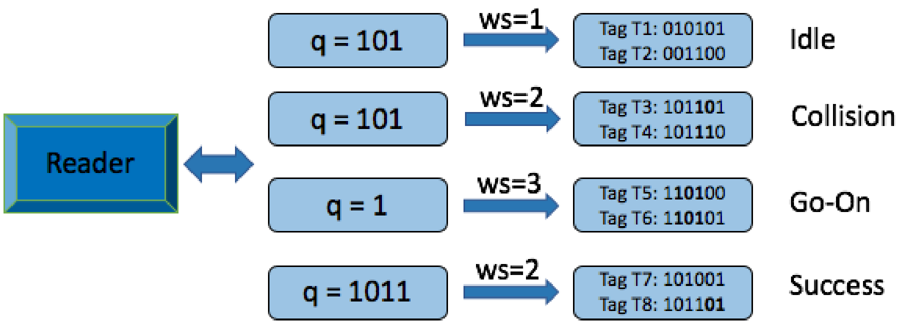

Figure 2.17: Window synchronized answer.

2.4.3.1 **Query Window Tree protocol**

The Query window Tree protocol (QwT) is a memoryless tree-based protocol that applies a dynamic bit window to QT [Landaluce 13] [Landaluce 16]. Tags respond directly depending on the current query. QwT tags compare their ID value with the query received and transmit a certain amount of bits, managed by the reader. This reduces the complexity of passive tags, their energy consumed, and the identification time. A reader and tag flow chart of QwT are given in Figure 2.18(a) and (b). When tags appear in the interrogation area, the reader will broadcast to them by transmitting a query length of L bits. Tags will respond if their ID prefix matches the query sent by the reader, but with the previously specified amount of bits. One of the main features is that the total number of collisions is decreased by transforming potential collisions into partial successful slots. This is a new type of slot, called go-on slots. The previously explained window methodology is implemented in the QwT protocol. The window allows tags to transmit just the bit-string instead of their full ID. If tags match a reader query, they will synchronously transmit the next adjacent ws bits of the ID. This protocol uses cyclic redundancy check (CRC) in order to differentiate between the types of tag responses. Accordingly, the slot types that can occur in the QwT protocol can be classified into 4 groups:

◇ Collision slot. When the reader cannot differentiate the answer, the reader will create two new queries by appending '0' and '1' to the former query [q_1,q_2...q_L]. The window size *ws*, will remain unchanged, with the value used in the previous query.

◇ Idle slot. When there is no response, the reader will discard the query and retain the same ws from the last command.

◇ Go-on slot. This occurs when at least one tag responds with a window and the reader is able to understand it. If *L*+*ws*<*k* is not true, the reader will transmit a new query created from the former query and received window. During this query the reader will append an updated ws value.

◇ Success slot. This is a type of go-on slot where the reader successfully receives the last part of the tag ID and *L*+*ws*=*k*. Then the reader can save the tag, calculate the new ws, and continue with the identification process.

Using the QwT protocol, the reader computes ws using the expression (2.1), where β is an adjustable parameter. This heuristic function is employed in order to provide dynamism to the value of ws. It can only be applied to the go-on and Success slots, since in a Collision or an Idle slot, ws will be held unchanged. The proposed protocol keeps the memoryless feature of QT in that it is an applied bit window procedure. It provides a decrease in the number of tag-transmitted bits, but increase the number of slots and reader-transmitted bits. Altogether, this tree-based protocol achieves significant energy savings and a reduction in the identification time.

$$f(L) = k(1 - e^{-\beta L}), \quad 0 < L \leq k \tag{2.1}$$

2.4.4 Manchester coding

Some tree-based protocols work using Manchester coding, which can be used to locate bits that have collided [Finkenzeller 10], [Klair 10], [Rohatgi 07]. The use of Manchester coding for the purpose of tracing the collision to an individual bit

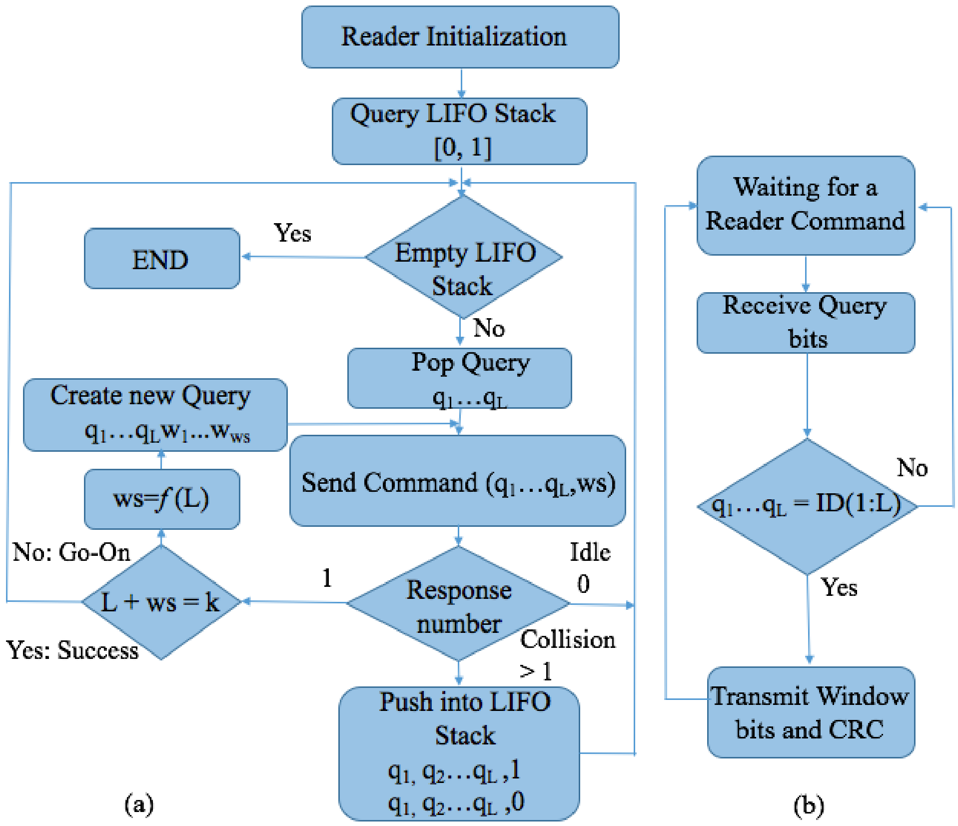

Figure 2.18: Flow chart of QwT protocol: (a) for reader; (b) for tags.

is called bit-tracking in the literature [Jia 12]. In Manchester coding, the value of a bit is defined by the change in the voltage level: a negative or positive transition. A logical 0 is coded by a positive transition; a logical 1 is coded by a negative transition. In the case where a minimum of two tags simultaneously transmit bits with different values (‘0’ and ‘1’), then the positive and negative transitions of the received bits violate the coding rules, and a collision can be tracked.

As shown in Figure 2.19, Tag 1 is 1100 and Tag 2 is 1010. Both tags are synchronously transmitting data. The reader can understand the first bit, but the second and the third bit cause a collision. The reader detects a violation of the Manchester codification on those bits, and so this is interpreted as a collision located at bits 2 and 3.

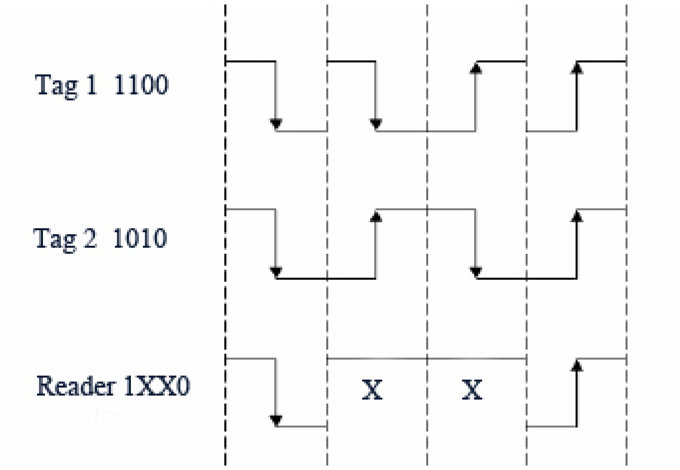

Figure 2.19: Example of Manchester codification showing a collision at the 2nd and the 3rd bits.

2.4.4.1 **Binary Search protocol**

The procedure in the Binary Search protocol (BS) algorithm [Finkenzeller 10], [Wang 06] involves transmitting a serial number from the reader to all the tags in the interrogation area. Only tags which have an equal or lower ID value than the received serial number will respond to the request. Then the reader checks the tags' responses bit by bit using Manchester coding and if a collision is detected, the reader divides the tags into subsets based on the collided bits.

Table 2.2 shows an example of BS being used to read four tags (Tag A to Tag D). The reader starts by interrogating tags with the maximum ID value 111. Tags with a value less than 111 will respond to the query. Their answer results in collision XXX, where all three bits have experienced a collision. In the next slot, the reader transmits a new query by replacing the most significant collided bit (MSB) with a 0. The reader transmits a new query, 011, in the next slot, and all tags compare their ID with the received value. Communication in this slot again results in a collision (01X). In the second slot, the reader replaces the third bit of the command with a 0 and transmits the next query, 010. In the new interrogation round (slot 3) only Tag A has a value equal to or lower than 010, and therefore it is identified successfully. After this slot, the reader restarts the query value with the initial value 111 and transmits it. This procedure is repeated until all tags are identified.

This protocol has two more versions: Enhanced BS protocol (EBSA) and Dynamic BS protocol (DBSA) [Yu 05]. The main difference from EBSA is that it does not restart the reading procedure after a tag is identified, as in the basic version of BS. In order to reduce bit consumption, the reader transmits in the initial slot only '1' instead of transmitting all '1's. In the DBSA version, the reader uses the knowledge from the last slot and reduces the number of transmitting bits. For example, if the reader has received 01X, it will request the tags to transmit only the last bit, since the initial prefix has been already identified.

Slot number	Reader command	Tag A (010)	Tag B (011)	Tag C (100)	Tag D (110)	Result	Type of slot
Slot 1	111	010	011	100	110	XXX	Collision
Slot 2	011	010	011			01X	Collision
Slot 3	010	010				010	Success
Slot 4	111		011	100	110	XXX	Collision
Slot 5	011		011			011	Success
Slot 6	111			100	110	1X0	Collision
Slot 7	101			100		100	Success
Slot 8	111				110	110	Success

Table 2.2: Example of the BS protocol.

2.4.4.2 Collision Tree protocol

The Collision Tree protocol (CT) is an improvement of QT which uses bit-tracking technology in order to find which bits collided and also where they are [Jia 10], [Jia 12]. The reader, using the bit-tracking technology, can trace a collision to an individual bit and get the correct bits successfully. This feature works using Manchester coding, which can locate the conflicting bits on the basis of voltage transitions.

The basic features of this protocol is that it decreases the collision slots and eliminates the idle slots. This contributes to better results in terms of the latency and the number of bits transmitted. The advantage of this protocol compared to the QT protocol is that CT has no idle slots and reduces collision slots.

Figure 2.20 presents how this protocol works in an environment with 4 tags. At the beginning of the identification process, the reader generates two queries '1' and '0' into a LIFO stack. Then, the reader pops query '0' from the stack and transmits it to the tags. In this case, one tag (010110) matches the query and responds with its ID, and the tag is identified. Then, the reader sends a new query from the stack '1' and a collision occurs. With bit-tracking, the reader can find the colliding bits and use this to resolve possible collisions. The reader pushes two new queries '11' and '10', and firstly transmits '10'. The second tag is identified (101010). On the next transmission, a collision occurs again. The reader is able to trace the collision to the fourth bit. Two new queries are made: '1111' and '1110'. These are the last queries in the interrogation round because both tags (111011, 111101) are identified. From this example it can be noted that there are no idle slots, the number of collision slots and the latency are reduced, which is the basic aim of the CT protocol.

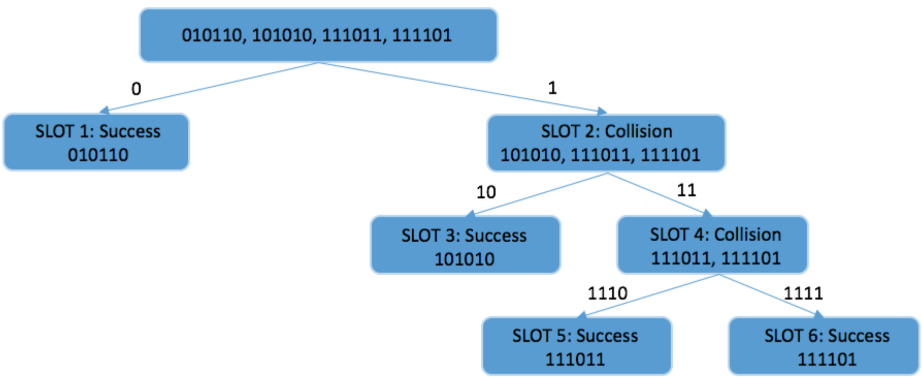

Figure 2.20: Example of the CT protocol.

The conclusion is that CT is a stable and efficient anti-collision protocol for RFID tag identification. The performance of CT is very dependent on the total number of tags in the interrogation area.

2.4.4.3 **Optimal Query Tracking Tree protocol**

The Optimal Query Tracking Tree protocol (OQTT) divides all the tags in the interrogation area into small tag sets in order to reduce the number of collisions at

the beginning of the identification process [Lai 13]. This protocol uses three main approaches: bit estimation, an optimal partition, and a query tracking tree.
With the bit estimation, the reader, using the bit tracking technology, can estimate the number of tags in the interrogation area with a small deviation. This phase detects the status of the bits to perform the estimation. The reader broadcasts the parameter *l*, which denotes the default value of the tag ID length. After receiving a command, all tags must randomly choose a value k between 0 and *l*-1. To simplify the procedure, the tags only respond with a bit string of length b, instead of a bit string of length l. All tags generate a *b*-string of all "0" and set the bit *k* mod *b* to "1". Accordingly, the reader can compute the number of selected bits (NSB), and the number of nonselected bits (NNB) from the tags' responses. The probabilities of bit's being selected or of being nonselected are calculated from the expressions presented in (2). Finally, the optimal estimation of the number of tags is denoted by *ñ* and calculated from (3).

$$\begin{aligned} Pr_{NB}(n,l) &= (1-\frac{1}{l})^n \\ Pr_{SB}(n,l) &= 1 - Pr_{NB}(n,l) \end{aligned} \tag{2.2}$$

$$\widetilde{n}(l, N_{NB}, N_{SB}) = \arg\min_n \left| \begin{pmatrix} l \times Pr_{NB}(n,l) \\ l \times Pr_{SB}(n,l) \end{pmatrix} - \begin{pmatrix} N_{NB} \\ N_{SB} \end{pmatrix} \right| \tag{2.3}$$

The next approach is an optimal partition which determines the number of initial sets. The reader divides the tags into different sets with initial queries.
The query tracking tree is the last procedure in OQTT, and splits the set of collided tags into two subsets using the first collided bit of the tags' responses. This procedure is followed until there are no more queries left in the stack.
The queries in the optimal partition are calculated using the equations $c_1 + c_2 = m$ and $2c_1 + c_2 = 2l$, where c_1 denotes the number of (*l*-1)-bit queries and c_2 the number of *l*-bit queries. The number of bits in each query is computed from the equations $2l - 1 < m \leq 2l$.

Table 2.3 shows an example of OQTT. In the interrogation area there are 5 tags, whose IDs are: '1010', '1100', '0100', '0010' and '0111'. At the beginning of the frame, the reader estimates the number of tags, which here is $\widetilde{n} = 5$. By using this estimatate for the number of tags, the initial number of sets is calculated with the formula *m* $= \lceil 0.595824 \times \widetilde{n} \rceil$, which yields 3. When $m = 3$, there are two 2-bit queries and one *1*-bit query. The reader generates the queries 00, 01 and 1 and pushes them into the stack. In the presented example, $c_2 = 2$ (queries 00 and 01) and $c_1 = 1$ (query 1). As presented in Table 2.3, when the reader pops a query, the tags answer with the value $k - q$, where *k* and *q* are the lengths of the ID and the query, respectively. When a collision occurs, this protocol splits the collided query according to the first collided bit. This is the case with slots 3, 4, 5, and 6.

Slot	**Query**			**Tag response**				
		Tag A (1010)	**Tag B (1100)**	**Tag C (0100)**	**Tag D (0010)**	**Tag E (0111)**	**Status**	**Stack**
	Est.							**00,01,1**
1	00				10		Identified	01,1
2	01			00		11	Collision	010,011,1
3	010			0			Identified	011,1
4	011					1	Identified	1
5	1	010	100				Collision	10,11
6	10	10					Identified	11
7	11		00				Identified	empty

Table 2.3: Example of the OQTT protocol.

However, OQTT may incorrectly estimate the tag number. But the estimation error is negligible, only producing an imperceptible difference in the final result of the number of queries. According to the literature, e.g. [Lai 13], this protocol provides an efficiency close to 0.614 and is one of the most efficient anti-collision protocols for tag identification. Although the slot efficiency obtained by OQTT is very high, the preprocessing increases the energy consumption of the protocol, especially in dense tag environments [Lai 13].

2.4.4.4 Collision window Tree protocol

The Collision window Tree protocol (CwT) is the second proposed window based protocol that applies a dynamic window to CT [Landaluce 14], [Landaluce 16]. This protocol adopts two techniques: bit tracking and bit windowing. The bit tracking uses Manchester coding with the aim of identifying the colliding bit in the tags' responses. This technique avoids using the CRC, which QwT used in order to identify the type of slot. This protocol does not remove idle slots as CT does, but instead decreases the total amount of bits transmitted by all the tags.

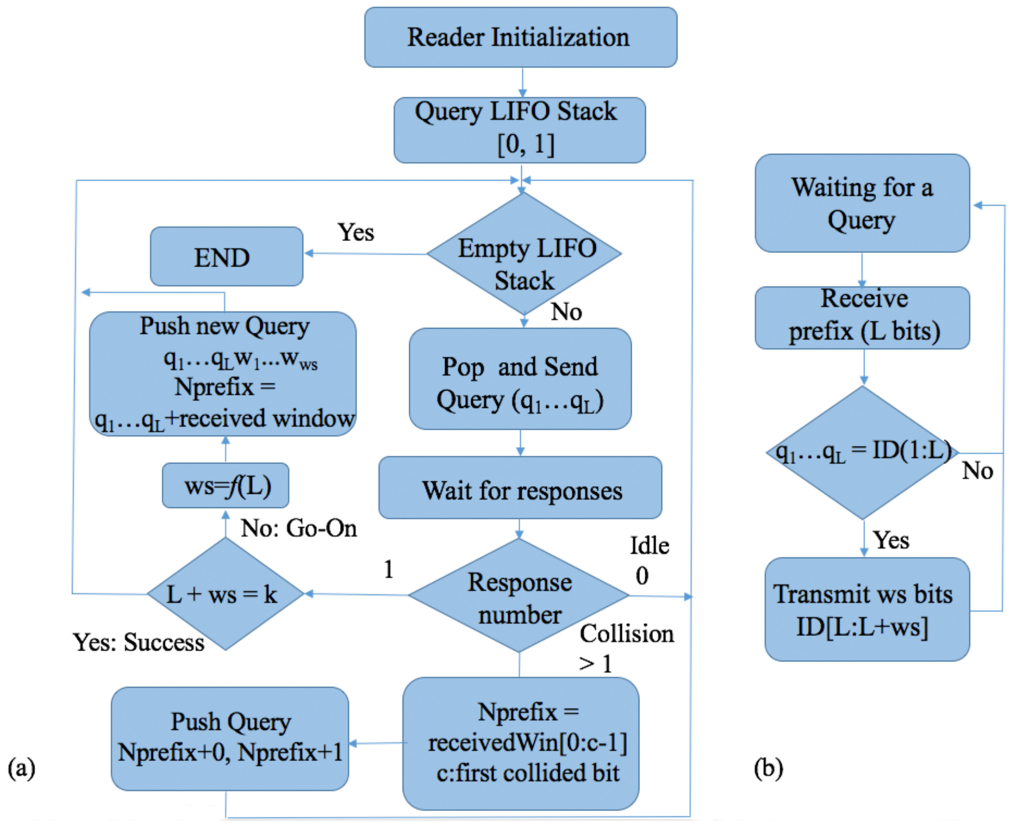

Figure 2.21: Flow chart of CwT protocol: (a) for reader; (b) for tags.

The reader interrogates tags by transmitting a query $[q_1...q_L]$ of length L, attached with the ws of length $[\log_2 ws]+1$ bits. The bit-string ws notifies the tags of the number of bits they must send in their reply. The variable ws is computed in every slot and transmitted together with the query. Only matching tags transmit ws to the last query bit received, [t_L+1...t_{ws}+L] of their ID [b_1,b_2...b_k]. When the reader transmits a query, three possible slot statuses can happen after a tag's response:

- ◇ A collision slot occurs when at least one colliding bit is found. Then the reader creates two additional queries [q_1,q_2...q_L,w_1...w_{col}-1, 0] and [q_1,q_2... q_L,w_1...w_{col}-1, 1], using bit tracking. The supplementary queries are made from the last transmitted query [q_1...q_L] appended with the received bits [q_L,w_1... w_{col}-1] indicating the first colliding bit.

- ◇ A go-on slot is when at least one tag responds and the expression L+ws<k is met. Then, the reader creates a new query based on the former query and the received window from the last slot. The next ws is calculated using the heuristic function in equation (2.1).

- ◇ A success slot occurs when the reader checks the expression L+ws=k and if it matches, the tag is successfully identified.

A flow chart of CwT is presented in Figure 2.21. The reader transmits the first query (0) with $ws = 1$. The matching tags respond with ws bits and the reader looks for a colliding bit. After slot identification, the reader creates a new query and calculates the value of ws, depending on the type of slot.
The CwT provides a significant decrease in the number of tag transmitted bits, but this benefit comes with some loss in the number of slots and the reader transmitted bits. This protocol achieves important energy savings due to a reduction in the time needed by the tag transmission process [Landaluce 16].
Table 2.4 compares some of the tree-based protocols.

Protocol name	QT	OQTT	STT	BS	CT	QwT	CwT
Protocol feature	They use random multi-access way to identify tags. In case of collision, the tags will be asked to send data later with a random time relay.	They identify the total number of tags in the interrogation zone. The reader controls every step of the protocol, using commands or queries to split colliding tags into subsets, and further repeatedly split those subsets until identifies all the tags.	They are mixture of Aloha and Tree-based protocols. They use two methods. The first is using randomized divisions in tree-based algorithms, and another is using tree strategies after a collision in Aloha algorithms.	It involves transmitting a serial number from the reader to all the tags. Only tags which have equal or lower ID value than the received serial number will respond on request.	It is an improvement of QT which uses Bit tracking technology in order to find which bits collided and also where they are.	It applies a dynamic bit window to QT. All the tags compare their ID value with the query received and transmit a certain bit amount managed by the reader.	It applies the dynamic bit window to CT and adopts two techniques: bit tracking and the bit window.
Disadvantages	The reader sends a query and tags, whose ID prefix match that query, respond their full ID.	Very complex protocol, uses three technologies. The preprocessing increases the energy consumption of the protocol, especially in dense tag environments.	On every collision, the full tag response, apart from the initial query bits, is wasted.	The reader restart the reading process after a tag is identified.	It wastes a high number of tag bits on every collision, which increases the energy consumed by the reader during the process.	When the calculated ws is high, the reader command needs a high number of bits to represent it. That leads to a wastage of the reader bits.	Increase the number of reader bits
RTF/TTF	RTF	RTF	RTF	RTF	RTF	RTF	RTF
Efficiency	34.6%	61.4%	58%		35%	80%	61%
System cost	Very low	Very expensive	Expensive	Medium	Low	Medium	Medium
Complexity	Very simple	Very high	High	Medium	Simple	Medium	Medium

Table 2.4: A comparison of tree-based protocols

2.5 Hybrid protocols

Hybrid protocols combine the advantages of tree-based and Aloha-based protocols to avoid their problems and to provide better features in tag identification [Zhang 13], [Bonuccelli 07], [Qian 10], [Porta 11]. Most of them first implement a tree-based procedure and tag estimation procedure in order to predict the number of tags. Consequently, the combined procedures of Aloha-based and tree-based protocols are known for their high complexity and hardware demands. This kind of protocol can significantly increase the performance in comparison with the previous ones.
Recent proposals include the Tree Slotted Aloha (TSA) and Binary Tree Slotted Aloha (BTSA). TSA uses a tree structure, and the tag's responses are organised in slots, as in FSA. In the BTSA protocol, tags randomly choose a slot after the reader query.

2.5.1 Tree Slotted Aloha

Tree Slotted Aloha (TSA) is a probabilistic protocol created with the aim of reducing the number of collisions that occur in FSA [Bonuccelli 07]. When more tags collide in a slot, FSA tries to solve this problem in the next frame. But in the new approach, if more tags collide in a frame, only those tags that are involved in that collision are queried in the next slot.
TSA uses $l_o -$ estimation for the initial frame size. This protocol provides very good efficiency despite the fact that this number can be far from the actual number of tags.
The initial query consists of a request for data by specifying the frame size l_i. Then, all tags in the interrogation area will generate a random number in the range [0, l_i] and transmit its ID in that randomly selected slot.
The protocol is organized in a tree structure. The first node in a tree is the first interrogation round. The reader sets the initial frame with the following data: l_0, N_i represents the number of transmitting tags in slot *i*, where $i \leq l_0$, $N_i \geq 0$, and $\sum_i N_i \geq n$ must hold. If $N_i \geq 2$, there is a collision in slot *i*.
At the end of each interrogation round, if the reader detects a collision, it starts a new frame from each slot where the collision has been detected. This is accomplished by adding new nodes to the tree: every new node is a son-frame of the collided

slot. In each round, the tags store the generated random number from the previous round and increase by 1 their tree level counter so they will know when they should transmit. Every time the reader detects a collision, it creates a new node in the tree and a new round involving only the tags that have collided in that slot. This procedure is shown for the case of the example in Figure 2.22.

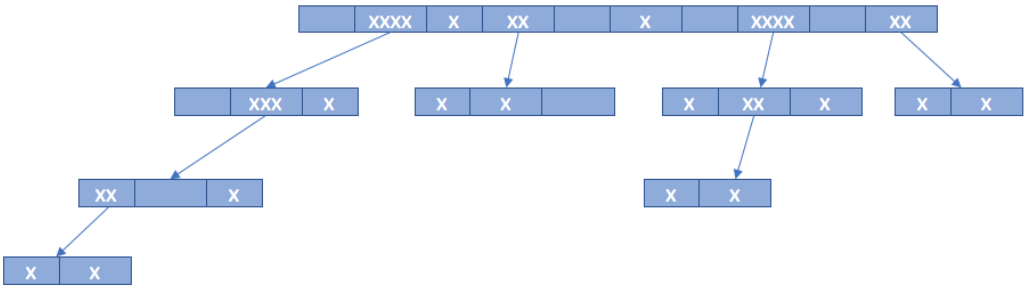

Figure 2.22: An example of TSA.

TSA is a modified version of the FSA protocol, created in order to reduce the number of collisions. This protocol behaves better than FSA. TSA achieves an efficiency between 37% and 41% [Bonuccelli 07].

2.5.2 Binary Tree Slotted Aloha

The reader in Binary Tree Slotted Aloha (BTSA) uses a dynamic frame length adjustment and BTSA algorithm [Wu 13]. Each tag from the interrogation area randomly chooses a slot and transmit its ID. If the reader identifies a tag successfully, it will not be activated in the subsequent slots. When a collision occurs, the collided tags are resolved by binary tree splitting, while the rest of the tags will wait until that process is successfully completed.
The collided tags are continually split into two sets until each set has only one tag. This operation is performed by binary tree (BinTree algorithm) [Hush 98]. The initial frame length is $L = 2^Q$ and the highest efficiency is achieved when the initial frame size is closed to the number of tags. Since BTSA has no estimation of the tag set size, the reader cannot set the initial frame size according to the number of tags. There are some protocols presented in order to achieve higher efficiency in a wide range of the number of tags.

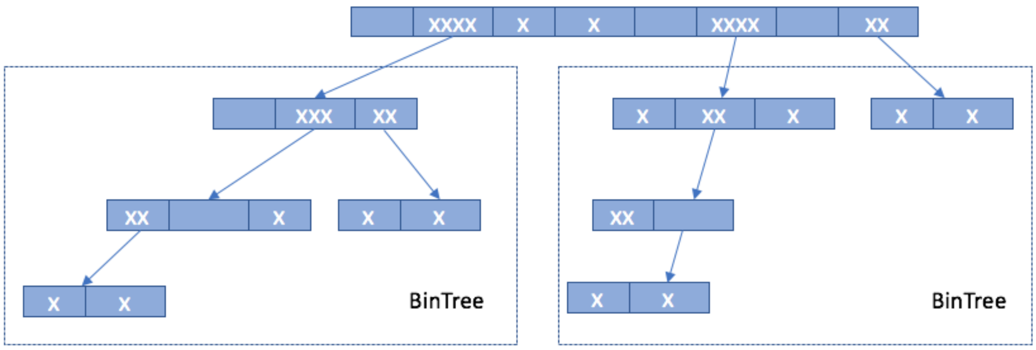

Figure 2.23: An example of BTSA.

2.5.2.1 Dynamic Binary Tree Slotted Aloha

Dynamic Binary Tree Slotted Aloha (dynamic BTSA) involves a dynamic frame adjustment and the basic BTSA algorithm [Wu 13]. The advantage of this protocol is that the reader can adjust its frame size by judging only the first slot type in the identification process.

Figure 2.24 shows the dynamic BTSA algorithm, where the initial frame length is $L = 2^Q$, and the initial $Q_0 = 4.0$. The procedure is very similar to the Q protocol. Firstly, the reader transmits a QueryAdjust command with frame length to all the tags in the interrogation area. Subsequently, each tag randomly chooses a random number between 0 and L-1. Tags whose Counter value is 0 transmit their ID. Then the reader transmits a new request with a new L and will be capable of receiving responses in the first slot of the following frame. If the first slot is idle, the reader decreases the value of Q by 1 (Q=Q-1) and the reader creates a new frame following the updated Q. If the reader successfully identifies a tag in the first slot, Q will not be changed and the reader will move to the BTSA algorithm [Wu 13].

In BTSA, the reader transmits the Query command in a frame. The reader has a slot counter (SC) that is set to 0 at the beginning of the frame, and is increased by 1 at the end of each slot. When the frame length is equal to SC, then the frame finishes. When the reader receives an ID in a slot, it knows the type of slot and will inform other tags by transmitting its feedback. If the reader detects a collision in a slot, it will resolve the collided tags by BinTree splitting [Porta 11]. In order for the reader to know when the binary tree has finished, it uses the variable B. The initial value of B is set to be 2. In the case of a collision, B is increased by 1, and if there is no collision, B is decreased by 1. Only if B=0 does the reader know that the binary tree has finished.

Dynamic BTSA reduces the number of collisions and improves identification efficiency [Wu 13].

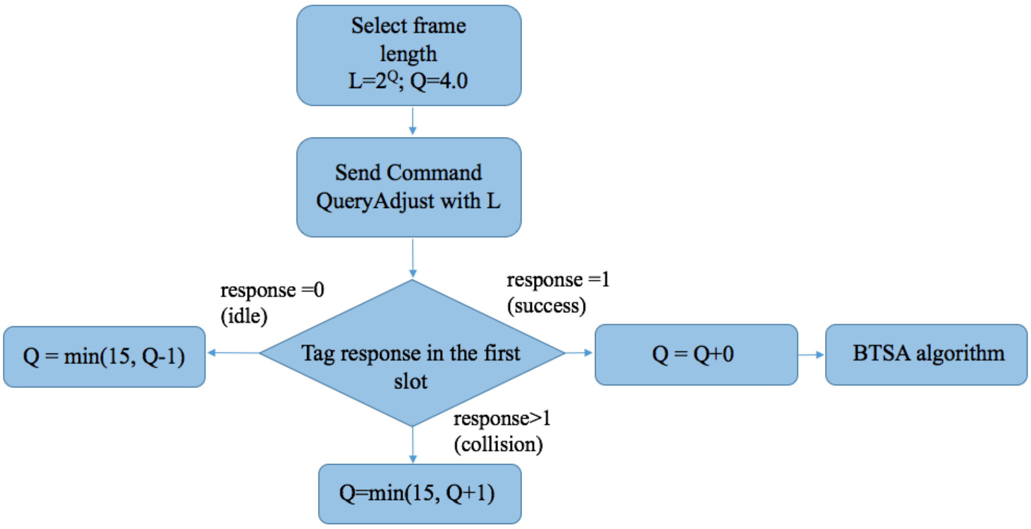

Figure 2.24: Flow chart of Dynamic BTSA.

2.5.2.2 Adaptive Binary Tree Slotted Aloha

Adaptive Binary Tree Slotted Aloha (Adaptive BTSA) presents an improvement on the Q protocol [Wu 13]. This protocol adjusts the frame size based on the tags' responses in a current slot. Adaptive BTSA first uses features from the Q protocol. If there are numerous collisions in a frame, the reader ends the frame earlier and transmits a new command with a new frame length. If there are excessive idle slots, the reader again ends the frame earlier and sends a new command with a smaller frame length.

The reader uses the parameters B and Q_{fp} in order to calculate the frame length. The initial frame length is to $L = 2^Q$ and Q=4. The Q algorithm can adjust the frame length by adjusting Q. The value of Q is the rounded value of Q_{fp}, which is a floating representation of Q. In the following process, the reader dynamically

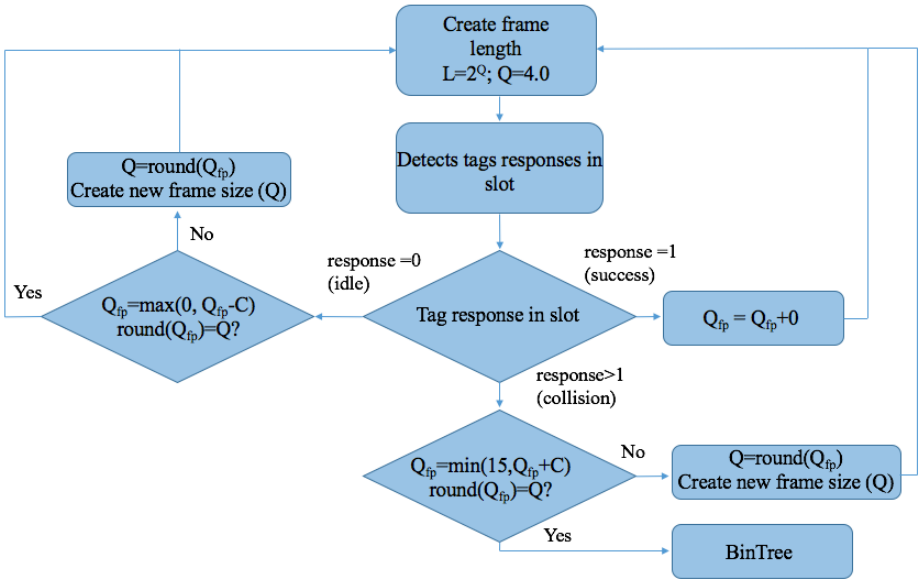

Figure 2.25: Flow chart of Adaptive BTSA

adjusts each slot using the presented values and c [Wang 09a]. In the first slot, if a collision occurs, the reader will calculate Q_{fp} by increasing it by c. In the case of an idle slot, the reader decreases Q_{fp} by c. When the reader identifies a tag, it will not change Q_{fp}. The flow chart of this protocol is shown in Figure 2.25.

The function framesize(Q) shown in Figure 2.25 denotes how a new frame has started and its length is 2^Q. Adaptive BTSA combines the Q algorithm and the BinTree strategy. The main difference between Adaptive BTSA and the Q protocol is that when a collision occurs in the slot, the collided tags will be resolved by BinTree. Table 2.5 shows observations pertaining to Aloha, tree-based and hybrid protocols.

Criterion	**Aloha protocols**	**Tree-based protocols**	Hybrid protocols
Protocol feature	They use random multi-access way to identify tags. In case of collision, the tags will be asked to send data later with a random time relay.	They identify the total number of tags in the interrogation zone. The reader controls every step of the protocol, using commands or queries to split colliding tags into subsets, and further repeatedly split those subsets until identifies all the tags.	They are mixture of Aloha and Tree-based protocols. They use two methods. The first is using randomized divisions in tree-based algorithms, and another is using tree strategies after a collision in Aloha algorithms.
Number of tags to reader commands	Low	High	Medium
Usage	Aloha protocols are commonly used in LF and HF RFID systems.	Tree-based protocols are commonly used in UHF and 18000-7 microwave RFID systems.	
Method	Probabilistic	Deterministic	Mixture (Aloha and Tree-based)
Tag starvation	Yes	No	No

Table 2.5: A comparison of Aloha, Tree-based and Hybrid protocols

2.6 Conclusion

Anti-collision protocols are a critical part of any RFID system. Every collision problem produces a series of deficiencies that must be resolved, since it to a large degree harms the system. The breadth of the literature proves that there has been a great amount of research. These problems result in wasting bandwidth and energy, and increasing latency. The newest protocols have become more sophisticated and achieve better results. Many of them have not been implemented in real systems. Each protocol uses some multi-access methods for identification in order to physically separate the transmitters' signals and accordingly they can be categorized into 4 different types: Space Division Multiple Access (SDMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA), and Time Division Multiple Access (TDMA).

Broadly, there are two main types of anti-collision protocols: deterministic and probabilistic. Nonetheless, these types can be combined to form hybrid protocols. In probabilistic protocols, the tags transmit their own ID in randomly selected slots in a frame in order to reduce the possibility of a collision. The tag answers are distributed into the slots and all of them have a chance of being identified. This type of protocol is highly adaptable to appearances in and disappearances from the interrogation area. Deterministic protocols are ideally designed to identify the whole set of tags in the interrogation area during each cycle. These protocols usually have a simple tag design and can work very well with uniform sets of tags, but are slower than probabilistic protocols.

In the literature, several protocols have been presented, and they can be classified into: Aloha based, tree based, and hybrid protocols. Tree-based protocols provide a deterministic approach to identifying the tags, while Aloha protocols are probabilistic in nature, simple, and promise dynamic adaptability to varying loads. Tree-based protocols must restart their reading process if a new tag appears in a reader's interrogation area while the tags are being read. Hybrid protocols have been created with the aim of avoiding the problems of the Aloha and tree-based protocols, but this comes at the expense of complex reader and tag designs. From these explanations, it cannot be concluded that some protocol type stands out above

the others. Aloha protocols are used more in HF RFID systems, but in UHF there are used both protocols.
The main conclusion is that because RFID systems have become more and more widely present, and because the number of tags has increased, and these system are faced with more important issues, the use of anti-collision protocols will be more omnipresent. This thesis will focus on the deterministic approach which have a bigger capacity for improvement.

3

Standardizing the window methodology

The tag collision problem forces the reader to repeat transmissions, which results in a loss of throughput, an increase in the number of transmitted bits, and increases the energy consumed by the reader as well as its latency. The window methodology is presented to resolve some of these problems by controlling the bits transmitted by the tags. This methodology is applied by some protocols in the literature: QT [Law 00] and CT [Jia 10], so that the number of transmitted bits can be controlled. The matching tags respond with the calculated number of bits - *ws*. QwT and CwT operate using a heuristic function in order to decrease the number of tag bits wasted on collisions and, therefore, the number of bits transmitted per tag [Landaluce 16]. As a result, tags transmit only the bits defined by the window instead of sending their full ID in every response, which results in a smaller total number of bits in the identification process.
The reader, in the previously explained protocols that use the window methodology (QwT and CwT), transmits a command that includes the query and the value of the window size, *ws*. When tags receive this command, these protocols assume the tag is able to differentiate the query bit string from the *ws* bit string. However, this is

not reliable in a real system, since the tag needs additional information to know the lengths of each of these fields in order to separate them. Consequently, they cannot locate the bit string representing the window size in the received command.
This section proposes a standarization of the window size to a power of 2, which allows keeping a fixed size for *ws*, covering all the needed window sizes, and allowing the tags to easily differentiate it from the query. In addition to this, the proposed solution reduces the number of bits transmitted by the reader compared to other protocols that use a window methodology. This results in a lower total number of bits in the identification process.

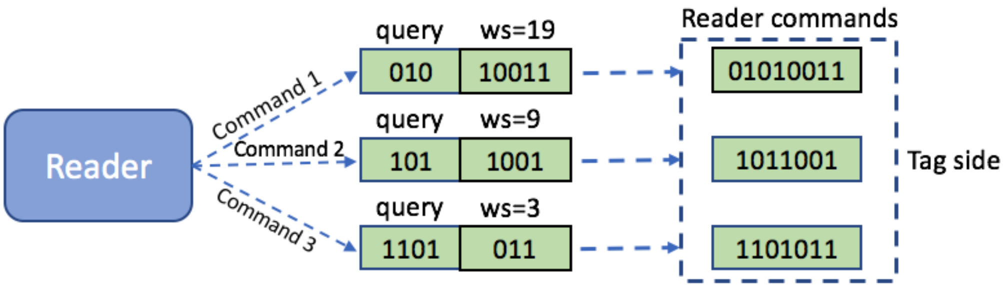

Figure 3.1: Example of the reader command on both sides.

3.1 Non-Standardized reader transmission

Tags in the window methodology proposed in Section 2.4.3 can not locate the bit string representing the window size in the received command. The query bits are attached to the bits for *ws* in the same reader command and transmitted to all tags in the interrogation area. An example of three reader commands using the QwT protocol is shown in Figure 3.1. These commands can be received in the same identification proces by the same tag. The tag must compare the received bits from a query with the corresponding part of its own ID. If the tags' ID matches the query, it must respond with *ws* bits. The tags do not have the ability to know which part of the reader command is intended for the comparison and which one gives the value of *ws* (which gives the number of bits that the tag must transmit in case it matches the query). The number of bits in the string that gives the value of *ws* can vary,

depending on the value of ws; the length of the query varies as well. Thus, tags are not able to know how many bits in the reader command are meant to describe *ws* and how many are for the query.
The example in Figure 3.1 shows a typical situation on the tag side. The tag receives the first reader command (01010011) and it cannot determine which part is meant for the query and which for *ws*. In this example, the first command consists of a query of bits (010) and *ws*=19 converted into the 5-bit binary number (10011). QwT and CwT assume the tags are able to identify what the transmitted value of *ws* is, since the length of this value is variable in the command. Tags using these protocols are not able to separate these two fields from the reader command.
These issues affect all the protocols based on the window methodology. They do not provide additional information in the command about the length of the query and of *ws*, and therefore tags are not able to distinguish between these two fields. In order to solve this problem, the Standardized Query window Tree protocol (SQwT) will be presented in the next section.

3.2 Standardized Query window Tree protocol

The Standardized Query window Tree protocol is proposed with the aim of standardizing the window methodology: it represents *ws* by a fixed string, allowing the tags to differentiate the query from the window size in the reader command [Cmiljanic 16]. SQwT is a memoryless protocol, since the current response of each tag only depends on the last query and *ws* received from the reader, and the tags do not need to store any variable from one interrogation round to the next one.
The reader calculates the number of bits with which the tags must respond to a matching query. Instead of sending the number of bits needed to represent *ws* bits, the reader using SQwT transmits a new value for *s*, which is the first higher power of 2 presented in (3.1). It will be limited to three bits and included in the reader command.
Tags receive the value of *s* and calculate the expression 2^s. That is the new value of *ws* that the tags are going to use. Tags will always transmit a power of 2 bits (1, 2, 4, 8, 16, 32 , 64 and 128).

E.g., if *ws*=128, the reader first uses (3.1), with the result 7. This is codified into the binary representation 111. The reader transmits this bit string attached to the predefined query. The ID length of 128 bits, currently, is the most common in RFID [Technology 15], and for this reason, SQwT uses three bits in the reader query for *s*, since three bits are enough to provide that maximum value. The process is depicted in Figure 3.2. For longer IDs, a larger value of *s* should be used, and a longer bit string would be needed. Such longer ID values have not been considered in this thesis.

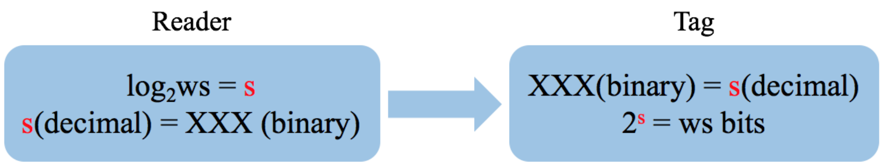

Figure 3.2: Converting ws between reader and tag.

$$s = log_2 ws \tag{3.1}$$

3.3 SQwT procedure

As shown in Figure 3.3, the reader interrogates the tags with a query $[q_1, q_2 \dots q_L]$ and the fixed string *s*. All the tags will receive this broadcast message, but only those tags whose ID matches the query will calculate *s* and respond to the reader's request. SQwT dedicates a constant location containing three bits inside the reader command. This allows the tags to locate the string containing the bits which indicate the size of the window in the received command. The matching tags calculate *ws* by using the expression (3.2) to obtain the size of the window.

$$ws = 2^s \tag{3.2}$$

The flow charts, for both the reader and the tags, of SQwT are given in Figure 3.4 (a) and (b). First, the reader is initialized by pushing two queries into a LIFO stack

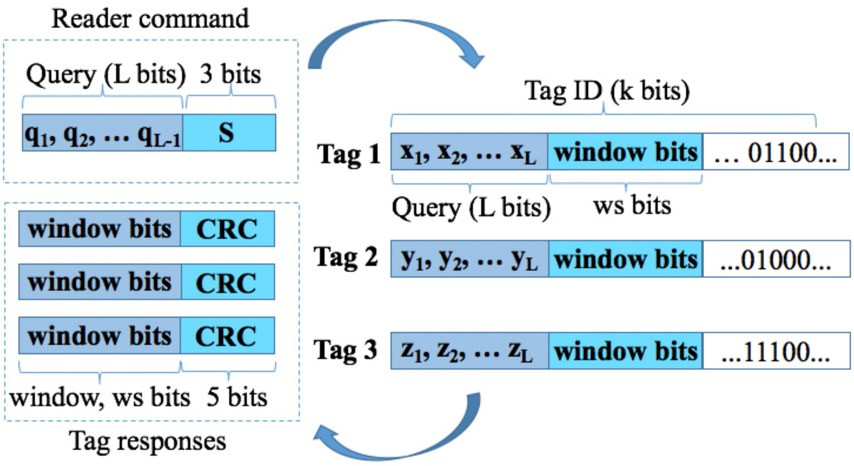

Figure 3.3: Format of the reader command and tag responses of the SQwT protocol.

and the reader pops the last pushed query. For the calculation of the initial *ws*, it uses *ws*=1. Subsequently, the reader obtains *s* with (3.1) and codifies it into three bits. Once *s* is calculated, a new reader command including a query and *s* will be transmitted to the tags. The reader waits for the tags' answers. The tags receive the reader's command (see Figure 3.4(b)) and compare the query with their IDs. Matching tags will calculate *ws* using (3.2) and will respond with the remaining *ws* bits from bit *L*; $x_{(L+1)} \ldots x_{(L+ws)}$, and a CRC. The reader will act, depending on the type of response, as follows:

- ◇ Upon a collision, the reader creates two new queries, $q_1, q_2 \ldots q_L, 1$ and $q_1, q_2 \ldots q_L, 0$, and leaves *ws* unchanged.
- ◇ When an empty window is received, the transmitted query is rejected, another one is popped from the stack, and *ws* remains unchanged.
- ◇ If the received CRC is positively checked, the reader checks the expression *L*+*ws*=*k*, and if it matches, the tag is successfully identified. Then, the ID is stored in a database and the query is discarded, to continue with the process. A new query is popped from the stack.
- ◇ Upon detecting a go-on, the expresion *L*+*ws*<*k* is met. The reader produces a new query, attaching the successfully received window $w_1, w_2 \ldots w_{ws}$ to the

last transmitted query until it receives the full ID value. The value of *ws* is calculated using (2.1).

This procedure is iterated until it gets the empty stack.

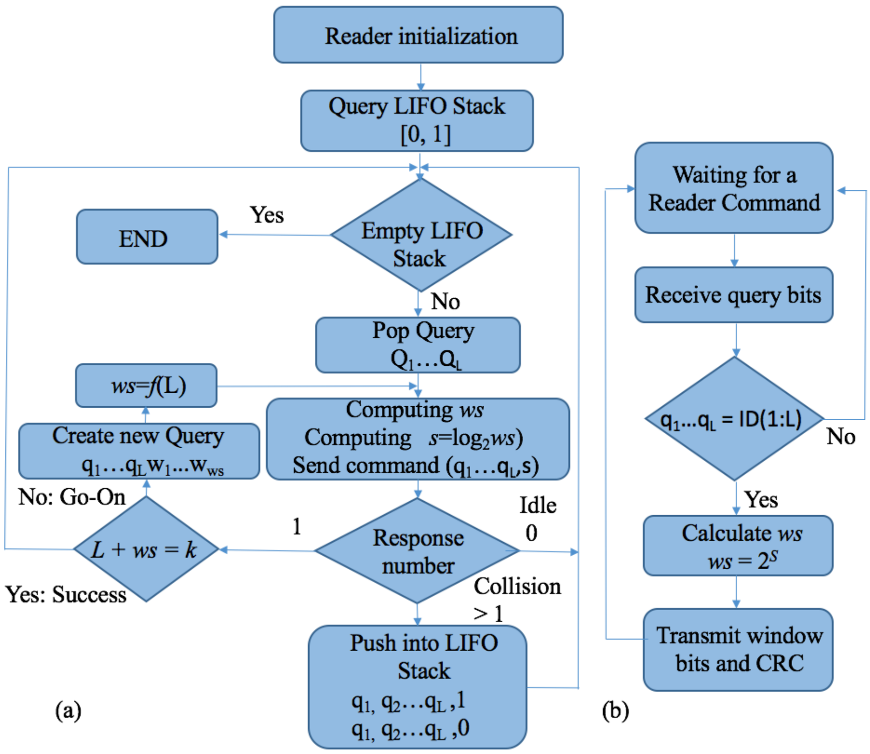

Figure 3.4: Flow chart of the proposed SQwT protocol: (a) for reader, (b) for tags.

3.3.1 Tuning the SQwT

SQwT needs a heuristic function in order to reduce the number of go-on slots while keeping low the number of bits transmitted by the tags. This function is affected by the standarization of the *ws* field and needs to be retuned. A heuristic function which computes *ws* is given in (3.3). An adequate value of β is 0.5 for the lowest number of transmitted bits in the identification process. This calculation is simulated for the range of β values (β=0.1–0.7), various groups of tags ($n = 200, 400, 600, 800, 1000$) and a tag ID length $k = 128$ bits. Figure 3.6 shows the results of the metric; the best value of β for the total transmitted bits is surrounded with a blue frame.

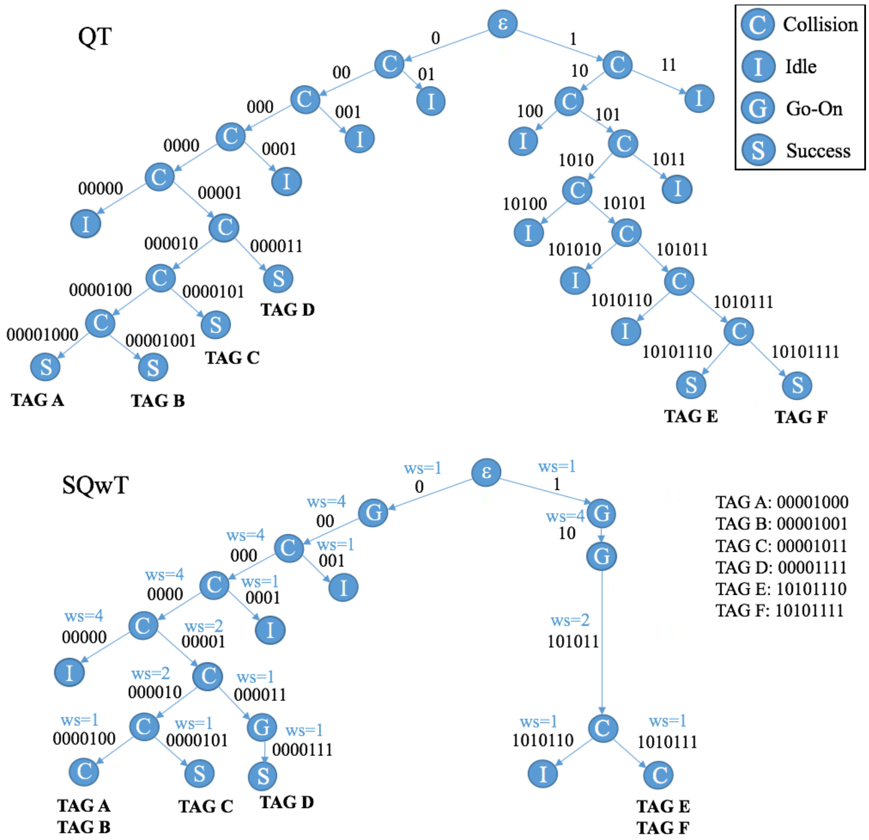

Figure 3.5: Example of the proposed SQwT and the QT protocol.

$$f(L) = k(1 - e^{-\beta L}), \quad 0 < L \leq k \tag{3.3}$$

3.4 Comparing SQwT with QT

An example involving the identification of 6 tags is depicted in Figure 3.5 using QT and the proposed SQwT protocol. The ID length *k* used is 8 bits and *ws* is computed using (3.3) with $\beta = 0.5$. The reader first sends the query (0). Tags A, B, C and D respond with the whole ID value and a collision occurs using the QT protocol. The difference from the SQwT protocol is in the tag response. After matching query (0) the same tags as in the QT protocol will transmit, but just with their first *ws* bits.

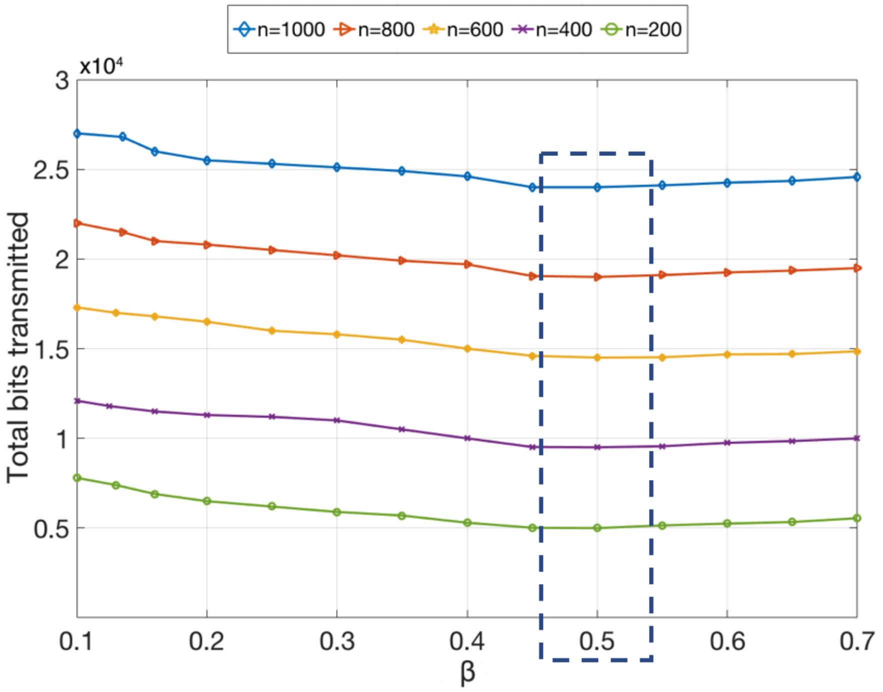

Figure 3.6: Selected β for n tags in the interrogation zone using the total bits transmitted in the identification process.

The reader receives a partially successful response, and it adds the bit '0' to the previous query, forming a new query (00), with which it continues the identification process. In this case, (1) calculates *ws* to be 4 bits. Again, 4 tags match the query (00) and transmit the following 4 bits. In this round, the reader can not understand the responses and a collision occurs. The following part of the tree is the same in both protocols, because three collided slots have occurred.

Protocol	Reader bits	Tag bits	Total bits	Collision slots	Idle slots	Go-on slots
QT	161	360	521	14	12	/
SQwT	98	83	181	5	1	5

Table 3.1: The statistics for the identification processes of the QT and SQwT protocols.

Every time a collision occurs, the reader pushes into the stack two new queries: $[q_1, q_2 \dots q_L, 0]$ and $[q_1, q_2 \dots q_L, 1]$. The left subtree under the 0 branch of both protocols is the same up to the point of transmitting the query 0000100. The QT protocol needs to carry out one more interrogation round because the tags respond with the whole of their ID values, as opposed to SQwT, which transmits just the last bits, and tags A and B can be identified. After this success slot, the reader pops the next query and can identify Tag C. The reader pops out a new query from the stack (000011), resulting in the identification of tag D. The QT protocol needs two more slots in order to identify the same number of tags. The right part of the tree is significantly reduced in the SQwT protocol. After transmitting the first query '1' from the right side of the tree, two go-on slots will occur. Afterwards, SQwT pops the next query (101011) and detects a collision. The last query created during the last collision results in the identification of the remaining two tags: Tag E and Tag F. Eventually, when the comparison of QT and SQwT is analysed, the results show a lower number of transmitted bits per tag, and a decrease in the number of collisions. The comparison of the statistics for both protocols is shown in Table 3.1.

3.5 Conclusion

The SQwT protocol has been presented in this section. Its aim is to carry out a standardization in the window methodology and to reduce the number of reader-transmitted bits. The existing protocols QwT and CwT are not able to differentiate the query bits from the *ws* field in the reader command.
A standardization of the window methodology has been proposed in this section, in which the value of *ws* is represented as a power of 2 with a string with a fixed length of three bits, included at the end of the reader command. This provides the tag with the ability to locate the bit string representing the window size in the received command, distinguishing it from the query. The mentioned problem from the QwT and CwT protocol is solved, and the presented SQwT protocol achieves a significant reduction in reader bits and accordingly reduces the total number of transmitted bits in the interrogation process.

Invention is the most important product of man's creative brain. The ultimate purpose is the complete mastery of mind over the material world, the harnessing of human nature to human needs.

Nikola Tesla

4

Influence of the distribution of the tag IDs on RFID anti-collision protocols

Every tag has a unique identification code (ID) that allows the object that it is attached to to be definitively identified by that number. All the existing tags in a reader's antenna range form a group of tag IDs, collections of 0's and 1's, that together consitute a distribution of bit strings. This distribution can be homogeneous, i.e. the likelihood of finding a 0 or 1 in the same position of all the IDs is equal, which is called the uniform distribution (UD). On the other hand, a heterogeneous or a non-UD distribution happens when there are shared parts of the ID among the different tags.

The standardization of the tag IDs with the EPC standard provides an improvement of RFID, allowing it to access global networks [EPCglobal 8]. The EPC standard defines IDs for all the tags in the world: the EPC codes. These codes contain information, such as the ID of the company, the region of the world, type of product, etc. The standardization of these bit strings using EPC causes situations where several objects that belong to the same company, or are of a similar type, share big portions of their EPC or ID code.

Anti-collision protocols must consider this factor in the identification process, specially those protocols that are based on the identification on the tag ID, such as tree-based protocols. Such protocols are severely affected by heterogeneity or non-uniformity, which causes an increase in the number of transmitted bits and energy consumption. Aloha protocols, on the other hand, are not affected by this issue since they randomly select the slot in which to transmit, without considering the ID. Moreover, only a few protocols in the literature have analysed or considered the impact of the ID distribution on the identification process: they only consider the UD [Lai 13], [Law 00], [Landaluce 16].

A study of the influence of the tag ID distribution on the protocols' behaviour is presented in this section. A new protocol will be presented that estimates the tag ID distribution, taking into consideration the partial responses of the tags', within the window. The aim is to create a flexible anti-collision protocol in order to identify a set of tags regardless of the type of distribution of tag IDs that is present.

4.1 Tag ID distribution problem with tree-based protocols

As mentioned above, the standardization of the tag IDs with the EPC standard provides an improvement of RFID, allowing it to access global networks. RFID technology is increasingly demanding larger tag IDs. Once this ID has been written in the electronic chip, it can be read and, in some tag solutions, can be changed.

The tag IDs are not always uniformly distributed. In the UD, the tags are randomly generated with their ID's between $[1, 2^k - 1]$, where k is the ID length. Figure 4.1 shows an example of a UD where the likelihood of obtaining '0' or '1' when generating the tag ID of distribution is the same. A UD is organized in such a manner that the tags' IDs are uniformly spread through the binary tree: using the left and righ sides of the tree equally. Figure 4.2, on the other hand, shows an example of a non-UD, where tag IDs are not equally distributed in the binary tree, in fact, they share some portions of their IDs.

The main problem with tree-based protocols is that their behaviour directly depends on the type of the distribution and on the length of the ID. These protocols may, with certain types of ID distributions, suffer a loss of performance in terms of the consumed bits, latency, and energy consumed. This happens because the tags' answers directly depend on a reader request and in a heterogeneous environment, tree-based protocols are forced to repeat unnecessary queries, hence increasing the number of collisions.

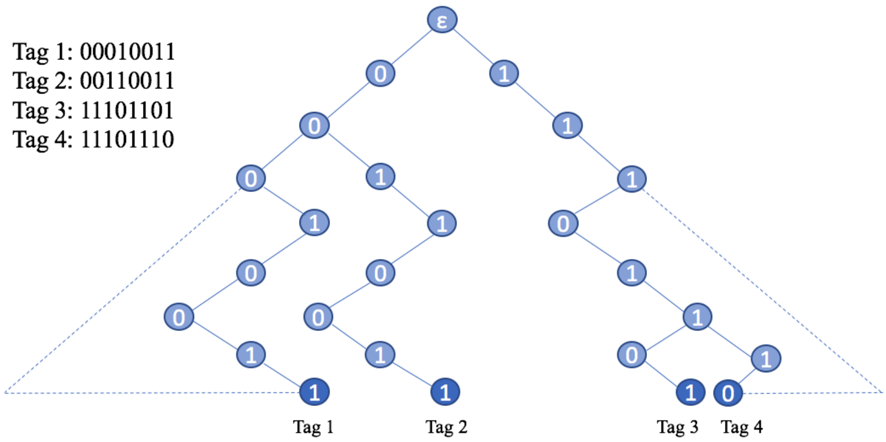

Figure 4.1: Example of a UD

In Figures 4.1 and 4.2, examples of two binary structures are shown: UD and non-UD, respectively. In order to understand the implications of the tag ID distribution for a tree-based protocol, and how these distributions affect the performance, two of this type of protocol, QT and QwT, are used with both the examples of tag ID distributions in these figures. Table 4.1. presents a comparison of the identification processes for both protocols, under these two distributions. Regarding Figure 4.1, QT needs 18 slots in order to identify this set of tags. The reader transmits 66 bits and all tags in the interrogation area transmit 86 bits. On the other hand, QwT needs 16 slots, the reader consumes 60 bits, and all tags from the interrogation area send 76 bits. The window methodology enables QwT to aggressively advance through the common parts of the tag IDs, causing a decrease of 2 total slots, 6 reader bits, and 10 tag bits in comparison with QT. The total consumption of bits for this comparison is 136 bits for QwT and 152 for QT.

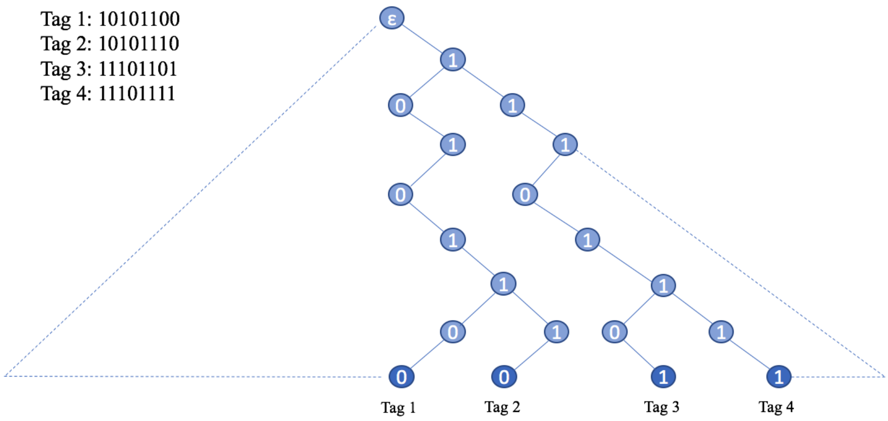

Figure 4.2: Example of non-UD

Comparing, now, the identification of the non-UD distribution shown in Figure 4.2 with QT and QwT, the QT protocol spends 24 slots, and QwT 23 slots in the whole identification process. The reader in the QT protocol needs 106 slots, while QwT needs 3 fewer slots to transmit all the commands. The most significant difference is in the tag-tranmitted bits: QT needs 100 bits and QwT only needs 78 bits. The total consumption of bits is 206 for QT and 181 for QwT.

Distribution	Protocol	Slots	Reader bits	Tag bits	Total bits
UD: Figure 4.1	**QT**	18	66	86	152
	QwT	16	60	76	136
Non-UD: Figure 4.2	**QT**	24	106	100	206
	QwT	23	103	78	181

Table 4.1: The statistics for the identification process of QT and QwT under UD and non-UD.

The statistics of the comparison for both protocols are shown in Table 4.1. With the uniform distribution, both protocols achieve better results and save more slots and consumed bits. The number of total transmitted bits with each distribution shows that the organization of the IDs in the tag environment is very important for the tested protocols. Summing up the obtained results, it can be concluded that QT and QwT are very dependent on the distribution of the IDs of the tags.

4.2 Representation of the ID distribution

The whole set of tag IDs constitutes a distribution of binary strings with the shape of a binary tree, starting from the most significant bit (MSB) to the least significant bit (LSB). The behaviour of some protocols depends significantly on the distribution that they encounter. In RFID systems, it is unrealistic to assume that the tag ID distribution is always uniform. Consider the goods in a supermarket, which belong to several specific categories. They will share several parts of their ID, since they may belong to the same supermarket, the same company, the same area, or be the same type of product, etc. Different tag ID distributions can greatly influence the performance of a protocol, as shown in the previous section.

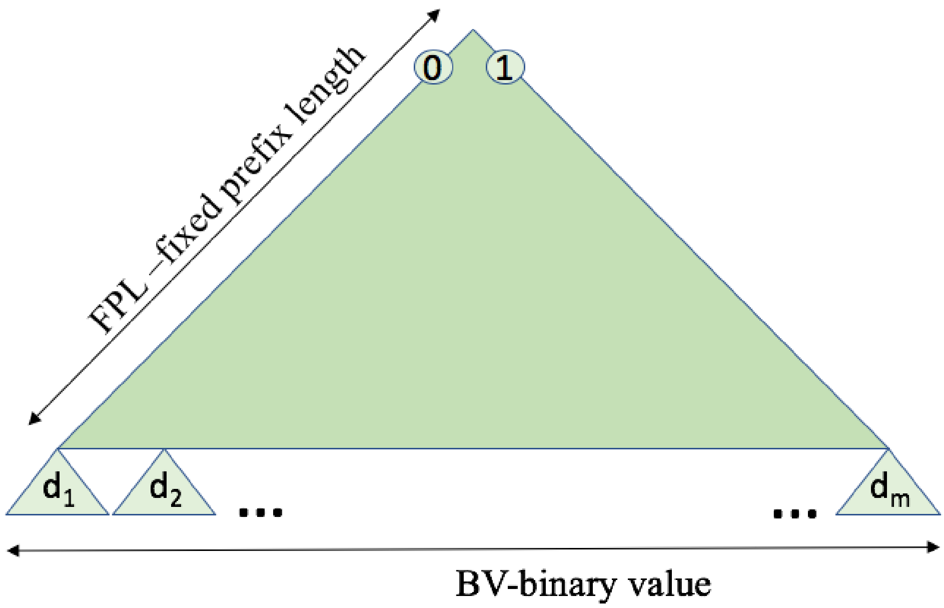

Figure 4.3: Types of tag ID distribution.

A new way to describe a distribution of the tag IDs is proposed here. Three main parameters have been defined, that completely describe the type of distribution: the fixed prefix length (FPL), the binary value (BV), and the number of uniform subdistributions (d_m). These parameters are shown in Figure 4.3, and detailed below:

- ◇ The fixed prefix length (FPL) defines a specific organization of the distribution where all the tags in the interrogation area share the initial part of the ID, of that length.

- ◇ Binary value (BV) denotes the horizontal position of the tag ID distribution at an FPL level. This horizontal position is given as a percentage of $2^{FPL} - 1$.
- ◇ The number of uniform subdistributions (d_m) assumes that the organization of the tag ID distribution is in several subdistributions *m* following UDs. The greater the number of subdistributions, the more similar the main distribution will be to a UD.

An example of varying FPL and BV, with a fixed d_m, is shown in Figure 4.4. If FPL = 3, BV = 28$\%$, and d_1, the first tag in this binary structure will have the initial binary value 010. BV is calculated as the percentage value from the largest value in the binary tree for the used FPL (111). In another case, when FPL = 4 and BV = 80$\%$ and 100$\%$, and d_2, the tags will have a fixed 4 bits for their initial parts and the tag ID sets will start with the fixed 4-bit value 1111. This organization has two different distributions, with the same FPL (4 bits), and two different values of BV. This distribution tends more towards the UD, unlike the previous one from the example. Therefore, different ID distributions can be considered in this thesis. Every type of distribution will have the same number of tags but be organized with many variations.
The influence of different tag ID distribution can degrade the behaviour of a deterministic protocols, resulting in a higher energy consumption, an increased number of collisions, and a prolonged identification time. In order to solve these problems, this section presents a protocol that estimates the ID distribution and further controls the protocol's behaviour, providing the flexibility to work with different ID distributions, yet with similar performances in terms of latency, energy consumed, and total bits transmitted.

4.3 Flexible Query window Tree protocol

This subsection proposes a novel Flexible Query window Tree (FQwT) and it analyses its flexibility in detail. This protocol employs the window method and an

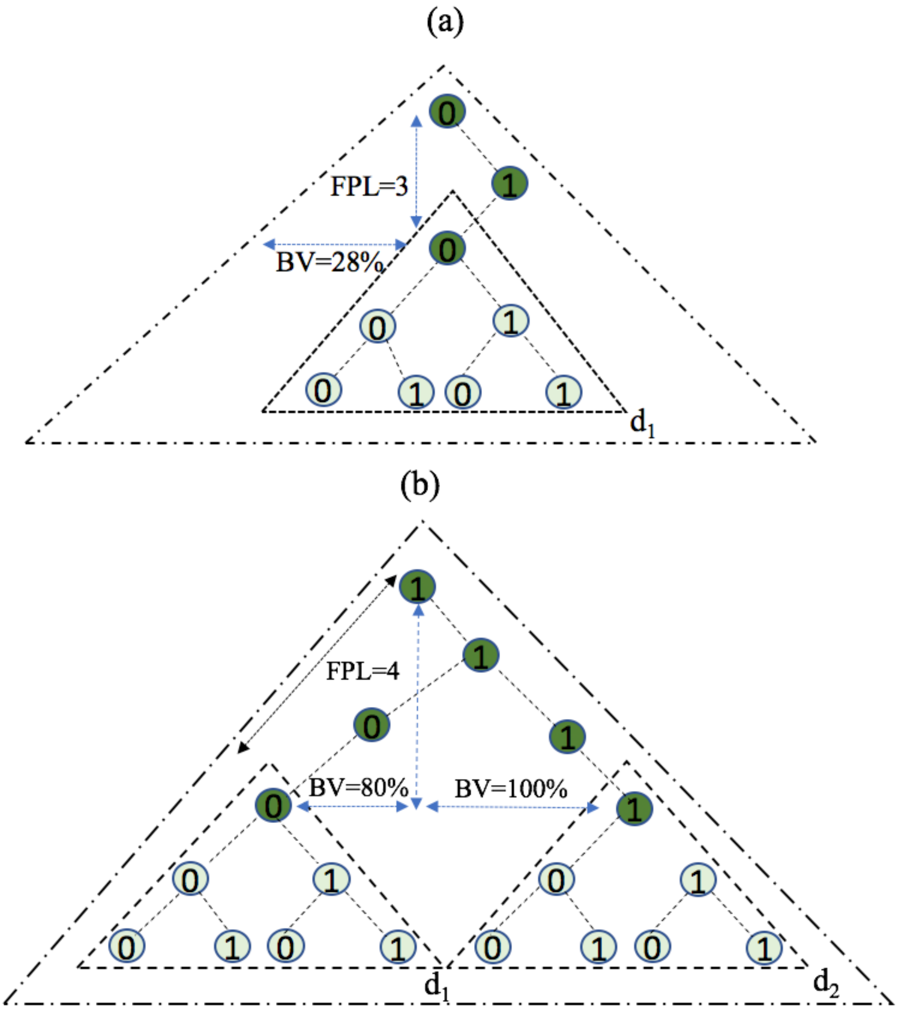

Figure 4.4: Examples of types of tag ID distribution with adjustable values: (**a**) FPL = 3; BV = 28%; d_1; (**b**) FPL = 4; BV = 80%; d_1 and 100%; d_2.

ID distribution estimator. It is also a memoryless protocol, since the tags do not need to save information in order to be identified, and the tags' responses directly depend on the current query and *ws*.

Excessive collisions increase energy consumption, wasting a large number of tag-transmitted bits. The FQwT protocol has been proposed to manage the length of the tags' responses, in order to reduce the energy wastage in scenarios where tags have different ID distributions. FQwT has the ability to estimate the ID distribution and reduce the number of the bits in the tags' responses, which provides significantly better behaviour regardless of the type of ID distribution followed by the tags. Consequently, this approach results in decreased: energy consumption, number of transmitted bits, and identification time.

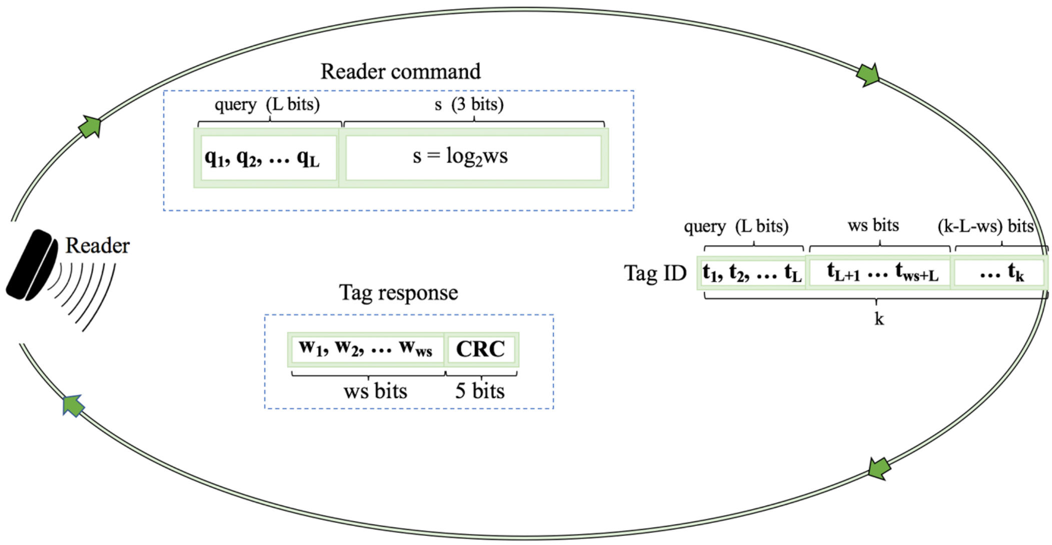

Figure 4.5: Format of the reader command and tag responses of the FQwT protocol.

A decrease in the number of transmitted bits in an heterogeneous tag environment is achieved by estimating the tag ID distribution and subsequently adjusting *ws* in the reader command. The window method provides a protocol to aggressively advance through the common parts of the IDs.

The reader calculates *ws* as an integer in the range of $1 < ws < k$ but then it is transmitted, together with a query, to the tags in the changed form *s* from (3.1). The string *s* is a standardized fixed 3-bit value that the reader must send in every interrogation round. As is shown in Figure 4.5, the proposed FQwT protocol sends a query and *s* to all the tags in the interrogation zone. The tags calculate *ws* using (3.2) from the received reader command, differentiating it from the query by using the last three bits of the reader command. The tags receive and compare the query with their ID; the matching tags respond only with the bits specified by *ws*. the tag ID distribution and calculates the next query and *ws*. The reader primarily estimates the tag ID distribution. Depending on the environment, the reader can determine the type of organization of the ID distribution. The functioning of the protocol is divided into two phases: the estimation of the distribution, and the identification process.

4.3.1 Phase 1: Estimation of the ID distribution of the tags

In order to obtain good results, the reader will, in the initial phase of the tag identification, estimate the type of distribution, store the obtained data, and use it in the subsequent operations. The first phase ends when the first tag is identified.
The reader flow chart of the initial procedures, called the ID distribution estimation, in FQwT is given in Figure 4.6. The reader initializes the procedure by pushing two new queries into a LIFO stack and then the reader starts the identification by popping the first query from the stack and transmitting it to the tags in the interrogation area. Initially, *ws* = 1. In the next interrogation rounds in the first phase, *ws* will be calculated using the same rules predefined just for this phase. Apart from the query of length L in the reader command, the length of *ws* is attached in the form of (3.1). Matching tags will calculate the value of *ws* (3.2) from the received string *s*. It will be the final value of *ws*.
The value of *ws* will be calculated during each phase: the estimation of the distribution and the identication of the tag sets. But the procedure for calculating it will be different, depending on the phase. Also, all responding tags attach CRC bits so that the reader can determine the type of the tag response by checking the consistency of the CRC. When the reader receives one or more responses from the tags, it will check their consistency. Depending on the type of response, the reader will act as follows:

- ◇ Upon a collision, the reader will check the value of *ws*. If *ws* is bigger than 1, it will restart at the beginning value (*ws* = 1). When *ws* = 1, the reader will calculate c_g, the difference between the ID length *k* and the current query length *L*, and locate the first group in a new type of ID subdistribution. The reader stores the first value of c_g into LIFO and continues with the interrogation. All c_g values are stored into the LIFO stack together with the the corresponding query, and used when a specific group is identified. Later, when the reader pops a query from the LIFO, it will use the same value of c_g for the whole identified group to calculate *ws*.

- ◇ In case of an empty response, the reader will continue the process with a new query from the stack and ws will be kept unchanged.

- ◇ A go-on slot is received if the reader understands the response but the ID is not complete (q + *ws* ¡ k). The reader will increase *ws* by 1 and store the received window in the stack and use it in the next query.

- ◇ Finally, when the reader receives the last window bits and completes the whole ID, a successful slot occurs. The reader saves this tag and completes the first phase. The following procedure will follow a new phase with addition calculation.

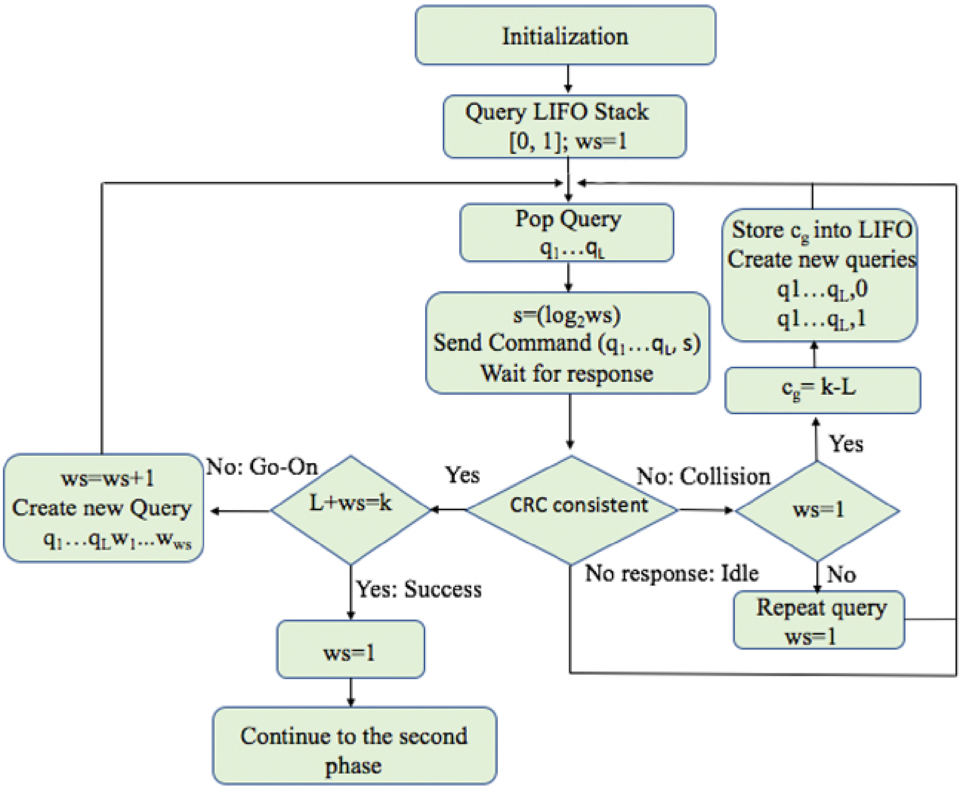

Figure 4.6: Flow chart of the first phase in the proposed FQwT protocol.

An example of ID distribution estimation in an environment of 10 tags using FQwT is shown in Figures 4.7 and 4.8. The initial *ws* is 1 and the ID length k is assumed to be 16 bits. The reader starts with a query, 0. All the tags from the example respond and a go-on slot occurs. A new query is created by appending the window to the last query. The reader recalculates *ws* (*ws* = *2*) and s, and attaches it to the query. On the new reader query (00), 7 tags will answer (Tag1–Tag7) and a collision occurs. In the subsequent interrogation round, the reader will repeat the same query but with decreased *ws*. In this slot, when a collision is detected and *ws* = 1, the reader can locate the first group (Group 1) in an unknown type of tag ID distribution and store c_g value in LIFO. From this example, the first branch is denoted by Group 1 and there are three tags (Tag8–Tag10). Subsequently, the reader checks the value of *ws*, reduces it, and creates two new queries by appending 0 and 1 (000 and 001). The same procedure is followed and Groups 2 and 3 are located and information about them will be stored into the stack.

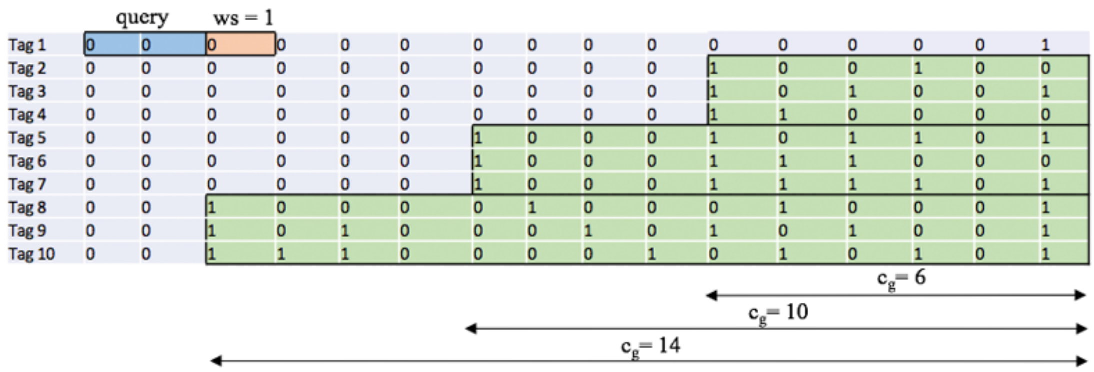

Figure 4.7: An example of the procedure of the estimation of the tag ID distribution.

After the last query (00000000000000), only one tag responds and transmits the last part of the ID. With this step, the estimation procedure is completed. The reader has located three groups in the interrogation area and identified the first tag ID. With this, the first phase of the process is finished. For comparison, Figure 4.8 presents the unused queries from the stack, from FQwT, QwT and QT. The novel protocol will continue with 22 bits from the stack, QwT needs 68 bits, while the QT reader needs to spend 66 bits in order to identify the rest of the tags. Finally, when the first

Slot	Query	ws	Group 1	Group 2	Group 3	First tag	Slot type	Distribution status
1	0	1	0	0	0	0	Go-On	
2	00	2	10, 11	00	00	00	Collision	
3	00	1	1	0	0	0	Collision	Located Group 1
3	000	1	-	0	0	0	Go-On	
4	0000	2	-	00	00	00	Go-On	
5	000000	2	-	10	00	00	Collision	
6	000000	1		1	0	0	Collision	Located Group 2
7	0000000	1	-	-	0	0	Go-On	
8	00000000	2	-	-	00	00	Go-On	
9	0000000000	2	-	-	10,10,11	00	Collision	
10	0000000000	1	-	-	1	0	Collision	Located Group 3
11	00000000000	1	-	-		0	Go-On	
12	000000000000	2	-	-		00	Go-On	
13	00000000000000	2 (k-q)	-	-		00	Success	Identified first tag

FQWT LIFO Stack after identification first tag
00000000001; 0000001; 001; 1

QwT LIFO Stack after identification first tag
0000000000101; 0000000001; 000000001; 00000001; 0000001; 000001; 00001; 0001; 001; 01; 1

QT LIFO Stack after identification first tag
00000000001; 0000000001; 000000001; 00000001; 0000001; 000001; 00001; 0001; 001; 01; 1

Figure 4.8: Comparison of FQwT, QwT, and QT protocols during the identification of the first tag.

tag is identified, the first phase is completed and then the reader can continue with the second phase.

After the last query, the first phase is completed and the reader is able to calculate the values of FPL and d_m. In the mentioned example, the presented RFID environment has three identified groups of tags with different values of c_g, and the reader will update the number of subdistributions to d_3. Also, in this example, the reader will allocate values for FPL to 2, 6, and 10. Using this data, the majority of the binary tree can be constructed. The conclusion is that the array of c_g after the first phase is in direct relation to FPL and d_m.

4.3.2 Phase 2: Identification Phase

This phase is based on the identification of all the tags in the interrogation zone by using the estimated values (c_g) from the first phase. An exponential heuristic function that links c_g and L to *ws* (here it is $f(c_g, L)$) is presented (4.1) in order to provide dynamism to the value of *ws* in order to maintain a balance by reducing the

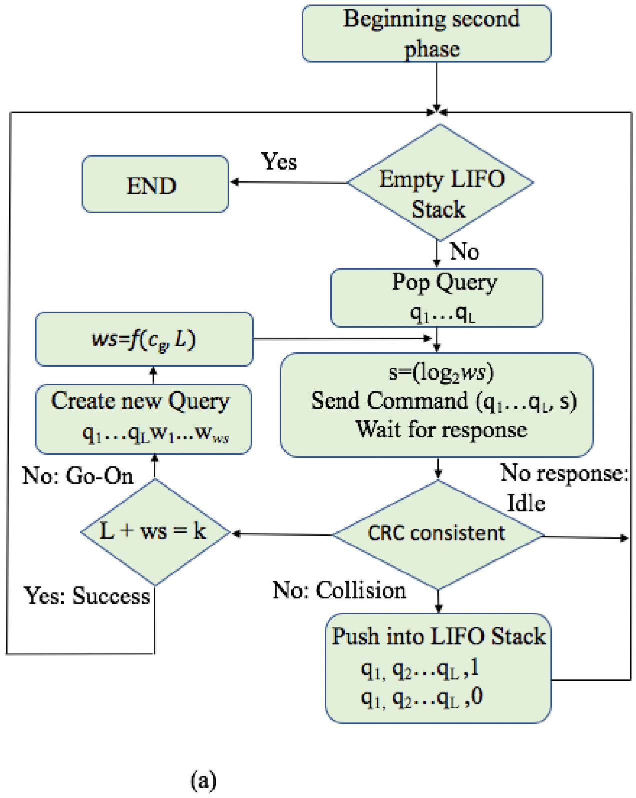

Figure 4.9: Flow chart of the proposed FQwT protocol: (**a**) for reader; (**b**) for tags.

identification time consumed, and limiting the number of go-on slots. This function is a heuristic proposed to provide better results for the adjustment of the window size when the distribution of the IDs is not homogeneous. It makes a balance by reducing the tag-transmitted bits and limiting the number of go-on slots. Every time the reader calculates *ws* it takes the appropriate c_g value from the stack and imports it into the exponential function. Also, this function takes into consideration the current length of the query *L*. The heuristic modifies ws to provide smaller values when the posibility of collision is higher, considering also the type of the distribution that the reader has identified with the parameter c_g. In the opposite situation, when *L* increases, the reader sets a larger *ws* in order to decrease the number of go-on slots. If the value of c_g is higher than *L*, the reader will adjust *ws* to c_g-*L*. *ws* can only be felt by the go-on slots, since during a collision, or an idle slot, *ws* will be held unchanged.

Equation (4.1) is adjusted with a value of the parameter β, preselected to decrease the energy consumed by the proposed protocol. In the following section, how to tune the parameter β will be detailed.

$$f(c_g, L) = \frac{c_g}{L}k(1 - e^{-\beta L}), \quad 0 < L \leq k \tag{4.1}$$

The process of identification for the second phase of the proposed FQwT protocol is depicted in Figure 4.9 in the form of a flow chart, and then described subsequently. The second phase starts by broadcasting a reader query, *[q_1...q_L]* of length *L*. In each round, the reader calculates the number of bits (*ws*) with which the tags must respond to a matching query, and converts it to the 3-bit string *s* which will be attached to the reader command.

Some pseudo-code for the FQwT reader and tags is shown for better clarity. Once the protocol begins, the reader transmits a query with appended value *s*, upon which 4 possible slot statuses can happen after a tag response: idle, collision, go-on, and success. In an idle slot, the reader will reject the last transmitted query and will pop the last pushed query from the stack for a new command. The reader will keep *ws* unchanged. A collision slot is detected when at least one colliding bit is found. The reader will not change the size *ws* from the last query, and creates two supplementary queries [q_1, q_2, . . . , q_L, 0] and [q_1, q_2, . . . , q_L, 1]. In a go-on slot, the reader creates a new query by appending the received window to the former query. The new *ws* is calculated with the heuristic function. Finally, a success slot is met with when the CRC validates the received window and the reader checks that *L* + *ws* = *k*. The reader will pop a new query from the stack and will calculate *ws* using the proposed heuristic function, according to the current distribution branch from the stack.

Reader Operation

1: $ws = 1$
2: FQwT([‘0’],ws); FQwT([‘1’],ws)
3: function $FQwT$(char [] $query$, int ws)
4: $k = ID.length$
5: $L = query.length$
6: $c_g = k - L$
7: $s = log2(ws)$

```
broadcast([query], s)
[winMatch, crcCheck] =receiveResponses()
if nSuccess = 0 then
    Phase1
    if isempty(winMatch) then
        nIdles++
    else {crcCheck = 0}
        nCollisions++
        if ws = 1 then
            Store c_g with query into LIFO (c_g = k − L)
            FQwT([query, '0'],ws); FQwT([query, '1'],ws)
        else
            ws = 1
            FQwT ([query],ws); FQwT([query],ws)
        end if
    else {crcCheck = 1}
        if L + ws < k then
            nGoons++
            ws = ws++
            FQwT([query, winMatch], ws)
        else {L + ws = k}
            ws = 1
            nSuccess++
        end if
    end if
else
    Phase2
    if isempty(winMatch) then
        nIdles++
    else {crcCheck = 0}
        nCollisions++
        FQwT([query, '0'],ws); FQwT([query, '1'],ws)
    else {crcCheck = 1}
        if L + ws < k then
            nGoons++
            ws = f(c_g, L)
```

```
            s = log2(ws)
            FQwT([query, winMatch],ws);
        else {L + ws = k}
            nSuccess++
        end if
    end if
end if
```

Tag Operation

```
receive(query, s)
L = query.length
ws = 2^s
if query = ID[0 : L − 1] then
    CRC = crc(ID[L : L+ws])
    backscatter(ID[L : L+ws], CRC)
end if
```

4.3.3 Tuning β in FQwT

The recalculation of *ws* is done using the exponential heuristic functions proposed in (4.1). This function includes a parameter β that varies the gradient of the function and must be tuned in order to minimize the number of go-on slots, energy consumed, and latency. The ideal β is the value where the protocol shows the most stable behaviour for the given simulation. For these simulations, there have been used various groups of tags (*n*=200, 400, 600, 800, 1000) and a tag ID length of *k*=128 bits. The value of c_g directly depends on the represented subdistribution in the identification process. Its influence on the heuristic function varies according to the organization of the bits in the current tag ID distribution.

The simulation results, assuming the UD, for the consumed go-on slots, latency, and energy, for a range of β values (β=0.1–0.7), using (4.1), are shown in Figure 4.10, which shows the metric results where the ideal β value is surrounded with a blue frame. The number of go-on slots used by the FQwT for different values of β and various groups of tags, *n*, is depicted in Figure 4.10 (a) and the consumed latency is shown in Figure 4.10 (b). The energy consumed by the reader using the mentioned heuristic function is shown in Figure 4.10 (c). The value of β that produces the minimum number of go-on slots, latency, and energy, is β=0.5.

The same simulations were executed for non-UDs, and their results are shown in Figure 4.11 for FPL=50, BV=20 and d_5. The results are very similar to the previous ones and prove that the ideal β is 0.5 regardless of the organization of the tag ID distribution for the

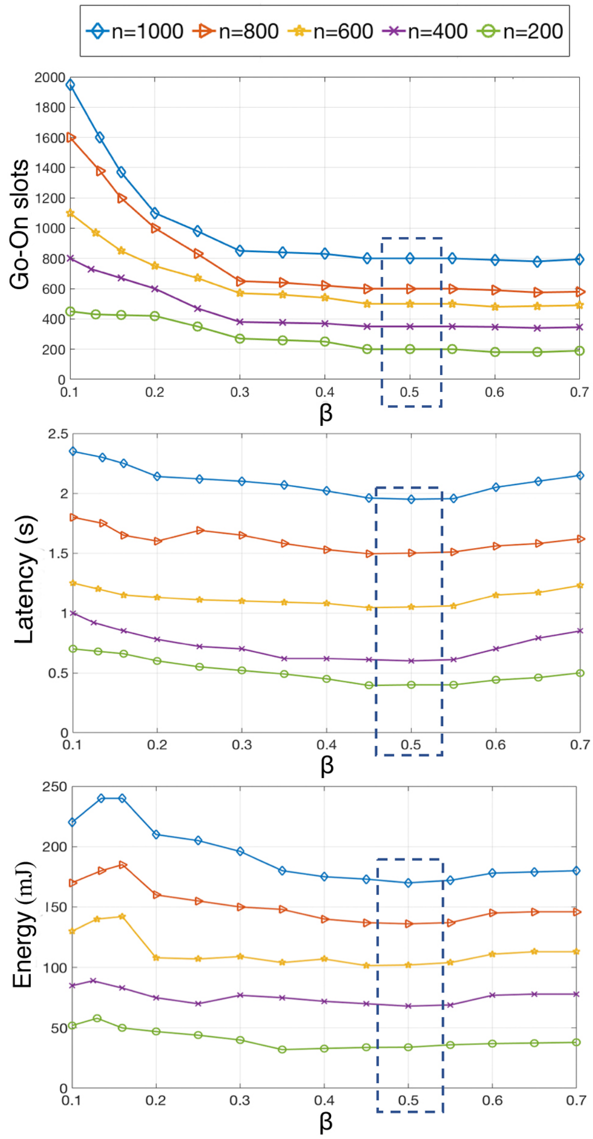

Figure 4.10: Selected β for the UD for n tags in the interrogation zone using the consumption of: (a) go-on slots, (b) latency, and (c) energy.

proposed heuristic function. This is one more piece of evidence that FQwT has a flexible behaviour under different organizations of the distributions. More comprehensive results will be presented in Chapter 5.

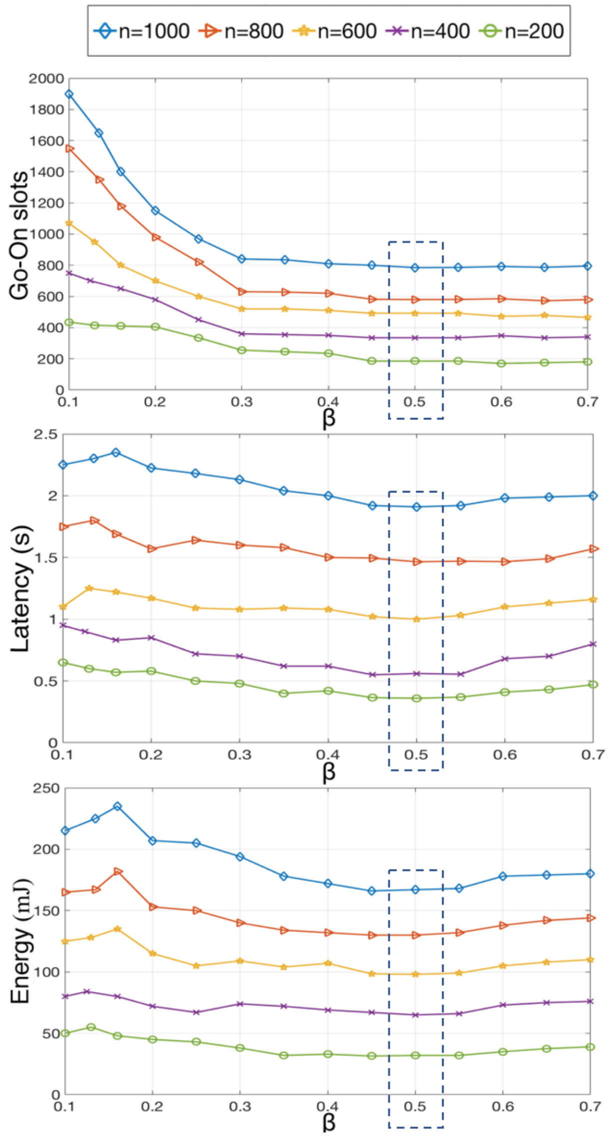

Figure 4.11: Selected β in non-UD for n tags in the interrogation zone using the consumption of: (a) go-on slots, (b) latency, and (c) energy.

4.4 Conclusion

The main requirement of the standard EPC is that all tags set in the interrogation area must have a unique ID. The tag IDs are not always uniformly distributed. Consequently, different tag ID distributions have been considered in this section. The main problem in some anti-collision protocols is that their behaviour directly depends on the type of the distribution

and on the length of the ID. Some protocols, in an RFID environment where the bits in the tag ID are randomly organized, suffer a decrease in terms of a greater consumption of bits, a longer latency, more energy consumed, as well as in terms of many other parameters.
The standardization of the tag IDs with the EPC standard provides an improvement of RFID, allowing it to access global networks. The whole set of tag IDs constitutes a distribution of binary strings with the shape of a binary tree, starting from the MSB and proceeding to the LSB. The tag IDs are structured in a binary tree, in which each node has at most two children, which are referred to as the left child and the right child. For the UD, the tag ID bits in the tree data structure are distributed with equal probability on the left and right sides of the tree. In real RFID systems, it is unrealistic to assume that the tag ID distribution is always uniform. In this thesis, the organization of a distribution is described in terms of three parameters: FPL, BV, and d_m, and it was shown in a three-cornered binary tree with potential variations.
A novel FQwT protocol with the ability to estimate the tag ID distribution has been presented and carefully analysed in this section. This protocol uses the estimation results in order to calculate the ideal number of bits with which the tags must respond to queries during the interrogation round. FQwT has flexible and efficient anti-collision features for RFID tag identification. This protocol consists of two phases: Estimation of the distribution of the IDs of the tags and identification phase. In the first phase, the reader estimates the type of distribution, stores the obtained data, and goes on to use it in the subsequent operations. This phase ends when the first tag is identified. After the last query in this phase, the reader is able to calculate the values of FPL and d_m. In the second phase, the reader uses the estimated values (c_g) from the first phase. This information is stored in LIFO. An exponential heuristic function that links c_g and L to ws is used in order to provide dynamism to the value of ws. A heuristic function is used in order to provide better results for the adjustment of the ws when the distribution of the ID is not uniform. It strikes a balance by reducing the number of tag-transmitted bits and limiting the number of go-on slots.

5

Experiments and results

This chapter contains the results of the experimentation performed to prove the previously stated hypothesis. A simulator with the aim of showing the results of the SQwT and FQwT protocols is presented. These results will be compared with the behaviour of the main memoryless anti-collision protocols in the literature. These chosen protocols are: QT, QwT, and STT. They were presented in Sections 2 and 3. They have been carefully tested using the variations of the parameters that form the distribution. All of these protocols have been simulated in the same environment, using the same conditions and parameters for a fair comparison between them. In this way, the comparison seems to be fairer and more correct. This Section is divided into 7 subsections. Firstly, the transmission model used in the simulated protocols will be presented. The second subsection presents how the hypothesis will be validated. Subsection 3 will present the simulation scenario. Subsection 4 will explain the measures used in the experimentation. In subsections 5 and 6, the simulation results of the SQwT and FQwT protocols will be presented. Subsection 7 will conclude this chapter.

5.1 Transmission Model

The transmission model used in the tested protocols is defined in Figure 5.1 and complies with the EPC global C1G2 standard. This figure shows the link times of four possible slots: collision, idle, go-on, and success. The reader interrogates tags using commands during the time period designated by t_R. Then it retains the RF downlink carrier, referred to as Continuous Wave (CW), and the tags in the interrogation area can gather the energy needed to respond to the reader. After every reader command, the tags have a period of time T_1 to generate their responses. Also, the reader needs the time T_2 to receive all the responses from the tags. The tags' responses are transmitted during the time t_T. An idle slot occurs when there is no answer to a reader command during the time period T_3.

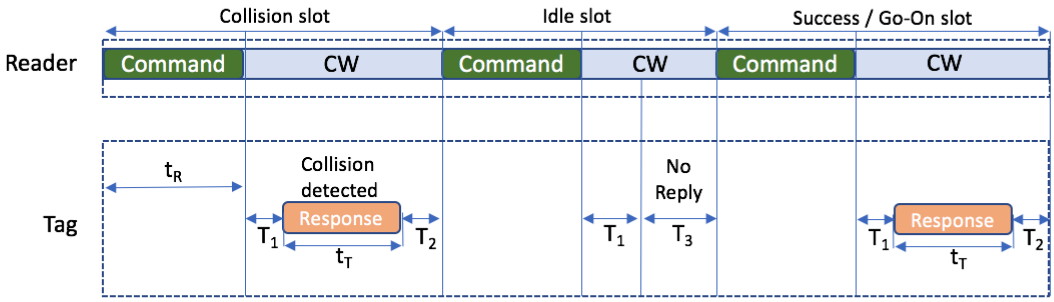

Figure 5.1: Example of a Collision/Idle/Success slot in the transmission model used.

5.2 Hypothesis validation

The main objective of the experimentation is the verification of the hypothesis. In order to define the working of the experiments, the hypothesis is again presented:

> ”It is possible to develop an RFID anti-collision protocol, using memoryless passive tags in order to reduce the energy consumption and decrease the identification time regardless the type of the tag ID distribution.”

As previously explained, this thesis deals with two different problems:

- The window does not work properly under different tag ID distributions.
- Related to the use of the window, this methodology assumes the reader can differentiate the *ws* from the query.

Two tree-based anti-collision protocols are presented in order to solve these two problems and prove, therefore, the hypothesis. The first solution, proposed in Chapter 3, the SQwT protocol, provides a standardization of the window. The presented protocol has the ability to locate the bit string representing the window size in the received command, distinguishing it from the query by using a fixed size bit-string at the end of the reader command. In the following section there will be presented the results of testing this protocol, with the aim of demonstrating a reduction of the number of bits transmitted by the reader, with respect to other protocols using the window.

The second protocol, FQwT, was presented in chapter 4 in order to estimate the tag ID distribution. This protocol has been tested under varying ID distributions to show its flexible behaviour and how it provides significant savings in identification time and energy consumption of the RFID passive system regardless of the type of ID distribution.

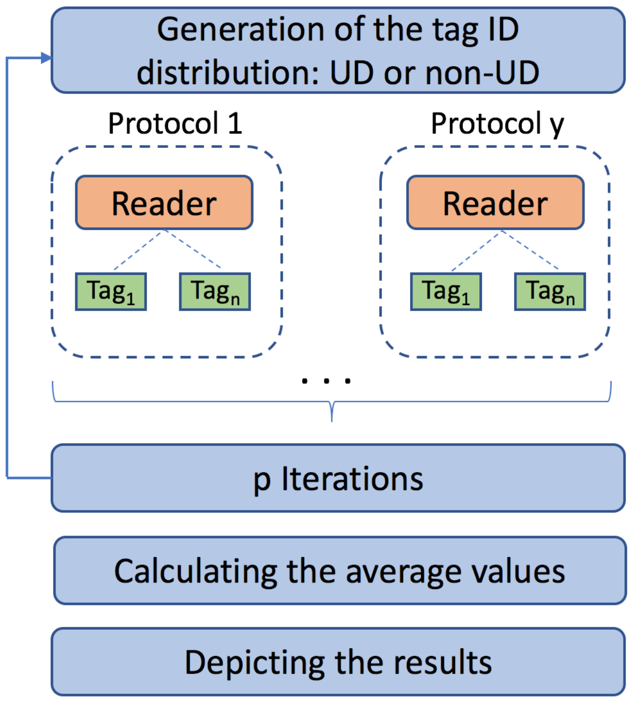

Figure 5.2: The representation of testing model.

5.3 Presentation of the simulation scenario

For the simulation of all the procedures in this thesis, the models were created and implemented in Matlab R2017b. These simulations provide the behaviour of some of the most important protocols presented in Section 2 and the ones proposed in this thesis, SQwT and FQwT, in order to obtain their performance under different simulated scenarios, such as different tag ID distributions.

All the performed simulations were averaged 100 times for accuracy in the results, and the tag IDs have a length of 128 bits. No simulation finished until all the set of tags had been identified. In the scenario of a UD, the tag IDs were randomly generated, but not in the case of a non-UD, where the distribution is organized using variable parameters that define the length, the value of the common prefix, and the number of subdistributions under different common prefixes: FPL, BV and d_m. The averages of the results of the measurements are used in order to get a graphical figure and easily make comparisons between them. Figure

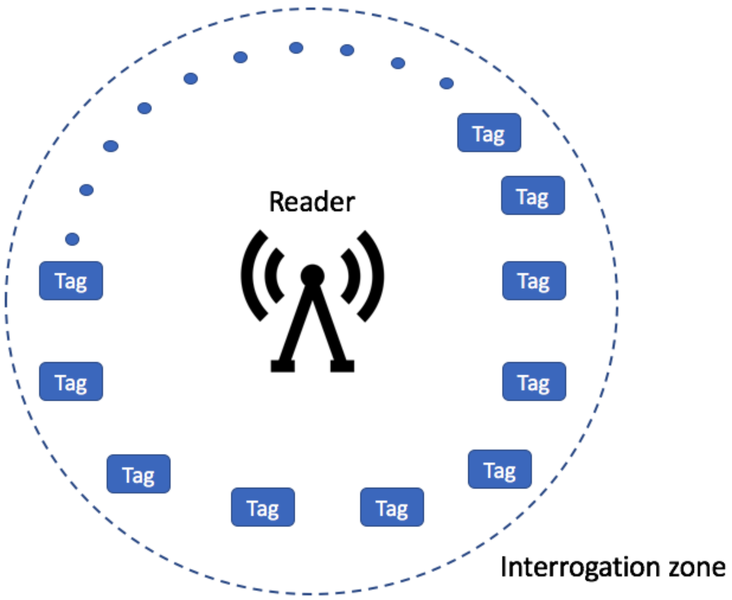

Figure 5.3: Scenario of the simulations.

The system used in this thesis is based on one reader surrounded by several tags. The tags are located at equal distances from the reader, to minimize the capture effect and the influence of detection error, in order to focus on the behaviour of the anti-collision protocol and the influence of the distribution of the tag IDs on the identification process. An example of the scenario is shown in Figure 5.3. Therefore, the following assumptions have been made in the experimentation scenario:

◇ It is assumed that the communication channel is perfect, without any disturbances in the interrogation area and the reader and tags can communicate without the influence of this issue. The aim of this thesis is to investigate the collision problem that occurs as a result of an issue with the bits rather than a problem due to the degradation of the communication signal. The signal is assumed to be powerful enough to provide unhindered transmission between the reader and the tags.

- ◇ The signals between the reader and tags are simultaneous and synchronously transmitted. In this way the tags' responses arrive at the reader at the same time.
- ◇ The distribution of tag IDs will be organised as:
  - ○ Uniform distribution (UD) is the type of distribution when there is the same probability of obtaining a 0 or 1 at every position of the tag ID. Under UD, tags are randomly generated with their ID's between [1, 2^k-1], where *k* is the tag ID length.
  - ○ Non-UD will be generated using the model defined in Section 4, using varied values for the parameters FPL, BV and d_m. This kind of distribution will be used in order to show the influence of the distribution of tag IDs on the compared anti-collision protocols.
- ◇ The communication channel can have three states: no tag answers the reader request, more than one tag answers, or just one tag responds to the reader. With the window methodology used in this thesis, another state is given – Go-On. This occurs when the tag ID received is not complete and the tag needs further transmissions to complete the sending of its ID.

5.4 Measures used in the experimentation

The first proposed protocol, SQwT, is expected to show a reduced number of reader transmitted bits in the whole identification process. Tags in the proposed window methodology can not locate the bit string representing the window size in the received command. SQwT solves this problem and provides a standardization of the window. The window is represented with a fixed size string of 3 bits.
A high priority in this section is to show the flexibility of the FQwT protocol in order to reduce latency and energy consumption regardless of the tag ID distribution. This protocol has been created in order to identify a set of tags that constitute an ID distribution. As a result, the reader classifies tags into groups determined by using a distribution estimator.
In this section the main terms used in the simulations will be explained. Moreover, the equations that are used in order to get graphical results will be presented. The measures chosen for the comparison are the following:

- ◇ The number of slots. This measure presents the total number of slots including colliding, idle, and success slots. The calculation for the protocols based on the window methodology includes also the number of Go-on slots.

◇ The number of transmitted bits. The calculation of the total transmitted bits involves: the number of bits that the reader sends to all tags in the interrogation area R_{bits}, and the number of bits received from all tags T_{bits}. Equation (5.1) shows R_{bits} from a total of *u* slots, where the number of reader transmitted bits per slot is denoted by R_{Bslot}. Equation (5.2) is determined for calculating the tag bits T_{bits}. Here, T_{Bslot} denotes the number of tag bits per slot. Finally, the average tag transmitted bits in the identification process is calculated using B_{avg} (5.3).

$$R_{bits} = \sum_{i=1}^{u} R_{Bslot} \tag{5.1}$$

$$T_{bits} = \sum_{i=1}^{u} T_{Bslot} \tag{5.2}$$

$$B_{avg} = \sum_{i=1}^{n} T_{bits}/n \tag{5.3}$$

◇ The latency *(Lat)*. The time required by an RFID system to identify all tags in the interrogation zone is known as the latency. This measure depends on the data rate of the RFID system. The total latency for this simulation is calculated by using (5.4), depending on the slot type, where Lat_c (5.5), Lat_s (5.6) and Lat_i (5.7) represent the time spent by a tag during a collision, success, and idle slot, according to the scenario shown in Figure 5.1. The number of collision, success, and idle slots are denoted by n_c, n_s and n_i. The calculation for success and go-on slots is the same for this mensure. Here, R_{Bslot} represents the number of reader bits in a slot, and T_{Bslot} the number of tag bits in each slot. On each reader command, the matching tags must respond within time T1. The reader has time T2 to receive all the transmissions. Lastly, when an idle slot occurs, the reader will wait for the tags for time T3. These intervals are indicated in Figure 5.1.

$$Lat = Lat_c + Lat_s + Lat_i \tag{5.4}$$

$$\begin{aligned} Lat_c &= \sum_{i=1}^{n_c}(t_{Ri} + T1 + t_{Ti} + T2) \\ &= \sum_{i=1}^{n_c}(\frac{R_{Bsloti}}{R_{DR}} + T1 + \frac{T_{Bsloti}}{T_{DR}} + T2) \end{aligned} \tag{5.5}$$

$$Lat_s = \sum_{i=1}^{n_s}(\frac{R_{Bsloti}}{R_{DR}} + T1 + \frac{T_{Bsloti}}{T_{DR}} + T2) \tag{5.6}$$

$$\begin{aligned} Lat_i &= \sum_{i=1}^{n_i}(T1 + T3 + t_{Ri}) \\ &= \sum_{i=1}^{n_i}(\frac{R_{Bsloti}}{R_{DR}} + T1 + T3) \end{aligned} \tag{5.7}$$

◇ The energy consumed. The energy that is wasted during the identification process is an unavoidable parameter when discussing protocol statistics. In every interrogation round, the reader will transmit the command to power up passive tags with power *Ptx*. When the reader receives a response from the tags, it will require extra power *Prx* to decode the responses from the tags. During the identification process, the total energy can be calculated using (5.8), where E_c (5.9), E_s (5.10) and E_i (5.11) represent the energy consumed during a collision, a success, and an idle slot.

$$\begin{aligned} E &= E_c + E_s + E_i \\ &= \sum_{j=0}^{n_c+n_s} [\mathrm{P}_{tx} \times (t_{Rj} + T1 + \mathrm{t}_{Tj} + T2) + \mathrm{P}_{rx} \times t_{Tj}] \\ &+ \sum_{j=0}^{n_i} [\mathrm{P}_{tx} \times (t_{Rj} + T1 + T3)] \end{aligned} \tag{5.8}$$

$$E_c = Lat_c \times Ptx + \sum_{j=0}^{n_c} Prx(\frac{T_{Bslotj}}{T_{DR}}) \tag{5.9}$$

$$E_s = Lat_s \times Ptx + \sum_{j=0}^{n_s} Prx(\frac{T_{Bslotj}}{T_{DR}}) \tag{5.10}$$

$$E_i = Lat_i \times Ptx \tag{5.11}$$

Parameter	**Definition**	**Value**
n	The number of tags	100–1000
k	The tag ID length	128 bits
CRC	Cyclic Redundancy Check procedure	5 bits
R_{DR}	Data rate for the reader	160 kbps
T_{DR}	Data rate for the tags	80 kbps
Tari	The reference time interval for a data-0 transmission	6.25 s
RTCal	Reader to tag preamble	18.75 s
TRCal	Tag to reader preamble	24.38 s
T1	Tags must respond within this time interval	26.75 s
T2	The time to receive all the transmissions	27.5 s
T3	The reader waits this time for the tags response	72 s
Ptx	Power of the reader in the transmission	825 mW
Prx	The reader receiving power	125 mW

Table 5.1: Values of the different variables used in the simulation.

5.4.1 Parametrization of the simulation to obtain measures

Table 5.1 shows all the values of the different variables used in the simulations. The number of tags *n* in the interrogation area varies from 100 to 1000 during the identification process. The tag length *k* is fixed to 128 bits since it is the most common ID length that is currently used in the standard EPC. *Tari* represents the reference time interval for a data-0 transmission. It is set to the standard's minimum of 6.25 s, conditioning, *RTCal*, *TRCal*, *T1*, *T2*, and *T3* in accordance with the EPC standard [EPCglobal 8]. *Ptx* and *Prx* were obtained from [Namboodiri 10]. Five bits are used for *CRC*. The simulation employs a data rate for the reader of 160 kbps and for the tags of 80 kbps.

Table 5.2 presents the calculation of the transmitted bits used in the simulation for the presented protocols. The number of tag bits transmitted in each slot is shown in *Tag response*. *Reader command* presents a dynamic size, which consists of the length of the current query *L* and the value *ws*, and tags would need to differentiate these values.

5.5 Simulations and results of SQwT protocol

This section presents the evaluation of the simulation results of the presented SQwT [Cmiljanic 16], with QT [Law 00], QwT [Landaluce 16] and STT [Pan 11].

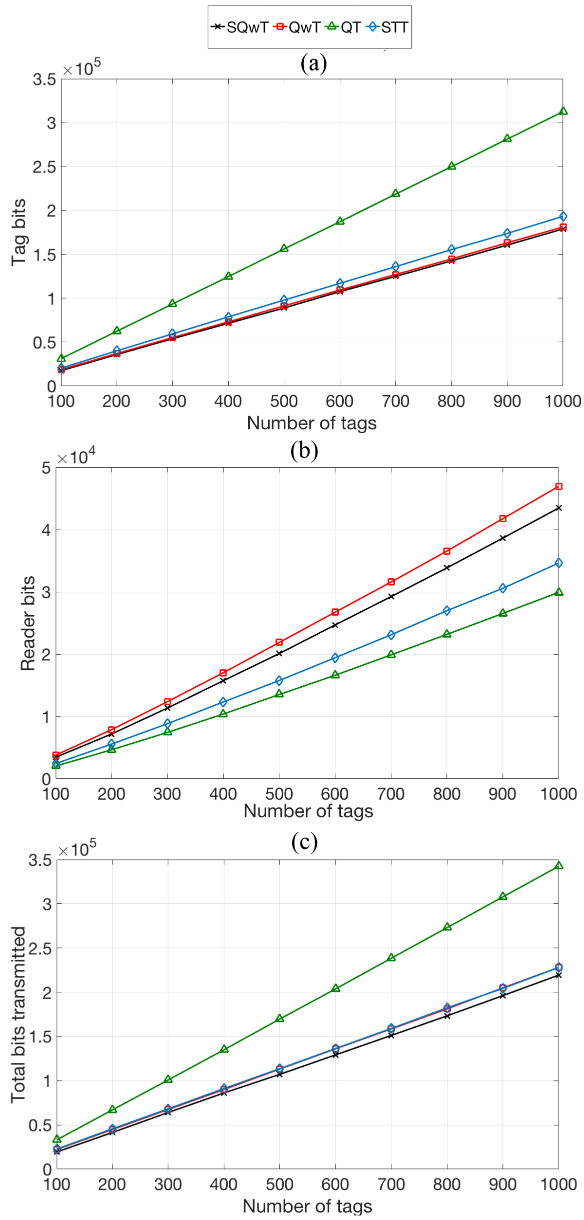

Figure 5.4: Performance of the SQwT: (a) tag bits, (b) reader bits, (c) total bits transmitted.

Protocol	Reader command	Tag response
SQwT	$L + 3$	$ws + CRC$
FQwT	$L + 3$	$ws + CRC$
QwT	$L + \lfloor \log_2 ws \rfloor + 1$	$ws + CRC$
QT	L	k
STT	L	$k - L$

Table 5.2: Calculation of transmitted bits used in simulation.

Figure 5.4(a) shows that SQwT and QwT are the least tag bit transmitting protocols, as opposed to QT and STT, which present the worst results. SQwT and QwT provide the highest savings since matching tags respond only with the number of bits specified by the window instead of the full ID length. Also, there is a minor difference between SQwT and QwT. The use of powers of 2 for *ws* causes a faster increase of *ws* in SQwT than in QwT. This involves a slight increase of tag transmitting bits at the expense of a decrease in go-on slots.

Simulated results in 5.4(b) present the reader bits used in the identification process. They evidence the decreased number of reader bits for the proposed SQwT protocol in comparison with QwT, especially in dense tag environments. The standardization of the window not only solves the issue of the differentiation between the query and the *ws* parameter, but also reduces the number of reader-trasmitted bits. The reduction of the reader bits is mainly caused by the use of 3 bits for *ws*, the slight decrease in slots, and the faster reaction to an increase of *ws*, thanks to its exponential values. This produces shorter queries than those of QwT. QwT achieves the worst results of all the protocols in the simulation. QT and STT consume the least reader bits, due to the low number of slots they need.

Figure 5.4(c) shows the total number of bits transmitted for identifying all the tags in the interrogation zone. The results indicate that the windowed protocols, SQwT and QwT, outperform QT and STT in terms of the total number of transmitted bits. The influence of the standardization of the window size reduces the number of reader transmitted bits. This keeps the total number of bits constant, so it can be seen that the standardized window contributes to a faster increase of *ws* on every query branch.

The total number of slots is shown in Figure 5.5(a). SQwT consumes fewer slots than QwT in the identification process. These protocols increase the total number of slots compared to QT and STT since the window based protocols demand Go-on slots. This happens since

the window based protocols manage to balance the number of tag transmitted bits and the number of Go-On slots, decreasing the number of bits transmitted at the expense of an increase in the total number of slots. Here, it is can also be seen that the number of slots is barely affected by the standardization of the window.
The simulation results in Figure 5.5(b) show the energy consumed by the compared protocols. The window based protocols, SQwT and QwT, present the lowest energy consumption. The worst result is obtained by the QT protocol. QT and STT waste a large number of tag bits on every collision, which increases the energy consumed by the reader during the whole process.

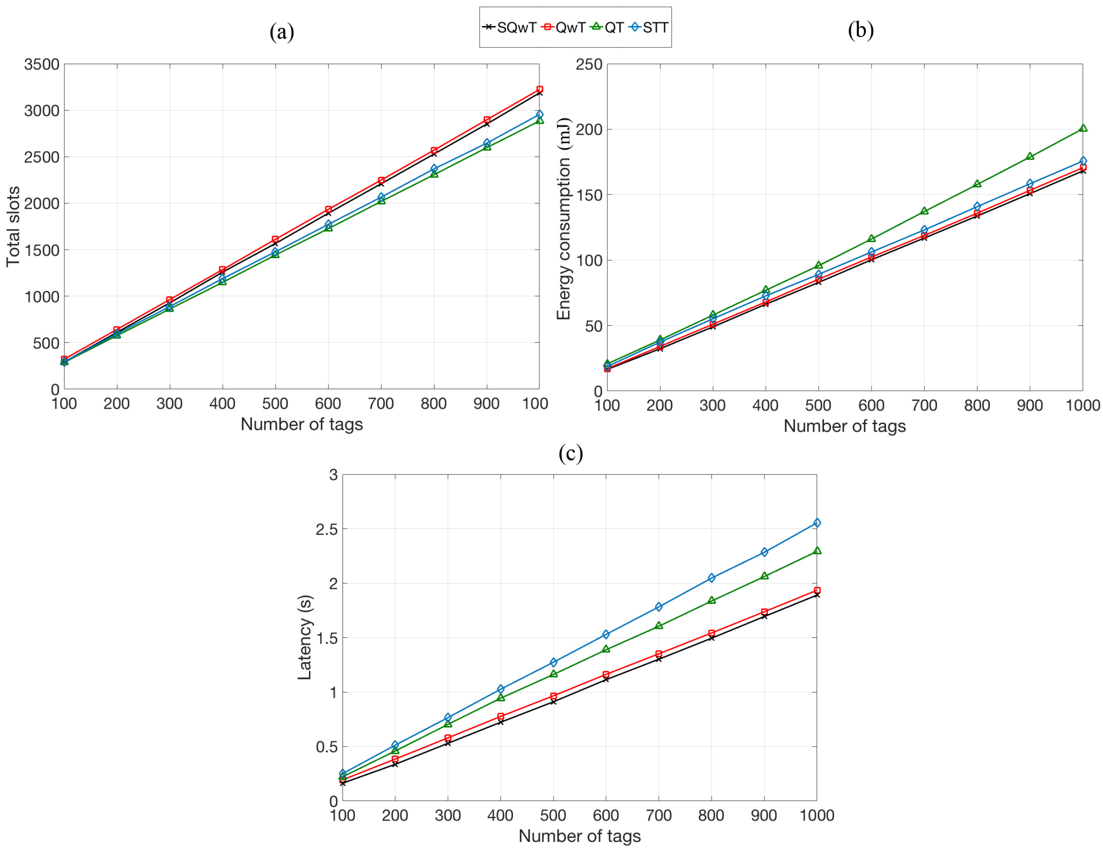

Figure 5.5: Performance of the SQwT: (a) total slots, (b) efficiency, (c) energy consumption, (d) latency.

The simulation results for the latency are shown in Figure 5.5(c). The concluding remarks will show the evidence for an improved latency for the SQwT and QwT protocols. Thus, it can be concluded that the already mentioned problem of the identification of the query and the *ws* on the reader command is now fixed, not affecting or damaging the performance of the anti-collision protocol.

5.5.1 Conclusion

The evaluation of the simulation results of the proposed SQwT protocol will be presented and carefully analysed in this section. This protocol is compared with the current anti-collision protocols: QwT, QT and STT. The basic approach of the SQwT is the ability of the tag to locate the bit string representing the window size in the received command, differentiating it from the query. Despite the fact that each reader command includes 3 bits, which involves tags responding window sizes of powers of 2, the number of tag transmitted bits is not greatly modified. Furthermore, the SQwT provides a noticeable decrease in the number of reader transmitted bits, without increasing the number of slots and tag transmitted bits. Therefore, the proposed SQwT solves the problem of differentiating the window size from the query in a reader command and also decreases the total number of bits transmitted in the identification process. The window manages to decrease the latency of the proposed protocol by reducing the number of bits transmitted on a collision.
In the following section, the influence of the distribution of tag IDs on the RFID memoryless anti-collision protocols and the results of FQwT is presented.

5.6 Simulations and results of FQwT protocol

This section presents an analysis of the influence of the tag ID distribution on the proposed FQwT protocol [Cmiljanic 17] and the following tree-based protocols: STT, QT, and QwT. The performance of the simulation protocols is compared with the aim of verifying the flexibility of FQwT in the presence of different types of ID distribution. As mentioned earlier in Section 4, three parameters have been proposed to define a tag ID distribution:

- ◇ FPL denotes a specific organization of the distribution where all tags in the interrogation area share the initial part of the ID, of that length.
- ◇ The value BV defines the horizontal position of the tag ID distribution at a given FPL level.
- ◇ The number of uniform subdistributions d_m assumes the organization of the tag ID distribution in several subdistributions, each one following an UD.

In the following subsection, there will be presented the results of several simulations, with the aim of better demonstrating the behaviour of the protocols under different tag ID

distributions. The QT, QwT, and STT protocols, from Chapter 2, are compared with FQwT while changing the adjustable parameters FPL, BV and d_m, so as to cover a wide range of possible tag ID distributions.
The rest of this section is organized as follows. In subsection 1.6.1 the influence of changing BV and FPL is presented and analysed. Subsection 1.6.2 presents experiments while varying FPL and d_m. It contains the results for: total slots, total transmitted bits, consumed latency and energy. Finally, subsection 1.6.3 concludes these experimentations.

5.6.1 Experiments under varying BV and FPL

The protocols have been tested under varying parameters BV and FPL. During these simulations, the number of subdistributions d_1 was fixed to 1. For the sake of clarity, Figure 5.6 shows a set of 4 tags, BV = 0% and BV = 100%, and FPL = 5 and FPL = 6. The first two tags have the same initial part (5 bits) of the ID, while the next 2 tags have another value in the initial part (6 bits). After these prefixes, the value of BV and the subdistribution are randomly and uniformly generated.

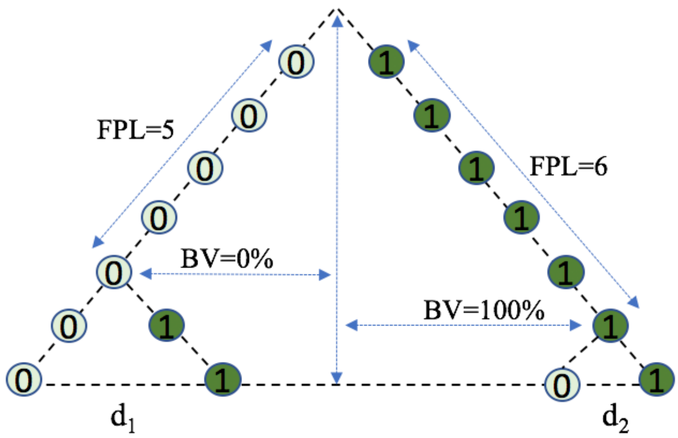

Figure 5.6: Example of set with 4 tags, 2 subdistributions with different values of FPL and BV.

The simulated results in Figure 5.7 present the total transmitted bits under d_1 and BV varying from 0% to 100%. These comparisons were performed with FPL equal to 20, 60, and 100. The simulation results for the total bits transmitted by the protocols evidences that FQwT has a very slight improvement in total bit consumption when varying FPL, in comparison with the other presented protocols. When FPL = 20, all protocols consume a similar number of transmitted bits. In the case when FPL is increased to 60, FQwT provides the highest savings compared with the other protocols. The last variation is performed when FPL = 100. FQwT, QT and STT transmitted similar number of total bits, while QwT achieves the worst results in terms of total transmitted bits. The influence of increasing FPL produces a higher number of total transmitted bits for all the protocols, but in general, FQwT provides the least bit consumption. The presented graphs in Figure 5.7 show that by changing BV, the behaviour of the FQwT, QwT, QT and STT protocol will not be affected. This means that a variation of BV does not have an influence on the bit consumption.

Figure 5.8 shows the total number of slots used in the identification process with the same scenario as in the previous figure. QwT is the most consuming protocol and shows the worst behaviour of all simulated protocols. STT wasted more slots than QT but consumes less slots than window based protocols. The use of a window increases the number of slots needed to identify the set of tags due to the generation of go-on slots. This is the main reason why FQwT and QwT need more slots than STT and QT. However, FQwT provides better behaviour than QwT in terms of slot consumption and saves more slots, since a heuristic function is applied to the go-on slots and provides an adapted *ws* to the current FPL. Also, this figure shows that BV does not affect the total slot consumption for all simulated protocols. Therefore, in the following simulations, this value was fixed to a random value.

5.6.2 Experiments under varying FPL and d_m

Here, the different types of tag ID distribution are generated by varying FPL and d_m. The tag sets are divided into a number of subdistributions and every group of subdistributions has the same initial ID part generated at the beginning of every single simulation, and fixed for all the sets of tags in the identification process. The rest of the tag ID is randomly generated. In Figure 5.6 a set of 4 tags; two subdistributions (d_2); FPL = 5 and FPL = 6 are shown. The value of BV is kept fixed since from the conclusions obtained from the previous analysis, the performance of the protocols is not affected by this parameter. BV is assigned randomly, for every type of subdistribution. In the case of one subdistribution, BV is randomly selected from a percentage of the range $[2^{FPL} - 1]$. If the tag ID distribution is organized with more subdistributions, every subdistribution has a different BV, randomly calculated from the same range. In the following simulations, FPL will be represented with three fixed values: 20, 60 and 100.

The results have also been evaluated from using one subdistribution to several subdistribution, the last case being considered as an aproximation to the UD when 11 or more subdistributions are in the tag ID distribution. The performed simulations were again parametrized using the data from Table 5.1.

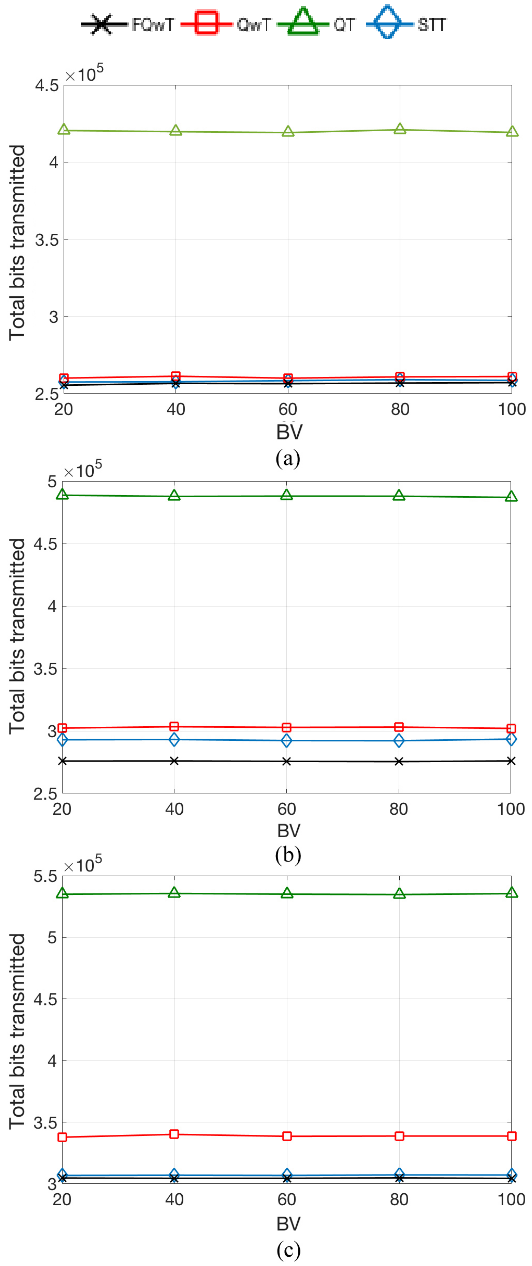

Figure 5.7: Simulation results obtained for total transmitted bits when d_1; FPL is 20, 60, and 100; and, BV varies from 20 to 100%. (**a**) FPL = 20; (**b**) FPL = 60; (**c**) FPL = 100.

In the following subsection, the results for the proposed protocols in terms of: total slots, total transmitted bits, total tag bits, total reader bits, total latency, and total energy consumed

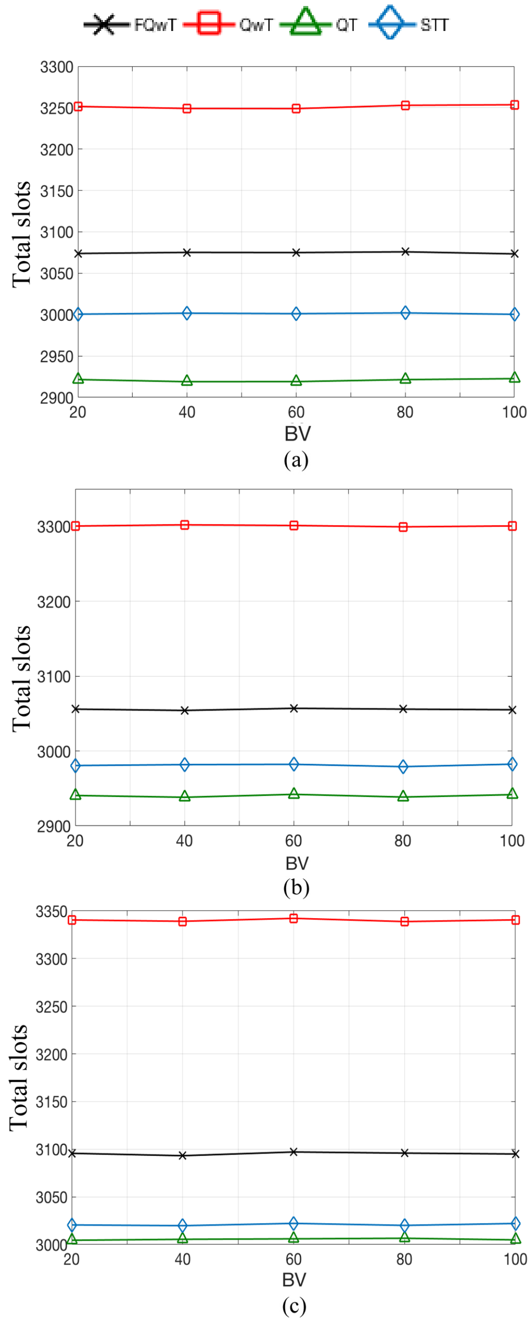

Figure 5.8: Simulation results obtained for total slots when d_1; FPL is 20, 60, and 100; and, BV varies from 20 to 100%. (**a**) FPL = 20; (**b**) FPL = 60; (**c**) FPL = 100.

during the whole identification process are presented.

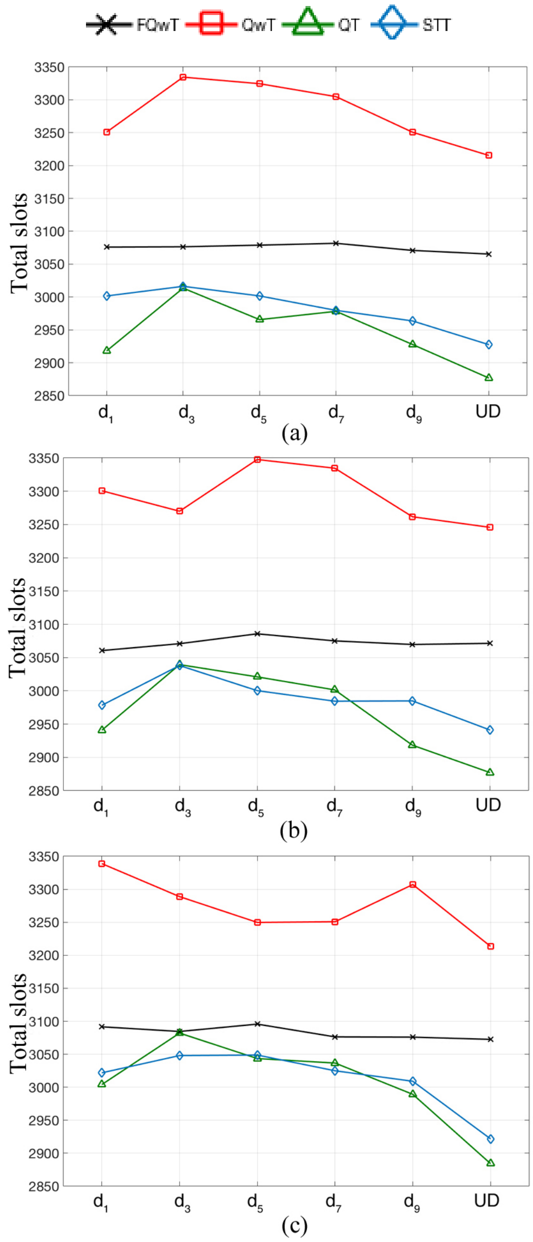

Figure 5.9: Simulation results obtained for the total slots consumption for different tag subdistributions, when FPL is 20, 60 and 100, and d_m varies from d_1 to UD. (**a**) FPL = 20; (**b**) FPL = 60; (**c**) FPL = 100.

5.6.2.1 **Total slots**

Figure 5.9 shows the number of total slots needed by each of the compared protocols. This number is the sum of the numbers of success, go-on, collision, and idle slots.

The results obtained when FPL = 20 in Figure 5.9 (a) show that FQwT provides a constant total number of consumed slots during the variation of the number of subdistributions. Unlike the rest of the protocols, QT uses less slots than the other protocols, but when d_3, QT consumes a similar number of slots as the STT protocol under the same organization of the distribution. STT has a decreasing number of slots with increasing number of subdistributions. In the case when FPL = 60, FQwT retains the same results as in the previous simulation. The highest number of consumed slots is detected when the number of subdistributions is d_5. The results from FPL = 100 show a slight increase in the slot consumption for all protocols with respect to the lower FPL values.
The heuristic function (4.1) provides a dynamism to the value of *ws* and provides a flexible behaviour during variation of d_m [Cmiljanic 17]. The ideal behaviour is when FPL = 20 from d_1 to UD. When FPL = 60 and d_5, FQwT consumes more slots than in the other cases. When FPL = 100, the proposed protocol increases slot consumption from d_1 to d_5. Under non-UD distributions, FQwT uses this function to limit the number of go-on slots. The worst results are obtained by the QwT protocol that has variable behaviour by changing d_m. This means that FQwT has improved the main problems the window methodology was suffering from under non-UD, and also provides a flexible behaviour. The last changes of FPL do not influence the behaviour of the protocols by a modification of d_m. In summary, QT provides the lowest slot consumption but FQwT has the most regular behaviour during the variation of the two variables d_m and FPL.

5.6.2.2 **Total tag bit**

This subsection evaluates the simulation results in terms of tag bits in the whole identification process. The results in terms of total tag bits are presented in Figure 5.10.
The highest consumption of tag bits for all protocols is achieved when FPL is the lowest, especially for the windowed protocols FQwT and QwT. For all FPL values, FQwT achieves constant behaviour, except for the UD case. The decrease of the tag transmitted bits with an increase of FPL means that the consumption of tag bits directly depends on the query length, since when FPL is higher, the reader uses a longer query and the tags transmit fewer bits. When the number of subdistributions varies from d_3 to d_9, all except FQwT decresed the number of tag bits. In the case of UD, FQwT consumes more bits than in the other cases. STT and QwT provide the least tag bit consumption for all the cases except for UD.
FQwT implements the window methodology and a distribution estimator, which preserves a flexible behaviour under different simulation conditions, except for UD. QwT and STT decrease the number of tag bits as d_1 goes to d_9, with respect to FQwT that shows a flexible behaviour. This is since the proposed protocol uses a standardization of the window and *ws*

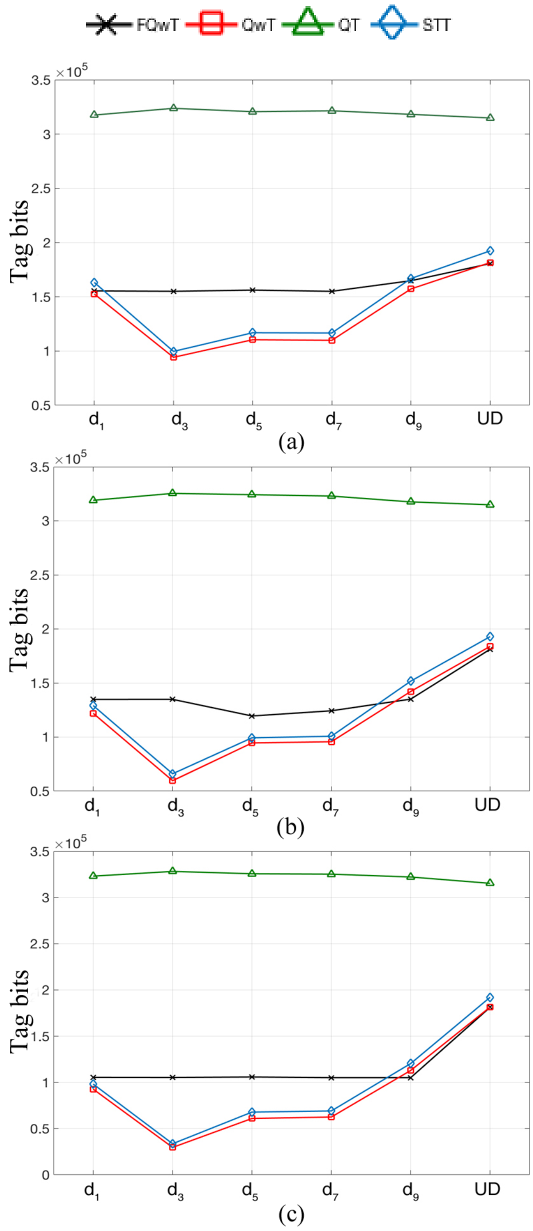

Figure 5.10: Simulation results obtained for the tag bits consumption for different tag subdistributions, when FPL is 20, 60 and 100, and d_m varies from d_1 to UD. (**a**) FPL = 20; (**b**) FPL = 60; (**c**) FPL = 100.

is transmitted as an exponent of a power of 2. When the distribution is uniformly organized, FQwT provides similar results as QwT and STT. The fact that FQwT transmit windows of a power of 2 increases the total tag bits with respect to the other windowed protocol and the rest of the protocols in the comparison.

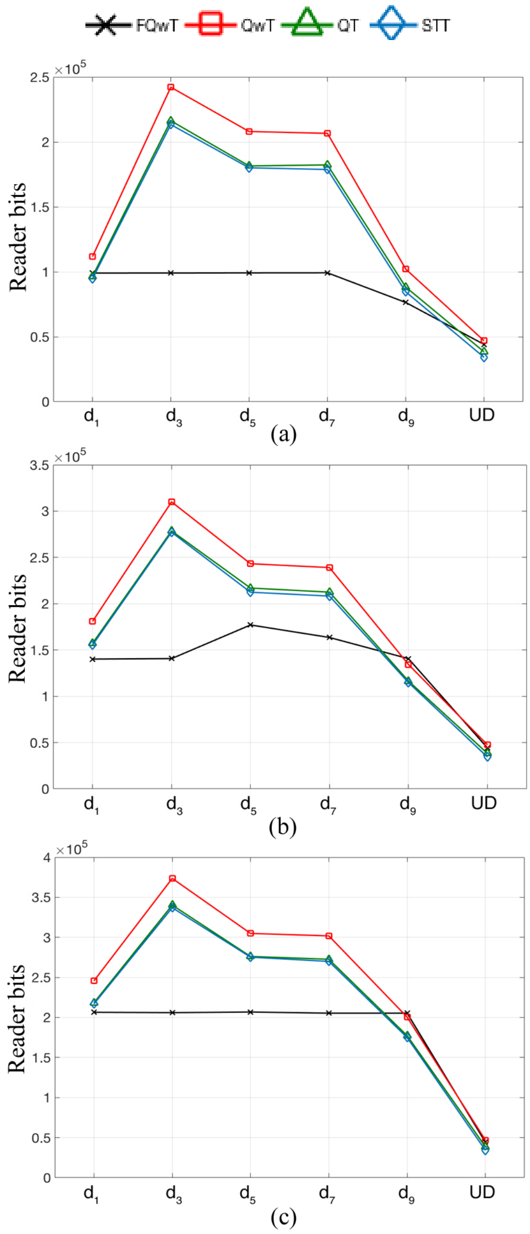

Figure 5.11: Simulation results obtained for the reader bits consumption for different tag subdistributions, when FPL is 20, 60 and 100, and d_m varies from d_1 to UD. (**a**) FPL = 20; (**b**) FPL = 60; (**c**) FPL = 100.

5.6.2.3 **Total reader bits**

In this subsection, the anti-collision protocols are validated using the total transmitted reader bits in the whole process. Figure 5.11 shows the performance of all proposed protocols in terms of this measure.

Here, FQwT outperforms the rest of the protocols, and consumes the lowest number of bits transmitted by the reader, except for the UD distribution. In terms of this metric, this protocol achieves significant savings in comparison with QwT, as opposed to the case of tag transmitted bits. Also, FQwT shows a regular behaviour between d_1 and d_9, particulary when FPL = 100. FQwT shows the highest difference between UD and the rest of the subdistributions. This is since FQwT uses an estimator that in the first phase identifies what the FPL is, and uses this to advance faster. By increasing FPL, FQwT consumes more reader bits, but still less than the other protocols in the comparison. The standardization of the window size affects beneficially the consumption of reader bits by decreasing them, but increases the tag transmitted bits instead. FQwT preserves its flexibility thanks to the estimation of the ID distribution and the heuristic function that is used in order to provide some dynamism to the value of *ws*. The other protocols have similar behaviour when changing both values. Again, QwT wastes the highest number of reader bits in order to identify the tag sets. This happens due to the higher query lengths used, particularly for high FPL values and the dynamic value of *ws*, which varies its length from one slot to another. FQwT, however, uses only 3 bits using the window standardization, and preserves the consumption of reader bits in every command.

Using the standardized window in FQwT, the number of tag bits increases, but at the same time the ID estimator contributes to decrease the number of go-ons, something which produces a decrease also in the number of reader bits. The presented results evidence a decreas in the reader bits for the proposed FQwT protocol in comparison with other protocols, especially when the number of subdistributions varies from 1 to 9 (d_1–d_9).

5.6.2.4 **Total transmitted bits**

The total number of bits transmitted by each protocol from the state of the art is depicted in Figure 5.12, when d_m and FPL vary. The presented results evidence a decrease in the total bits for the proposed FQwT protocol in comparison with the rest of the protocols from the comparison.

When FPL = 20 the best performance is provided by the FQwT protocol. By increasing this parameter, all the protocols increase the number of transmitted bits. The QT protocol achieved the worst results and wasted the highest number of bits in the identification process. FQwT underwent small changes as FPL increased to 60, but despite these changes, it still

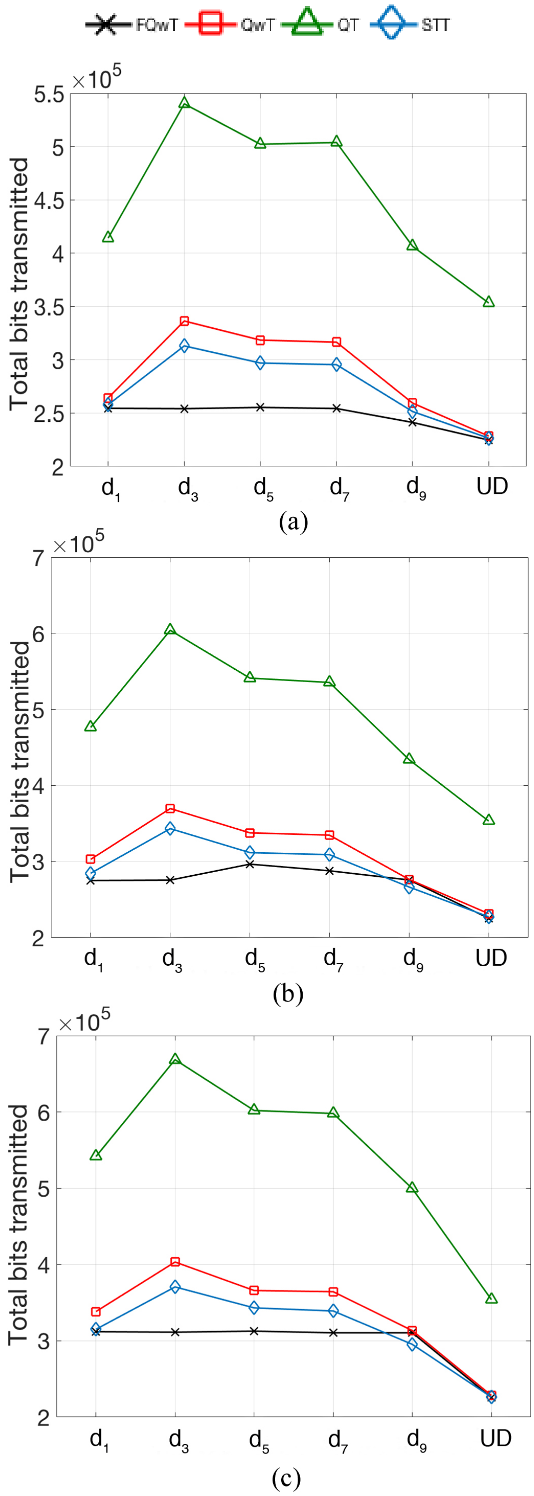

Figure 5.12: Simulation results obtained for total transmitted bits for different tag subdistributions when BV varies from 0% to 100%; FPL is 20, 60 and 100; and, d_m varies from d_1 to UD. (**a**) FPL = 20; (**b**) FPL = 60; (**c**) FPL = 100

provides the best results in comparison with other protocols. The last case, when FPL = 100, shows how FQwT has the most regular behaviour between d_1 and d_9 since the estimator works better when a clear FPL is recognizable. FQwT outperformed the other window based protocol QwT in terms of total transmitted bits, since the proposed protocol uses a standardized *ws* and therefore consumes fewer bits in the reader command.

By changing the number of subdistributions, FQwT preserves its flexibility. QwT and STT also prove to have good results under the UD but not particularly for the rest of the cases. The variation of distribution has the greatest impact on the QT protocol, which is the most affected protocol, especially when using 3, 5 or 7 subdistributions. These results indicate that FQwT significantly outperformed the other protocols when the number of subdistributions varies from 1 to 9, and provides similar results when using the UD. Despite the fact that FQwT consumes more tag bits than reader bits in comparison with other protocols, in terms of the total transmitted bits, it provides a significant reduction.

5.6.2.5 **Total Latency**

A comparison of the FQwT protocol with QT, STT, and QwT in terms of latency is presented in Figure 5.13. The reduction that occurs in the total number of bits transmitted shown in Figure 5.12 is reflected in the latency of the FQwT. Latency is, thus, affected by the total tranmitted bits and *T1, T2, T3*.

The proposed FQwT achieves a reduction in terms of total transmitted bits but consumes more slots than STT and QT. This is implicit in the reduction of the total transmitted bits. For all values of FPL, the FQwT protocol has the best results and achieves the most flexible behaviour since an increase in the number of subdistributions does not have an effect on its behaviour. The other protocols suffer from the influence of d_m and yield variable results. QwT shows a similar behaviour to that of STT when FPL = 20 and 60. However, STT improved its latency in comparison with FQwT for the case of UD. QT shows the worst behaviour and needs significantly more time to identify all the tags than all other protocols

FQwT demonstrates an almost constant behaviour during changes of FPL and d_m. The other protocols show their best behaviour when the distribution is organized uniformly. The presented protocol improves on the other window based protocol (QwT) since FQwT uses an improved heuristic function together with a distribution estimator, providing the flexibility to work under different ID distributions. All the presented variations show that FQwT decreases the identification time regardless of the type of the tag ID distribution, which proves the proposed hypothesis.

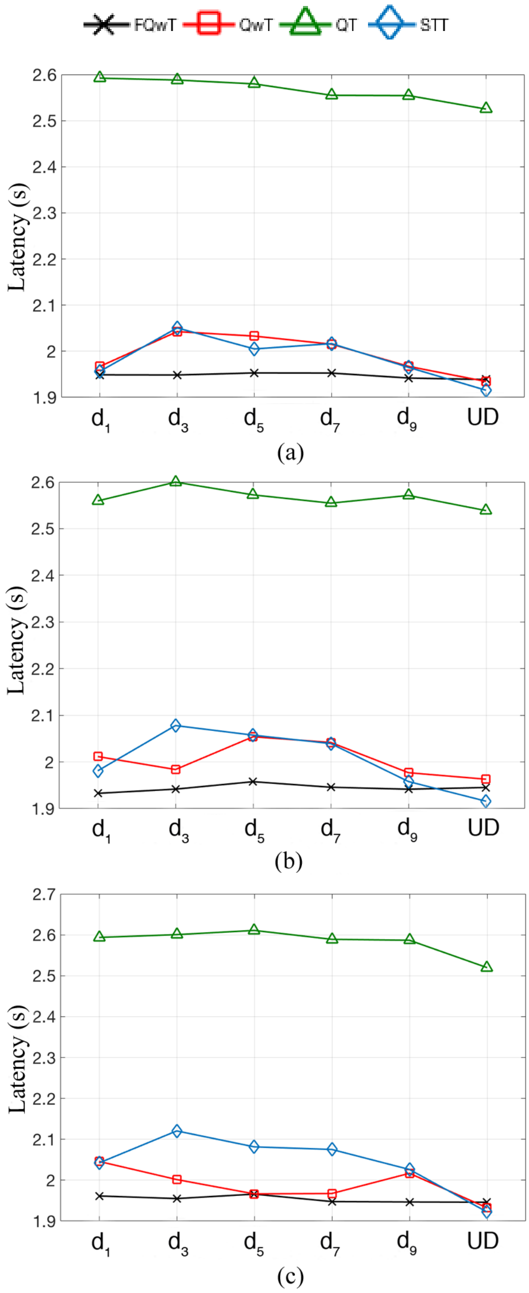

Figure 5.13: Simulation results obtained for the latency for different tag subdistributions, when FPL is 20, 60 and 100, and d_m varies from d_1 to UD. (a) FPL = 20; (b) FPL = 60; (c) FPL = 100.

5.6.2.6 Total energy consumed

The energy consumed by the reader is represented by E (5.8) and calculated during the time of transmitting and receiving information. Figure 5.14 shows the energy consumption of all presented protocols as FPL varies from 20 to 100 and the number of subdistributions from d_1 to UD.

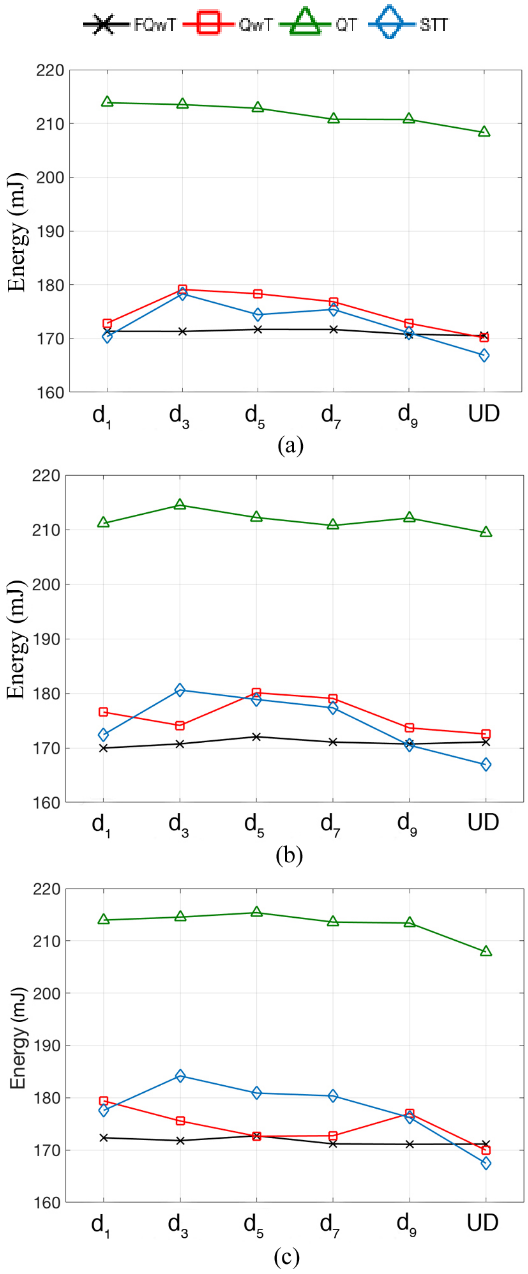

Figure 5.14: Simulation results obtained for the energy consumption for different tag subdistributions, when FPL is 20, 60 and 100, and d_m varies from d_1 to UD. (**a**) FPL = 20; (**b**) FPL = 60; (**c**) FPL = 100.

The presented figure evidences how FQwT outperforms, in terms of energy consumption, the others in environments with 3, 5, 7, and 9 subdistributions for all compared values of FPL values. Only QwT achieves similar results when the number of subdistribution is 5 and 7, when FPL = 100. STT presents the best performance under the UD for all simulated values of FPL. QT achieves the worst performance of the protocols. The other window based protocol, QwT, shows variable behaviour, only when FPL = 20 did it ourperform QT. When FPL = 60, QwT provides a similar energy consumption to that of STT under d_5 and d_7. In the last case, when FPL = 100, it has a similar behaviour to that of STT under d_1 and d_9. In other cases, QwT saves more energy than STT and QT.
As shown in the previous subsection, FQwT provides good results in terms of the identification time. These results are reflected in the total energy consumption in the whole process. The most important observation is that FQwT is the only protocol of those compared that provides a similar energy consumption for different numbers of subdistribution. Therefore, it can be confirmed that FQwT is a flexible protocol able to work under different tag ID distributions while providing a low energy consumption. From this simulation, it can be said that the hypothesis is proved.

5.6.3 Conclusion

In this chapter the results and conclusions from the experimentation of how tag ID distributions can influence tree-based memoryless protocols has been presented. A novel FQwT protocol with the ability to estimate the tag ID distribution has been presented and carefully analysed. This protocol has been compared with QwT, QT and STT in order to show the influence of changing distributions in the environment. These protocols have been simulated under different organizations of the ID distributions, and their results have been analysed. The FQwT protocol uses the estimation results in order to calculate the ideal number of bits with which the tags must respond to queries during the interrogation round. Also, FQwT demonstrates a flexibility for RFID tag identification.
The results obtained show that during a change in the number of subdistributions, FQwT maintains its flexibility, similar to that of UD, in all the metrics employed. FQwT presents the lowest number of total transmitted bits for all variations of the ID distributions modifying BV, FPL and d_m. The highest savings the proposed protocol achieves are in the reader transmitted bits, thanks to the adoption of the standardization of the window.
In addition, simulation comparisons showed that the FQwT is a protocol that outperforms the state of the art protocols in terms of reducing the number of transmitted bits, the latency, and increasing energy savings, and is thus to be considered as a good anti-collision solution in

passive RFID systems. The proposed solution proves the hypothesis that a better flexibility in tree-based protocols can be achieved, providing a fast and low energy consumption protocol for any kind of tag ID distribution.

Let the future tell the truth, and evaluate each one according to his work and accomplishments. The present is theirs; the future, for which I have really worked, is mine.

Nikola Tesla

6 Conclusions

At the beginning of the dissertation, an overview of RFID technology was presented. The related literature has been analysed in Chapter 2 and the focus was directed to tree-based protocols, since they keep the computational and memory requirements for each tag minimal; and particularly to the query based protocols. These protocols usually increase the complexity of the reader, and increase the total number of transmitted bits. Chapter 3 presented one of the main procedures of this thesis, the window standardization. Chapter 4 provided the explanations of the flexible anti-collision protocol with the aim of estimating the tag ID distribution and therefore provide improved results in terms of identification time and energy consumed by the RFID system, regardless of the ID distribution. Simulations presented in Chapter 5 provided a great amount of information in order to explain the benefits and drawbacks of the proposed protocols and compare it with the current protocols from the literature.

The thesis addresses two important issues of anti-collision protocols: the impossibility of tags differentiating between the query and the window size on every reader command, when using the window methodology, and the fact that the behaviour of the protocol is strongly affected by heterogeneous distributions.

This chapter contains the general conclusions of this thesis and consists of the following sections. Section 1 gives a general overview of the thesis. Sections 2 and 3 summarize the conclusions about the proposed protocols, SQwT and FQwT. Section 4 presents future lines of work. Finally, Section 5 collects all the conclusions of the previous sections.

6.1 General overview of the thesis

One of advantages of the RFID technology, with respect to other AutoID technologies, is that the reader can communicate with several tags at the same time. However, the reader cannot correctly read their messages at the same time due to message collisions, something that leads to retransmitting their messages until the reader successfully receives them. This problem is known as the tag collision problem, and it increases the identification time, the energy consumed, and the total transmitted bits.
For the aforementioned problem, many procedures have been developed in order to separate the individual signals from another. These are called anti-collision protocols. These protocols arbitrate tags' responses to be read in a fast and energy efficient manner and can be categorized in three groups: Aloha-based, tree-based, and hybrid protocols.

- ◇ Aloha-based protocols use a random-access strategy, where the tags transmit their own ID in randomly selected slots in a frame, significantly reducing the possibility of collision. However, these protocols do not guarantee that the whole tag set will be identified, due to the tag starvation problem. In the literature, the focus is on DFSA protocols, particularly those using estimators with the aim of more precisely predicting the number of tags in the interrogation area.

- ◇ Tree-based protocols are known as deterministic methods since they ideally finish when they have identified the whole set of tags in the interrogation area. These protocols have tags with a simple design, and work very well with a uniform set of tags. The reader manages every step in the identification process by using query prefix commands and separates responses into smaller groups until every tag is isolated to be successfully identified.

- ◇ Hybrid protocols try to avoid the problems of Aloha and tree-based protocols. They use a combination of these two methods in order to avoid their problems and to provide better features in tag identification. These protocols require complex tags, and do not completely solve the tag starvation problem found in Aloha-based protocols.

In the majority of tree-based protocols, tags respond with their full ID or with the following bits to the query received, when the query sent by the reader matches the tag ID prefix. In order to reduce the number of bits transmitted by the tag, a window methodology was presented in [Landaluce 16]. Window based protocols assume the ability of the tag to separate the query from the attached *ws*. This, however, is practically impossible, and requires the reader to indicate either the length of the query or the *ws*. This would cause

an increase of the reader transmitted bits. In order to solve this problem, this thesis has presented the Standardized Query Window Tree (SQwT) protocol.
Moreover, the tag IDs in the interrogation area are structured in a binary tree, and form the tag ID distribution. The behaviour of tree-based protocols is highly dependent on the type of this distribution and on the ID length. They are severely affected when this distribution is non-uniform. In addition, most of the protocols from the literature have been tested only under a uniform distribution. This thesis makes a second contribution in flexibilizing the anti-collision protocol under different tag ID distributions. This is called the Flexible Query window Tree (FQwT) protocol. FQwT estimates the ID distribution, so as to aggressively advance through the common parts of the IDs during the identification process. This estimator is incorporated in a tree-based protocol in order to provide savings in identification time and energy consumption of the RFID passive system.
The conclusions obtained from the two main contributions of the thesis are gathered below.

6.2 Standardized Query Window Tree protocol (SQwT)

The first presented protocol in this thesis is SQwT. This protocol applies a standardized window to the QT. Instead of transmitting an unknown number of bits for *ws* in the reader command, SQwT represents *ws* as a fixed string of 3 bits and allows tags to differentiate it from the query. SQwT transmits a new value *s*, which is the first higher power of 2 to obtain the closest value to *ws*. The reader attaches this bit string to the predefined query and transmits it. The reason why the reader uses only 3 bits for *s* is that it can provide 7 digits in the binary system when used as an exponent of 2. This covers up to 128-bit IDs, which is the most common length in the usual RFID systems.
The main advantages of the SQwT protocol are:

- ◇ SQwT solves the issue of differentiating the query bit string from the *ws* bit string, by employing a window standardization.
- ◇ There is a decrease in the reader transmitted bits and accordingly a reduction in the total transmitted bits in the identification process.
- ◇ A standardized window allows decreasing the latency of the proposed protocol by reducing the number of bits transmitted on a collision.

The basic approach of the SQwT is the tag's ability to locate the bit string representing the window size in the received command, distinguishing it from the query. In SQwT, the reader uses only three bits in every slot in order to inform the tags of how many bits to answer. The main and most clear conclusion is obtained in the reader transmitted bits, since SQwT provides an noticeable decrease in the number of reader transmitted bits, without increasing the number of slots and tag transmitted bits. Each reader command includes 3 bits, which involves tags responding window sizes of powers of 2, and the number of tag transmitted bits is not greatly increased. Futhermore, the total number of bits transmitted in the identification process also is decreased.

6.3 Flexible Query Window Tree Protocol

The second proposed protocol in this thesis is the flexible anti-collision (FQwT) protocol. FQwT estimates the tag ID distribution in order to calculate the ideal number of bits with which the tags must respond to queries during the interrogation round. The window method used in this protocol enables the protocol to aggressively advance through the common parts of the IDs. FQwT controls the protocol's behaviour, providing flexibility to work under different ID distributions, yet with similar performances in terms of latency, energy consumed, and total bits transmitted.

The main advantages of the FQwT protocol are the estimation of the distribution and the calculation of the number of bits that tags need to transmit on each reader command. After using the distribution estimator, the reader stores the obtained data and uses it with a heuristic function that provides a dynamism to the value of *ws* when the distribution is non-uniform in order to maintain a balance by reducing the identification time, consumed energy, and limiting the number of Go-On slots. A success slot involves the identification of the tag, then no change in *ws* is needed.

The experimentations presented in Chapter 5 proved that FQwT provides flexible behaviour, since the results obtained show that during a change in the number of subdistributions, FQwT keeps flexibility features similar to that of UD in all the metrics used. The highest savings the proposed protocol achieves are in the reader transmitted bits thanks to the adoption of the standardization of the window. FQwT outperformed some of the most significant tree-based protocols in terms of latency and energy, providing a flexible behaviour during the modification of the types of ID distributions.

6.4 Future work

After presenting the conclusions, promising avenues for future work for RFID anti-collision protocols are hereby proposed. The main proposals for future work are:

- ◇ Improving the existing function from the QwT protocol. The QwT protocol operates using three heuristic functions: exponential, linear, and quadratic. The proposed protocol FQwT, with the possibility of estimating the tag ID distribution, only uses the exponential function. The plan is to improve the linear and quadratic functions and test the protocol's behaviour when using them.

- ◇ Implementing an estimation of the tag ID distribution in the CwT protocol. CwT works properly under a uniform distribution and it has been tested. The proposed heuristic function for use by the CwT does not consider the possibility of a heterogeneous distribution, and the results would not be efficient. This function needs to be modified according to the feedback of the responses from the tags. Also, this protocol will provide the reader with a better insight into the interrogation area before the identification process. With these features, a novel proposed protocol would improve results such as identification time, the number of transmitted bits, and consumed energy.

- ◇ Implementting the window methodology on tags with internal memory. This option will reduce the number of transmitted bits from the reader. The reader in the proposed protocols uses the queries in order to instruct the tags on which operations to perform. With the new solution, the tags will record which part of the ID needs to be transmitted in the following slots.

6.5 Concluding remarks

RFID has been deployed almost worldwide now, for tracking valuables and personal assets as well as for safety reasons. RFID has replaced the age-old concept of barcodes, which have been in use since 1974, and it has made incredible progress, especially in the last 10 years, in which many scientific papers have been published as well as projects in areas such as: access management, tracking of goods, tracking of persons and animals, toll collection and contactless payment, machine readable travel documents, smartdust (for massively distributed sensor networks), airport baggage tracking logistics, timing sporting events, tracking and billing processes, and so on. RFID is used extensively today, around the world, and three key factors have driven a significant increase in RFID use: the reduced cost of the equipment and tags, the increased performance with a reliability of 99.9%, and a stable international standard around UHF passive systems.
RFID technology will be kept and further developed. This is obvious, since some of the world's most important companies are investing in this technology. It is offered as a tool

for businesses and consumers alike. Moreover, it will continue to reduce costs and improve security and privacy.

Acronyms

ACK *Acknowledgment command*

AIS *Automatic Identification Systems*

BS *Binary Search*

BTSA *Binary Tree Slotted Aloha*

BV *Binary Value*

CDMA *Code Division Multiple Access*

CRC *Cyclic Redundancy Check*

CRFID *Computational Radio Frequency Identification*

CT *Collision Tree*

CW *Continuous Wave*

CwT *Collision Window Tree*

DBSA *Dynamic Binary Search*

DBTSA *Dynamic Binary Tree Slotted Aloha*

DFSA *Dynamic Frame Slotted Aloha*

DR *Data Rate*

EAN *European Article Number*

EBSA *Enhanced Binary Search Algorithm*

EOF *End of Frame*

Acronyms

EPC *Electronic Product Code*

FDMA *Frequency Division Multiple Access*

FPL *Fixed Prefix Length*

FQwT *Flexible Query window Tree*

FSA *Frame Slotted Aloha*

HF *High Frequency*

IC *Integrated Circuit*

LF *Low Frequency*

LIFO *Last Input First Output*

LSB *Least Significant Bit*

MSB *Most Significant Bit*

NNB *The Number of Nonselected Bits*

Non-UD *Non Uniform Distribution*

NSB *Number of Selected Bits*

OCR *Optical Character Recognition*

OQTT *Optimal Query Tracking Tree*

PA *Pure Aloha*

PN *Pseudo-random Sequence*

QT *Query Tree*

QTP *Query Traversal Path*

QwT *Query Window Tree*

RF *Radio Frequency*

RFID *Radio Frequency Identification*

RN *Random Number*

SA *Slotted Aloha*

SC *Slot Counter*

SDMA *Space Division Multiple Access*

SN *Slot Number*

SOF *Start of Frame*

SQwT *Standardized Query window Tree*

STT *Smart Trend Transversal*

TDMA *Time Division Multiple Access*

TSA *Tree Slotted Aloha*

UD *Uniform Distribution*

UHF *Ultra High Frequency*

WS *Window Size*

Bibliography

[Abraham 2] C. Abraham, V. Ahuja, A.K. Ghosh and P. Pakanati et al. *Inventory management using passive RFID tags: A survey*. Department of Computer Science, The University of Texas at Dallas, Richardson, pp. 1–16, 2002.

[Abramson 70] N. Abramson. *THE ALOHA SYSTEM: Another alternative for computer communications*. In Proceedings of the November 17-19, 1970, Fall Joint Computer Conference, pp. 281–285. ACM, 1970

[Azambuja 08] M.C. Azambuja, C.A. Marcon and F.P. Hessel. *Survey of Standardized ISO 18000-6 RFID Anti-collision Protocols*. In 2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008), pp. 468–473, Aug 2008

[Banks 05] J. Banks, M. Pachano, L. Thompson and D. Hanny. *RFID anti-collision system using the spread spectrum technique*. Technical Report, http://www.propagation.gatech.edu/Archive/PG TR 050425 AR/PG TR 050425 AR.pdf., 2005

[Barthel 06] H. Barthel. *Regulatory status for using RFID in the UHF spectrum*. EPC Global, 2006

[Bhandari 06] N. Bhandari, A. Sahoo and S. Iyer. *Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency*. In Information Technology, 2006. 9th International Conference on, pp. 46–51, Dec 2006

[Bonuccelli 07] M.A. Bonuccelli, F. Lonetti and F.Martelli. *Tree slotted aloha: A new protocol for tag identification in RFID networks*. In Instant collision resolution for tag identification in RFID networks, Elsevier, Ad Hoc Netw., p. 8 pp.1220–1232, Nov 2007

[Bueno-Delgado 09] M.V. Bueno-Delgado, J. Vales-Alonso and F.J. Gonzalez-Castaño. *Analysis of DFSA anti-collision protocols in passive RFID environments*. In 2009 35th Annual Conference of IEEE Industrial Electronics, pp. 2610–2617, Nov 2009

[Burdet 04] L. A. Burdet. *RFID multiple access methods*. In Smart Environments Seminar, Zurich, URL: http://www. vs. inf. ethz. ch/edu/SS2004/DS/reports/06RFID-macreport.pdf, 2004

[Cha 05] J. R. Cha and J. H. Kim. *Novel Anti-collision Algorithms for Fast Object Identification in RFID System*. In 11th International Conference on Parallel and Distributed Systems (ICPADS'05), volume 2, pp. 63–67, July 2005

[Chen 09] W. T. Chen. *An Accurate Tag Estimate Method for Improving the Performance of an RFID Anticollision Algorithm Based on Dynamic Frame Length Aloha*. IEEE Transactions on Automation Science and Engineering, Vol. 6, No. 1, pp. 9–15, Jan 2009

[Choi 06] J.H. Choi, D. Lee, Y. Youn, H. Jeon and H. Lee. *Scanning-Based Pre-Processing for Enhanced Tag Anti-Collision Protocols*. In 2006 International Symposium on Communications and Information Technologies, pp. 1207–1211, Oct 2006

[Choi 07] J.H. Choi, D. Lee and H. Lee. *Query tree-based reservation for efficient RFID tag anti-collision*. IEEE Communications Letters, Vol. 11, No. 1, pp. 85–87, Jan 2007

[Cmiljanic 16] N. Cmiljanic, H. Landaluce, A. Perallos and L. Arjona. *Reducing Transmitted Bits in a Memoryless RFID Anti-Collision Protocol*. Winsys 2016, pp. 152–157, August 2016

[Cmiljanic 17] N. Cmiljanic, H. Landaluce, A. Perallos and L. Arjona. *Influence of the Distribution of Tag IDs on RFID Memoryless Anti-Collision Protocols*. Sensors, Vol. 17, No. 8, 2017

[ElectroSome 17] ElectroSome. https://electrosome.com/RFID-radio-frequency-identification/, 2017

[Eom 08] J.B. Eom, T.J. Lee, R. Rietman and A. Yener. *An efficient framed-slotted Aloha algorithm with pilot frame and binary selection for anti-collision of RFID tags*. IEEE Communications Letters, Vol. 12, No. 11, pp. 861–863, November 2008

[EPCglobal 8] EPC EPCglobal. *Radio-Frequency Identity Protocols Class- 1 Generation-2 UHF RFID Protocol for Communications at 860–960 MHz, Version 1.2.0.* EPC EPCglobal., 2008.

[Finkenzeller 10] K. Finkenzeller. RFID handbook. Wiley, 2010

[Floerkemeier 07] C. Floerkemeier. *Bayesian Transmission Strategy for Framed Aloha Based RFID Protocols*. In 2007 IEEE International Conference on RFID, pp. 228–235, March 2007

[Gou 10] H. Gou, H. Jeong and Y. Yoo. *A Bit collision detection based Query Tree protocol for anti-collision in RFID system*. In 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, pp. 421–428, Oct 2010

[GS1 11] GS1 and EPCglobal. *GS1 EPC Tag Data Standard 1.6*. GS1, EPC-global, Sept. 2011

[Hush 98] D.R. Hush and C. Wood. *Analysis of tree algorithms for RFID arbitration*. In Proceedings. 1998 IEEE International Symposium on Information Theory, pp. 107–125, Aug 1998

[Idtechex 15] Idtechex. *RFID Forecasts, Players and Opportunities 2016-2026*. http://www.idtechex.com/research/reports/rfid-forecasts-players-and-opportunities-2016-2026-000451.asp, 2015

[Jia 10] X. Jia, Q. Feng and C. Ma. *An Efficient Anti-Collision Protocol for RFID Tag Identification*. IEEE Communications Letters, Vol. 14, No. 11, pp. 1014–1016, November 2010

[Jia 12] X. Jia, Q. Feng and L. Yu. *Stability Analysis of an Efficient Anti-Collision Protocol for RFID Tag Identification*. IEEE Transactions on Communications, Vol. 60, No. 8, pp. 2285–2294, August 2012

[Klair 07] D.K. Klair, K.W. Chin and R. Raad. *An Investigation into thie Energy Eficiency of Pure and Slotted Aloha Based REID Anti-Collision Protocols*. In 2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, pp. 1–4, June 2007

[Klair 10] D.K. Klair, K.W. Chin and R. Raad. *A Survey and Tutorial of RFID Anti-Collision Protocols*. Communications Surveys Tutorials, IEEE, Vol. 12, No. 3, pp. 400–421, Third 2010

[Lai 09] Y.C. Lai and C.C. Lin. *Two Blocking Algorithms on Adaptive Binary Splitting: Single and Pair Resolutions for RFID Tag Identification*. IEEE/ACM Transactions on Networking, Vol. 17, No. 3, pp. 962–975, June 2009

[Lai 10] Y.C. Lai and L.Y. Hsiao. *General binary tree protocol for coping with the capture effect in RFID tag identification*. IEEE Communications Letters, Vol. 14, No. 3, pp. 208–210, March 2010

[Lai 12] Y.C. Lai and C.C. Lin. *Two Couple-Resolution Blocking Protocols on Adaptive Query Splitting for RFID Tag Identification.* IEEE Transactions on Mobile Computing, Vol. 11, No. 10, pp. 1450–1463, Oct 2012

[Lai 13] Y.C. Lai, L.Y. Hsiao, H.J. Chen, C.N. Lai and J.W. Lin. *A Novel Query Tree Protocol with Bit Tracking in RFID Tag Identification*. IEEE Transactions on Mobile Computing, Vol. 12, No. 10, pp. 2063–2075, Oct 2013

[Landaluce 4] H. Landaluce. *Ventana de bits: Procedimiento para reducir la latencia y el consumo energetico de protocolos de anti-colision RFID basados en queries y tags pasivos*. Doctoral thesis, University of Deusto, 2014.

[Landaluce 13] H. Landaluce, A. Perallos and I.J.G. Zuazola. *A Fast RFID Identification Protocol with Low Tag Complexity*. IEEE Communications Letters, Vol. 17, No. 9, pp. 1704–1706, September 2013

[Landaluce 14] H. Landaluce, A. Perallos and I. Angulo. *Managing the Number of Tag Bits Transmitted in a Bit-Tracking RFID Collision Resolution Protocol*. Sensors, Vol. 14, pp. 1010–1027, 2014

[Landaluce 16] H. Landaluce, A. Perallos, E. Onieva, L. Arjona and L. Bengtsson. *An Energy and Identification Time Decreasing Procedure for Memoryless RFID Tag Anticollision Protocols*. IEEE Transactions on Wireless Communications, Vol. 15, No. 6, pp. 4234–4247, June 2016

[Landt 05] J. Landt. *The history of RFID*. IEEE Potentials, Vol. 24, No. 4, pp. 8–11, Oct 2005

[Law 00] C. Law, K. Lee and K.Y. Siu. *Efficient memoryless protocol for tag identification*. In Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications,, 2000

[Lee 07] D. Lee, K. Kim and W. Lee. *Q+-Algorithm: An Enhanced RFID Tag Collision Arbitration Algorithm*. In Ubiquitous Intelligence and Computing: 4th international conference (UIC 2007), Hong Kong, China, pp. 23–32, 2007

[Leong 05] K.S. Leong, M.L. Ng and P.H. Cole. *The reader collision problem in RFID systems*. In 2005 IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, volume 1, pp. 658–661, Aug 2005

[Li 11] B. Li and J. Wang. *Efficient Anti-Collision Algorithm Utilizing the Capture Effect for ISO 18000-6C RFID Protocol*. IEEE Communications Letters, Vol. 15, No. 3, pp. 352–354, March 2011

[Lin 10] C.F. Lin and F.Y.S. Lin. *Efficient Estimation and Collision-Group-Based Anti-collision Algorithms for Dynamic Frame-Slotted Aloha in RFID Networks*. IEEE Transactions on Automation Science and Engineering, Vol. 7, No. 4, pp. 840–848, Oct 2010

[Liu 09] H. C. Liu and J.P Ciou. *Performance analysis of multi-carrier RFID systems*. In 2009 International Symposium on Performance Evaluation of Computer Telecommunication Systems, volume 41, pp. 112–116, July 2009

[Loeffler 10] A. Loeffler, F. Schuh and H. Gerhaeuser. *Realization of a CDMA-based RFID System Using a Semi-active UHF Transponder*. In 2010

6th International Conference on Wireless and Mobile Communications, pp. 5–10, Sept 2010

[Maguire 09] Y. Maguire and R. Pappu. *An Optimal Q-Algorithm for the ISO 18000-6C RFID Protocol*. IEEE Transactions on Automation Science and Engineering, Vol. 6, No. 1, pp. 16–24, Jan 2009

[Myung 06] J. Myung, W. Lee and T.K. Shih. *An Adaptive Memoryless Protocol for RFID Tag Collision Arbitration*. IEEE Transactions on Multimedia, Vol. 8, No. 5, pp. 1096–1101, Oct 2006

[Namboodir 12] V. Namboodir, M. DeSilva, K. Deegala and S. Ramamoorthy. *An extensive study of slotted Aloha-based RFID anti-collision protocols*. Computer Communications, Vol. 35, No. 16, p. 1955–1966, 2012

[Namboodiri 10] V. Namboodiri and L. Gao. *Energy-Aware Tag Anticollision Protocols for RFID Systems*. IEEE Transactions on Mobile Computing, Vol. 9, No. 1, pp. 44–59, Jan 2010

[Pan 11] L. Pan and H. Wu. *Smart Trend-Traversal Protocol for RFID Tag Arbitration*. Wireless Communications, IEEE Transactions on, Vol. 10, No. 11, pp. 3565–3569, November 2011

[Peeters 10] G.T. Peeters and B.V. Houdt. *Interference Cancellation Tree Algorithms with k-Signal Memory Locations*. IEEE Transactions on Communications, Vol. 58, No. 11, pp. 3056–3061, November 2010

[Porta 11] T.F. La Porta, G. Maselli and C. Petrioli. *Anticollision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization*. IEEE Transactions on Mobile Computing, Vol. 10, No. 2, pp. 267–279, Feb 2011

[Qian 10] C. Qian, Y. Liu, H. Ngan and L.M. Ni. *ASAP: Scalable Identification and Counting for Contactless RFID Systems*. In 2010 IEEE 30th International Conference on Distributed Computing Systems, pp. 52–61, June 2010

[Roberts 75] L.G. Roberts. *ALOHA packet system with and without slots and capture*. In ACM SIGCOMM Computer Communication Review, volume 5, p. 28–42, 1975

[Rohatgi 07] I. Rohatgi and G.D. Durgin. *RFID Applied.* John Wiley Sons, Inc., 2007

[Sayeed 09] M.S.I. Sayeed and Y.S. Kim. *A Simple LMS Algorithm Based Smart Antenna to Solve the Reader Collision Problems in RFID System.* In 2009 International Conference on Information and Multimedia Technology, pp. 426–430, Dec 2009

[Schoute 10] F. Schoute. *Dynamic Frame Length Aloha.* in IEEE Transactions on communications, vol. COM-31, NO 4, pp. 565-568, 2010

[Shih 06] D.H. Shih, P.L. Sun, D.C Yen and S.M. Huang. *Taxonomy and survey of RFID anti-collision protocols.* Computer Communications, Vol. 29, No. 11, p. 2150–2166, 2006

[T. W. Hwang 06] Y. S. Kim D. Y. Suh T. W. Hwang B. Gyo Lee and J. S. Kim. *Improved Anti-collision Scheme for High Speed Identification in RFID System.* In First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06), volume 2, pp. 449–452, Aug 2006

[Tang 07] Z. Tang and Y. He. *Research of Multi-access and Anti-collision Protocols in RFID Systems.* In 2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID), pp. 377–380, April 2007

[Technology 15] A. Technology. *EPCglobal class 1 gen 2 RFID specifications.* http://www.alientechnology.com/docs/AT wp EPCGlobal WEB.pdf, 2015

[TSL 17] TSL. *Technology Solutions.* https://www.tsl.com/, 2017

[Vales-Alonso 11] J. Vales-Alonso, V. Bueno-Delgado, E. Egea-Lopez, F. J. Gonzalez-Castano and J. Alcaraz. *Multiframe Maximum-Likelihood Tag Estimation for RFID Anticollision Protocols.* IEEE Transactions on Industrial Informatics, Vol. 7, No. 3, pp. 487–496, Aug 2011

[Vogt 02a] H. Vogt. *Efficient object identification with passive RFID tags.* In IEEE PerCom, (TX, USA), Oct 2002

[Vogt 02b] H. Vogt. *Multiple object identification with passive RFID tags*. In IEEE International Conference on Systems, Man and Cybernetics, volume 3, pp. 1–6, Oct 2002

[Wang 06] T. P. Wang. *Enhanced binary search with cut-through operation for anti-collision in RFID systems*. IEEE Communications Letters, Vol. 10, No. 4, pp. 236–238, Apr 2006

[Wang 09a] C. Wang, M. Daneshmand, K. Sohraby and B. Li. *Performance analysis of RFID Generation-2 protocol*. IEEE Transactions on Wireless Communications, Vol. 8, No. 5, pp. 2592–2601, May 2009

[Wang 09b] W. Wang, Y. Zhang, Y. Sang and S. Wang. *Analysis of anti-collision algorithms in RFID system*. In 2009 IEEE International Conference on Communications Technology and Applications, pp. 58–62, Oct 2009

[Web 15] Web. The Next Web, https://thenextweb.com/wp-content, 2015

[Wieselthier 89] J.E. Wieselthier, A. Ephremides and L.A. Michaels. *An exact analysis and performance evaluation of framed Aloha with capture*. IEEE Transactions on Communications, Vol. 37, No. 2, pp. 125–137, Feb 1989

[World 15] World. *Just Web World, Harness the Benefits of Optical Character Recognition for Your Business*. https://www.justwebworld.com/benefits-optical-character-recognition/, 2015

[Wu 13] H. Wu, Y. Zeng, J. Feng and Yu Gu. *Binary Tree Slotted Aloha for Passive RFID Tag Anticollision*. IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 1, January 2013

[Yan 08] F. Yan, X. Shao and Q. Sun. *The dynamic anti-collision algorithm based on the similar binary in RFID system*. In 2008 2nd International Conference on Anti-Counterfeiting, Security and Identification, pp. 444–447, Aug 2008

[Yan 12] X.Q. Yan, R.X. Zhang and B. Li. *Smart Trend-Traversal Protocol with Shortcutting for Memoryless RFID Tag Collision Resolution*. In 2012

9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing, pp. 857–862, Sept 2012

[Yang 11] C.N. Yang and J.Y. He. *An Effective 16-bit Random Number Aided Query Tree Algorithm for RFID Tag Anti-Collision*. IEEE Communications Letters, Vol. 15, No. 5, pp. 539–541, May 2011

[Yeh 11] M.K. Yeh, J.R. Jiang and S.T. Huang. *Parallel Response Query Tree Splitting for RFID Tag Anti-collision*. In 2011 40th International Conference on Parallel Processing Workshops, pp. 6–15, Sept 2011

[Yu 05] S. Yu, Y. Zhan, Z. Wang and Z. Tang. *Anti-collision algorithm based on jumping and dynamic searching and its analysis*. Computer Engineering, Vol. 31, No. 1, pp. 19–20, Jan 2005

[Yu 08] J. Yu, K. Liu and G. Yan. *A Novel RFID Anti-Collision Algorithm Based on SDMA*. In 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4, Oct 2008

[Zein 10] W. Zein and E. Shaaban. *An enhanced binary tree anti-collision technique for dynamically added tags in RFID systems*. In 2010 2nd International Conference on Computer Engineering and Technology, volume 7, pp. V7–349–V7–353, April 2010

[Zhang 13] L. Zhang, J. Zhang and X. Tang. *Assigned Tree Slotted Aloha RFID Tag Anti-Collision Protocols*. IEEE Transactions on Wireless Communications, Vol. 12, No. 11, pp. 5493–5505, November 2013

[Zhou 04] F. Zhou, C. Chen, D. Jin, C. Huang and H. Min. *Evaluating and Optimizing Power Consumption of Anti-Collision Protocols for Applications in RFID Systems*. In Low Power Electronics and Design, 2004. ISLPED '04. Proceedings of the 2004 International Symposium on, pp. 357–362, Aug 2004

[Zhu 10a] L. Zhu and T.S.P. Yum. *Optimal Framed Aloha Based Anti-Collision Algorithms for RFID Systems*. IEEE Transactions on Communications, Vol. 58, No. 12, pp. 3583–3592, December 2010

[Zhu 10b] L. Zhu and T.S.P. Yum. *The Optimal Reading Strategy for EPC Gen-2 RFID Anti-Collision Systems*. IEEE Transactions on Communications, Vol. 58, No. 9, pp. 2725–2733, September 2010