PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH



**UNIVERSITY YAHIA FARES OF MEDEA**

**F**aculty of **T**echnology

Department of Electrical Engineering

# Master's thesis

In : telecommunications

Specialty: systems of telecommunications

---

## Performance Comparison of Tree Based Anti-Collision Protocols in RFID Systems

---

Presented by:

TEBBAL Amina

LAGHOUATI Ibtissam

Proposed and directed by:

Dr. Kalache Mohamed Aissa

Dr. Gacemi Abdelrrezak

College year :2020-2021

## *Acknowledgments*

This dissertation was carried out in the Faculty of Technology Department of Electrical Engineering, university Yahia Fares of Medea

I would like to thank my supervisors Dr. Gacemi Abdelrrezak and Dr. Kalache Mohamed Aissa for all their help and support, which made the completion of this work possible.

I would like to thank the jury members for accepting to read and evaluate this thesis.

## *Dedication*

*This work is dedicated to:*

*My dear mother and father for their prayers ,support, sacrifice ,care, and patience. May Allah grant you a long life and good health*

*My dear sisters for their encouragement and my dear brother for his support and care.*

*My nieces and nephews*

*My grandmother*

*All my family*

*All my beloved*

*My friends Sara Dallel Nesrine and especially my partner in this thesis Ibtissam*

*and all my classmates*

***TEBBAL Amina***

## *Dedication*

*This work is dedicated to:*

*First, I would like to thank our GOD for giving me the strength to finish this study.*

*Now, how can I forget about the people who made this journey even possible,*

*In the first place, Mom and Dad, sisters "MAROUA, RIHABE, DOUAA" and brother "YOUCEF" from the bottom of my heart, I thank you for giving me support weather by encouraging me mentally or keeping me safe physically, and I can promise you the sacrifice that you made won't go in vain, and I dedicate this memory to you and I will keep striving to make you proud of me.*

*To the memory of my grand-mother.*

*To all my friends and classmates, especially my partner "AMINA TEBBAL" and "SOUHILA, IHCENE, NASSIMA, RIMA" who stuck with me through the good and the bad, who laughed with me when I'm happy and cried, supported me when I was sad, I'm lucky to have a group of supportive friends and sisters like you and I appreciate every moment we had together.*

***LAGHOUATI Ibtissam***

# *Abstract*

The radio frequency identification (RFID) system is a supporting technology of the internet of things (IoT), which allows all physical devices to be connected to IoT. It is now used in many applications. The main problem of RFID is tag collisions that occur when several tags appear in the same area of the reader at the same time. In order to solve this problem, there are many anti-collision protocols. In this thesis, we compare the performance of the tree-based anti-collision protocols and we illustrate the best protocol among the tree-based anti-collision protocols.



***Keywords:*** *RFID, Tag collision, Tree based anti-collision protocols*

## ***الملخص***

نظام تحديد الهوية بموجات الراديو هو تقنية داعمة لإنترنت الأشياء (***IoT***) ، والتي تتيح توصيل جميع الأجهزة المادية بإنترنت الأشياء. يتم استخدامه الآن في العديد من التطبيقات. المشكلة الرئيسية في نظام تحديد الهوية بموجات الراديو هي تصادم العلامات والتي تحدث عندما تظهر عدة علامات في نفس المنطقة من القارئ في نفس الوقت. لحل هذه المشكلة، هناك العديد من البروتوكولات المضادة للتصادم. في هذه المذكرة، قمنا بمقارنة أداء البروتوكولات المضادة للتصادم القائمة على الأشجار ونوضح أفضل بروتوكول بين البروتوكولات المضادة للتصادم القائمة على الأشجار.



***الكلمات المفتاحية:*** *أنظمة تحديد الهوية بموجات الراديو ،تصادم العلامات ،بروتوكولات المضادة للتصادم القائمة على الاشجار.*

## *Résumé*

Le système d'identification par radiofréquence (RFID) est une technologie de support de l'internet des objets (IoT), qui permet à tous les appareils physiques d'être connectés à l'IoT. Il est maintenant utilisé dans de nombreuses applications. Le principal problème de la RFID est la collision des tags qui se produisent lorsque plusieurs tags apparaissent en même temps dans la même zone du lecteur. Afin de résoudre ce problème, il existe de nombreux protocoles anticollision. Dans cette thèse, nous comparons les performances des protocoles anticollision basé sur des arbres et nous illustrons le meilleur protocole parmi les protocoles anticollision basé sur des arbres.



***Mots clés** : RFID, Collision de tag, les protocoles anticollision basé sur des arbres.*

## *Table of contents*

Acknowledgments ..... I
Dedication ..... II
Abstract ..... IV
Table of contents ..... VII
List of Figures ..... X
List of Tables ..... XII
Acronyms ..... XIII
General Introduction ..... 1

### *Chapter I: RFID Generality*

1. Introduction ..... 3
2. Automatic Identification Systems ..... 4
2.1. Barcode Identification Systems ..... 4
2.2.Optical character recognition ..... 5
2.3.Smart cards ..... 6
2.3.1. Memory cards ..... 7
2.3.2 Microprocessor cards ..... 7
2.4.RFID ..... 8
2.4.1. History of RFID ..... 8
2.5.RFID systems ..... 9
2.4.2. Comparison of different ID systems ..... 9
3. Components of an RFID system ..... 10
3.1.The transponder or tag ..... 11
3.1.1. Types of RFID tag ..... 11
3.1.2. Smart tags ..... 13
3.1.3. Classes of RFID tag ..... 14
3.2.The interrogator or reader ..... 14
4. Backscatter communications systems. ..... 15
4.1.Monostatic Backscatter Communications Systems ..... 15
4.2.Bistatic Backscatter Communications Systems ..... 16
4.3.Ambient Backscatter Communications Systems ..... 16
5. Operating Frequency ..... 17
5.1.Low Frequency ..... 17
5.2.Ultra-High Frequency ..... 18
5.3.Ultra-Wideband ..... 18
6. Advantages and disadvantages of RFID systems ..... 19
7. RFID applications ..... 20
7.1.Security and control applications ..... 21

7.2.Patrolling log applications ..... 21
7.3.Baggage application ..... 22
7.4.Toll road applications ..... 22
7.5.Payments in Means of Transport: Public Transports and Tolls ..... 22
7.6.Asset Management: Location, Identification and Transport Trace ..... 23
7.7.Logistics ..... 23
7.8.Inventories and Warehouses ..... 23
7.9.Identification of Animals ..... 23
7.10. Surgeries ..... 23
7.11. Access Control: Sports Facilities, Buildings, Tourism ..... 24
7.12. Passports ..... 24
7.13. The Supermarket of the Future ..... 24
7.14. Libraries and Museums ..... 24
7.15. Timing of Sports Events ..... 25
8. Challenges of RFID Technology ..... 25
8.1.Collision Problems ..... 25
8.2.Security and Privacy Concerns ..... 25
8.3.Other Challenges ..... 26
9. Conclusion ..... 26

## *Chapter II: Anti-collision Protocol*

1. Introduction ..... 27
2. The collision problem ..... 27
3. Multi-Access Methods ..... 29
3.1.SDMA ..... 29
3.2.FDMA ..... 30
3.3.CDMA ..... 31
3.4.TDMA ..... 31
4. Aloha Protocols ..... 32
4.1.Pure Aloha ..... 32
4.2.Slotted Aloha ..... 33
4.3.Framed Slotted Aloha ..... 33
4.4.Dynamic Slotted Aloha ..... 34
4.5.Q Protocol ..... 35
5. Tree-Based Protocols ..... 37
5.1.Query Tree Protocol ..... 38
5.2.Smart Trend Traversal Protocol ..... 39
5.3.Window Based Protocols ..... 41
5.4.Query Window Tree Protocol ..... 42
5.5.Binary Search Protocol ..... 46
5.6.Collision Tree Protocol ..... 47
5.7.Optimal Query Tracking Tree Protocol ..... 48
5.8.Collision Window Tree Protocol ..... 50

5.9.Tree Splitting Protocol ..... 53
6. Hybrid Protocols ..... 55
6.1.Tree Slotted Aloha ..... 55
6.2.Binary Tree Slotted Aloha ..... 56
6.3.Dynamic Binary Tree Slotted Aloha ..... 57
6.4.Adaptive Binary Tree Slotted Aloha ..... 58
7. Discussion ..... 59
8. Conclusion ..... 61

## *Chapter III: Simulation Results and discussion*

1. Introduction ..... 62
2. Performance metric ..... 62
2.1.Throughput ..... 62
2.2.Tag Identification Speed ..... 62
3. EPC generation 2 ..... 63
3.1.Communication Protocol ..... 63
3.2.Physical layer of EPC generation 2 ..... 64
3.2.1 Forward Link ..... 64
3.2.2 Backscatter Link ..... 66
3.2.3 Link Timing ..... 66
4. Frame Structure ..... 68
4.1.Query Tree or Smart Trend-Traversal Protocol ..... 68
4.2.Binary Search Protocol ..... 69
4.3.Tree Splitting Protocol ..... 69
5. Simulation Results ..... 70
6. Conclusion ..... 73
General Conclusion ..... 74
Bibliography ..... 75

# *List of figures*

Figure I.1: Overview of the most important auto-ID procedures ..... 3
Figure I.2: EAN-Code ..... 5
Figure I.3: The different areas of character recognition ..... 5
Figure I.4: Optical Character Recognition (OCR) ..... 6
Figure I.5: Typical architecture of a memory card with security logic ..... 7
Figure I.6: Components of RFID Systems ..... 10
Figure I.7: RFID Tags ..... 11
Figure I.8: Configuration of a passive RFID ..... 12
Figure I.9: Passive, semi passive and active tag a respectively ..... 13
Figure I.10: RFID reader logical components ..... 15
Figure I.11: Paradigms for backscatter communications ..... 17
Figure I.12: The RFID frequencies ..... 19
Figure I.13: General RFID applications according to its capabilities ..... 20
Figure II.1: Reader collision ..... 28
Figure II.2: Tag collision ..... 28
Figure II.3: Multi-access methods ..... 29
Figure II.4: Space Division Multiple Access (SDMA) procedure ..... 30
Figure II.5: Frequency Division Multiple Access (FDMA) procedure ..... 30
Figure II.6: Code Division Multiple Access (CDMA) procedure ..... 31
Figure II.7: Time Division Multiple Access (TDMA) procedure ..... 31
Figure II.8: An example of PA ..... 32
Figure II.9: An example of SA ..... 33
Figure II.10: Example of FSA ..... 34
Figure II.11: Example of DFSA ..... 34
Figure II.12: Flow chart of the Q ..... 36
Figure II.13: Example of the QT protocols ..... 39
Figure II.14: An example of STT protocol ..... 40
Figure II.15: Example of a communication slot between the reader and one tag ..... 41
Figure II.16: Window synchronized answer ..... 42
Figure II.17: Flow chart of QWT protocol ..... 43
Figure II.18: Flow chart of the tag ID estimation phase of the protocol FQwT protocol ..... 45
Figure II.19: Example of the CT ..... 48
Figure II.20: Flow chart of CwT protocol ..... 52
Figure II.21: Tree diagram of the BTS protocol ..... 55
Figure II.22: An example of tree slotted Aloha ..... 56
Figure II.23: An example of Binary Tree Slotted Aloha ..... 57
Figure II.24: Flow chart of Dynamic ..... 58
Figure II.25: Flow chart of Adaptive BTSA ..... 59
Figure III.1 : EPC-96 structure code ..... 63
Figure III.2: Communication protocol between a tag and a reader ..... 64

Figure III.3: PIE encoding ..... 65
Figure III.4: Forward link preamble ..... 65
Figure III.5: Frame structure of the 'ACK' command response ..... 66
Figure III.6: Link timing of the EPC G2 standard ..... 67
Figure III.7: the frame structure of query command QT protocol ..... 69
Figure III.8: the frame structure of query command BS protocol ..... 69
Figure III.9: the frame structure of unselect command ..... 69
Figure III.10: the frame structure of fail command ..... 69
Figure III.11: the frame structure of success command ..... 69
Figure III.12: Throughput of BS, QT, TS and STT versus the number of tags in a small Scale ..... 71
Figure III.13: TIS of BS, QT, TS and STT versus the number of tags in a small scale ..... 71
Figure III.14: Throughput of BS, QT, TS and STT versus the number of tags in a large Scale ..... 72
Figure III.15: TIS of BS, QT, TS and STT versus the number of tags in a large scale ..... 73

## *Liste of tables*

Table I.1: Comparison of different RFID systems showing their advantages and Disadvantages ..... 9
Table I.2: Advantages and disadvantages of RFID system ..... 20
Table II.1: A comparison of Aloha protocols ..... 37
Table II.2: A comparison of standard tree-based protocols ..... 45
Table II.3: Example of the BS protocol ..... 46
Table II.4: Example of the OQTT protocol ..... 50
Table II.5: A comparison of tree-based protocols using bit-tracking ..... 53
Table II.6: Example of identification process for BTS protocol ..... 54
Table II.7: A comparison of Aloha, Tree-based and Hybrid protocols ..... 60
Table III.1: Minimum and maximum values of T1, T2 and T3 ..... 67
Tableau III.2: Air interface of EPC G2 standard parameters used in our simulations ..... 68

## *Acronyms*

| | |
|---|---|
| AIM | Association of the Automatic Identification and Data Capture Industry |
| ACK | Acknowledge. ASC Allocated Slot Counter |
| ATM | Automated Teller Machine |
| ATP | Available to Promise AVI Automatic Vehicle Identification |
| BS | Binary Search |
| CASPIAN | Consumers against Supermarket Privacy Invasion and Numbering |
| EAN | European Article Number |
| ECR | Efficient Consumer Response |
| EPC | Electronic Product Code |
| EPC G2 | EPC Generation 2 |
| ERP | Enterprise Resource Planning |
| ETC | Electronic Toll Collection |
| HF | High Frequency |
| ID | Identification |
| IoT | Internet of things |
| ISO | International Organization for Standardization |
| LF | Low Frequency |
| MAC | Medium Access Control |
| QT | Query Tree |
| RF | Radio Frequency |
| RFID | Radio Frequency Identification |
| RN16 | Random Number 16 |
| SA | Slotted ALOHA |
| STT | Smart Trend-Traversal |
| TDMA | Time Division Multiple Access |
| TIS | Tag Identification Speed |
| TS | Tree splitting |
| UHF | Ultra High Frequency |
| UPC | Universal Product Code |
| WLAN | Wireless Local Area Network |
| WORM-tag | Write Once, Read Many – tag |

## *General Introduction*

In recent years, Auto-ID has become very popular in many service industries, procurement and distribution logistics, industrial, manufacturing companies and material flow systems. There is an automatic recognition program to provide information about people, animals, goods and products in transit.

The ubiquitous barcode label triggered a revolution in the identification system a long time ago; it was found that there were fewer and fewer places with insufficient numbers case. Bar codes may be very cheap, but their stumbling block is low prices storage capacity and the fact that it cannot be reprogrammed.

The technically best solution is to store the data in a silicon chip. This most common electronic data transmission equipment used in daily life is smart cards based on contact fields (phone smart cards, bank cards). However, the mechanical contacts used in smart cards are often impractical. The contactless data transmission between the data transmission device and its reader is much more flexible. Ideally in this case, the power required to operate the electronic data transmission equipment will also be use contactless technology to transmit from the reader. Due to the process used to transmit power and data, non-contact ID systems are called RFID systems (Radio Frequency Identification).

The number of companies actively participating in the development and sales of RFID systems shows that this is a market that should be taken seriously. And global sales of RFID systems were approximately US$900 million in 2000, and US$2.65 billion in 2005 (Krebs, n.d.). Therefore, the RFID market is the fastest growing field in the radio technology industry. Including mobile phones and cordless phones.

An RFID system is typically composed of a reader (also called interrogator), tags (also called transponders) and the host application. Tags store unique identifications (ID) and are attached to objects. A reader acts as a master for the tag and as a slave for the host application. It performs the tag interrogation process to identify objects by issuing wireless radio frequency (RF) signals. When a tag and a reader are close enough, they can communicate with each other. For such a situation, we say that the tag is in the interrogation zone of the reader.

The RFID systems suffer from tag collision, which happens when multiple tags are within the interrogation zone of a reader and simultaneously reply to the reader's commands. To handle tag collision issue, two main classes of anti-collision protocols exist: ALOHA and tree based protocols. The anti-collision protocols decrease the probability of tags collision occurrence by allowing them to transmit their own identification at a distinct time.

This thesis presents various ALOHA and tree based anti-collision protocols. In addition, we programed a simulation of tree based anti-collision protocols and comparing their performance has been also carried out. The rest of this thesis is organized as follows. Chapter I presents a generality of RFID systems. Chapter II presents different ALOHA and Tree anti-collision protocols. Chapter III shows and discuss the simulation results.

# *Chapter I*

## *RFID Generality*

## 1. Introduction

Although the foundation of the radio frequency identification (RFID) technology was laid by past generation, only recent advances opened an expanding application range to its practical implementation.

RFID is only of the numerous technologies grouped under the tem automatic identification (Auto ID), such as bar cod, magnetic inks, optical charter recognition, touch memory, smart cards, biometric etc. Auto ID technologies are a new way of controlling information and material flow, especially suitable for large productions networks.

The RFID technology is a means of gathering data about a certain item without the need of touching or seeing the data carrier, through the use of inductive coupling or electromagnetic waves. The data carrier is a microchip attached to an antenna (together called tag), the latter enabling the chip to transmit information to a host computer. The middleware (software for reading and writing tags) and the tag can be enhanced by data encryption for security-critical application at an extra cost, and anti-collision algorithms may be implemented for the tags if several of them are to be read simultaneously.

As an introduction, the following section briefly outlines different automatic ID systems that perform similar functions to RFID (Figure I.1)

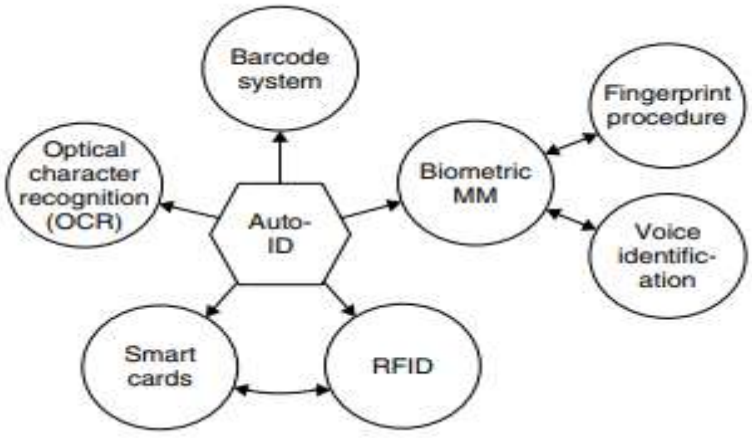



***Figure I.1:*** Overview of the most important auto-ID procedures.

# 2. Automatic Identification Systems

## 2.1 Barcode Identification Systems

The barcode has successfully corresponded with other identification systems. According to experts in the past twenty years, the total turnover of the barcode system in the early 1990s; there were approximately three billion german marks in western europe [1].

A barcode is a binary code that includes bar-shaped areas arranged in the form of bars and Gaps. Parallel configuration. They are arranged in a predetermined pattern, and represent a data element that references an associated symbol. Sequence, by the wide and narrow bars and gaps can be interpreted numerically and alphanumerically.

It is read by optical laser scanning, that is, read by different reflections of the laser beam Black stripes and white gaps [2]. However, even though they are in the same way in physical design, there are considerable differences between code layouts. There are about ten different barcode types currently in use.

The most popular barcode by some margin is the EAN code (European Article Number), this was designed specifically to fulfil the requirements of the grocery industry in 1976.

The EAN code represents a development of the universal product code (UPC) from the USA, which was introduced in the USA as early as 1973. Today, the UPC represents a subset of the EAN code, and is therefore compatible with it [1]. The EAN code is made up of 13 digits: the country identifier, the company identifier, the manufacturer's item number and a check digit. In addition to the EAN code, the following barcodes are popular in other industrial fields (see Figure I.2):

- Code Code bar: medical/clinical applications, fields with high safety requirements.
- Code 2/5 interleaved: automotive industry, goods storage, pallets, shipping containers and heavy industry.
- Code 39: processing industry, logistics, universities and libraries.

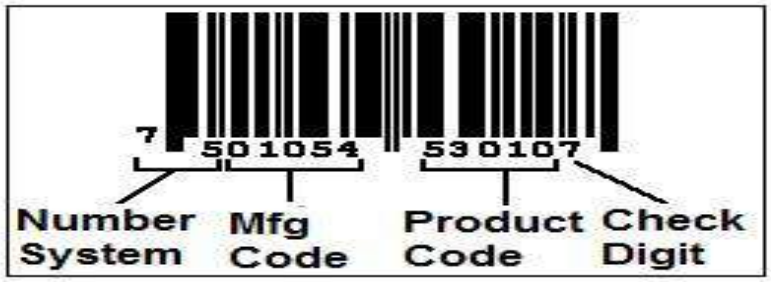



***Figure I.2:*** EAN-Code.

## 2.2 Optical Character Recognition

Optical character recognition (OCR) deals with the problem of recognizing optically processed characters. Optical recognition is performed off-line after the writing or printing has been completed, as opposed to on-line recognition where the computer recognizes the characters as they are drawn. Both hand printed and printed characters may be recognized, but the performance is directly dependent upon the quality of the input documents. The more constrained the input is, the better will the performance of the OCR system be.

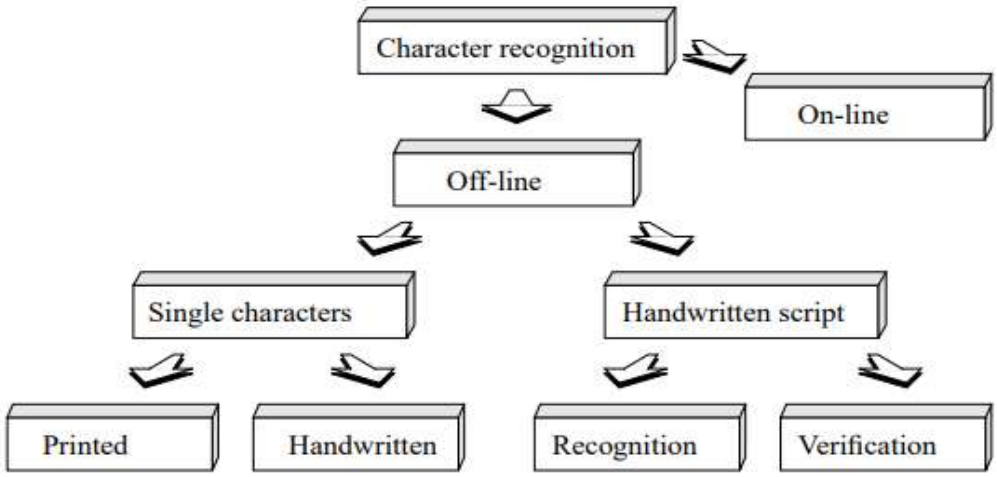



***Figure I.3:*** The different areas of character recognition.

However, when it comes to totally unconstrained handwriting, OCR machines are still a long way from reading as well as humans. However, the computer reads fast and technical advances are continually bringing the technology closer to its ideal.




***Figure I.4:*** Optical Character Recognition (OCR).

## 2.3 Smart Cards

A smart card is a small portable computer, usually the size of a credit card, without a display and a keyboard. It integrates a microprocessor, some memory, and some apps. The circular metal contact is vital to connect to the chip below and activate the card electrically. It's used in conjunction with a contact or contactless card reader (POS for payments, at the ATM, or even on your mobile phone). Smart cards contain unique features that bring many benefits to both consumers and issuing organizations.

Smart cards provide:

- Security (the physical layout of the chip is designed to be tamper-resistant, and a large portion of the operating-system code is dedicated to self-protection.)
- Confidentiality (the card and its data are with you, and you decide what to do with them.)
- Portability (you carry it in your wallet or phone.)
- Convenience (a consequence of all the above)

Smart cards offer more security and confidentiality than other financial information or transaction storage vehicles, making them a perfect solution for e-commerce transactions. As they include a tamper-resistant microprocessor, they have substantial processing pow to protect the information, encrypt and execute instructions from specific programs. Here's where smart cards have a massive advantage over magistrate cards. A smart card is a safe place to store valuable information such as private keys, account numbers, passwords, or personal information.

It's also a secure place to perform processes that one doesn't want to be exposed to the world, for example, performing a public key or private key encryption. The benefit of the smart card is that you can verify the PIN or fingerprint securely, off-line.

Smart cards come in two varieties: memory and microprocessor.

### 2.3.1 Memory Cards

In memory cards the memory — usually an EEPROM — is accessed using a sequential logic (state machine) (Figure I.5). It is also possible to incorporate simple security algorithms, e.g. stream ciphering, using this system. The functionality of the memory card in question is usually optimised for a specific application. Flexibility of application is highly limited but, on the positive side, memory cards are very cost effective. For this reason, memory cards are predominantly used in price sensitive, large-scale applications [3]. One example of this is the national insurance card used by the state pension system in Germany [4].

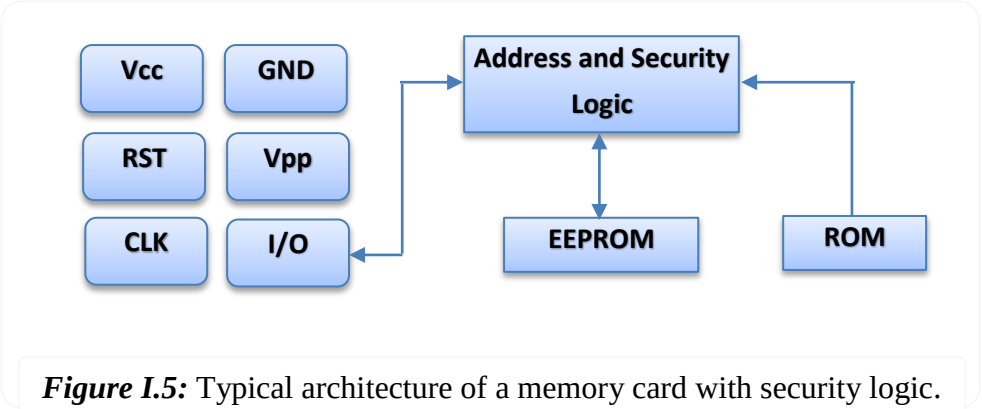



***Figure I.5:*** Typical architecture of a memory card with security logic.

### 2.3.2 Microprocessor Cards

As the name suggests, microprocessor cards contain a microprocessor, which is connected to a segmented memory (ROM, RAM and EEPROM segments). The mask programmed ROM incorporates an operating system (higher programme code) for the microprocessor and is inserted during chip manufacture. The contents of the ROM are determined during manufacturing, are identical for all microchips from the same production batch, and cannot be overwritten. The chip's EEPROM contains application data and application-related programme code. Reading from or writing to this memory area is controlled by the operating

system. The RAM is the microprocessor's temporary working memory. Data stored in the RAM are lost when the supply voltage is disconnected (Figure I.6).

Microprocessor cards are very flexible. In modern smart card systems it is also possible to integrate different applications in a single card (multi-application). The application-specific parts of the programme are not loaded into the EEPROM until after manufacture and can be initiated via the operating system. Microprocessor cards are primarily used in security sensitive applications. Examples are smart cards for GSM mobile phones and the new EC (electronic cash) cards. The option of programming the microprocessor cards also facilitates rapid adaptation to new application (Rankl and Effing, 1996) [3].

## 2.4 RFID

### 2.4.1 History of RFID

Many important scientists can be seen as the pioneers of RFID such as Benjamin Franklin in the 18th century (electricity), Michael Faraday, James Clerk Maxwell and Heinrich Rudolf Hertz in the 19th century and Albert Einstein for theirs studies in electromagnetism. Hertz was the first to transmit and receive radio waves at the turn of the 19th century and in 1896 Guglielmo Marconi transmitted radiotelegraphy across the Atlantic (Encyclopaedia Britannica 2004)[5].

The beginning of modern radio communication with Ernst F. W. Alexanderson and the birth of radar in the early 20th century pathed the way for RFID as this technology is the combination of radio broadcast technology and radar. Technical developments in radio and radar in the 1930s and 40s led to an "era of exploration of RFID" (AIM 2001)[6] in the 1950s.

The first well-known and often-mentioned application of RFID was in World War II. Allied forces used RFID in FoF (Friend or Foe) systems to identify friendly aircrafts from enemy aircrafts. In the 1960s the first widespread commercial use of RFID was the counter theft measure of electronic article surveillance (EAS). Utilizing "1-bit" tags, where only the presence or absence of a tag is detected, was and still is an inexpensive and effective measure of detecting theft on the shop floor (AIM 2001) [6].

In the 1980s the full implementation of RFID started. Interestingly the intentions of Americans and Europeans where somehow different. In the US, the studies on and the implementation of RFID concentrated on transportation, personnel access and only to a small extent on animals. Europe's focus was more on animal tagging, industrial and business

applications (AIM 2001) [6]. Electronic toll collection was a first major field of implementation of RFID mainly in the Americas.

### 2.4.2 RFID Systems

RFID System is an abbreviation of Radio Frequency Identification System. It is an "Identification system using wireless communication" that enables transferring data between "RF Tags (or Data Carriers)" that are held by men or attached to objects and "Antenna (or Reader/Writers)". It is a kind of radio communication system.

RFID systems are used in various applications. Using an RFID system allows consolidated management of objects and information. Purposes of using RFID in a production site mainly comprise the following applications.

### 2.4.3 Comparison of Different ID Systems

**Table *I.1:*** Comparison of different RFID systems showing their advantages and disadvantages.

| System parameters | Barcode | OCR | Voice recog. | Biometry | Smart card | RFID systems |
|---|---|---|---|---|---|---|
| Typical data quantity (bytes) | 1-100 | 1-100 | - | - | 16-64 k | 16-64 k |
| Data density | Low | Low | High | High | Very high | Very high |
| Machine readability | Good | Good | Expensive | Expensive | Good | Good |
| Readability by people | Limited | Simple | Simple | Difficult | Impossible | Impossible |
| Influence of dirt/damp | Very high | Very high | - | - | Possible (contact) | No influence |
| Influence of (opt.) covering | Total failure | Total failure | - | Possible | - | No influence |
| Influence of direction and position | Low | Low | - | - | Unidirectional | No influence |
| Degradation/Wear | Limited | Limited | - | - | Contacts | No influence |
| Purchase cost / reading electronics | Very low | Medium | Very high | Very high | Low | Medium |
| Operating costs (e.g. printer) | Low | Low | None | None | Medium (contacts) | None |
| Unauthorised copying / modification | Slight | Slight | Possible (audio tape) | Impossible | Impossible | Impossible |
| Reading speed (including handling of data carrier) | Low ~4s | Low ~3s | Very low >5s | Very low >5s-10s | Low ~4s | Very fast ~0.5s |
| Maximum distance between data carrier and reader | 0-50 cm | <1 cm Scanner | 0-50 cm | Direct contact | Direct contact | 0-5m, microwave |

## 3. Components of an RFID System

An RFID system is always made up of three components (Figure I.6) [7] [8]:

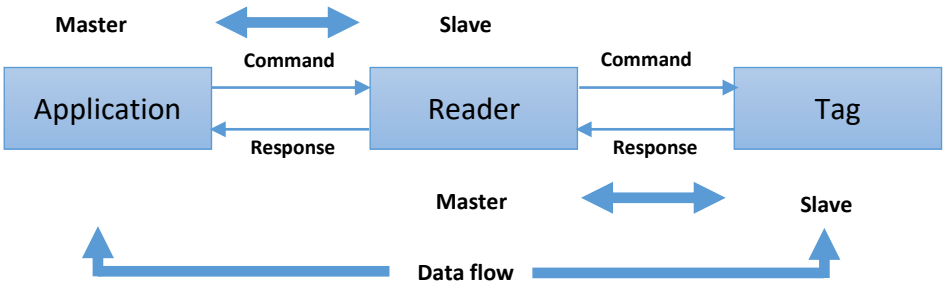



***Figure I.6:*** Components of RFID Systems.

- RFID tags also known as transponders (transmitter/ responder), are attached to the objects to count or identify. Tags could be either active or passive. Active tags are those that have partly or fully battery powered, have the capability to communicate with other tags, and can initiate a dialogue of their own with the tag reader. Passive tags, on the other hand, do not need any internal power source but are powered up by the tag reader. Tags consist mainly of a coiled antenna and a microchip, with the main purpose of storing data.
- Reader also known as transceiver (transmitter/receiver) made up of a radio frequency interface (RFI) module and control unit. Its main functions are to activate the tags, structure the communication sequence with the tag, and transfer data between the application software and tags.
- Application system also called data processing system, which can be an application or database, depending on the application. The application software initiates all readers and tags activities. RFID provides a quick, flexible, and reliable way for electronically detecting, tracking and controlling a variety of items. RFID systems use radio transmissions to send energy to a RFID tag while the tag emits a unique identification code back to a data collection reader linked to an information management system. The data collected from the tag can then be sent either directly to a host computer, or stored in a portable reader and up-loaded later to the host computer.

### 3.1 The transponder or Tag

Contain tiny semiconductor chips and miniaturized antennas inside some form of packaging [9]. They can be uniquely identified by the reader/host pair and, when applied or fastened to an object or a person, that object or person can be tracked and identified wirelessly and on the move [9]. RFID tags come in many forms. For example, some look like paper labels and are applied to boxes and packaging; others are incorporated into the walls of injection moulded plastic containers; and still others are built into wristbands and worn by people (Figure I.7).




***Figure I.7:*** RFID tags.

#### 3.1.1 Types of RFID Tags

- ***Passive RFID tags***

In passive RFID systems, the reader and reader antenna send a radio signal to the tag. The RFID tag and then uses the transmitted signal to power on, and reflect energy back to the reader.

Passive RFID systems can operate in the low frequency (LF) , high frequency (HF) or ultra-high frequency (UHF) radio bands .As passive system rages are limited by the power of the tags backscatter (the radio signal reflected from the tag back to the reader) , they are typically less than 10 m . Because passive tags do not require a power source or transmitter, and only require a tag chip and antenna, they are cheaper, smaller, and easier to manufacture than active tags.

Passive tags can be packaged in many different ways [10], depending on the specific RFID application requirement. For instance, they may be mounted on a substrate, or sandwiched between an adhesive layer and a paper label to create smart RFID labels. Passive tags may also be embedded in a variety of devices or packages to make the tag resistant to extreme temperatures or harsh chemicals.

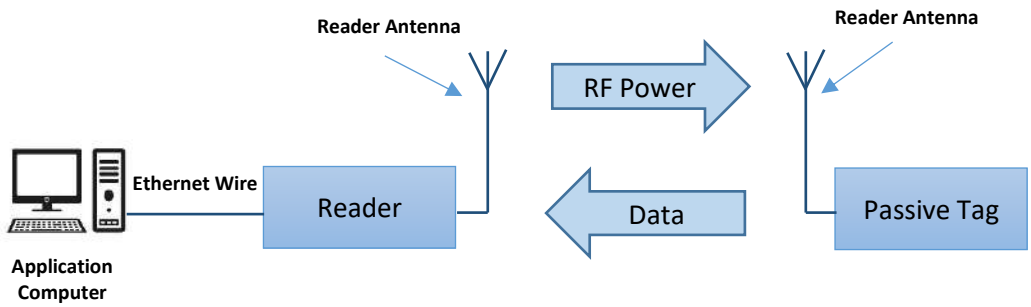



***Figure I.8:*** Configuration of a passive RFID.

- ***Active RFID tags***

In active RFID systems, tags have their own transmitter and power source. Usually, the power source is a battery. Active tags broadcast their own signal to transmit the information stored on their microchips.

Active RFID systems typically operate in the ultra-high frequency (UHF) band and offer a range of up to 100 m. In general, active tags are used on large objects [10], such as rail cars, big reusable containers and other assists that need to be tracked over long distances. (see Figure I.9)

There are two main types of active:

*Transponders:* In system that uses an active transponder tag, the reader (like passive systems) will send a signal first, and then the active transponder will send a signal back with the relevant information. Transponder tags are very efficient because they conserve battery life when the tag is out of range of the reader.

Active RFID transponders are commonly used in secure access control and in toll booth-payment systems.

*Beacons:* In a system that uses an active beacon tag, the tag will not wait to hear the reader's signal .Instead, true to its name, the tag will beacon's, or send out its specific information every 3-5 seconds. Beacons tags are very common in the oil and gas industry, as well as mining and cargo tracking applications. Active tag's beacons can be read hundreds of meters away, but , in order to conserve battery life, they may be set to a lower transmit power in order to reach around 100 meters read range .

- ***Battery Assisted Passive (BAP) or Semi-passive RFID tags***

Is a type of passive tag which incorporates a crucial active tag feature [10]. While most passive RFID tags use the energy from the RFID reader's signal to power on the tag's chip and backscatter to the reader, semi active tags use an integrated power source (usually a battery) to power on the chip, so all of the captured energy from the reader can be used for backscatter .Unlike transponders, semi active tags don't have their own transmitters.

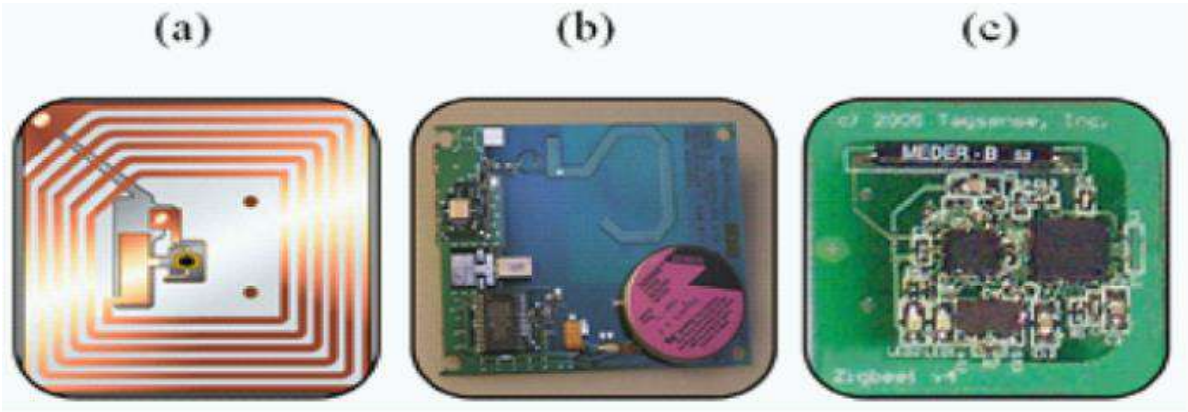



***Figure I.9:*** Passive, semi passive and active tags respectively.

### 3.1.2 Smart Tags

- ***Read only tags:*** Information is programmed onto chip during manufacturing, no overwriting, and information constant, least expensive.
- ***Write Once Read Many (WORM) tags:*** Information added only once along with unique identifier but can be read many times.

- ***Read-Write tags:*** Open to data manipulation by user's system without restrictions. It contains a unique identifier but carry an updateable memory for that to be added. It is expensive also.

### 3.1.3 Classes of RFID Tags

Tags are built to comply with a categorization called a class. Classes progressively have greater capability. EPCglobal has defined six classifications for RFID tags (0 to 5). A general description of functionality that each class is required to comply is as follows.

- **Class 0/class 1:** These classes provide the basic radio frequency (RF) passive capability. Class 0 is factory programmed. Beyond class 0, including class 1, the tags are user-programmable.
- **Class 2:** Additional functionality is added, which includes encryption and read-write RF memory.
- **Class 3:** Batteries are found on board that will power logic in the computer circuit. Class 3 provides longer range and broadband communications.
- **Class 4:** Active tags are part of the definition of class 4 tags. Peer-to-peer communications and additional sensing are also included.
- **Class 5:** Class 5 tags contain enough power to activate other tags and could be effectively classified as a reader.

Passive tags, which have no built-in power source and the power is provided by the radio frequency wave created by the reader, are usually classified in the class 0 to 3 range. Class 4 describes active tags, which have an internal power source (a battery), that provides the necessary power for the operation of the tag over a period of time. Class 5 is reserved for tag readers and active tags that can read data from other tags.

## 3.2 The interrogator or Reader

RFID reader is an external powered equipment used in RFID system for producing and accepting radio signals (GAO, 2005). A single reader can operate on multiple frequencies and this functionality depends on the vendor (Application Notes CAENRFID, 2008; Frank et al., 2006), it can have anti-collision algorithm/procedures for deducting multiple tags at one time. RFID reader works as middle-ware between tag and user application. Reader is the central part of the RFID system and communicates with tags and computer program, it supply tags information to a computer program after reading each tags unique ID. It can also perform writing onto tag, if the tag is supported. Although the reader can have multiple frequency

capability but it works on a single frequency at a time. The reader can communicate with the computer program and need either wired or wireless connection with the computer. This reader can use a wire connection with any of the following: USB, RS-232, and RS485. Otherwise, the reader can connect with the computer through Wi-Fi (known as network reader) (Sandip, 2005; Zeisel & Sabella, 2006). The reader provides various management techniques and functionality to computer programs (Zeisel & Sabella, 2006) through various built-in functions/components, these components can be understood with following figure I.10.

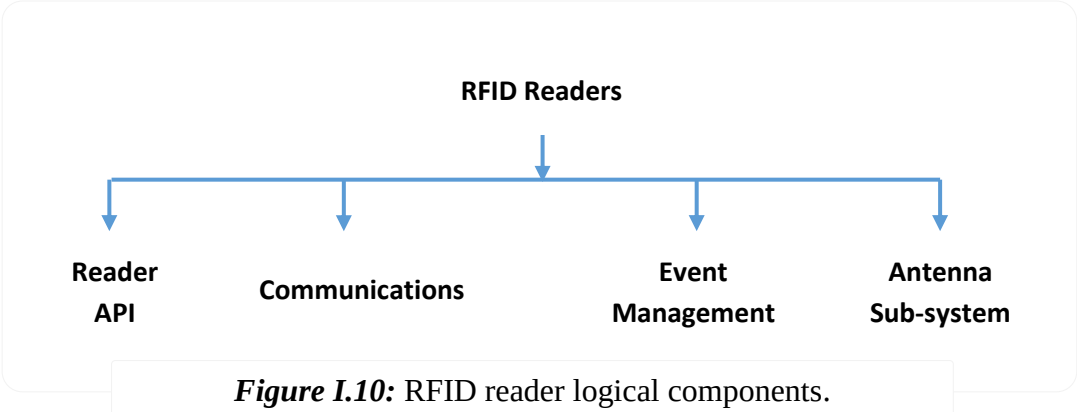



***Figure I.10:*** RFID reader logical components.

## 4. Backscatter Communications Systems

Backscatter communications systems can be classified into three major types based on their architectures: monostatic backscatter communications systems (MBCSs), BBCSs, and ABCSs as shown in Fig. I.11.

### 4.1 Monostatic Backscatter Communications Systems

In an MBCS, e.g., an RFID system, there are two main components: a backscatter transmitter, e.g., an RFID tag, and a reader as shown in Fig. I.11(a). The reader consists of, in the same device, an RF source and a backscatter receiver. The RF source generates RF signals to activate the tag. Then, the backscatter transmitter modulates and reflects the RF signals sent from the RF source to transmit its data to the backscatter receiver. As the RF source and the backscatter receiver are placed on the same device, i.e., the tag reader, the modulated signals may suffer from a round-trip path loss [11]. Moreover, MBCSs can be affected by the doubly near-far problem. In particular, due to signal loss from the RF source to the backscatter transmitter, and vice versa, if a backscatter transmitter is located far from the reader, it can

experience a higher energy outage probability and a lower modulated backscatter signal strength [12]. The MBCSs are mainly adopted for short-range RFID applications.

### 4.2 Bistatic Backscatter Communications Systems

Different from MBCSs, in a BBCS, the RF source, i.e., the carrier emitter, and the backscatter receiver are separated as shown in Fig. I.11(b). As such, the BBCSs can avoid the round-trip path loss as in MBCSs. Additionally; the performance of the BBCS can be improved dramatically by placing carrier emitters at optimal locations. Specifically, one centralized backscatter receiver can be located in the field while multiple carrier emitters are well placed around backscatter transmitters. Consequently, the overall field coverage can be expanded. Moreover, the doubly near-far problem can be mitigated as backscatter transmitters can derive unmodulated RF signals sent from nearby carrier emitters to harvest energy and backscatter data [12]. Although carrier emitters are bulky and their deployment is costly, the manufacturing cost for carrier emitters and backscatter receivers of BBCSs is cheaper than that of MBCSs due to the simple design of the components [13].

### 4.3 Ambient Backscatter Communications Systems

Similar to BBCSs, carrier emitters in ABCSs are also separated from backscatter receivers. Different from BBCSs, carrier emitters in ABCSs are available ambient RF sources, e.g., TV towers, cellular base stations, and Wi-Fi APs instead of using dedicated RF sources as in BBCSs. As a result, ABCSs have some advantages compared with BBCSs. First, because of using already-available RF sources, there is no need to deploy and maintain dedicated RF sources, thereby reducing the cost and power consumption for ABCSs. Second, by utilizing existing RF signals, there is no need to allocate new frequency spectrum for ABCSs, and the spectrum resource utilization can be improved. However, because of using modulated ambient signals for backscatter communications, there are some disadvantages in ABCSs compared with BBCSs. First, modulated ambient RF signals are unpredictable and dynamic, and act as direct interference to backscatter receivers, which largely limits the performance of an ABCS, unlike unmodulated ones of the BBCS, which can easily be eliminated before backscattered signal detection. Second, since ambient RF sources of ABCSs are not controllable, e.g., transmission power and locations, the design and deployment of an ABCS to achieve optimal performance is often more complicated than that of an BBCS.

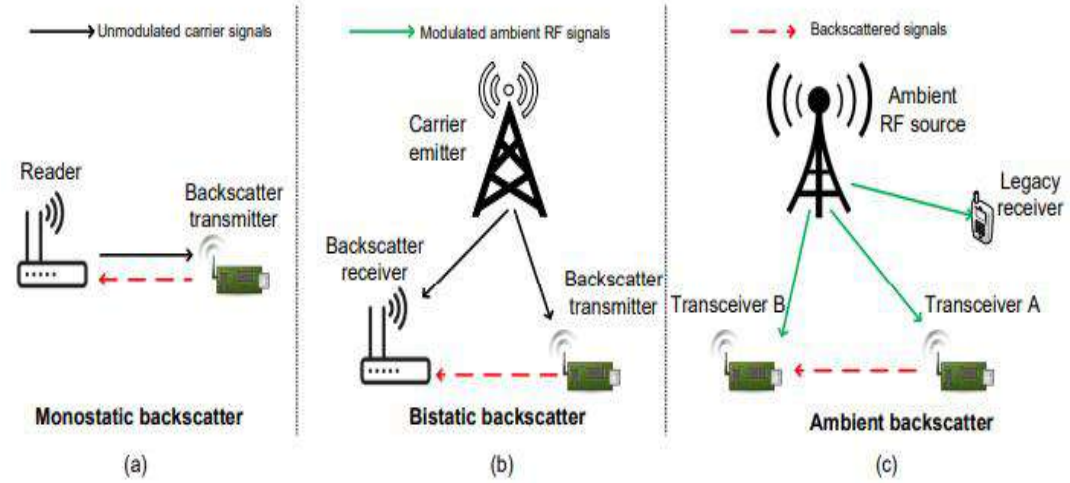



***Figure I.11:*** Paradigms for backscatter communications

## 5. Operating Frequencies

There are three main varieties of RFID tags in use today. They all operate at industrial, scientific and medical (ISM) band.

### 5.1 Low Frequency

In the early days of RFID, low-frequency (LF) tags were the most common. The LF Tags operate at 125 kHz and 134.2 kHz. Because of the electromagnetic properties at LF frequencies, those tags can be read while attached to objects containing water, animal tissues, metal, wood, and liquids. They are only suitable for proximity applications, because they can be interrogated from a very short range of only a few centimetres [15].

They have the lowest data transfer rate among all the RFID frequencies and usually store a small amount of data. The LF tags have no or limited anti-collision capabilities, therefore, reading multiple tags simultaneously is almost impossible. The LF tag antennas are usually made of a copper coil with hundreds of turns wound around a ferrite core.

Because of these properties of LF tags, they are well-suited for specific applications such as access control, asset tracking, animal identification, automotive control, vehicle immobilizer, health-care, and various point-of-sale applications. In particular, LF tags have been intensively used for animal tracking since the early 1980s.

Nowadays, the automotive industry is the largest user of LF tags. For example, in an automobile vehicle immobilizer system, an LF tag is embedded inside the ignition key. When that key is used to start the car, an RFID interrogator placed around the key slot reads the tag ID. The car can be started only if the correct ID can be read from the key.

### 5.2 Ultra-High Frequency

In the ultra-high-frequency (UHF) band, 433 MHz and 860–960 MHz are used for UHF RFID applications. The 433-MHz frequency is used for active tags, while the 860 to 960-MHz range is used mostly for passive tags. In contrast to LF and HF tags, UHF tags and interrogators use far-field coupling or backscatter coupling to communicate with one another. Therefore UHF tags have a read range of up to 20 m under good conditions. All the protocols in the UHF range have some type of anti-collision capability, allowing multiple tags to be read simultaneously [15].

The UHF tag antennas are mostly based on dipole antennas and made of copper, aluminium, or silver deposited on the substrate. The length of a resonant half-wave dipole antenna at 900 MHz is approximately 15 cm. However, the overall antenna size can be reduced through proper design techniques such as folding, fractals, double layering, and so on. The UHF antennas are easy to manufacture and, as LF tags, can be made thin using planar design, and they are almost two-dimensional.

While UHF tags offer more memory size and better read range, their performance is severely degraded when attached to objects containing water, biological tissues, and metals. The proximity of those materials will degrade the efficiency of the tag through absorption and detuning. The range will also be affected by propagation effects in unfavourable environments. UHF tags cannot be read if water or any conductive material is placed between the interrogator antenna and the tags.

### 5.3 Ultra-Wideband

According to the federal communications commission (FCC) regulations, any radio technology that communicates over a bandwidth exceeding the lesser of 500 MHz or 20% of the arithmetic centre frequency is classified as ultra-wideband (UWB). Due to the extremely low emission levels currently allowed by regulatory agencies, UWB systems tend to be short-range. The benefit of a UWB system is a high data rate that can enable wireless communication between devices at a very high speed. UWB is also used in real-time location systems.

UWB operates by emitting a series of extremely short pulses (billionths of a second or shorter) across a band of frequencies simultaneously. The FCC has cleared 3.1 GHz to 10.6 GHz to be used by UWB devices [15]. UWB devices are less prone to interfere with existing devices (usually less than 1 GHz) because of their operation frequency range. All the UWB RFID tags currently in the market are active.

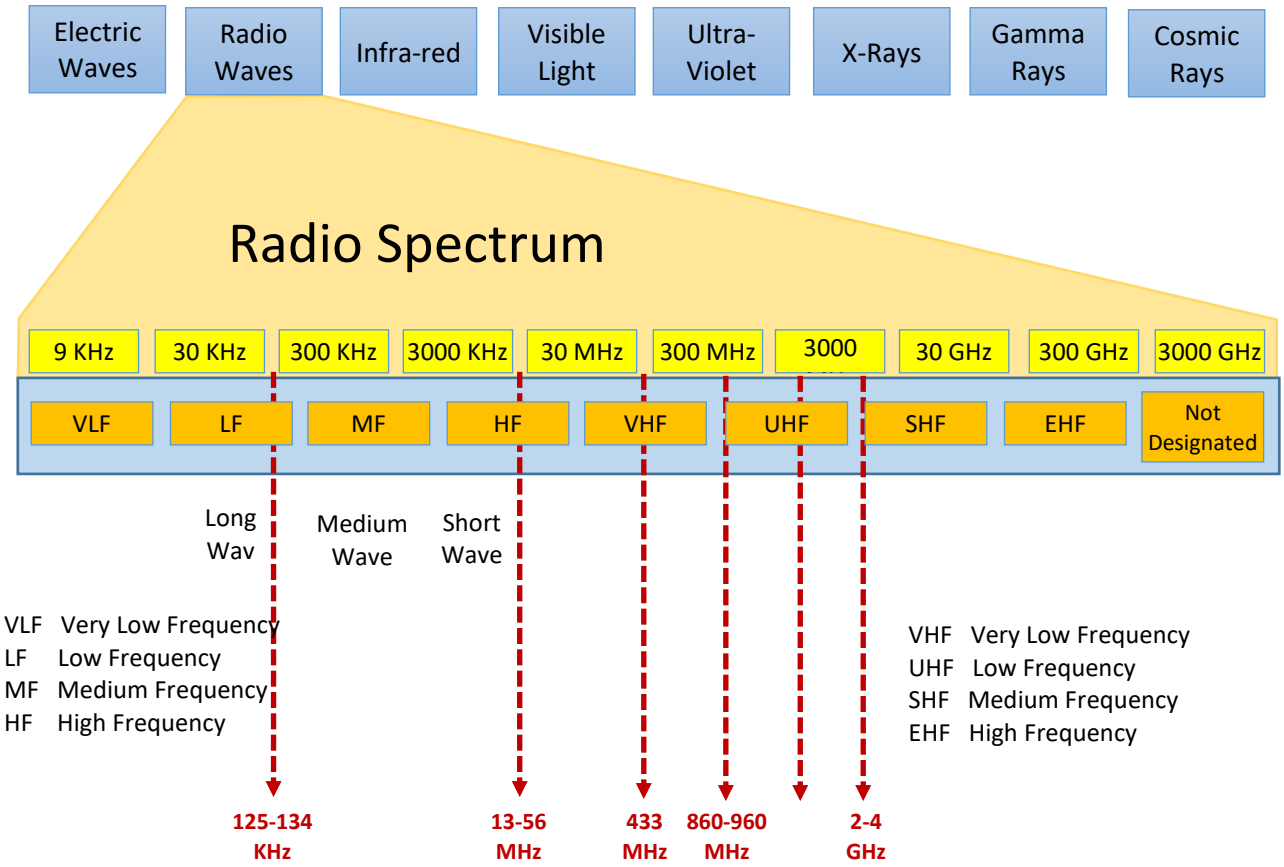



***Figure I.12:*** The RFID frequencies

## 6. Advantages and disadvantages of RFID systems

The use of RFID solutions have been recognize by many industry. However, the appropriate level of RFID components combination and selection of these components according the suitability of organisational situation and environment can make it beneficial. Otherwise, RFID system with the inappropriate combination and selection of RFID components may generates error or does not work effectively which could be increase organisational operational cost and may affect customers' good will. The list of advantages and disadvantages can be seen in table I.2:

**Table *I.2:*** Advantages and disadvantages of RFID system

| Advantage | Disadvantage |
|---|---|
| High speed | Interference |
| Multipurpose and many format | High cost |
| Reduce man-power | Some materials may create signal problem |
| High accuracy | Overloaded reading (fail to read) |
| Complex duplication | |
| Multiple reading (tags) | |

## 7. RFID Applications

Before understanding the use of RFID system for contextual knowledge management, it is appropriate to study the use of RFID applications for other scenario or situations with respect to short and long range RFID applications [7]. For this purpose a generalised RFID application model has been presented as shown in figure I.13.

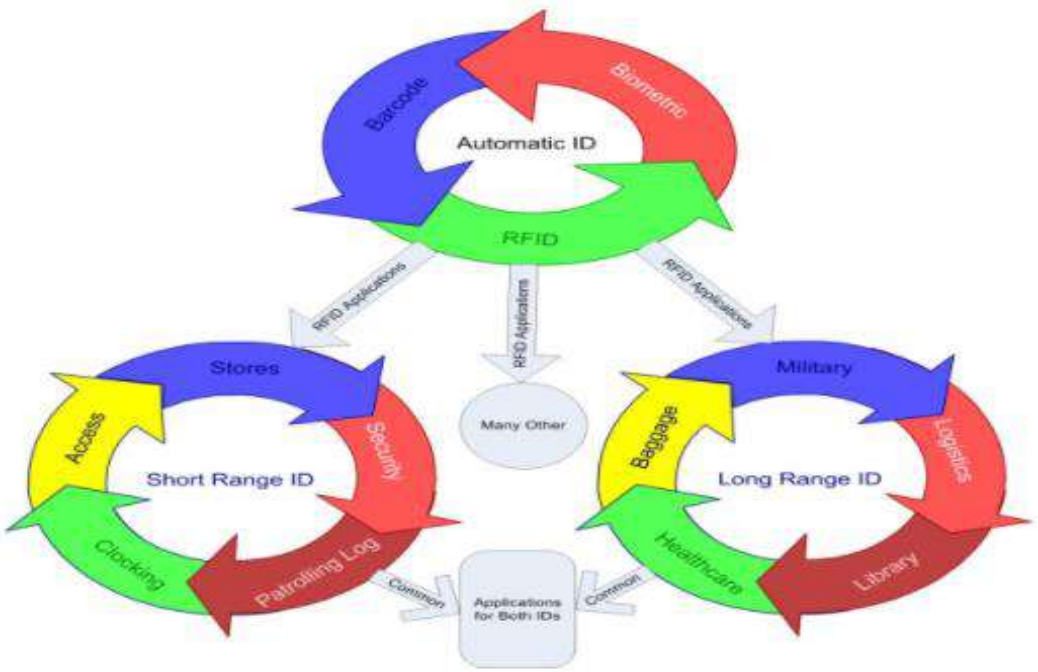



***Figure I.13:*** General RFID applications according to its capabilities.

From a RFID application perspective, RFID has two categories: short range and long range. In short range application tag need to show to reader. Application works perfectly in short range if object or tag access the reader one by one. For example, in access control, only employee or specified person can access the secure building and building has several divisions each division has particular set of authorized people, who can access certain divisions. Person need to show the tag near the reader before every secure door for enabling the system to decide whether door should open for that person or not. In long range applications, tags do not need reader near, as compare to short range. For example, tags place on every book in the library and user can easily access the exact book shelf for the required book. This enable automated inventory control. Readers can read the multiple items simultaneously from the distance. Reading distance can vary depending on the frequency and type of reader. This section further discusses the most common use within short and long range applications [7].

### 7.1 Security and Control Applications

RFID system can be used for control access and security; it is also useful for audit purposes. These applications are not only granted permission to access a particular secure zone but also record who is accessing, from which location/areas, at what time and for how long duration. These types of RFID systems can maintain building and departmental security including privacy [16]. This type of RFID solution is also workable for equipment and objects controls.

### 7.2 Patrolling log Applications

In this application, security firms use RFID system to control their security guards and use RFID data for various purposes [7]. These purposes include performance checks, data can be used in reference to unexpected event and audits for a secure area etc. The main difference in this type of application with other applications is that the reader is variable and the tag is fixed, i.e. the reader go near to the tag rather tag swap near to the reader. In security patrolling several numbers of tags can be fixed throughout the building and the security guard needs to swap the reader with each tag (checkpoint) in a sequential order within an allocated time and repeat this process throughout his/her shift. The reader records each swap which can be transferable to a computer program later on for audit or other purposes.

### 7.3 Baggage Applications

Baggage handling and package delivery is a complex task and needs a large number of human involvement which is an expensive resource. Humans do various operations from receiving packages, sorting, assembling and distribution [7]. Due to human involvement the error rate can be high. While the use of RFID tagging system not only reduces human involvement but also automates the process for certain extend which enable fast packages delivery. RFID solutions for baggage and packaging firm including airline industry are better for their effective operation and it reduce the complexity for overall system.

### 7.4 Toll Road Applications

RFID can provide the automated toll collection and maintain the traffic flow without stopping vehicles for payment. In these type applications, vehicles either pre-pay their toll yearly quarterly or monthly or other kind of scheme can be applicable such as pay-as-you go. In any case, the reader can recognize and record the vehicle entry at each toll which can be calculated later on by an application program. These applications not only help in toll collection and maintain traffic flow but also provide statistical data for any road which can be utilized for analysis and improvements:

- An antenna sends out and receives radio signals.
- These signals are received and returned by an RFID tag with information added.
- A reader which is integrated with a system accepts and stores these data called events and finally, they trigger actions.

The main purposes of RFID nowadays are identification and location.

### 7.5 Payments in Means of Transport

In order to eliminate the delays on public transport and tolls, electronic tolls collection (ETC) collects tolls electronically. Cars don’t need to stop when they pass through tolls, the payment is automatic if the car is enrolled in the program and if it’s not, the system sends an event which triggers an action like don’t let the car pass or it takes a photo [16].

If you don’t drive, when you are travelling around the city you also use RFID in public transport, because of the entrance payment and the exit control can be done thanks to RFID tags in the prepaid transport passes.

### 7.6 Asset Management

Nowadays, the most modern and effective organizations are using RFID technology to automate assets tracking location, identification and transport trace. Systems based on RFID solutions avoid many problems caused by manual tracking. With RFID they increase security and accuracy of the information which company has about the goods in a constant and instantly way [7].

As an example, when you're in a hotel, the tools, machines, and other items and disposable consumables may be controlled by RFID tags.

### 7.7 Logistics

Tagging goods and pallets, you can instantly know what goods you have, how many they are and how much space they occupy in your warehouse or back store just when they are entered.

Can you imagine what would happen if huge food companies didn't have constant and instant control of the products they have in stock? When you shop at the supermarket you are not aware of how much the product has travelled, but thanks to RFID technology that provides a trace of its movements [7].

### 7.8 Inventories and Warehouses

The main objective to use RFID is to increase the efficiency of warehouses by reducing work and logistic costs [7]. Likewise, to get instantly an accurate inventory of goods with all kind of details, like size, quality, country, and so on. Physical inventory counts which are expensive and inaccurate are not necessary anymore. This way you save money and time.

### 7.9 Identification of Animals

Tagging animals with RFID is an important tool for a farmer, in order to identify each animal with its information like origin, pedigree, medical details, etc.[16]. As well as with the help of software, to keep updated the information by uploading the new information like veterinary visits.

### 7.10 Surgeries

RFID technology in hospitals is present in many forms, from tracking surgical tools to tracking persons –patients, visitors and staff. There are several important reasons to use RFID technology in the healthcare industry from reducing medical errors like to lost essential

surgery tools or forget surgery sponges within patients, as well as reduce economic costs or increase the security at healthcare buildings [17].

The most common RFID applications in hospitals are inventory tracking, control access, staff and patients tracking, tracking tools, tracking disposable consumables, tracking large/expensive equipment, laundry tracking, etc.

### 7.11 Access Control

The RFID access control systems work to identify whose, where and when it's entering a building or a room. It's useful to have information about individuals in an organization or event like a sports match or music festival, as well as to allow or deny the entry of individuals in particular places [16].

The same function can be fulfilled in a company to have control over the schedules of employees, to what places they access inside the company and the control of people who are visiting the company offices.

### 7.12 Passports

Passports with an embedded chip are called biometric passport, e-passport or digital-passport. This chip has biometric information that is used to authenticate the identity of the passport holder. The information stored in the RFID chips of e-passports depends on the country's policy [7]. Data usually stored are name, date and place of birth, sex, nationality and a digital version of the photograph. Data about the passport are in the chip too, such as the number, issue date and place, and the expiration date. The standardized biometrics used for identification system is facial recognition, fingerprint recognition or iris recognition.

### 7.13 The supermarket of the Future

Imagine entering a store and just grab the neither products you want to buy and finally leave the store without whether lines nor checkout. The store stock would be updated instantly while you are taking the goods and you could check your total bill on your mobile at the same moment having better control of your expenses [16]. RFID is changing the grocery retailing.

### 7.14 Libraries and Museums

RFID is being applied in the museums, libraries and other related settings mostly in three ways:

- Objects tracking: managing full inventories of collections is a huge and time-consuming work.
- Security systems: protecting assets from theft an essential issue in museums, galleries. RFID provides security by automatically tracking the movement of objects tagged which can be located instantly.
- Visitor experience: there is too much information to put in a single label on the wall. The visitor's experience gets complete and amusing if they are directed to a virtual resource linked with a tag.

### 7.15 Timing Of Sports Events

In many sports events like races the participants must be timed accurately, the best way to do this is tagging them and tracking when they pass across control points [7].

## 8. Challenges of RFID Technology

Although promising, RFID is not without its challenges, which arise from both a technological and usage point of view.

### 8.1 Collision Problems

Communication between tags and readers are inherently susceptible to electromagnetic interference. Simultaneous transmissions in RFID lead to collisions as readers and tags typically operate on a same wireless channel. Therefore, efficient anti-collision protocols for identifying multi-tags simultaneously are of great importance for the development of large-scale RFID applications [17]. Many anti-collision protocols for RFID tag identification have been proposed, such as query tree protocol (QT), binary tree protocol (BT), frame slotted ALOHA protocol (FSA), etc., but almost all known protocols exhibit an overall identification efficiency small than 50%. Besides, uniform IDs distribution has always been assumed in the past. Furthermore, it is very useful for pointing out the best performing features of RFID tag identification protocols, and for designing new and better protocols. In [17], we present a novel and efficient anti-collision protocol for RFID tag identification, i.e., collision tree protocol (CT), which outperforms all the other anti-collision protocols proposed so far.

### 8.2 Security and Privacy Concerns

Security and privacy issues [7] of RFID tags can affect both organizations and individuals. Unprotected tags may be vulnerable to eavesdropping, traffic analysis, spoofing or denial of service and many more. Even unauthorized readers can affect the privacy by accessing tags

without enough access control. Even if the tag content is secure then also it can be tracked by the predictable tag responses; “location privacy” can be affected by a traffic analysis attack.

Attacker can also threaten the security of systems, which depends on RFID technology through the denial of service attack. Due to its cost and resource constraint limitations, RFID system does not have a sufficient security and privacy support. Many researcher and scientist work to implement low cost security and privacy protocol to increase the applicability. Lots of lightweight solutions have been proposed for RFID, but they are still expensive and vulnerable to the security and do not fully resolve the security issues. So there is a good research scope in the field of designing an efficient ultra-lightweight cryptographic protocol for low-cost RFID system [18].

### 8.3 Other Challenges

Three other issues also are mainly holding back RFID’s widespread adoption. The first one is the cost. The RFID tags are still more expensive than printed labels. The second issue is design. It is still needed to engineer tags and readers so that they guarantee highly reliable identification. Another challenge to RFID is its integration into existing systems. It is significant to develop effective RFID middleware which used to link new RFID systems into existing back-end infrastructures. Despite these challenges, it’s only a matter of time before these issues could be solved. RFID’s potential benefits are large, and many novel applications will be see in the future, even some of which cannot begin to imagine.

## 9. Conclusion

RFID is a technology with the potential to improve the way we live our lives and the way we conduct business .however, for this potential to be realized the challenges listed above, particularly those relating to security and privacy will have to be thoroughly addressed. A lot of research on RFID tags is ongoing including on embedding these with other devices, especially mobile devices. RFID manufacturers and users are looking for proper standardization and regulation of RFID. As prices fall further and technological improvements continue to occur, RFID technology is expected to become economically and technically more viable impact our daily lives as more application are developed.

# *Chapter II*

## *Anti-Collision Protocols*

## 1. Introduction

IoT is a set of millions of physical devices around the world that are connected to the Internet, collecting and sending data. It enables objects and machines to connect to the Internet and share collected information. Thanks to very cheap procedures and wireless networks, it is possible to turn anything, from a chip to a big building, into part of the IoT. This is an additional level of digital intelligence of devices that would otherwise be dumb, permitting communication without humans, and merging the digital and physical worlds. The term IoT was first coined by British entrepreneur Kevin Ashton in 1999, while working at auto-ID Labs, referring specifically to a global network of objects connected by RFID [19]. Nowadays, the IoT has been the focus of numerous research studies. Moreover, although IoT extends beyond the RFID, this technology is one of the key cores to implement the IoT.

RFID technology uses radio frequency to identify tags, which are small stickers attached to objects that need to be identified [20, 21]. The RFID system operates with a reader sending and receiving information from numerous tags in the interrogation range of its antenna at the same time. That is, the reader broadcasts messages using electromagnetic waves to tags in the interrogation area, while receiving responses from the tags through backscattering. Therefore, a multi-access arbitration procedure is necessary in order to prevent the response from being garbled [20, 22].

When more than one tag transmits simultaneously to a reader, their backscattered signals cancel each other out, resulting in an illegible message by the reader. This is called collision and it causes a loss of identification time and an increase in the power consumed by the reader.

## 2. The Collision Problem

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

1) Reader collision: occurs when the reader attempts to make communication with tags that are in the coverage area of another reader [25]. This type of collision is shown in Figure II.1. 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.

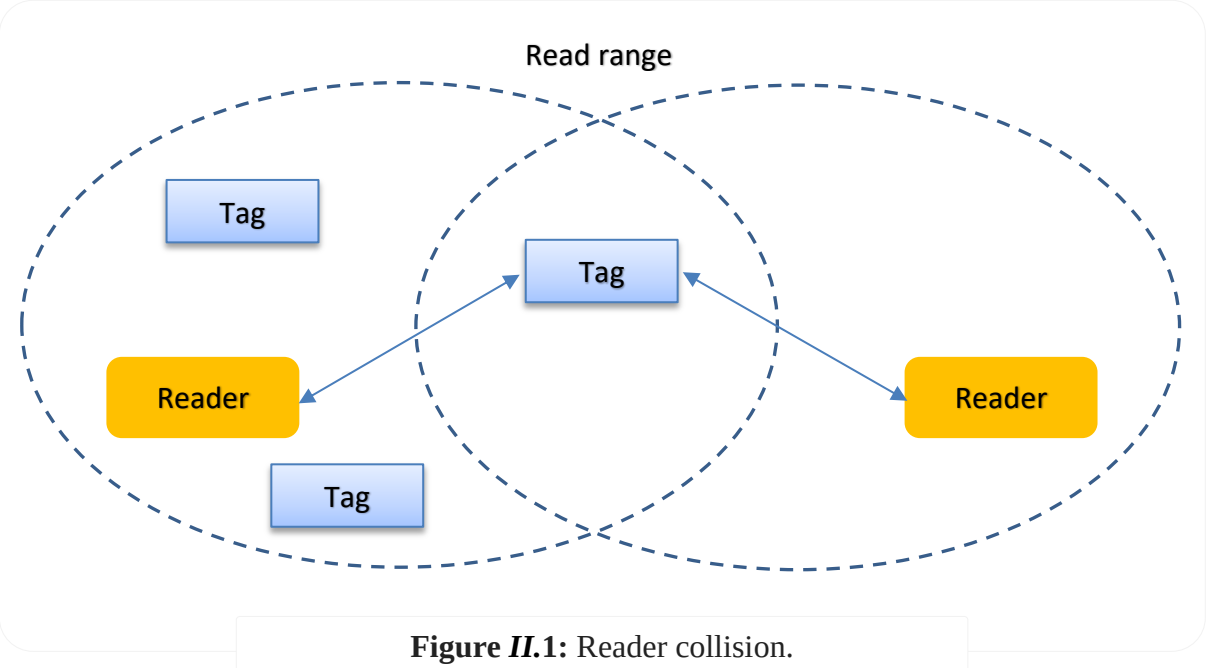



**Figure *II*.1:** Reader collision.

2) 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 tag's signals and cannot understand it. This type of collision is shown in Figure II.2.

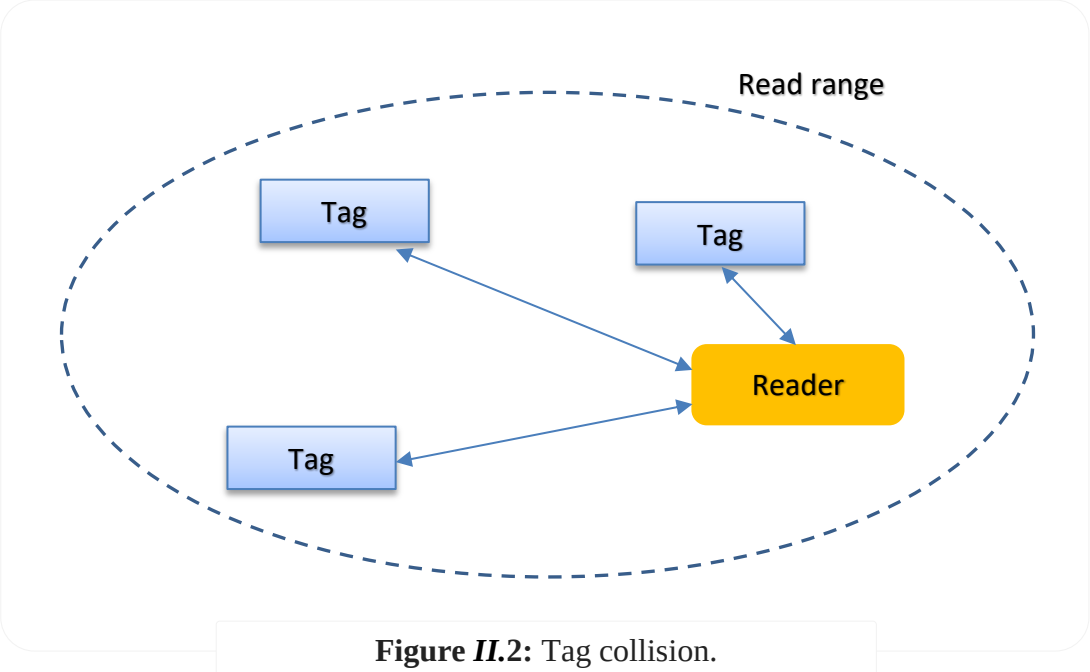



**Figure *II*.2:** Tag collision.

Simultaneous responses from numerous tags prevent the reader from correctly translating the signal, which decreases throughput. No tag is aware of the activity of any other tag, and so they cannot prevent the simultaneous transmission of tags. The transmissions of three tags, shown in Figure II.1, are not synchronized, but in many cases, the reader is synchronized with at least one tag in the interrogation zone. In the presented illustration, the reader is prevented from decoding the entire transmission and it has experienced a collision. To solve this problem, anti-collision protocols [23, 24] are very influential.

## 3. Multi-Access Methods

Each anti-collision protocol uses certain multi-access methods for identification in order to physically separate 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) [20,23,26]. Figure II.3 shows various multiple access and anti-collision procedures.

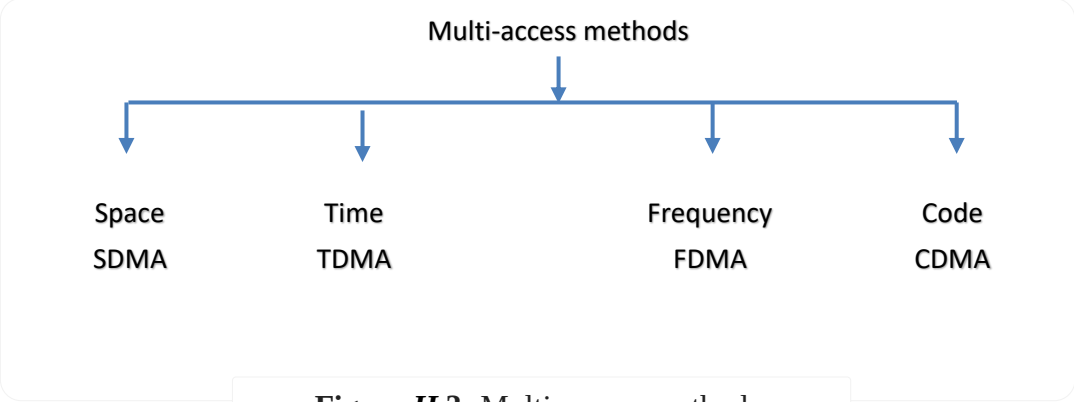



**Figure *II*.3:** Multi-access methods.

### 3.1 SDMA

The term space division multiple accesses relate to dividing of the channel capacity into separate 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 means of achieving this is through the use of 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 quite expensive and requires complex antenna design. The use of this type of method is restricted to a few specialized applications [27, 28, 29]. This technique is shown in Figure II.4.

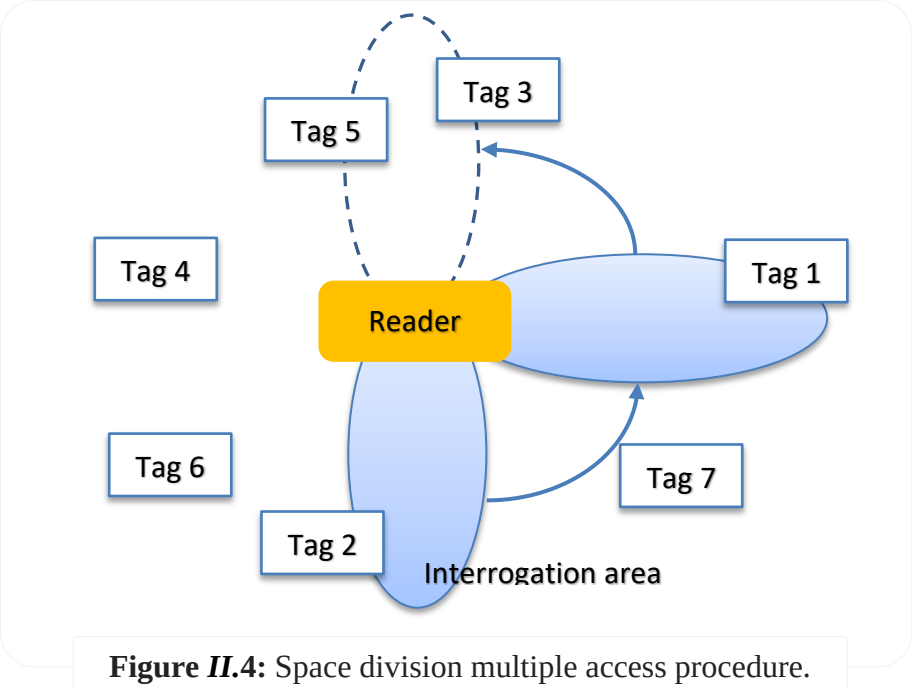



**Figure *II*.4:** Space division multiple access procedure.

### 3.2 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 433–435 MHz range. However, this technique is expensive and is only intended for certain specific applications [30]. Figure II.5 shows FDMA procedure.

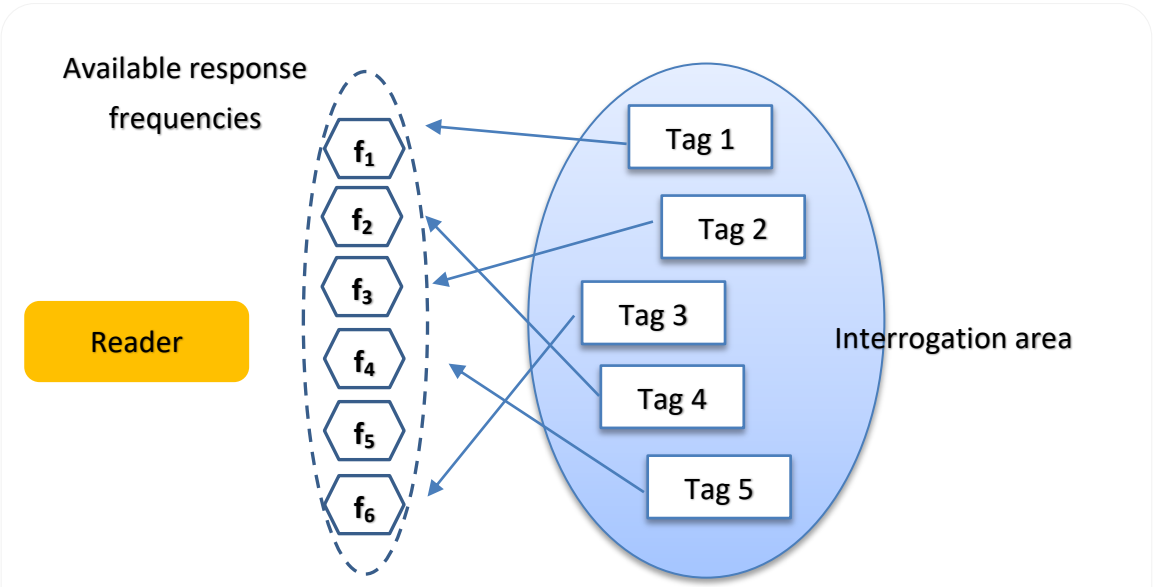



**Figure *II*.5:** frequency division multiple access procedure.

### 3.3 CDMA

Requires tags to multiply their ID by a pseudo-random sequence (PN) before transmission. CDMA is quite good in many ways, such as the security of the communications between the RFID tags and the reader, and multiple tag identification. It adds great complexity and is expensive for RFID tags. Furthermore, this method consumes a great deal of power and can be classified as a group with elevated demands [31]. Figure II.6 shows this procedure.

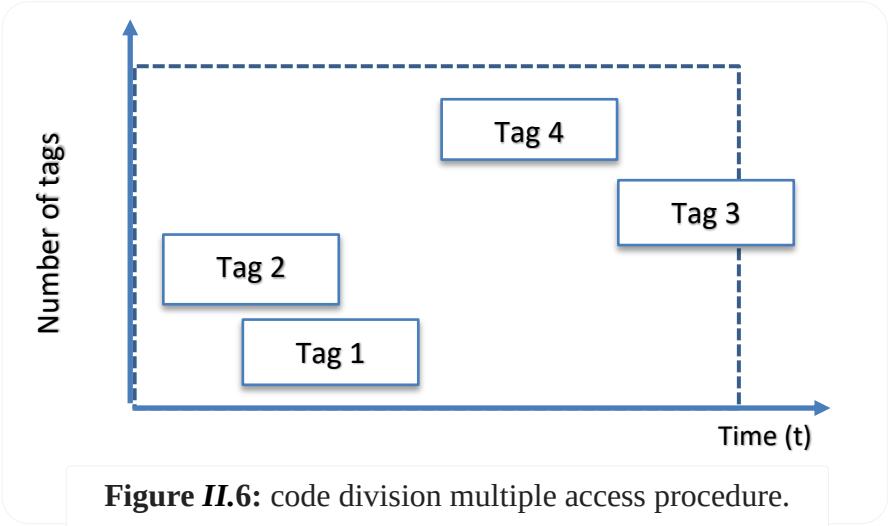



**Figure *II*.6:** code division multiple access procedure.

### 3.4 TDMA

Given that it is less expensive, this is the most widely used method. 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 for the transmitting tag. After that, the tags are activated one by one [32, 33]. TDMA is shown in Figure II.7.

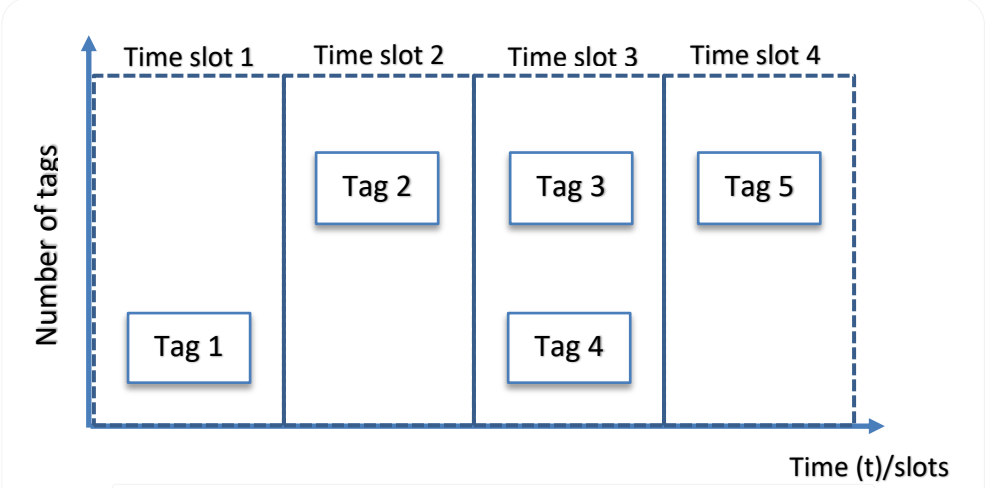



**Figure *II*.7:** time division multiple access procedure.

## 4. Aloha Protocols

Aloha-based protocols use a random-access strategy in order to successfully identify the number of tags in an interrogation area [34, 35, 36, 37, 38]. 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, there is no guarantee that all of the tags will be identified in the interrogation process. 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 [34].

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.

### 4.1 Pure Aloha

Pure Aloha (PA) is one of the simplest anti-collision protocols. It is based on TDMA [39, 40]. 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, resulting in complete or incomplete collisions. A complete collision occurs when the messages of two tags fully collide; an incomplete collision, however, takes place when only part of the tag message collides with another tag message. This procedure is shown in Figure II.8 and will be repeated until all tags are successfully identified.

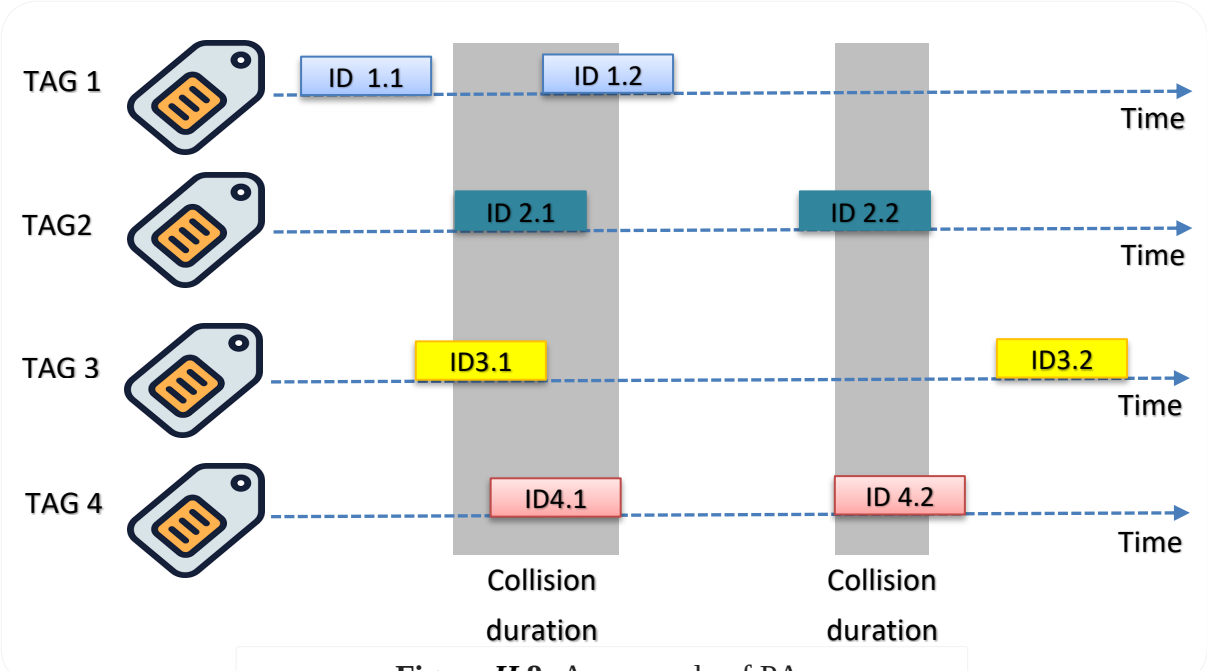



**Figure *II*.8:** An example of PA.

PA has been presented using different extra-features [26,41] such as: muting for silencing tags after being identified; the 'Slow down' for decreasing a tag response rate after identification; the 'Fast Mode' for sending a silence message before a tag begins transmission; and combinations of these different features.

### 4.2 Slotted Aloha

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 [26,39,41,42]. The communication between the reader and the tag is now synchronous. An example of communication with this protocol is presented in Figure II.9.

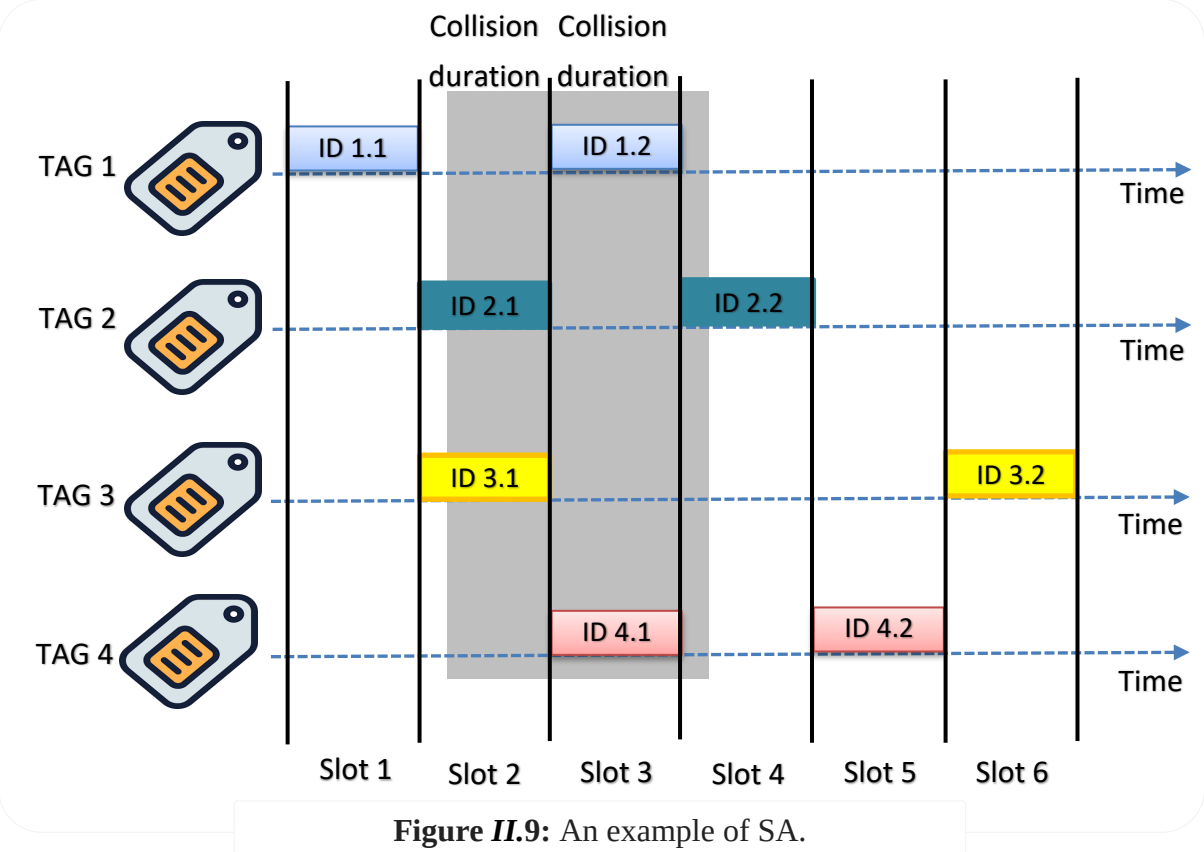



**Figure *II*.9:** An example of SA.

Also, SA can use features similar to those presented for PA. The muting of slow down features is used to silence or decrease the rate of the tags; the early end closes the slot earlier than normal; and also combinations of the types.

### 4.3 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 [34,43,44,45,46]. 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. This protocol

significantly reduces the probability of collision since tags can only respond once in a frame. Before the onset of communication, the reader generates a random time that is less than the fixed frame size, to select just one slot in a frame. If a collision occurs, the involved tags will once again choose a slot in which to respond in the next frame. The main inconvenience of

FSA is slot wastage when the number of tags is small and the size of the frame is significantly larger [47, 48]. An example of this protocol is shown in Figure II.10.

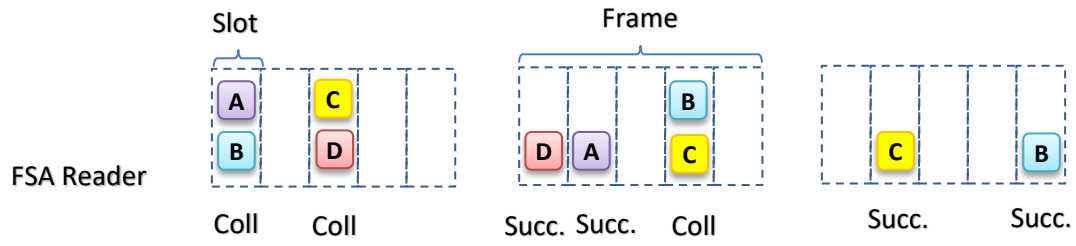



**Figure *II*.10:** Example of FSA.

### 4.4 Dynamic Framed Slotted Aloha

To ameliorate the disadvantage of (FSA), the dynamic frame slotted aloha (DFSA) protocol was developed [16, 31, 32]. DFSA is capable of changing 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. There are certain disadvantages to tag estimation, such as: increased computational costs in the identification process and errors that degrade the protocol's efficiency. An example of this protocol is shown in Figure II.11.

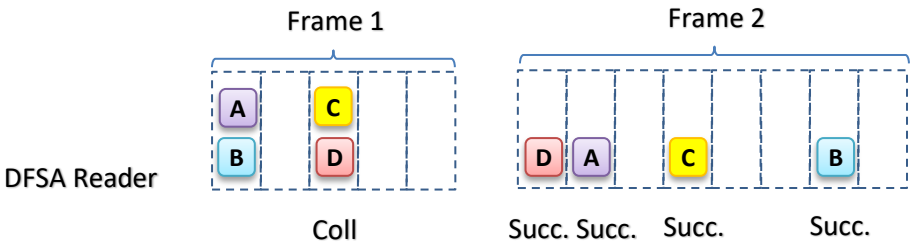



**Figure *II*. 11:** Example of DFSA.

### 4.5 Q Protocol

The Q protocol is used in the EPCglobal Generation-2 (Gen-2) standard [36,51]. The Q protocol is a DFSA-type protocol that modifies frame size using feedback from the last frame accomplished [52]. The Q algorithm can jump into the following frame without finishing the current one [52].

The Q algorithm operates with two basic parameters: Q, and a constant c that can be modified depending on the situation [42,43]. The Q. variable 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, 2Q−1]. This command initiates the identification process by providing a new value of Q.
- Query Adjust is the command used to instruct all tags to increase, decrease, or maintain the Q value unchanged and to 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.
- Query Rep is used in order to notify all tags to decrease their SN by 1.

The procedure of the Q protocol appears in the flow chart in Figure II.12. If it is time to initiate a new inventory round, the reader will transmit a query command. If the tags receive query or query adjust, they need to choose SN from [0, 2Q−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 will respond with an RN16 to the reader. There are three possibilities, depending on the tags' response:

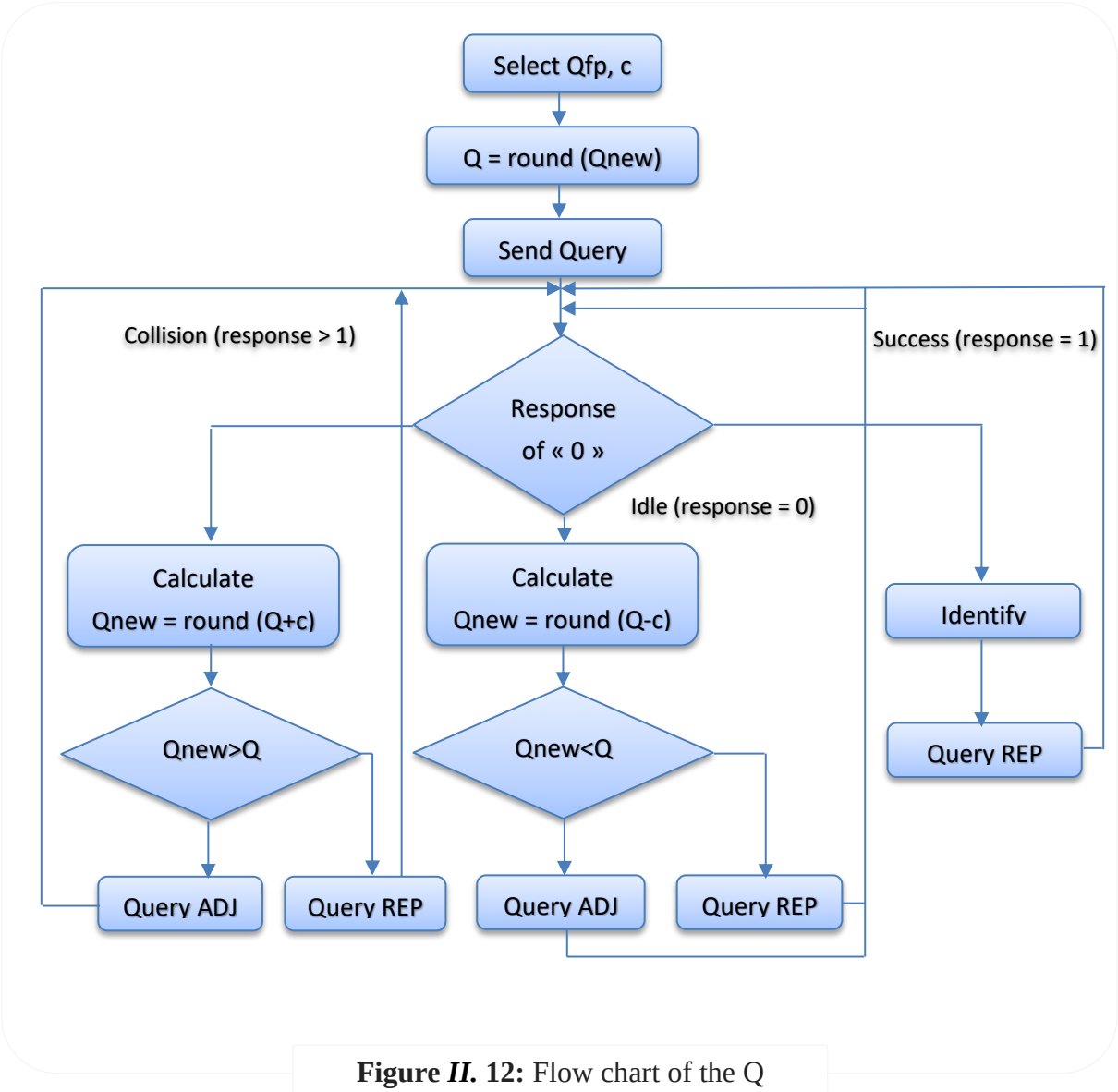



**Figure *II.* 12:** Flow chart of the Q

- Successful reply: If only one tag responds and the reader successfully received RN16. Subsequently, the reader sends the ACK and only the tag that successfully responded recognizes the ACK and reports its EPC to the reader.
- Collided reply: If 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 [46,53]. Table II.1 summarizes the observations regarding the Aloha

protocols, where the efficiency measures the exploitation of the tags' responses and the influence of a collision on a tag response. It is calculated using the expression nslt×100, where n is the number of tags, and slt denotes the total number of consumed slots.

**Table *II*.1:** A comparison of Aloha protocols.

| 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 number of collision increases significantly. | In a dense tag environment 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 |
| **Efficiency** | 18.4% | 36.8% | 36.8% | 42.6% | 36.8% |
| **System cost** | Very low | Low | Expensive | Expensive | Expensive |
| **Complexity** | Very simple | Simple | Medium | High | Medium |

## 5. Tree-Based Protocols

One of the main features of tree-based protocols is that they are deterministic, since ideally, they are designed to identify the whole set of tags in the interrogation area [54,55,56,57,58]. 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 the identified tags to remain quiet after their identification. These protocols usually work using queries, which are broadcast commands transmitted by a reader to require the tags to respond. If a tag's ID does not match the query, the reader command is rejected.

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).

### 5.1 Query Tree Protocol

The query tree protocol (QT) is one of the most representative memoryless protocols, in which the reader must provide the tags with a query and the matching tags must respond with their full ID [59]. Tag response depends directly on the current query, ignoring the prior communication history. 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 response 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 II.13 shows the QT protocol being used to read 6 tags (Tag A–Tag F). Each tag uses an ID length of k=6 bits. Initially, the LIFO stack is empty, and the reader begins with a 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 II.13, 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 only 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 no tags contain that prefix. In round 7 the reader transmits 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.

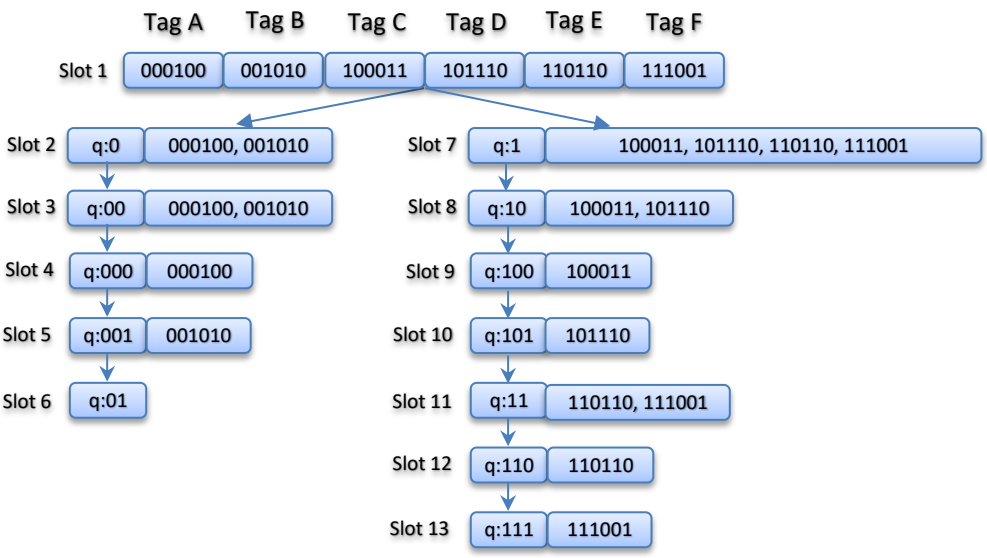



**Figure *II*.13:** Example of the QT protocol.

In the literature, numerous extensions to the QT protocol have also been proposed [56,60,61,62,63].

### 5.2 Smart Trend Traversal Protocol

The Smart Trend Traversal protocol (STT) is a deterministic and memoryless protocol that was created with the aim of reducing the number of collisions in the QT protocol [64].

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 that results in a 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 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=q1, q2, q3, ⋯, qn, where qn is the last query used in the identification process [65].

It is difficult to achieve, but 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 the QTP is at too high a certain level and should be moved 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 + ncol− 1. Let s denote the minimum increase, and ncol 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 + nemp− 1 and nemp is the number of consecutive idle slots.
- Upon a successful response, a single node is visited, indicating that the tag has been successfully identified by the reader. Then QTP moves to the symmetric node if the query finishes with a 0, but it returns one level if the query finishes with a 1.

The identification process of the STT protocol, which was explained above, is depicted in Figure II.14 with 4 tags.

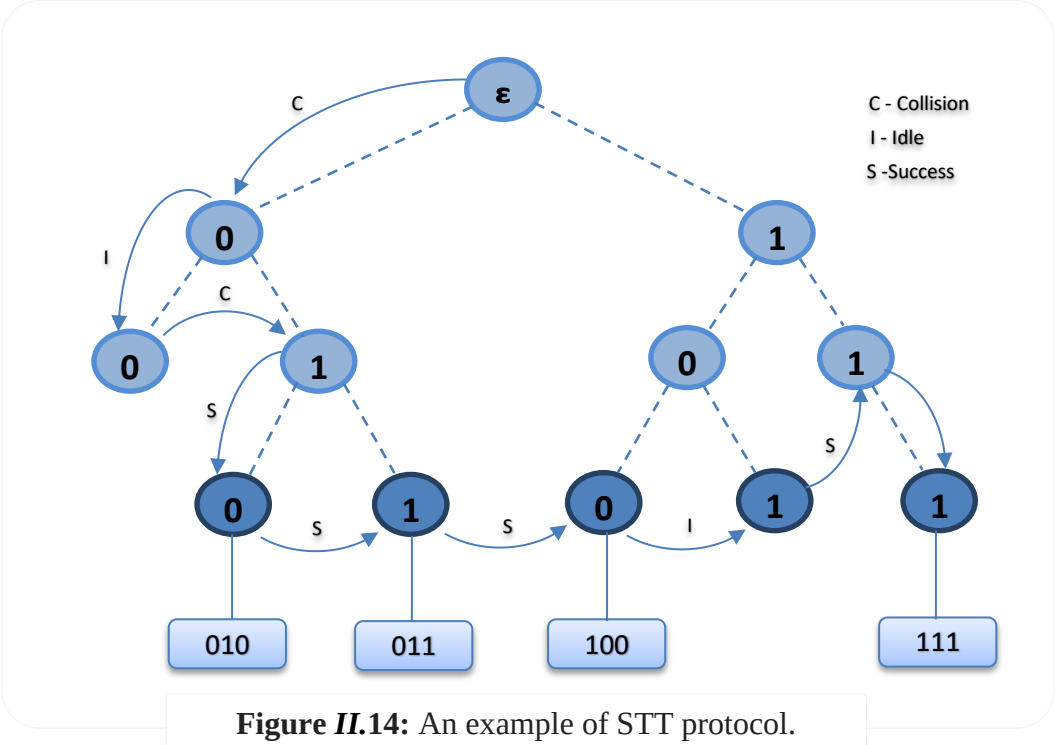



**Figure *II.*14:** An example of STT protocol.

In conclusion, STT significantly reduces the number of collisions, the identification time, and the energy consumption as compared to the existing Aloha-based and tree-based protocols.

### 5.3 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. Figure 15 shows an example of a communication slot between the reader and the tag. To reduce the number of bits transmitted by the tag, a window method has been proposed [66,67,68]. In the identification process, many slots ultimately collide, resulting in a huge waste of bits. Protocols using 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 on the reader side, respecting the condition 0<ws<k. It is shown in Figure II.16. Most tree-based protocols use a fixed tag response during the identification process, but some use different operational process methods with a dynamic response that is based on window synchronization.

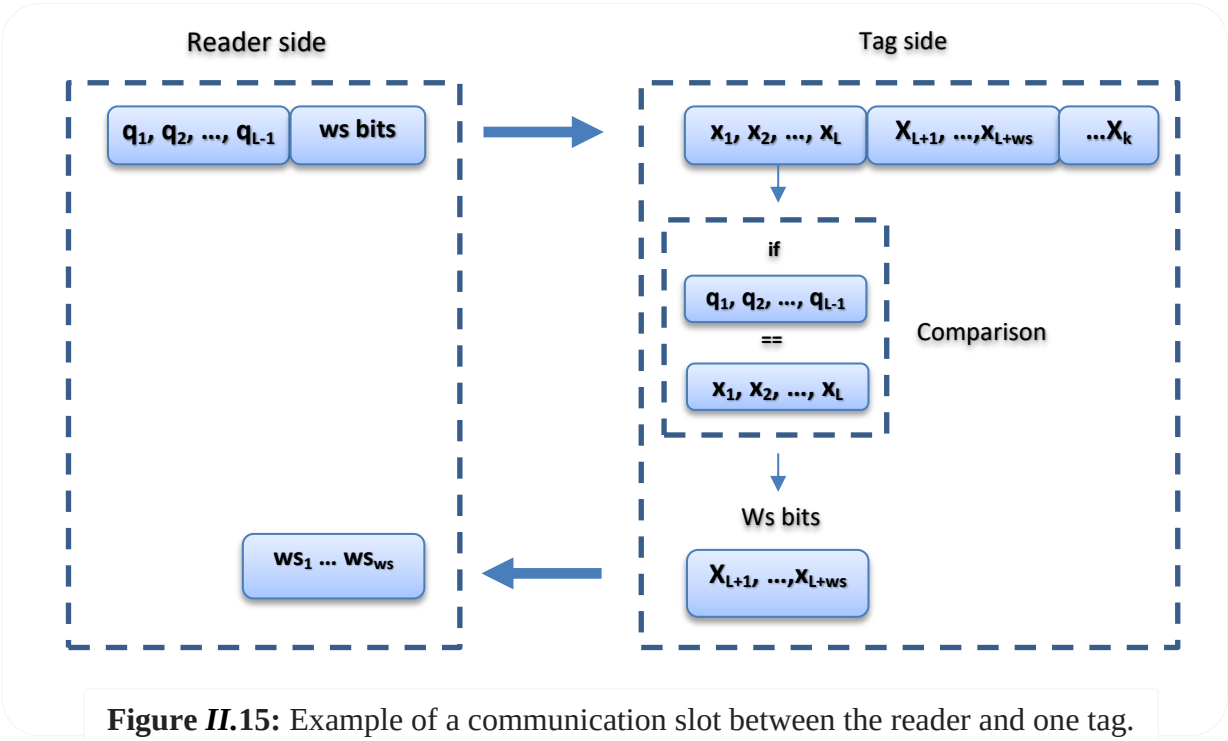



**Figure *II*.15:** Example of a communication slot between the reader and one tag.

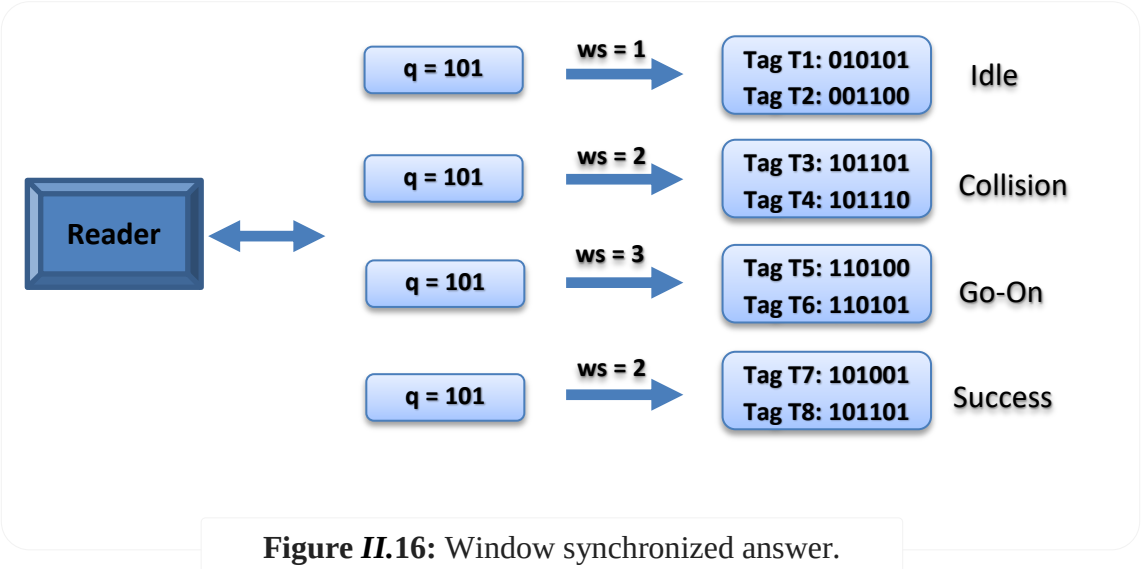



**Figure *II*.16:** Window synchronized answer.

### 5.4 Query Window Tree Protocol

The Query window Tree protocol (QwT) is a memoryless tree-based protocol that applies a dynamic bit window to QT [66,68]. Tags respond directly depending on the current query. QwT tags compare their ID value with the query received and transmit a certain number 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 for QwT are shown in Figure II.17 a, 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 number 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 only 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:

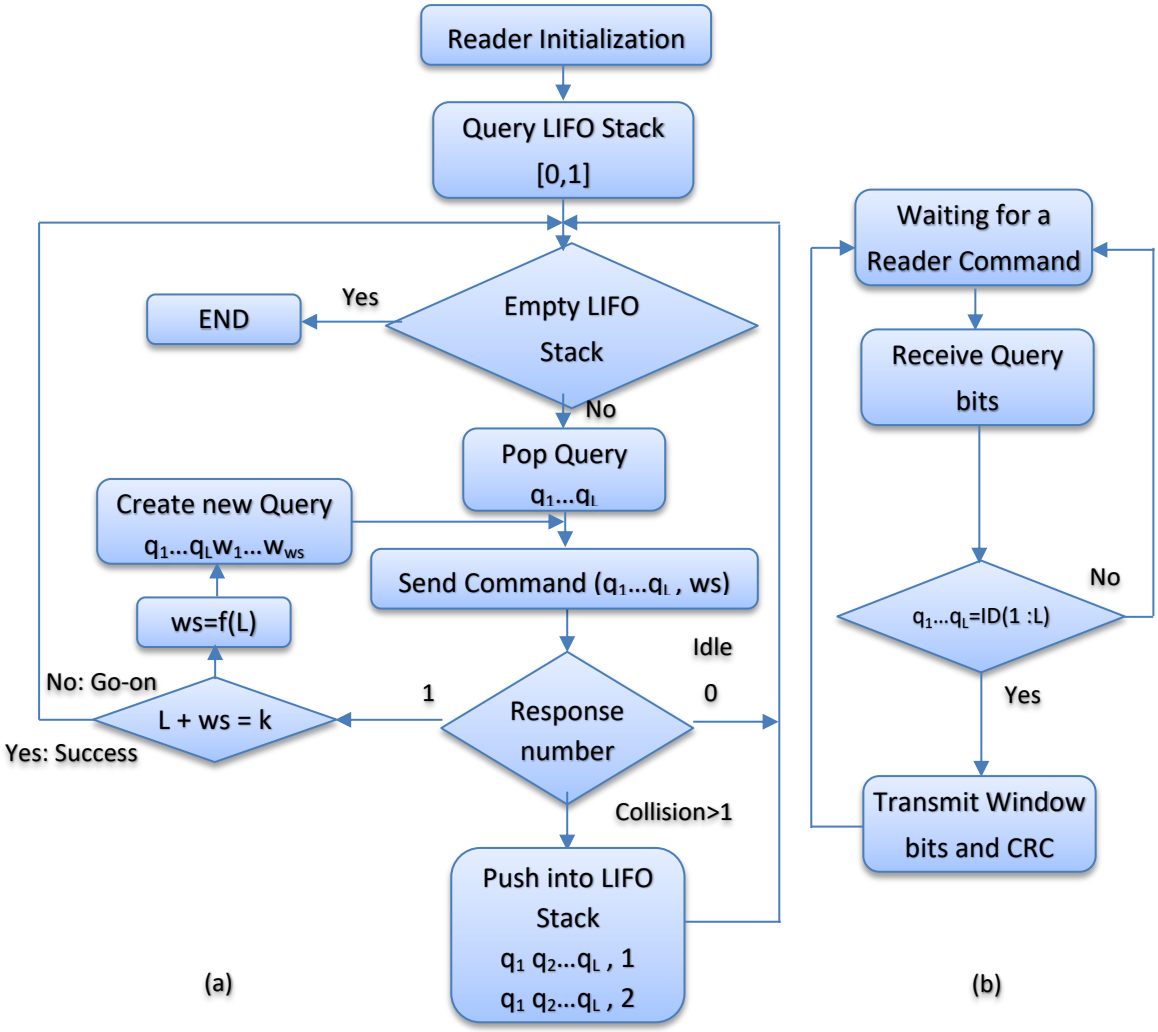



**Figure *II*.17:** Flow chart of QwT protocol: (**a**) for reader; (**b**) for tags.

- Collision slot. When the reader cannot differentiate the answer, they 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* as that of 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 (*II.*1), where β is an adjustable parameter. This heuristic function is used 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 Idle slot, ws will be held unchanged. The proposed protocol maintains 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 increases the number of slots and reader-transmitted bits. Altogether, this tree-based protocol achieves significant energy savings and a reduction in identification time.

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

- A modification of the QwT is presented in [69] called Standardized Query window Tree protocol (SQwT). This protocol aims to reduce the number of bits that define ws by using a standardized value of 3 bits and approximating it to the nearest power of 2, using s= $\log_2$ ws instead. Tags calculate the number of bits to respond by using ws=2s. By using only 3 bits, SQwT can cover window size values from 1 to 128 bits.
- Another modification of QwT is presented in [70] and is called flexible query window tree protocol (FQwT). This modification takes advantage of the window to perform an estimation of the tag ID distribution in the interrogation area and improve the identification time of the protocol. The functioning of the protocol is divided into two phases: the estimation of the distribution, and the identification process. During the former phase, the reader estimates the tag ID distribution until the first tag is identified (see Figure II.18).

After the first tag is identified, the reader begins the identification phase, similar to that of the SQwT. This phase however uses a different heuristic function when a go-on slot occurs, taking advantage of the cg parameter (see Equation (*II.*2) related to the type of tag ID distribution).

$$f(c_g, L) = \frac{c_g}{L} k\left(1 - e^{-\beta L}\right), \quad 0 < L \leq k \qquad (II.2)$$

Equation (*II.*2) is adjusted with a value of parameter β, preselected to decrease the energy consumed by the proposed protocol. Table II.2 compares some of the standard tree-based protocols.

**Table *II*.2:** A comparison of standard tree-based

| Protocol Name | QT | STT | QwT |
|---|---|---|---|
| **Protocol Feature** | Tags respond when their ID match the query transmitted by the reader. | Defines the generation of the new queries according the several predefined rules. These rules take into account the number of consecutive collisions and slots. | The reader transmits the number of bits tags must respond in addition to the query. This number is calculated using a heuristic function at the reader side. |
| **Disadvantages** | It produces a high number of collisions, particularly in the beginning of the identification procedure. | It transmits the full tag ID with each response; therefore a high number of bits are wasted with every collision. | The reader command needs a high number of bits to represent the size of the tags responses. |
| **Efficiency** | 34.6% | 34% | 35% |
| **System cost** | Very Low | expensive | Medium |
| **Complexity** | Low | High | Medium |

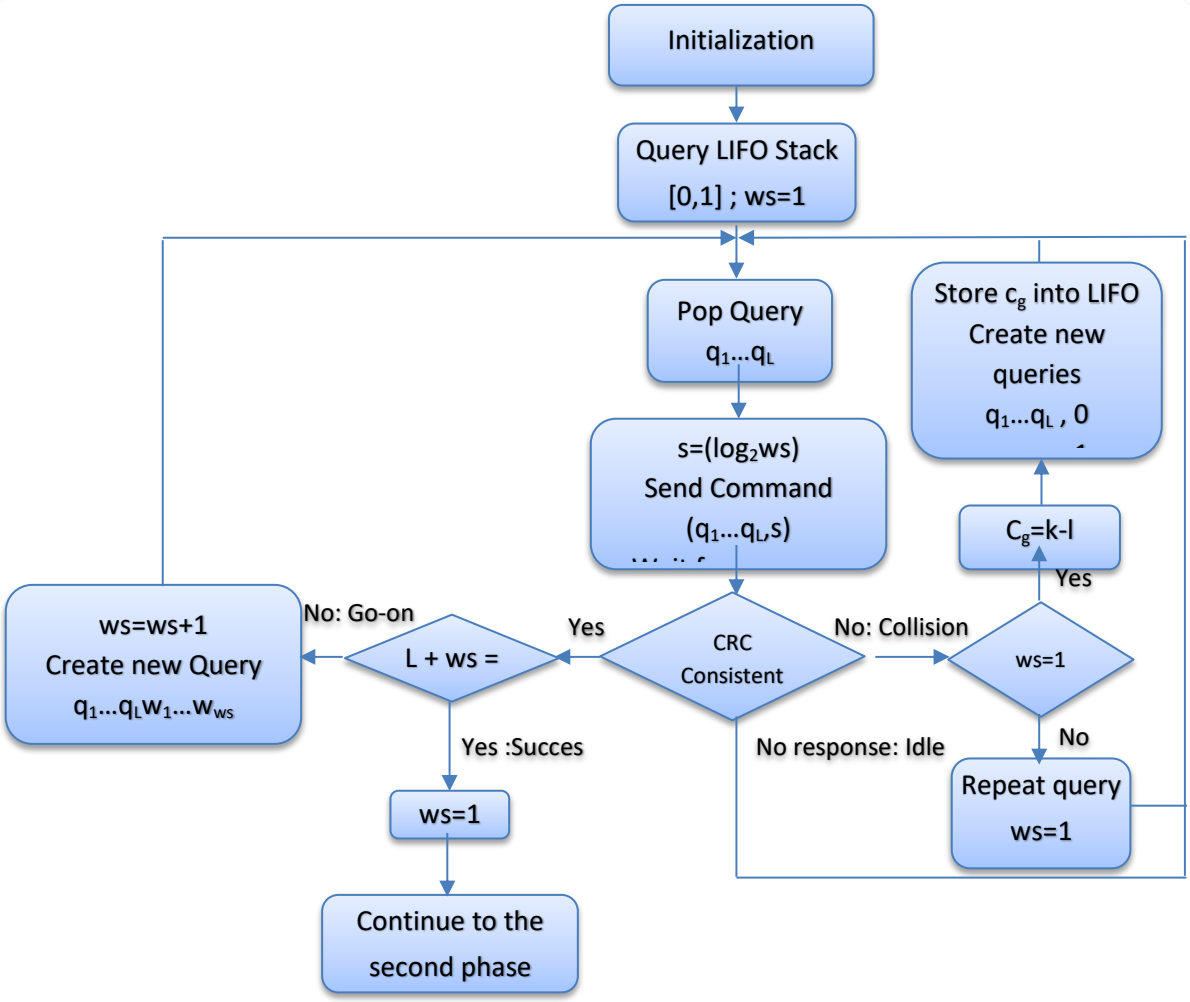



**Figure *II*.18:** Flow chart of the tag ID estimation phase of the proposed FQwT protocol : (**a**) for reader; (**b**) for tags.

### 5.5 Binary Search Protocol

The procedure in the Binary Search protocol (BS) algorithm [20,71] 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 II.3 shows an example of BS being used to read four tags (Tag A to Tag D). The reader begins by interrogating tags with the maximum ID value 111. Tags with a value of 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 successfully identified. After this slot, the reader restarts the query value with the initial value 111 and transmits it. This procedure is repeated until all of the tags are identified.

**Table *II.* 3:** Example of the BS protocol.

| Slot Number | Reader | Tag A (1010) | Tag B (1100) | Tag C (0100) | Tag D (0010) | Result | Type of Slot |
|---|---|---|---|---|---|---|---|
| Slot 1 | 111 | 010 | 011 | 100 | 110 | XXX | Collisio |
| Slot 2 | 011 | 010 | 011 | | | 01X | Collision |
| Slot 3 | 010 | 010 | | | | 010 | Success |
| Slot 4 | 111 | | 011 | 100 | 110 | XXX | Collisio |
| Slot 5 | 011 | | 011 | | | 011 | Success |
| Slot 6 | 111 | | | 100 | 110 | 1X0 | Collisio |
| Slot 7 | 101 | | | 100 | | 100 | Success |
| Slot 8 | 111 | | | | 110 | 110 | Success |

This protocol has two additional versions: Enhanced BS protocol (EBSA) and Dynamic BS protocol (DBSA) [72]. 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. To reduce bit consumption, in the initial slot, the reader transmits only '1' instead of 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 already been identified.

### 5.6 Collision Tree Protocol

The collision tree protocol (CT) is an improvement of QT which uses bit-tracking technology in order to find which bits have collided as well as where they are [73]. 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 based on voltage transitions.

The basic feature of this protocol is that it decreases collision slots and eliminates idle slots. This contributes to improved results in terms of 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 II.19 reveals 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 pop 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 stack '1' and a collision occurs. Through bit-tracking, the reader can find the colliding bits and thereby resolve potential 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 once again occurs. The reader can 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 may be noted that there are no idle slots and the number of collision slots and latency are reduced, which is the basic aim of the CT protocol.

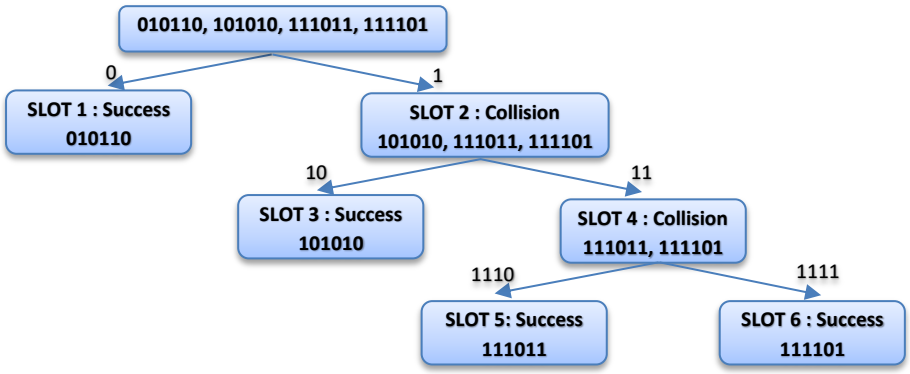



**Figure *II*.19:** Example of the CT.

In conclusion, 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.

### 5.7 Optimal Query Tracking Tree Protocol

The optimal query tracking tree protocol (OQTT) divides all of 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 [74]. This protocol uses three main approaches: bit estimation, an optimal partition, and a query tracking tree.

With bit estimation, the reader, using 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 non-selected bits (NNB) from the tags’ responses. The probabilities of bits being selected or of being non-selected are calculated from the expressions presented in (*II*.3). Finally, the optimal estimation of the number of tags is denoted by ñ and is calculated from (*II*.4).

$$Pr_{NB}(n,l) = \left(1 - \frac{1}{l}\right)^{n}$$
$$Pr_{SB}(n,l) = 1 - Pr_{NB}(n,l) \qquad (II.3)$$

$$\tilde{n}(l, N_{NB}, N_{SB}) = \underset{n}{\operatorname{argmin}} \left| \binom{l \times Pr_{NB}(n,l)}{l \times Pr_{SB}(n,l)} - \binom{N_{NB}}{N_{SB}} \right| \quad (II.4)$$

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 no more queries remain 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 c1 denotes the number of (l−1)-bit queries and c2 the number of l-bit queries. The number of bits in each query is computed from the equations 2l−1<m≤2l.

Table II.4 shows an example of OQTT. In the interrogation area there are 5 tags, whose IDs are as follows: '1010', '1100', '0100', '0010' and '0111'. At the beginning of the frame, the reader estimates the number of tags, which here is $\tilde{n}$=5. By using this estimate for the number of tags, the initial number of sets is calculated with the formula m=⌈ 0.595824×$\tilde{n}$⌉ , 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 Figure 12, 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.

**Table *II*.4:** Example of the OQTT protocol.

| 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 | Collisio | 010,011, |
| 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 |

However, OQTT may incorrectly estimate the tag number. The estimation error however is negligible, only producing an imperceptible difference in the final result of the number of queries. According to the literature, e.g., [74], this protocol provides an efficiency of approximately 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 pre-processing increases the energy consumption of the protocol, especially in dense tag environments [74].

- A modification of the OQTT is presented in [59] called optimal binary tracking tree (OBTT). This modification implements the estimator of the OQTT with a simple binary tree (BT) protocol [76]. The estimator establishes the initial upper bound for tags' counters. This produces a separation of the existing tags in groups avoiding excessive responses in the beginning of the interrogation procedure.

### 5.8 Collision Window Tree Protocol

The collision window tree protocol (CwT) is the second proposed window-based protocol that applies a dynamic window to CT [67, 68]. This protocol adopts two techniques: bit tracking and bit windowing. The bit tracking uses Manchester coding in order to identify 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.

The reader interrogates tags by transmitting a query $[q_1...q_L]$ of length $L$, attached to the ws of length $[\log_2 ws] + 1$ bit. The bit-string ws inform the tags of the number of bits that they must send in their reply. The variable ws is computed in every slot and is 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 take place 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 occurs 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 one and the received window from the last slot. The next ws is calculated using the heuristic function in Equation (*II*.4) .

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

A CwT flow chart is presented in Figure 20. 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.

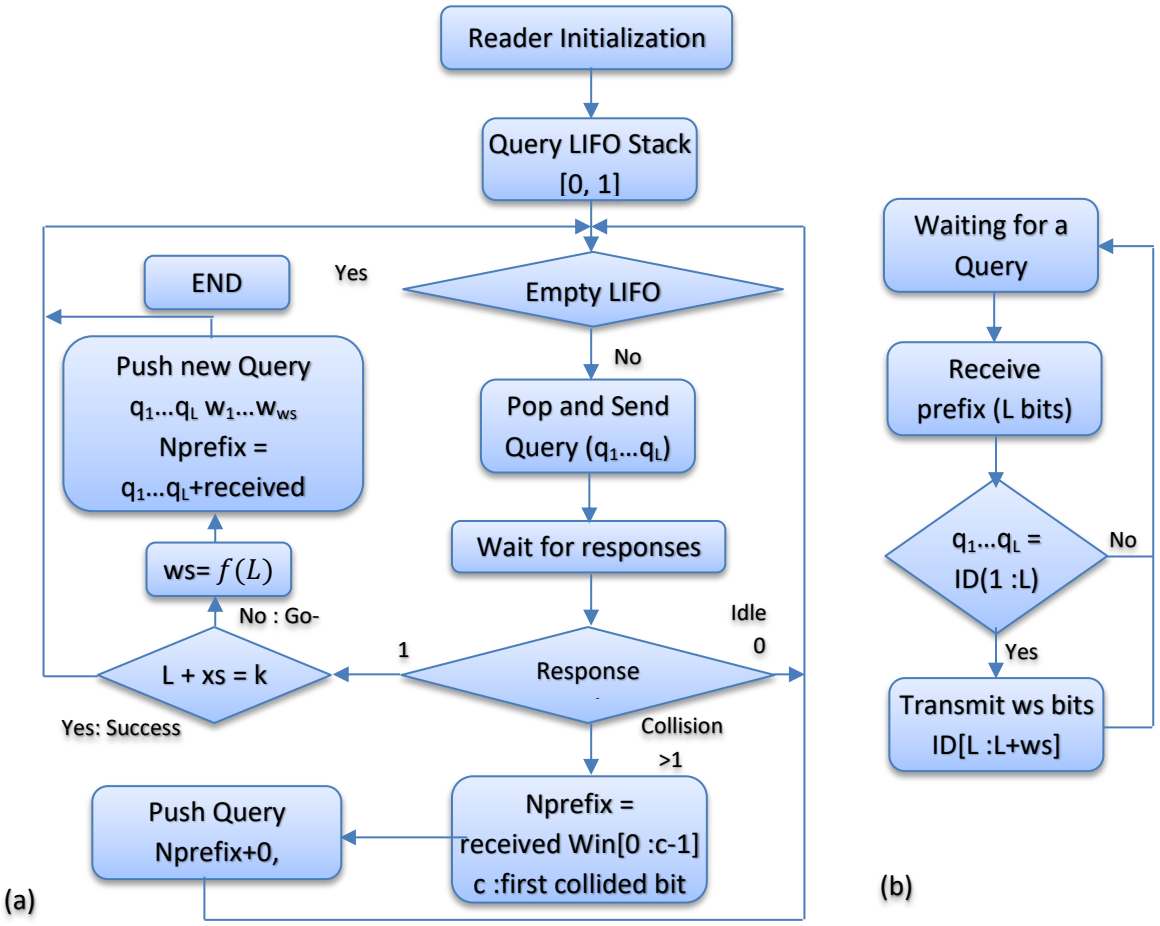



**Figure *II*.20:** Flow chart of CwT protocol:
(**a**) for reader; (**b**) for tags.

The CwT provides a significant decrease in the number of tag-transmitted bits, but this benefit comes with a certain decrease in the number of slots and the reader-transmitted bits. This protocol achieves important energy savings due to the reduction in the time required by the tag transmission process [68].

Table II.5 compares some of the tree-based protocols.

**Table *II.* 5:** A comparison of tree-based protocols using bit-tracking.

| Protocol Name | OQTT | BS | CT | CwT |
|---|---|---|---|---|
| **Protocol Feature** | The reader performs an initial estimation to split the tags in groups. These groups are identified using the CT protocol. | The reader transmits a serial number. Only tags having an 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 as well as where they are. | The reader is able to identify the collide bits from the tags responses, and uses them to calculate the new tag response size. |
| **Disadvantages** | Very complex protocol. Hard to be physically implemented. | The reader restarts the reading process after a tag is identified | It wastes a high number of tag bits on every collision, increasing the energy consumed by the reader during the process. | The reader command needs a high number of bits to represent the size of the tags responses. |
| **Efficiency** | 61.4% | 30% | 50% | 35% |
| **System cost** | Very expensive | Low | Medium | Medium |
| **Complexity** | Very High | Low | Medium | Medium |

## 5.9 Tree Splitting Protocols

In tree splitting (TS) protocols, the reader splits tags into multiple groups using a random number generator. We will describe, in the following, two variants of this protocol.
Basic tree splitting protocol resolves collision by splitting collided tags into b groups. These groups become increasingly smaller until they contain one tag. Each tag has a random bit generator r and a counter to save its position in the resulting tree. Tags with a counter value of zero are considered to be in the transmit state; otherwise tags are in the wait state. After each slot, the reader informs the tags in the transmit state to transmit their ID. If a collision occurs, the reader sends the ’FAIL’ command. All tags receiving a ’FAIL’ command with counter not equal to 0, shall increment counter by one. If tag’s counter is equal to 0, it generates a random bit and adds it to their counter. When counter equals 0 the tag enter the transmit state otherwise it enters the wait state. If an empty or single slot appears, the reader transmits the ’SUCCESS’ command and every tag in the wait state decrement their counter by one. After identification, tags enter the sleep state.

An example for the identification process for BTS protocol is summarized in Table II.6 for four tags named A, B, C and D. This table gives the counter and the random bit for each tag. In slot 1 tags A, B, D have selected binary zero (r = 0) and they enter the transmit state, while tag C enter the wait state, which causes a collision. Then, the reader transmit the 'FAIL' command. At slot 2, the tags A, B, D generate another r and add it to their counter whereas tag C increments its counter. Here, tags A and D have selected r = 1 and enter the wait state, only tag B has a counter value of zero, it is identified successfully. After the receiving a success slot, the reader informs tags in the wait state (Tag A, C, D) to decrement their counter by one by sending the command 'SUCCESS'. As a result, tags A and D have a counter value of zero, which indicates a collision in slot 3. They are not identified until slots 5 and 6 respectively.

Finally, after an empty slot, tag C is identified in slot 8. Overall, the reader uses eight time slots to identify all four tags. The Figure II.21 depicts the tree diagram for this example. The standard ISO/IEC 18000-6B adopts this basic tree splitting protocol [83].

**Table *II*.6:** Example of identification process for BTS protocol.

| | Slot 1 | | Slot 2 | | Slot 3 | | Slot 4 | |
|---|---|---|---|---|---|---|---|---|
| | r | counter | r | counter | r | counter | r | counter |
| Tag A | 0 | 0 | 1 | 1 | | 0 | 0 | 0 |
| Tag B | 0 | 0 | 0 | 0 | - | | - | |
| Tag C | 1 | 1 | | 2 | | 1 | | 2 |
| Tag D | 0 | 0 | 1 | 1 | | 0 | 0 | 0 |
| Reader | Collision | | Single (Tag B) | | Collision | | Collision | |
| | Slot 5 | | Slot 6 | | Slot 7 | | Slot 8 | |
| Tag A | 0 | 0 | - | | - | | - | |
| Tag B | - | | - | | - | | - | |
| Tag C | | 3 | | 2 | | 1 | | 0 |
| Tag D | 1 | 1 | | 0 | - | | - | |
| Reader | Single (Tag A) | | Single (Tag D) | | Empty | | Single (Tag C) | |

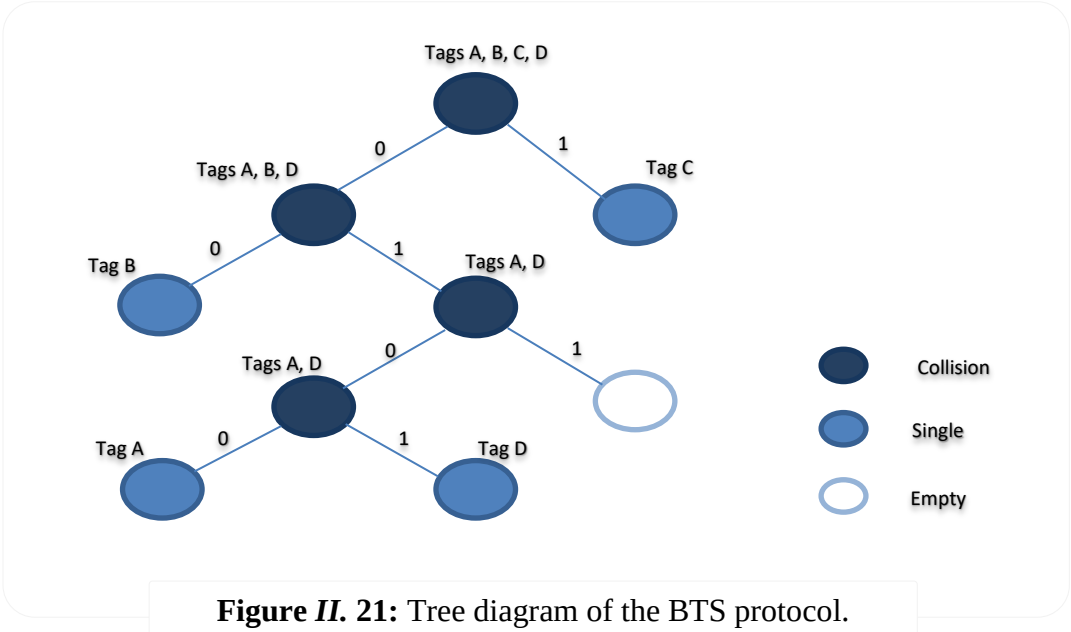



**Figure *II.* 21:** Tree diagram of the BTS protocol.

## 6. Hybrid Protocols

Hybrid protocols combine the advantages of tree-based and Aloha-based protocols to avoid their problems and provide better features in tag identification [77,78,79,80]. Most of them first implement a tree-based procedure and tag estimation procedure in order to predict the number of tags. Therefore, the combined Aloha-based and tree-based protocol procedures are known for their high complexity and hardware demands. This kind of protocol can significantly increase performance as compared to 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 organized in slots, as in FSA. In the BTSA protocol, tags randomly choose a slot after the reader query.

### 6.1 Tree Slotted Aloha

Tree Slotted Aloha (TSA) is a probabilistic protocol created to reduce the number of collisions occurring in FSA [78]. When more tags collide in a slot, FSA attempts to solve this problem in the next frame. However, 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 lo 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 li. Then, all tags in the interrogation area will generate a random number in the range [0, li] 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: $l_0$, Ni represents the number of transmitting tags in slot i, where i≤l0, Ni≥0, and ∑iNi≥n must hold. If Ni≥2, there is a collision in slot i.

At the end of each interrogation round, if the reader detects a collision, it begins a new frame from each slot where the collision was 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 their tree level counter by 1 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 example in Figure II.22.

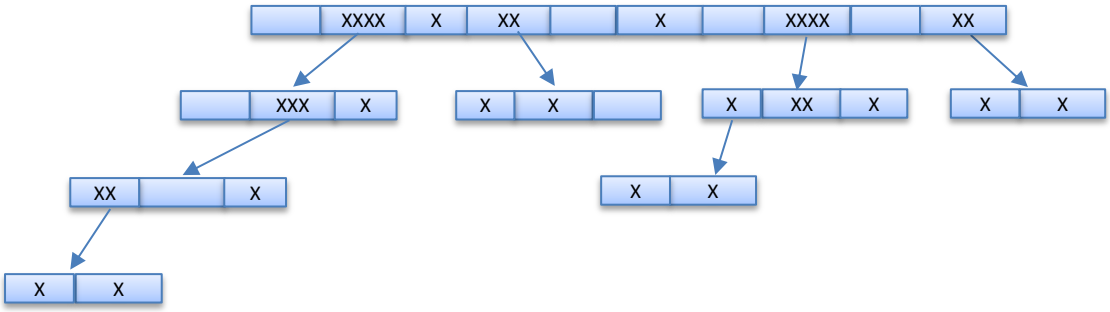



**Figure *II*. 22:** An example of tree slotted aloha.

TSA is a modified version of the FSA protocol, created to reduce the number of collisions. This protocol behaves better than FSA. TSA achieves an efficiency of between 37% and 41% [78].

### 6.2 Binary Tree Slotted Aloha

The reader in the binary tree slotted aloha (BTSA) uses a dynamic frame length adjustment and BTSA algorithm [81]. Each tag from the interrogation area randomly chooses a slot and transmits its ID. If the reader successfully identifies a tag 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 the BT. The initial frame length is L=2Q and the highest efficiency

is achieved when the initial frame size is close 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. Some protocols are presented in order to achieve higher efficiency in a wide range of number of tags. An example of BTSA is shown in Figure II.23.

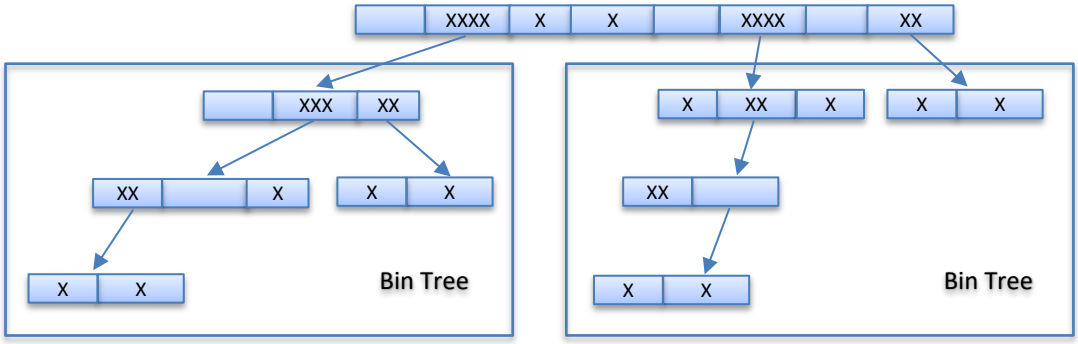



**Figure *II*.23:** An example of binary tree slotted aloha.

### 6.3 Dynamic Binary Tree Slotted Aloha

Dynamic Binary Tree Slotted Aloha (dynamic BTSA) involves a dynamic frame adjustment and the basic BTSA algorithm [81]. 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 II.24 shows the dynamic BTSA algorithm, where the initial frame length is L=2Q, 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 of 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 based on 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 [81].

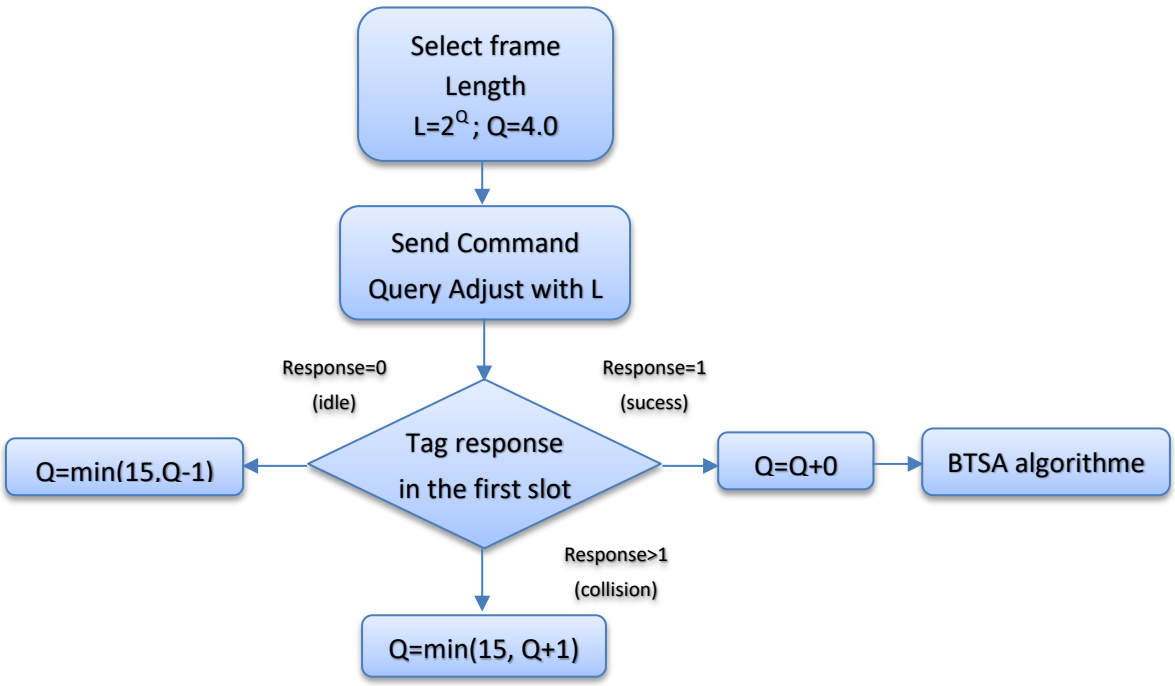



**Figure *II*. 24:** Flow chart of Dynamic

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, 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 [82]. 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 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 [81].

### 6.4 Adaptive Binary Tree Slotted Aloha

Adaptive binary tree slotted aloha (Adaptive BTSA) offers an improvement to the Q protocol [81]. 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 once again ends the frame earlier and sends a new command with a smaller frame length.

The reader uses the parameters B and Qfp in order to calculate the frame length. The initial frame length is to L=2Q and Q = 4. The Q algorithm can adjust the frame length by adjusting Q. The value of Q. is the rounded value of Qfp, which is a floating representation of Q. In the following process, the reader dynamically adjusts each slot using the presented values c [52]. In the first slot, if a collision occurs, the reader will calculate Qfp by increasing it by c. In the case of an idle slot, the reader decreases Qfp by c. When the reader identifies a tag, it will not change Qfp. The flow chart of this protocol is shown in Figure II.25.

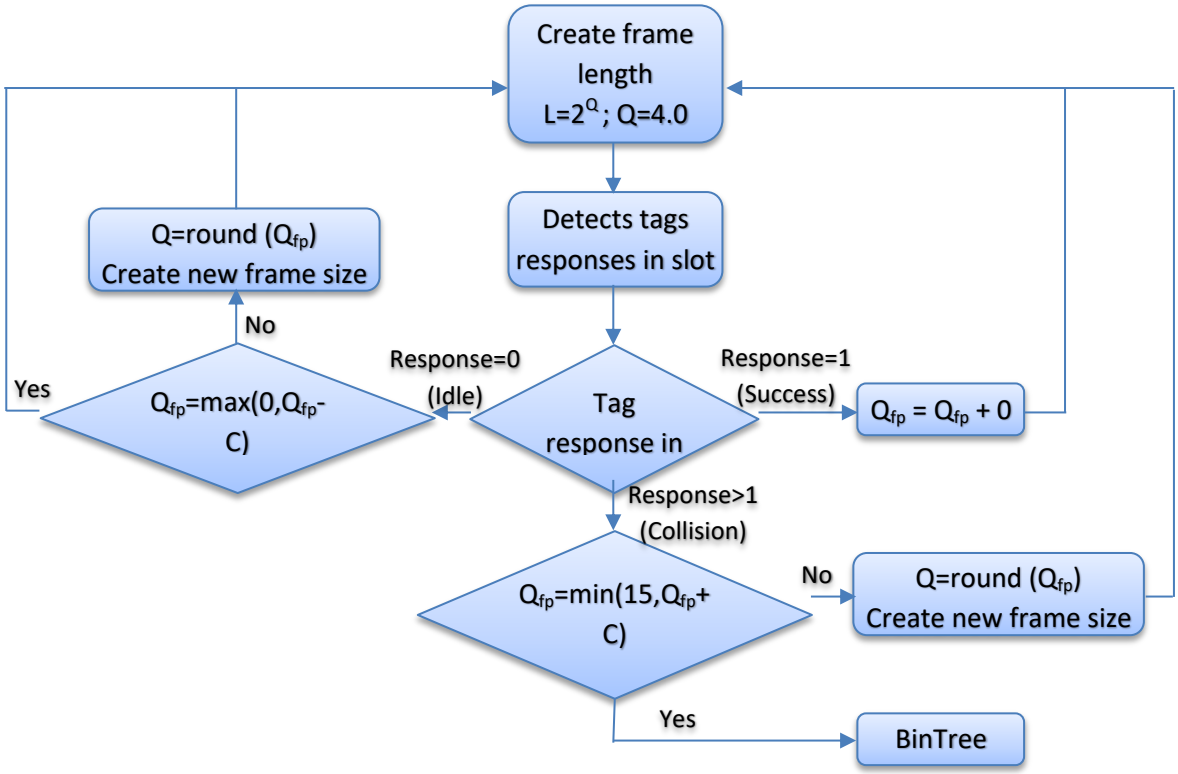



**Figure *II*.25:** Flow chart of Adaptive BTSA.

The function frame size (Q) shown in Figure II.25 denotes that a new frame has started and its length is 2Q. 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.

## 7. Discussion

Anti-collision protocols are a critical part of any RFID system. This section offers a critical analysis of the different protocols presented in the previous sections. The tag collision problem results in the wasting of bandwidth and energy, and an increased latency. Thus, an optimized anti-collision protocol is essential for a competitive RFID system.

The breadth of the literature reveals that there has been a great amount of research conducted in this area. There are two main types of anti-collision protocols: deterministic and probabilistic. 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. These types of protocols are highly adaptable to the appearances and disappearances of tags from the interrogation area. Deterministic protocols, on the other hand, 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. However, these protocols do not admit unexpected appearances and disappearances as easily as Aloha-based protocols. 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.

And finally, these types can be combined to form hybrid protocols, which provide very competitive protocols. Hybrid protocols have been created in order to avoid the problems of the Aloha and tree-based protocols, but this comes at the expense of complex reader and tag designs. Table II.7 shows observations regarding Aloha, tree-based and hybrid protocols.

**Table *II.* 7:** A comparison of Aloha, Tree-based and Hybrid protocols.

| Criterion | Aloha Protocols | Tree-Based Protocols | Hybrid Protocols |
|---|---|---|---|
| **Protocol Feature** | They use random multi-access means to identify tags. In the case of collision, 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 identifying all of the tags. | Tree-based protocols. They use two methods. The first uses randomized divisions in tree-based algorithms, and another uses 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, HF and UHF RFID (18000-6C) systems. | Tree-based protocols are commonly used in HF, UHF and micowave (18000-6B and 18000-7) RFID systems. | Not implemented on any standard |
| **Method** | Probabilistic | Deterministic | Mixture (Aloha and Tree-based) |
| **Tag starvation** | Yes | No | No |

From these explanations of the protocols, it cannot be concluded that certain protocol types stand out from the rest. However it should be noted that the newest protocols have become more sophisticated and attain better results in simulations. This also contrasts with the ability to implement these solutions in real hardware. RFID systems are very constrained systems, and their hardware needs to be very simple in order to comply with tags' needs. That is why many of these solutions have yet to be tested under real hardware conditions.

## 8. Conclusions

The term IoT, as established in RFID and the supply chain, involves the global information service architecture for RFID tags, that is, networked services that discuss things, rather than services. RFID is a key opportunity for the IoT due to its cost-effectiveness, high readability rates, automatic identification and, importantly, its energy efficiency benefits.

This paper presents some of the main RFID procedures and proposes some of the most up-to-date anti-collision protocols. In the literature, these may be classified into Aloha-based, tree-based, and hybrid protocols. The breadth of the literature reveals that there has been considerable research carried out in this field. However, further research needs to be conducted in order to ultimately implement all these solutions. RFID systems have become more and more widely present. Given that the number of tags has increased, these systems are faced with more important issues, and therefore, the use of anti-collision protocols will be more omnipresent.

# *Chapter III*

## *Simulation Results and Discussion*

# 1. Introduction

This chapter evaluate and compare the performance of some tree based anti-collision protocols presented in Chapter II. Then, it presents the assumption and the simulation scenario. The simulation is carried out using MATLAB software. The simulation results are presented in term of throughput and tag identification speed criteria to compare the performances of the tree based anti-collision protocols.

# 2. Performance metric

Before presenting our simulation results, in this section, we briefly introduce our performance evaluation metrics:

## 2.1 Throughput

In general term, throughput is the rate of production or the rate at which something is processed. In this work, the throughput of an anti-collision protocol is the ratio between the successfully read tags and the total number of slot during the identification process, or it is the number of successfully read tags per slot. The higher throughput, the more efficient protocol. The throughput formula is given by:

$$\text{Throughput} = \frac{\text{successfully read tags}}{\text{total number of slots}} \qquad (III.1)$$

## 2.2 Tag identification speed

The tag identification speed (TIS) is the ratio between the successfully read tags and the required time to identify all tags. It defines by [tags/second].

$$\text{TIS} = \frac{\text{successfully read tags}}{\text{required time to identify all tags}} \qquad (III.2)$$

# 3. EPC generation 2

## 3.1 Communication Protocol

The commonly used standard in RFID applications is the electronic product code generation 2 (EPC G2) standard, which defines the physical and logical requirement and describes the communication protocol for the passive RFID systems in the rang 860 MHZ - 960 MHz .The electronic product code (EPC) is designed as a universal identifier that provides a unique identity for every physical object anywhere in the world, for all time. The EPC structure is defined in the EPC G2. The Figure III.1. represents the structure of EPC-96, which is deliberately partitioned into 4 distinct parts to allow flexibility in object identification using RFID tags. The header defines the type of EPC. The 28 bit manager element allows for 268 million different users or product manufacturers, who are able to uniquely define 16 Million different types of object (24 bits). Each of these is then assigned a 36 bit serial number, giving a total of 68 billion potential unique identification numbers per object. In total the EPC-96 number space can uniquely identify 2 88 ≈ 3.09 × 1026 (309 septillion) different entities.

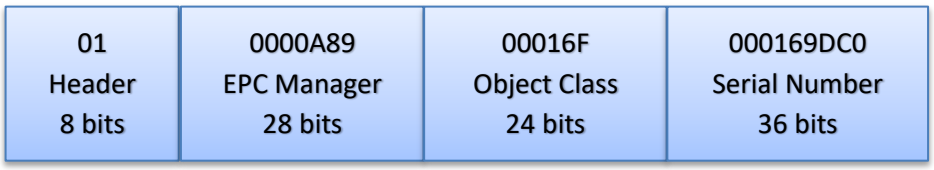



**Figure *III*.1:** EPC-96 structure code.

In the EPC G2, the reader sends various commands to ensure the communication between a tag and a reader. In addition these commands allow also resolving the tag collision problem. For example, if we have a single tag in the reader's interrogation zone. First the reader transmit a 'Query' command, containing the physical layer information such as: the data rate and the coding of the backscatter link (tag to reader). Then, when the tag receive the 'Query' command, it reply a generated 16 bit random number (RN16). After that, the reader sends a 'ACK' command to acknowledge the tag. This command contains the received RN16 from the tag. At the tag, if the RN16 of 'ACK' equals the RN16 memorized in the tag, then, it sends the identifier EPC to reader (see Figure III.2).

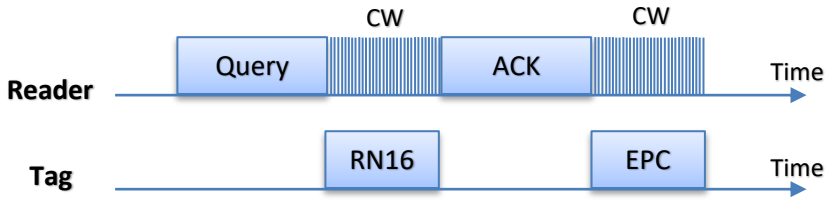



**Figure *III*.2:** Communication protocol between a tag and a reader.

### 3.2 Physical layer of EPC generation 2

In this thesis, we assumed that all simulated anti-collision protocols are based on EPC G2 standard [21]. We evaluated these protocols in term of TIS. Therefore, to compute TIS we need to define the physical layer of the forward link (reader to tag) and the backscatter link (tag to reader) of the EPC G2 and the frame structure of their reader commands and tag response.

#### 3.2.1 Forward Link

The forward link uses the pulse interval encoding (PIE) coding to transmit the reader commands. In this coding, the duration of data-0 equals to Tari, and the duration of data-1 equals to 2×Tari as shown in Figure III.3. Tari is the reference time interval for forward link, its values is in the range of 6.25µs to 25µs. According to EPC G2 standard, the maximum value of the pulse width (PW) shown in the Figure III.3 shall be:

$$\mathrm{Max}(0.265\mathrm{Tari}, 2\mu s) \qquad (III.3)$$

The minimum value of PW equals to 0.525Tari. In this work, we choose Tari and PW as: 6.25µs and 0.5Tari, respectively.

The coded commands are preceded by a preamble. In this work we use the preamble, while the others begin with a frame-sync. Figure III.4. shows the preamble and frame-sync of the forward link. A preamble and frame-sync comprise a fixed length start delimiter, a data-0 symbol and a reader-to-tag calibration (RTcal) symbol. However, the preamble comprises in addition a tag-to-reader calibration (TRcal) symbol.

The tag measures the length of Tari sent in the preamble, which is useful information for the PIE decoding. RTcal is the length of a data-0 symbol plus the length of a data-1 symbol (RTcal = 0length + 1length). Tag measures the length of RTcal and uses it to decode the reader signal. The reader shall specify a backscatter link data rate using the TRcal. The length of this latter shall be also measured by the tag. The values of delimiter, RTcal and TRcal set in this work are 12.5µs, 3Tari and 33.3µs, respectively.

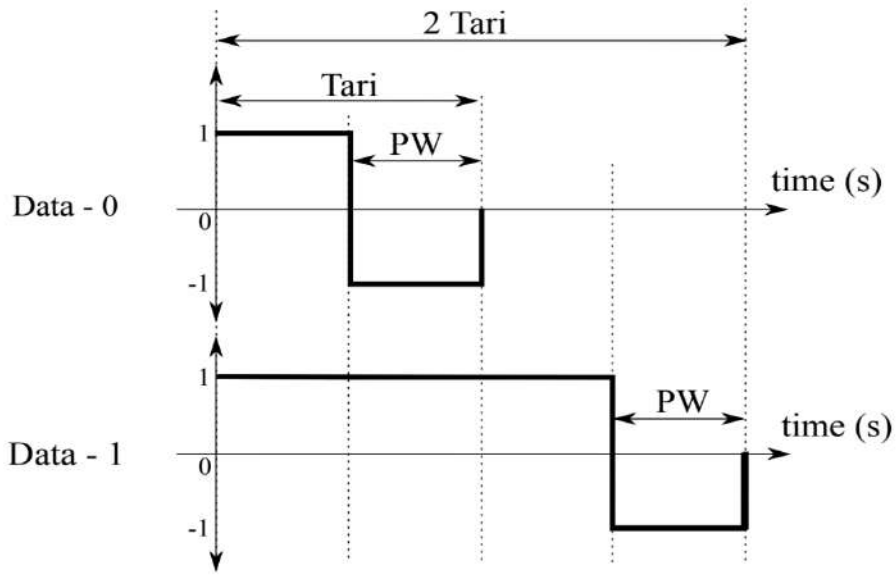



**Figure *III*.3:** PIE encoding.

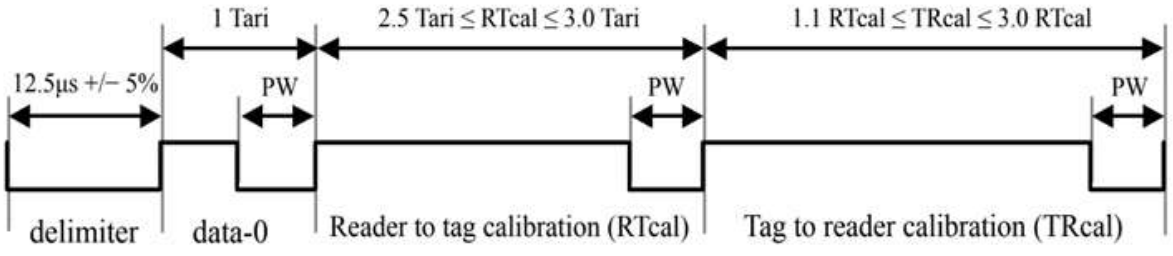



**Figure *III*.4:** Forward link preamble.

### 3.2.2 Backscatter Link

The data encoding of the backscatter link used in our simulation is the FM0 encoding. The FM0 coding is preceded by a 18 bits of backscatter link preamble. The backscatter link data rate BLF is given by using the DR field of the 'Query' command and the TRcal duration of the forward link preamble. Where the latter is measured by the tag. Therefore, BLF is specified as:

$$\mathrm{BLF} = \frac{\mathrm{DR}}{\mathrm{TRcal}} \qquad (III.4)$$

In our simulation, we set DR to 64/3 and TRcal to 33.3µs. Consequently, BLF = 640kbit/s. The tag response of 'Query', 'Query Adjust', or 'Query Rep' commands is a RN16, and the response of the 'ACK' command used in our simulation is composed of: 16 bits of protocol control (PC) , 96 bits of electronic product code (EPC) , and 16 bit of CRC-16 with a polynomial of $x^{16} + x^{12} + x^5 + 1$. Figure III.5 shows the frame structure of the response of the 'ACK' command.

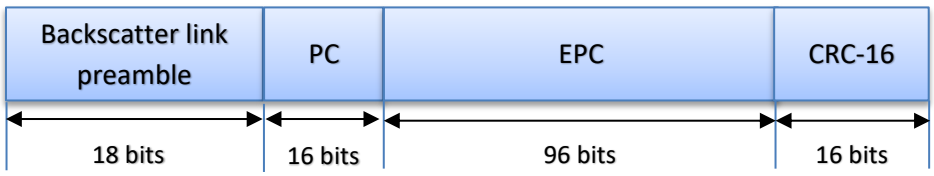



**Figure *III*.5:** Frame structure of the 'ACK' command response.

### 3.2.3 Link Timing

The time from reader command to tag is defined as T1 while the time from tag to reader command is defined as T2. T3 is the time between the reader commands as shown in Figure 6. According to EPC G2 the maximum and the minimum values of these times are summarized in Table III.1. Where, Tpri is the period of an FM0 symbol, which is equal 1/BLF, and FrT is the backscatter link data rate tolerance specified in [21]. In the simulation, we set T1, T2 and T3 to RTcal, 3Tpri and 2RTcal, respectively.

**Table *III*.1:** Minimum and maximum values of T1, T2 and T3.

| | **Minimum** | **Maximum** |
|---|---|---|
| $T_1$ | $Max(RTcal, 10T_{pri}) \times (1 - \lvert FrT \rvert) - 2\mu s$ | $Max(RTcal, 10T_{pri}) \times (1 - \lvert FrT \rvert) - 2\mu s$ |
| $T_2$ | $3T_{pri}$ | $20T_{pri}$ |
| $T_3$ | $2RTcal$ | $Max(RTcal, 10T_{pri}) \times (1 - \lvert FrT \rvert) - 2\mu s$ |

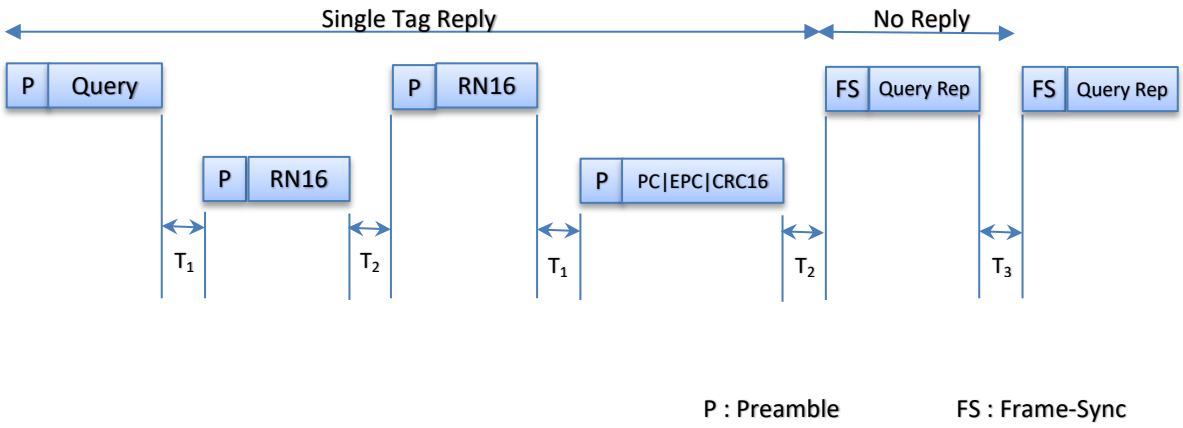



**Figure *III*.6:** Link timing of the EPC G2 standard.

The parameters of the air interface of the EPC G2 standard used in our simulation are summarized in the following table:

**Table *III*.2:** Air interface of EPC G2 standard parameters used in our simulations.

| | Parameters | Values |
|---|---|---|
| | | |
| Forward link | Coding | PIE |
| | *Tari* [µs] | 6.25 |
| | *PW* [µs] | 3.125 |
| | Delimiter [µs] | 12.5 |
| | RTcal | 3 *Tari* |
| | TRcal [µs] | 33.33 |
| | Duration of preamble [µs] | 70.8 |
| | Duration of frame-sync [µs] | 37.5 |
| | | |
| Backscatter link | Coding | FM0 |
| | DR | 64 / 3 |
| | Date rate (BLF) | 640 Kbit / s |
| | Preamble length [bits] | 18 |
| | EPC length [bits] | 96 |
| | | |
| Link timing [µs] | $T_1$ | 18.75 |
| | $T_2$ | 4.6875 |
| | $T_3$ | 37.5 |

## 4. Frame structure

For compute the required time to identify all tag, we need to define the frame structure of reader command and the tag response of the tree based anti-collision protocols. We cannot use the frame structures of the EPC Generation 2 standard in the tree based protocols, because the EPC Generation 2 standard is used in ALOHA protocols. Therefore, we propose a new form of the frame structure of reader command and the tag response of some tree based anti-collision protocols such as: QT, BS, TS and STT protocol.

### 4.1 Query Tree or Smart Trend-Traversal protocol

The query tree or smart trend-traversal protocol has one reader command, which is the "query" command. This command informs tags to reply their RN16 and it contain a parameter q . The frame structure of "query" command is shown in Figure III.7.

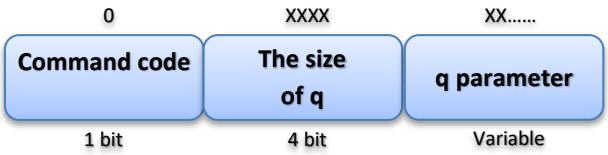



**Figure *III*.7:** the frame structure of query command QT protocol.

## 4.2 Binary search protocol

The Binary search protocol has the commands: “query” and “UNSELECT”. The “query” command contains a parameter q. If the tag’s RN16 less than q replies its RN16. The “UNSELECT” command makes the tag inactive. The frame structure of “query” and “UNSELECT” command are shown in Figure III.8 and Figure III.9 respectively.

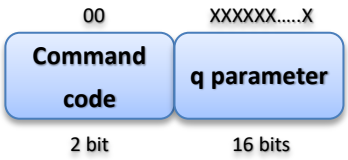



**Figure *III*.8:** the frame structure of query command BS protocol.




**Figure *III*.9:** the frame structure of unselect command.

## 4.3 Tree splitting protocols

The tree splitting protocol has two commands: “FAIL” and “SUCCESS” as described in Section V.9. The frame structure of “FAIL” and “SUCCESS” command are shown in Figure III.10 and Figure III.11 respectively.

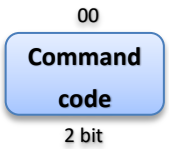



**Figure *III*.10:** the frame structure of fail command.




**Figure *III*.11:** the frame structure of success command.

## 5. Simulation results

In this thesis, we program and simulate the tree based anti-collision protocol such as: QT, BS, TS and STT protocols. Then we show their simulation results in term of throughput and tag identification speed depending on number of tags. These protocols follow the EPC Generation 2 standard in the physical layer. In the simulation, we simulate the protocols several iterations. The execution trace of one iteration is terminated, when the protocol identifies all tags.

The throughput and the tag identification speed in the simulation are obtained by computing the throughput and the tag identification speed during one iteration of the anti-collision process until all tags are identified. Then, the final throughput and tag identification speed are given by calculating the average throughput and the tag identification speed for 1000 iterations in the case of small and large scale. In small scale the numbers of tags are from 10 to 90 with an increment of 10 and in large scale the numbers of tags are from 100 to 500 with an increment of 50.

In the simulation, we set the number of ID to 16 bits which is the RN16. In addition, we assume that the channel model between the reader and tag is ideal with free noise. Moreover, we assume that the collision detection is perfect in all protocols.

Figure III.12 shows the throughput of QT, BS, TS and STT protocols depending on the number of tags in small scale (i.e. from 10 to 90 tags with an increment of 10). We can observe that, the BS protocol outperforms the other protocols in term of throughput. In addition, the QT and TS protocols have the same performance and are better than STT protocol. Moreover, we can see that the throughput is slightly decreased when the number of tag increase. The maximum value of throughput is provided by the BS protocol for $n = 10$ tags, which is equal to 0.526 read tags per slot.

Figure III.13 shows the tag identification speed (TIS) of QT, BS, TS and STT protocols versus the number of tags in small scale. We can see that, the TS protocol is better than the others. In addition, the tag identification speed of QT and STT are decreased when the number of tag increase. The TS and BS protocols are more stable than the QT and STT protocols. The maximum value of TIS is provided by QT and TS protocols for $n = 10$ tags and it equals to 1087 read tags / second.

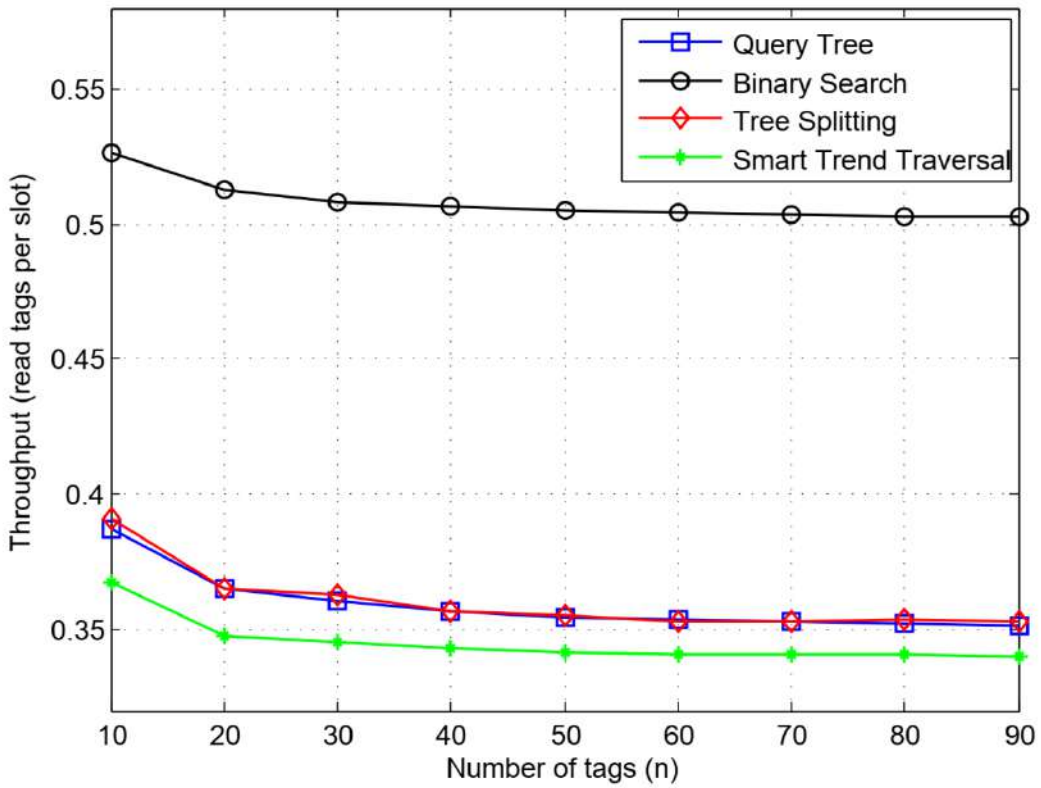



**Figure *III*.12:** Throughput of BS, QT, TS and STT versus the number of tags in a small scale.

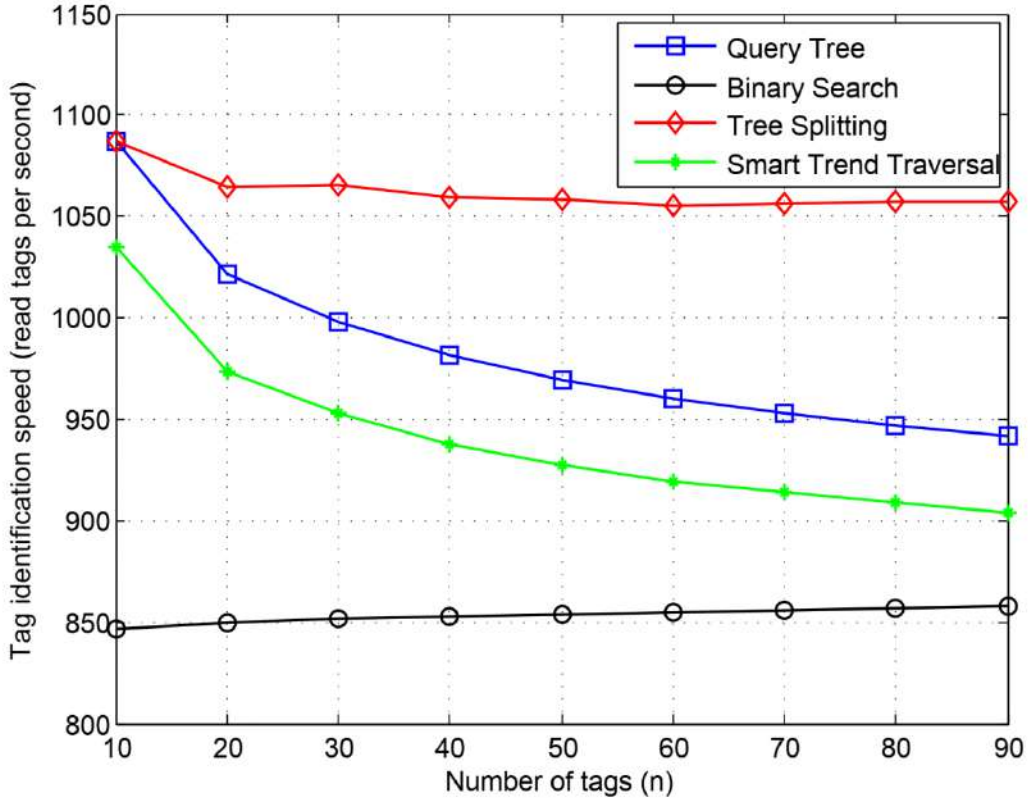



**Figure *III*.13:** TIS of BS, QT, TS and STT versus the number of tags in a small scale.

Figure III.14 shows the throughput of QT, BS, TS and STT protocols depending on the number of tags in large scale. We can observe that, BS protocol gives the best results compared with the others in term of throughput. QT and TS have the same performance for number of tags of 100 and 150 tags and for $n > 150$ tags, QT is better than TS protocol. For $n = 450$ and 500 tags the TS and STT protocol have the same performance. Moreover, we can observe that, the performance of STT increase with the number of tags.

Figure III.15 shows the tag identification speed of QT, BS, TS and STT protocols versus the number of tags in large scale. We can see that, the TS protocol outperforms the others and the QT protocol is better than BS and STT protocols. In addition, we can see that, the QT and STT protocols are decreased when the number of tag increase. In contrary, the BS protocol is increased when the number of tag increase. The maximum value of TIS is provided by TS protocol for $n = 100$ tags and it equal to 1054 read tags / second.

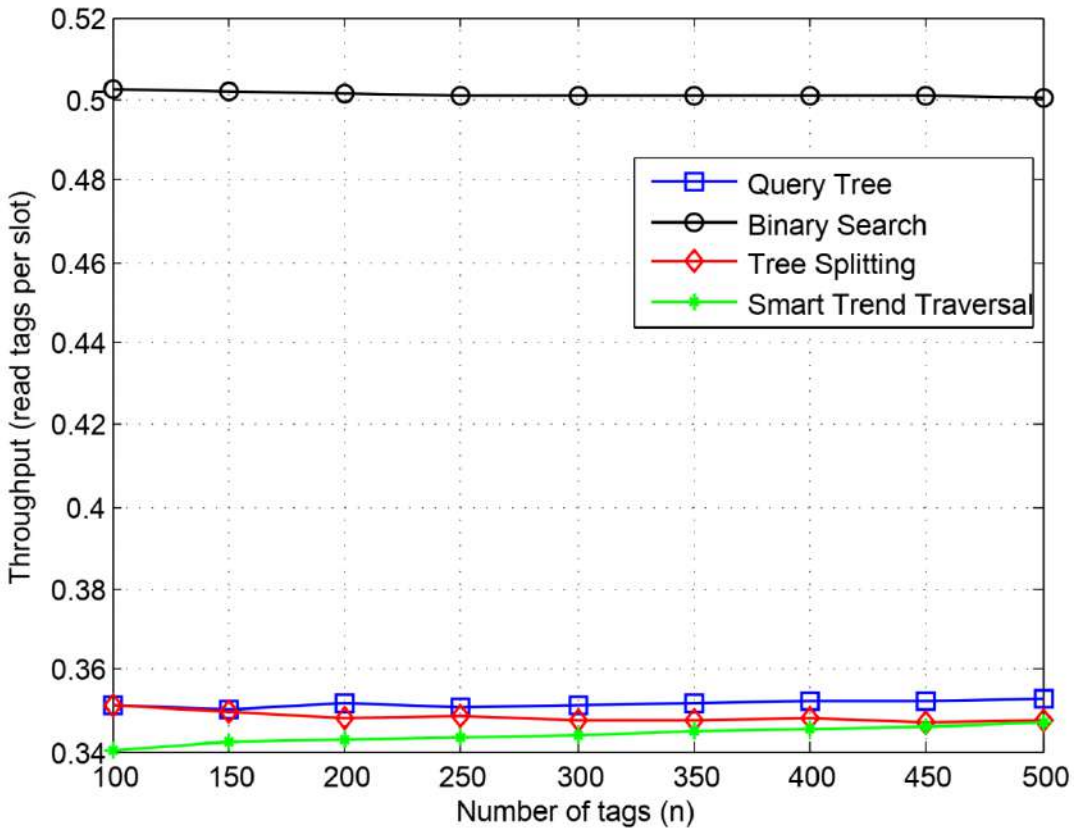



**Figure *III*.14:** Throughput of BS, QT, TS and STT versus the number of tags in a large scale.

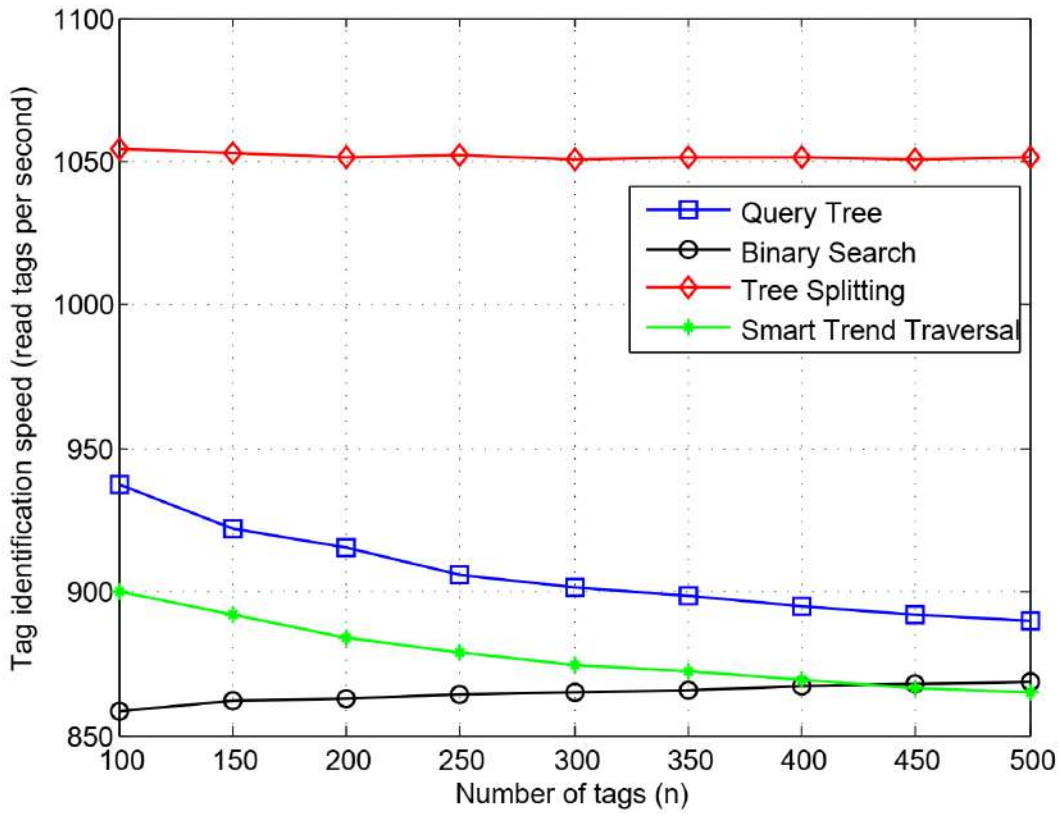



**Figure *III.*15:** TIS of BS, QT, TS and STT versus the number of tags in a large scale.

## 6. Conclusion

In this chapter, we simulated and compared the performance of the tree based anti-collisions protocols between each others in term of throughput and tag identification speed depending on the number of tags in small and large scale. In the simulation, the number of iteration chosen for running the protocols is 1000 iterations. The simulation result shows that, the binary search protocol outperforms the other protocols in term of throughput for small and large scale and the tree splitting protocols outperforms the other protocols in term of tag identification speed for small and large scale.

## *General Conclusion*

The major problem encountered with RFID systems during identification process is tag-collision, which occurs when multiple tags are within the interrogation zone of a reader and simultaneously reply to the reader's commands. This encouraged researchers to develop various MAC layer anti-collision protocols that can be divided into three categories: ALOHA, tree based and hybrid protocols. In this work, we focus on the tree based anti-collision.

In this thesis, we described different MAC layer anti-collision protocols such as: ALOHA based and tree-based anti-collision protocols. In the ALOHA-based, we presented DFSA and Q protocols, which are dynamic adaptability to varying tag numbers, unlike PA, SA and FSA. Also, we presented nine tree-based anti-collision protocols.

We developed and simulated under MATLAB software some of presented tree-based protocols such as: query tree, binary search, tree splitting and smart trend traversal protocols. In this work, we propose a new form of the frame structure of reader command and the tag response of some tree based anti-collision protocols such as: Query Tree, Binary Search, Tree Splitting and Smart Trend-Traversal protocol.

We compared the performance of these four anti-collision protocols, and we found that, the binary search protocol give better performance than the other protocols in term of throughput and the tree splitting protocols outperforms the others in term of tag identification speed.

## *Bibliography*

[1] Virnich, M. and Posten, K. (1992) Handbuch der codierten Datentr ¨ager, Verlag TUV Rheinland ¨ GmbH, Cologne, ISBN 3-8249-0044-0.

[2] ident (1996) ident 1, UMSCHAU Zeitschriftenverlag, Frankfurt

[3] Rankl, W. and Effing, W. (1996) Handbuch der Chipkarten, 2nd edn, Carl Hanser Verlag, Munich ISBN 3-446-18893-2.

[4] Lemme, Helmut (1993) Der Mikrorechner in der Brieftasche, Elektronik, 20, 22, 26, FranzisVerlag, Munich

[5] Encyclopaedia Britannica Deluxe Edition 2004, Copyright by Encyclopaedia Britannica, Inc.

[6] AIM (2001): "Shrouds of Time – The history of RFID", The Association of the Automatic Identification and Data Capture Industry, www.aimglobal.org, 10/01/2001.

[7] K. Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, New York: Wiley, 2010.

[8] O.Bang, J.H.Choi, D.Lee and H.Lee, "Efficient Novel Anti-collision Protocols for Passive RFID Tags," Auto-ID Labs White Paper WPHARDWARE-050, MIT, Mar.2009

[9] J.Banks, M.Pachano, L.Thompson and D.hanny, RFID Applied, New York: John Wiley & Sons, Inc., 2007.

[10] AHSAN, Kamran. RFID components, applications and system integration with healthcare perspective. Chapter in the book titled" Deploying RFID–Challenges, Solutions and Open Issues", Edited by C. Turcu, I-Tech Education and Publishing KG, 2011, p. 27-41.

[11] J. Kimionis, A. Bletsas, and J. N. Sahalos, "Increased range bistatic scatter radio," IEEE Transactions on Communications, vol. 62, no. 3, Mar. 2014, pp. 1091-1104.

[12] S. H. Choi, and D. I. Kim, "Backscatter radio communication for wireless powered communication networks," in Proc. of 21st AsiaPacific Conference on Communications (APCC), Kyoto, Japan, Oct. 2015, pp. 370-374.

[13] X. Lu, D. Niyato, H. Jiang, D. I. Kim, Y. Xiao, and Z. Han, “Ambient Backscatter Networking: A Novel Paradigm to Assist Wireless Powered Communications”, IEEE Wireless Communications.

[14] “RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification”, Second Edition, John Wiley & Sons, Ltd. ISBN 0-470-84402-7.

[15] AHSON, Syed A. et ILYAS, Mohammad. RFID handbook: applications, technology, security, and privacy. CRC press, 2017.

[16] J. Kimionis, A. Bletsas, and J. N. Sahalos, “Increased range bistatic scatter radio,” IEEE Transactions on Communications, vol. 62, no. 3, Mar. 2014, pp. 1091-1104.

[17] X.L. Jia, Q.Y. Feng, C.Z. Ma, “An efficient anti-collision protocol for RFID tag identification,” IEEE Communications Letters, vol.14, no.11, pp.1014-1016, 2010.

[18] R.K.Pateriya, S.Sharma, “The Evolution of RFID Security and Privacy: A Research Survey,” 2011 International Conference on Communication Systems and Network Technologies (CSNT), pp.115-119, June 2011.

[19] RFID-Journal. That Internet of Things. 2017. Available online: http://www.rfidjournal.com/articles/ (accessed on 15 April 2018).

[20] Finkenzeller, K. RFID Handbook; Wiley: Hoboken, NJ, USA, 2010.

[21] Azambuja, M.; Marcon, C.; Hessel, F. Survey of Standardized ISO 18000-6 RFID Anti-collision Protocols. In Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008), Cap Esterel, France, 25–31 August 2008; pp. 468–473.

[22] Shih, D.; Sun, P.; Yen, D.; Huang, S. Taxonomy and survey of RFID anti-collision protocols. Comput. Commun. 2006, 29, 2150–2166.

[23] Klair, D.; Chin, K.; Raad, R. A Survey and Tutorial of RFID Anti-Collision Protocols. IEEE Commun. Surv. Tutor. 2010, 12, 400–421.

[24] Abraham, C.; Ahuja, V.; Ghosh, A.; Pakanati, P.P. Inventory Management Using Passive RFID Tags: A Survey; Department of Computer Science, The University of Texas at Dallas: Richardson, TX, USA, 2002; pp. 1–16.

[25] Leong, K.; Ng, M.; Cole, P. The reader collision problem in RFID systems. In Proceedings of the 2005 IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, Beijing, China, 8–12 August 2005; pp. 658–661.

[26] Burdet, L.A. RFID Multiple Access Methods. Smart Environments Seminar: Zurich, 2004. Available online: http://www.vs.inf.ethz.ch/edu/SS2004/DS/reports/06RFID-macreport.pdf (accessed on 20 April 2018).

[27] Yu, J.; Liu, K.; Yan, G. A Novel RFID Anti-Collision Algorithm Based on SDMA. In Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, China, 12–17 October 2008; pp. 1–4.

[28] Sayeed, M.; Kim, Y. A Simple LMS Algorithm Based Smart Antenna to Solve the Reader Collision Problems in RFID System. In Proceedings of the 2009 International Conference on Information and Multimedia Technology, Jeju Island, Korea, 16–18 December 2009; pp. 426–430.

[29] Banks, J.; Pachano, M.; Thompson, L.; Hanny, D. RFID Anti-Collision System Using the Spread Spectrum Technique; Technical Report; Georgia Institute of Technology: Atlanta, GA, USA, 2005.

[30] Liu, H.C.; Ciou, J. Performance analysis of multi-carrier RFID systems. In Proceedings of the 2009 International Symposium on Performance Evaluation of Computer Telecommunication Systems, Istanbul, Turkey, 13–16 July 2009; Volume 41, pp. 112–116. [

[31] Loeffler, A.; Schuh, F.; Gerhaeuser, H. Realization of a CDMA-based RFID System Using a Semi-active UHF Transponder. In Proceedings of the 2010 6th International Conference on Wireless and Mobile Communications, Valencia, Spain, 20–25 September 2010; pp. 5–10.

[32] Tang, Z.; He, Y. Research of Multi-access and Anti-collision Protocols in RFID Systems. In Proceedings of the 2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID), Xiamen, China, 16–18 April 2007; pp. 377–380.

[33] Wang, W.; Zhang, Y.; Sang, Y.; Wang, S. Analysis of anti-collision algorithms in RFID system. In Proceedings of the 2009 IEEE International Conference on Communications Technology and Applications, Boston, MA, USA, 16–18 October 2009; pp. 58–62.

[34] Schoute, F. Dynamic Frame Length Aloha. IEEE Trans. Commun. 2010, 31, 565–568.

[35] Zhu, L.; Yum, T. The Optimal Reading Strategy for EPC Gen-2 RFID Anti-Collision Systems. IEEE Trans. Commun. 2010, 58, 2725–2733.

[36] Maguire, Y.; Pappu, R. An Optimal Q-Algorithm for the ISO 18000-6C RFID Protocol. IEEE Trans. Autom. Sci. Eng. 2009, 6, 16–24.

[37] Zhu, L.; Yum, T. Optimal Framed Aloha Based Anti-Collision Algorithms for RFID Systems. IEEE Trans. Commun. 2010, 58, 3583–3592.

[38] Chen, W.T. An Accurate Tag Estimate Method for Improving the Performance of an RFID Anticollision Algorithm Based on Dynamic Frame Length Aloha. IEEE Trans. Autom. Sci. Eng. 2009, 6, 9–15.

[39] Abramson, N. THE ALOHA SYSTEM: Another alternative for computer communications. In Proceedings of the Fall Joint Computer Conference, Houston, TX, USA, 17–19 November 1970; pp. 281–285.

[40] Roberts, L. ALOHA packet system with and without slots and capture. ACM SIGCOMM Comput. Commun. Rev. 1975, 5, 28–42.

[41] Klair, D.; Chin, K.; Raad, R. An Investigation into thie Energy Eficiency of Pure and Slotted Aloha Based REID Anti-Collision Protocols. In Proceedings of the 2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Espoo, Finland, 18–21 June 2007; pp. 1–4.

[42] Namboodir, V.; DeSilva, M.; Deegala, K.; Ramamoorthy, S. An extensive study of slotted Aloha-based RFID anti-collision protocols. Comput. Commun. 2012, 35, 1955–1966.

[43] Bueno-Delgado, M.; Vales-Alonso, J.; Gonzalez-Castaño, F. Analysis of DFSA anti-collision protocols in passive RFID environments. In Proceedings of the 2009 35th Annual Conference of IEEE Industrial Electronics, Pisa, Italy, 3–5 November 2009; pp. 2610–2617.

[44] Vogt, H. Efficient object identification with passive RFID tags. In Proceedings of the 2002 International Conference on Pervasive Computing, Zürich, Switzerland, 26–28 August 2002.

[45] Vogt, H. Multiple object identification with passive RFID tags. In Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia, 6–9 October 2002; Volume 3, pp. 1–6.

[46] Cha, J.R.; Kim, J.H. Novel Anti-collision Algorithms for Fast Object Identification in RFID System. In Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS'05), Fuduoka, Japan, 22–25 July 2005; Volume 2, pp. 63–67.

[47] Hwang, T.W.; Lee, B.G.; Kim, Y.S.; Suh, D.Y.; Kim, J.S. Improved Anti-collision Scheme for High Speed Identification in RFID System. In Proceedings of the First International Conference on Innovative Computing, Information and Control, Beijiang, China, 30 August–1 September 2006; Volume 2, pp. 449–452.

[48] Wieselthier, J.; Ephremides, A.; Michaels, L. An exact analysis and performance evaluation of framed Aloha with capture. IEEE Trans. Commun. 1989, 37, 125–137.

[49] Yan, F.; Shao, X.; Sun, Q. The dynamic anti-collision algorithm based on the similar binary in RFID system. In Proceedings of the 2008 2nd International Conference on Anti-Counterfeiting, Security and Identification, Guiyang, China, 20–23 August 2008; pp. 444–447.

[50] Lin, C.; Lin, F. Efficient Estimation and Collision-Group-Based Anti-collision Algorithms for Dynamic Frame-Slotted Aloha in RFID Networks. IEEE Trans. Autom. Sci. Eng. 2010, 7, 840–848.

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

[52] Wang, C.; Daneshmand, M.; Sohraby, K.; Li, B. Performance analysis of RFID Generation-2 protocol. IEEE Trans. Wirel. Commun. 2009, 8, 2592–2601.

[53] Lee, D.; Kim, K.; Lee, W. Q+-Algorithm: An Enhanced RFID Tag Collision Arbitration Algorithm. In Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing (UIC 2007), Hong Kong, China, 11–13 July 2007; pp. 23–32.

[54] Peeters, G.; Houdt, B. Interference Cancellation Tree Algorithms with k-Signal Memory Locations. IEEE Trans. Commun. 2010, 58, 3056–3061.

[55] Yeh, M.; Jiang, J.; Huang, S. Parallel Response Query Tree Splitting for RFID Tag Anti-collision. In Proceedings of the 2011 40th International Conference on Parallel Processing Workshops, Taipei, Taiwan, 13–16 September 2011; pp. 6–15.

[56] Myung, J.; Lee, W.; Shih, T. An Adaptive Memoryless Protocol for RFID Tag Collision Arbitration. IEEE Trans. Multimed. 2006, 8, 1096–1101.

[57] Gou, H.; Jeong, H.; Yoo, Y. A Bit collision detection based Query Tree protocol for anti-collision in RFID system. In Proceedings of the 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, Chengdu, China, 23–25 September 2010; pp. 421–428.

[58] Zein, W.; Shaaban, E. An enhanced binary tree anti-collision technique for dynamically added tags in RFID systems. In Proceedings of the 2010 2nd International Conference on Computer Engineering and Technology, Cengdu, China, 16–18 April 2010; Volume 7, pp. 349–353.

[59] Law, C.; Lee, K.; Siu, K. Efficient memoryless protocol for tag identification. In Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Boston, MA, USA, 11 August 2000. [

[60] Choi, J.; Lee, D.; Youn, Y.; Jeon, H.; Lee, H. Scanning-Based Pre-Processing for Enhanced Tag Anti-Collision Protocols. In Proceedings of the 2006 International Symposium on Communications and Information Technologies, Damascus, Syria, 24–28 April 2006; pp. 1207–1211.

[61] Zhou, F.; Chen, C.; Jin, D.; Huang, C.; Min, H. Evaluating and Optimizing Power Consumption of Anti-Collision Protocols for Applications in RFID Systems. In Proceedings of the 2004 International Symposium on Low Power Electronics and Design, Newport Beach, CA, USA, 9–11 August 2004; pp. 357–362.

[62] Choi, J.; Lee, D.; Lee, H. Query tree-based reservation for efficient RFID tag anti-collision. IEEE Commun. Lett. 2007, 11, 85–87.

[63] Bhandari, N.; Sahoo, A.; Iyer, S. Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency. In Proceedings of the 9th International Conference on Information Technology, Orissa, India, 18–21 December 2006; pp. 46–51.

[64] Pan, L.; Wu, H. Smart Trend-Traversal Protocol for RFID Tag Arbitration. IEEE Trans. Wirel. Commun. 2011, 10, 3565–3569.

[65] Yan, X.; Zhang, R.; Li, B. Smart Trend-Traversal Protocol with Shortcutting for Memoryless RFID Tag Collision Resolution. In Proceedings of the 2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing, Fukuoka, Japan, 4–7 September 2012; pp. 857–862.

[66] Landaluce, H.; Perallos, A.; Zuazola, I. A Fast RFID Identification Protocol with Low Tag Complexity. IEEE Commun. Lett. 2013, 17, 1704–1706.

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

[68] Landaluce, H.; Perallos, A.; Onieva, E.; Arjona, L.; Bengtsson, L. An Energy and Identification Time Decreasing Procedure for Memoryless RFID Tag Anticollision Protocols. IEEE Trans. Wirel. Commun. 2016, 15, 4234–4247.

[69] Cmiljanic, N.; Landaluce, H.; Perallos, A.; Arjona, L. Reducing Transmitted Bits in a Memoryless RFID Anti-Collision Protocol. IEEE Commun. Lett. 2016, 14, 713–715.

[70] Cmiljanic, N.; Landaluce, H.; Perallos, A.; Arjona, L. Influence of the Distribution of Tag IDs on RFID Memoryless Anti-Collision Protocols. Sensors 2017, 17, 1891.

[71] Wang, T.P. Enhanced binary search with cut-through operation for anti-collision in RFID systems. IEEE Commun. Lett. 2006, 10, 236–238.

[72] Yu, S.; Zhan, Y.; Wang, Z.; Tang, Z. Anti-collision algorithm based on jumping and dynamic searching and its analysis. Comput. Eng. 2005, 31, 19–20.

[73] Jia, X.; Feng, Q.; Ma, C. An Efficient Anti-Collision Protocol for RFID Tag Identification. IEEE Commun. Lett. 2010, 14, 1014–1016.

[74] Lai, Y.; Hsiao, L.; Chen, H.; Lai, C.; Lin, J. A Novel Query Tree Protocol with Bit Tracking in RFID Tag Identification. IEEE Trans. Mob. Comput. 2013, 12, 2063–2075.

[75] Lai, Y.C.; Hsiao, L.Y.; Lin, B.S. Optimal Slot Assignment for Binary Tracking Tree Protocol in RFID Tag Identification. IEEE/ACM Trans. Netw. 2015, 23, 255–268.

[76] Hush, D.; Wood, C. Analysis of tree algorithms for RFID arbitration. In Proceedings of the 1998 IEEE International Symposium on Information Theory, Cambridge, MA, USA, 16–21 August 1998; pp. 107–125.

[77] Zhang, L.; Zhang, J.; Tang, X. Assigned Tree Slotted Aloha RFID Tag Anti-Collision Protocols. IEEE Trans. Wirel. Commun. 2013, 12, 5493–5505.

[78] Bonuccelli, M.; Lonetti, F.; Martelli, F. Tree slotted aloha: A new protocol for tag identification in RFID networks. In Instant Collision Resolution for Tag Identification in RFID Networks; Elsevier: New York, NY, USA, 2007; pp. 1220–1232.

[79] Qian, C.; Liu, Y.; Ngan, H.; Ni, L. ASAP: Scalable Identification and Counting for Contactless RFID Systems. In Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems, Genova, Italy, 21–25 June 2010; pp. 52–61.

[80] Myung, J.; Lee, W.; Srivastava, J. Adaptive binary splitting for efficient RFID tag anticollision. IEEE Commun. Lett. 2006, 10, 144–146.

[81] Wu, H.; Zeng, Y.; Feng, J.; Gu, Y. Binary Tree Slotted Aloha for Passive RFID Tag Anticollision. IEEE Trans. Parallel Distrib. Syst. 2013, 24, 19–31.

[82] Porta, T.L.; Maselli, G.; Petrioli, C. Anticollision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization. IEEE Trans. Mob. Comput. 2011, 10, 267–279.

[83] Kalache, M. A. “RFID Signals Separation by Independent Component Analysis,” Ph.D. dissertation, University of science and technology Houari Boumediene, 2021.