Universidade de Aveiro
2008

Departamento de Electrónica,
Telecomunicações e Informática

Pedro Miguel
Oliveira da Silva

# Protocolo MAC para Sistemas de Localização

# MAC Protocol for Location Systems

Universidade de Aveiro
2008

Departamento de Electrónica,
Telecomunicações e Informática

Pedro Miguel
Oliveira da Silva

# Protocolo MAC para Sistemas de Localização

# MAC Protocol for Location Systems

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor José Alberto Fonseca, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro. !

Dedico este trabalho aos meus Pais, pelo Exemplo de Vida que me deram.
À minha Família, pelo apoio e carinho incondicional.
A todos os Amigos que me apoiaram durante o desenrolar deste projecto.
Aos meus Orientadores agradeço a disponibilidade, o apoio e a forma como me souberam ensinar e motivar.

**o júri**

presidente

Prof. Doutor Valeri Skliarov
professor catedrático da Universidade de Aveiro

Prof. Doutor José Alberto Gouveia Fonseca
professor associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Mestre Paulo Jorge de Campos Bartolomeu
Especialista em comunicações wireless, Micro I/O

Prof. Doutor Mário Jorge de Andrade Ferreira Alves
professor coordenador do Departamento de Engª Electrotécnica do Instituto Superior de Engenharia do Porto

**palavras-chave**

ZigBee, FTT, FTT-L, Comunicações sem fios, Localização, Flexibilidade, IEEE 802.15.4, LOPES

**resumo**

Na ultima década as redes de comunicações sem fios sofreram uma evolução sem precedentes, e as suas características particulares potenciaram a sua aplicação em vários contextos. No caso caso específico da localização em ambientes interiores, pretende-se que através de dados recolhidos por um conjunto de sensores sem fios, seja possível detectar com relativa precisão um objecto móvel devidamente identificado. Este tipo de sistemas permitem por exemplo, monitorizar um paciente de risco num hospital, dando informação precisa do seu paradeiro no momento da ocorrência de algum incidente.
Outra possível aplicação será, por exemplo, dentro de uma empresa ou instituição, recolher informação precisa sobre o paradeiro de cada trabalhador, visitante ou produto. Isto permite melhorar a logística a gestão dos recursos.
As soluções comuns de localização não endereçam devidamente a problemática de acesso ao meio para a realização de transmissões. Isto tem impacto na eficiência de utilização do canal de comunicações e resulta num número inferior de localizações por unidade de tempo.
Estes sistemas revelam assim a necessidade de um protocolo de acesso ao meio específico que permita reduzir o tempo necessário à localização de cada dispositivo móvel permitindo a integração de um número mais elevado dispositivos móveis ou fixos numa rede de localização.
Esta tese explora a utilização de Zigbee para implementar um protocolo master/multi-slave (FTT-L) aplicado a um sistema de localização.
Este protocolo bem como a sua implementação é descrita neste documento.
São também conduzidos vários testes para determinar alguns parâmetros chave. Os resultados são utilizados para derivar o tempo mínimo de localização que é validado experimentalmente.

**keywords**

ZigBee, FTT, FTT-L, System, Indoor, Wireless, Location, System, Network, Flexible, Time, Trigger, IEEE 802.15.4, Sensor, Localization, LOPES

**abstract**

In the last decade the proliferation of wireless communication networks has reached unprecedented values, and their features empowered the application of wireless networks to the most varied contexts.
In the specific case of indoor location, the target is to detect with relative precision an identified mobile object with the data collected from a wireless sensor network. This system allows us to monitor a risk patient in an hospital, giving the precise information about its location at the moment of a possible accident. Other possible application is, for example, to gather inside a company or institution the precise location of every worker, visitor or product. This can improve the logistics and the management of the personnel.
The common location solutions do not address the problematic of medium access for transmitting messages. This has impact on the transmission channel occupation and indirectly lowers the system efficiency, which results on a smaller number of locations per unit of time.
These systems reveal the necessity of a specific protocol for accessing the medium to reduce the necessary time to locate each mobile device, allowing the integration of a larger number of mobile devices or fixed devices in a location network.
This thesis explores the use of ZigBee to implement a master/multi-slave protocol (FTT-L) supporting a localization system. This protocol as well as its implementation are described throughout the thesis and an assessment of several key parameters is conducted. Results are used to derive the delay of a localization round, which was validated experimentally.

# Contents

# Figure Index

# Table Index

# Acronyms

| | |
|---|---|
| AB | Active Badge |
| ADC | Analog to Digital Converter |
| AES | Advanced Encryption Standard |
| BE | Backoff Exponent |
| CAP | Contention Access Period |
| CCA | Clear Channel Assessment |
| CFP | Contention Free Period |
| CLS | Cisco Location Solution |
| CSMA-CA | Carrier Sense Multiple Access with Collision Avoidance |
| CW | Contention Window length |
| DSSS | Direct Sequence Spread Spectrum |
| ED | Energy Detection |
| FFD | Full Function Device |
| FID | Frame Identifier |
| FTT | Flexible Time Trigger |
| FTT-L | Flexible Time Trigger for Location |
| FTT-SE | Flexible Time Trigger – Switched Ethernet |
| GTS | Guaranteed Time Slot |
| ILS | Indoor Location System |
| IR | Infra-Red |
| IWLS | Indoor Wireless Location System |
| LC | Location Cycle |
| LCD | Liquid Crystals Display |
| LEC | Location Elementary Cycle |
| LED | Light Emitting Diode |
| LOPES | LOcalização de PESsoas |
| LQI | Link Quality Indication |
| MAC | Medium Access Control |
| MBS | Master Base Station |
| MFR | MAC Footer |
| MGU | Master Gateway Unit |
| MHR | MAC Header |
| MSDU | MAC Service Data Unit |
| MT | Mobile Transceiver |

| | |
|---|---|
| NB | Number of Backoffs |
| OFDM | Orthogonal Frequency Division Multiplex |
| OFDM | Orthogonal Frequency Division Multiplexing |
| PAN | Personal Area Network |
| PC | Personal Computer |
| PHY | PHYsical Layer |
| PSDU | PHY Service Data Unit |
| RF | Radio Frequency |
| RFD | Reduced Function Device |
| RFID | Radio-Frequency Identification |
| ROI | Return of Investment |
| RTL | Real Time Location |
| RTLS | Real-Time Location System |
| RU | Reference Unit |
| SBS | Slave Base Station |
| SNR | Signal-to-Noise Ratio |
| SoC | System on Chip |
| TCP | Transmission Control Protocol |
| TM | Trigger Message |
| USART | Universal Synchronous and Asynchronous Receiver Transmitter |
| USB | Universal Serial Bus |
| ZDO | Zigbee Device Objective |

# Chapter 1

# Introduction

## 1.1. Motivation

Wireless Locations Systems experienced great development in the last years. The possibility to determine the location of a mobile object in a confined space or room is of great interest. Indoor Wireless Location Systems (IWLS) can be applied to many fields. In Industry, Security can be improved by knowing the exact location of every worker or device inside a High Security facility. Implementing this type of system can trigger an alarm if, for example, a visitor (or a worker) enters an unauthorized area. In Logistics, IWLS can be applied to stock management. A location mobile device can be applied to a product, informing the stock management system about the exact location of that product. If for some reason the product is misplaced, there is no problem, its location is always correct and the product is never lost. During medical surgery, a Doctor can always be sure of the location of the tools that are used during the intervention. There is no risk that any tool is left inside the patient. In other case, as a smart home can track a person, it can activate or deactivate devices as he/she walks by the house, adjusting it's behaviour to his/her needs, such as turning off the lights of a room when it is empty, or changing the music according to who is in that room.

The possibility to track objects and people has various applications and a coherent real-time wireless location system can revolutionize the sensor industry in the way we see it. Wireless Location can also be applied to many other purposes; some of them are approached in Chapter 2, where a state-of-the-art on wireless location systems is provided.

There are many academic and commercial solutions, though most solutions despise the medium access problematic. These kinds of system clearly reveal the need for a specific Medium Access Control (MAC) protocol to manage the medium occupation and reduce the necessary time to locate each mobile device, so the system can achieve a location process that is closer to Real-Time. Flexible Time Trigger (FTT-L) for Location allows location systems to integrate a larger number of mobile and fixed devices in a location network.

This research resulted on a paper “Implementig the FTT-L protocol with Zigbee” [SIL01].

## 1.2. Objectives

The main focus of this work is to implement a protocol for IWLS based on FTT-L specification [JAF01]. This protocol intends to optimise the medium usage in wireless location systems by controlling the medium access transmission control message sent by the network’s master.

In order to design the protocol, several studies were conducted, namely an analysis of the State-of-the-art of IWLS, with focus on location techniques.

It is also an objective of this research to determine ultimately the effective values for location cycles depending on the system characteristics and considering different contexts.

The goal is to experimentally validate the FTT-L as a protocol that provides efficient communications in location systems.

## 1.3. Thesis Structure

This thesis is organized in 8 chapters, including the Introduction (Chapter 1). Chapter 2 provides a study on the State-of-the-art of Wireless Location systems where several projects are approached including commercial and non-commercial solutions. Chapter 3 is an overview on IEEE 802.15.4 protocol, with special concern regarding the MAC and PHY layer properties of the stack. In Chapter 4 an overview on the LOPES project is provided with focus on the used communications protocol. The Hardware and Software features of the LOPES system are also described. Chapter 5 introduces and specifies the FTT-L protocol. This Chapter also presents the device’s state diagrams and exposes the primary functions and characteristics of the main elements that compose the location system. Chapter 6 presents the main coding functions used to program the Chipcon Z-Stack, the protocol stack where an experimental version of the FTT-L was implemented.

In Chapter 7, all the performance practical tests are presented, analysed and discussed. Chapter 8 concludes this thesis and provides some lines of future work.

# Chapter 2

# State-of-the-art

Wireless Location has been an area where more investments have been in the later years. Global Positioning System (GPS) technology was not designed to operate inside buildings, therefore Indoor Real-Time Location represents one of the most desired achievements in the current wireless sensor business.

An Indoor Location System (ILS) could monitor and optimize a process by locating instantly an asset, an authorized wireless device, or even an unauthorized one anywhere in an industry's wide facilities. This kind of system could even give a precise location of a worker in a possible risk situation. If a system could provide location aware support to workers, reducing defects caused by human error and cutting down the time needed to execute a particular task, it would lead to improvements in quality and lower the company’s production costs. When considering of more accurate location, with a location uncertainty of 1 cm, it would be possible to determine in real time the precise location of a medical tool during a surgery.

The following sub-sections provide an overview of a few representative wireless location systems.

## 2.1. eShepherd

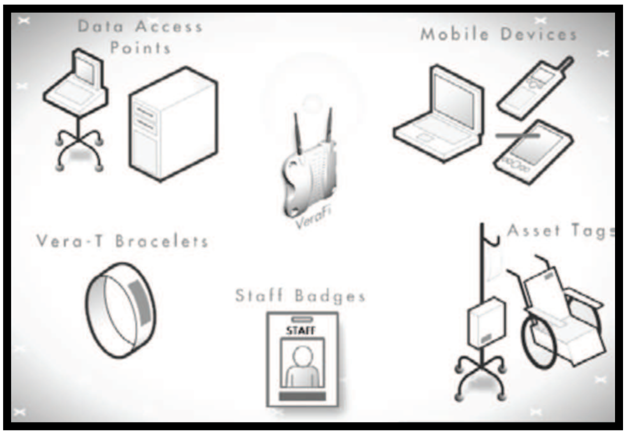


*Figure 1: eShepherd solution for location in Healthcare Management Systems [ESH01]*

**Overview**

eShepherd is wireless information-delivery solution applied to healthcare [ESH01]. This solution integrates RFID devices, such as patient bracelets, staff badges and asset tags like the ones represented in Figure 1 with a secure wireless network, providing patient identification; patient, staff and asset location and data-delivery to wireless devices such as PDAs or tablet PC. As the information is instantly available, time, costs and lives can be saved.

RFID readers are conveniently spread over the hospital and each IEEE 802.11 access point is equipped with a long range RFID Reader, powered over Ethernet. In eShepherd, a single Access Point is capable of supporting up to 2500 devices. The system communicates with wireless devices, such as PDAs, tablet PCs, laptops, desktop computers, WiFi telephones, and printers, via IEEE 802.11.

**Operation**

The system is composed of Passive and Active Tags, long range RFID and proximity based RFID. Passive Tags receive the energy that uses to transmit it's signal from a magnetic field generated by an RFID reader. Without this field, the tag is inactive. An Active Tag receives the energy its uses to transmit its signal from a battery or other direct power source. As a patient passes near a RFID reader, the patient’s Tag (bracelet) is detected and the information about the patient’s approximate location is updated. The system also possesses battery-assisted Tags and Long Range RFID technology to communicate with tags that are away from the reader.

| | |
|---|---|
| **Range** | 50 m |
| **Precision** | Room |
| **Technology** | WiFi - IEEE 802.11 and RFID |
| **Purpose** | Hospital and Healthcare Assistance |
| **Commercial** | Yes |

*Table 1: eShepherd Features*

## 2.2. Ekahau RTLS

*Figure 2: Photo of a Identifier used by the Ekahau RTLS [EKA01]*

**Overview**

Ekahau's Wi-Fi based RTLS uses patented software-based algorithms to compute location of tracked objects, eliminating the need for exciters, choke points, receivers and other proprietary hardware infrastructure [EKA01]. This system is a complete location solution. Mobile TAGs like the one in Figure 2 are placed on the objects that are to be tracked.

**Operation**

Ekahau system includes battery powered Wi-Fi Tags, a Positioning Engine and an application for network verification and for creating positioning models during the system set-up. Though the Positioning Engine provides it's own location algorithm it allows the user to calculate the position using different algorithms. This system enables full visibility across geographically dispersed campuses without the need to install software or hardware in remote sites.

The system is already a complete versatile commercial solution that encompasses a flexible database and features like API support to integrate XML.

| | |
|---|---|
| **Range** | 50 m |
| **Precision** | not defined |
| **Technology** | Wi-Fi |
| **Purpose** | Asset Location |
| **Commercial** | Yes |

Table 2: Ekahau RTLS features

## 2.3. Cisco Location Solution

*Figure 3: Picture on Cisco Location Solution system [CLS01]*

**Overview**

The Cisco Location Solution (CLS) was built to enable the location of Wi-Fi devices on a given network, such as Wi-Fi laptops or Wi-Fi phones and also Wi-Fi tags added to mobile assets [CLS01]. CLS allows locating unauthorized wireless devices to increase network and to track a variety of mobile assets within the facilities to monitor processes and find opportunities to optimize them.

The system can be used for searching a lost Wi-Fi capable device that can be anywhere in the facilities. Users can track equipment at any time, for example, when a mobile device is needed or when it is lost. The system also supports remote inventory functions. When a vehicle arrives at the Wi-Fi-enabled facilities, its identification and the condition of the Wi-Fi tagged goods can be retrieved without manual intervention, with a user-friendly interface like the one shown in Figure 3.

**Operation**

In CLS, location has to belong to a predefined set of places and the location information only has to be known at certain events. The system uses Wi-Fi devices, also known as Tags to provide a regular update of the location and telemetry information sent between the locatable device and the access point. The CLS can determine the precise location of a given device using an adaptive algorithm that reflects the environmental characteristics. This algorithm enables locating an asset within 10 meters with a maximum accuracy close to 60 cm. Some Wi-Fi tags designed for the system have telemetry capabilities and can sense information such as temperature, pressure, and humidity.

As multiple-frequency tags come close to a chokepoint, their lower frequency side receives energy from it and thus they transmit their location and telemetry information through the Wi-Fi side of the tag to the access point. A chokepoint location system allows location and telemetry information is updated on an event basis, such as when the tag is in proximity of the chokepoint. The locatable device and the chokepoint must be within a close distance, which provides deterministic and limited location to certain zones.

| | |
|---|---|
| **Range** | 50 m |
| **Precision** | 60 cm maximum |
| **Technology** | Wi-Fi - IEEE 802.11<br>RFID chokepoints (optional) |
| **Purpose** | Asset Tracking, Network planning |
| **Commercial** | Yes |

*Table 3: Cisco Location Solutions features*

## 2.4. Ubisense

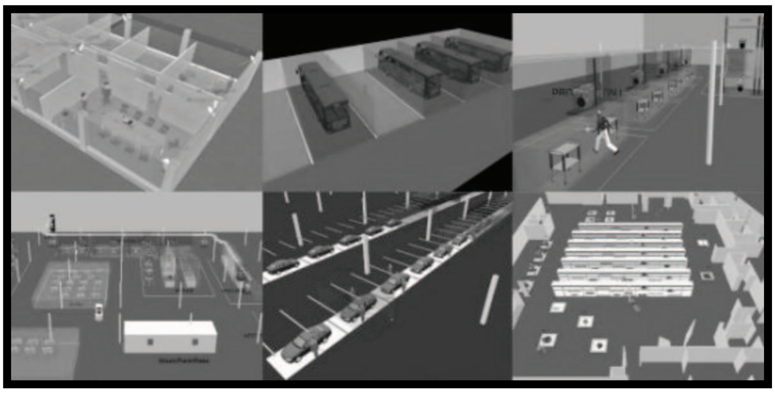

*Figure 4: Scheme on Ubisense Sistem Application [UBI01]*

**Overview**

The Ubisense Location System provides a 15cm 3D positional accuracy in real-time enabling a large number of new applications such as precise device location in space and time, for example to associate the data from a bar code scanner with its location in a warehouse, represented in Figure 4.

The system can also be used to provide the location of a pallet of goods placed in a warehouse by a fork-lift truck by associating the precise location of the truck's fork with the pallet or shipment ID at the time of placement. Other applications are automatic context sensitive proximity detection between objects, for example to select the correct program to drive an automatic tool based on its proximity to a car of a particular type and model without any manual input and identification and location of cars in finishing bays during final quality assurance.

Other example is for determining if, during an emergency procedure, employees have all reached designated areas or if there is a "man down" requiring assistance.

**Operation**

Ubisense applied Ultra-WideBand (UWB) to build a Real-Time Location System (RTLS) which delivers very high positional accuracy in traditionally challenging environments at reliability levels unachievable by legacy technologies such as conventional RFID or Wi-Fi [UBI01]. As Ubisense represents a commercial solution, it wasn't possible to obtain more details on the system architecture.

| | |
|---|---|
| **Range** | 50 m |
| **Precision** | 15 cm (maximum) |
| **Technology** | UWB |
| **Purpose** | Precise Asset Tracking |
| **Commercial** | Yes |

*Table 4: Ubisense features*

## 2.5. Cricket

**Overview**

Cricket is an indoor location system that provides fine-grained location information to applications running on handhelds, laptops or sensor nodes [CRI02]. Its software stack runs on TinyOS, and supports continuous object tracking, with various auto-configuration algorithms.

Location may be specified as a coordinate position in some coordinate system, a geographic space such as a room or portion of a room, and as the orientation of a device within some coordinate system. There are many examples of location-aware applications that can be developed using Cricket including resource discovery, human/robot navigation, physical/virtual computer games, location-aware sensing, hospital/medical applications, etc.

Cricket is intended for use indoors or in urban areas where outdoor systems like GPS doesn't work. Cricket can provide a positioning precision up to 3 cm.


*Figure 5: Cricket Deployment [CRI01]*

**Operation**

Cricket uses a combination of RF and ultrasound technologies to provide location information to attached host devices. Wall or ceiling beacons like the ones in Figure 5 are placed through a building, and publish information on an RF channel.

| | |
|---|---|
| **Range** | 20 m |
| **Precision** | 3cm (maximum) |
| **Technology** | RF and Ultrasounds |
| **Purpose** | Fine Location Utilities |
| **Commercial** | No |

*Table 5: Cricket features*

When a pulse sent by the beacon arrives, the listeners obtain a distance estimate for the corresponding beacon, by evaluating the difference in propagation speeds between RF and

Ultrasound [CRI01] using an algorithm to correlate RF and Ultrasound samples. The tracking system consists on a centralized controller with a database that receives transmissions from users and devices and tracks them.

## 2.6. PanGO

**Overview**

The PanGO Integrated Asset Tracking Solution (IATS) operates using a 802.11 infrastructure through a control platform that centralizes information [PAN01]. The system encompasses a group of IEEE 802.11 access points that measure the RSSI values of the tags. At the same time, the RFID notices when a tag goes through checkpoints like doors.

PanGO IATS allows organizations to transform their wireless networks in an intelligent application platform that can improve the efficiency of the organization's functional processes, as represented in Figure 6.

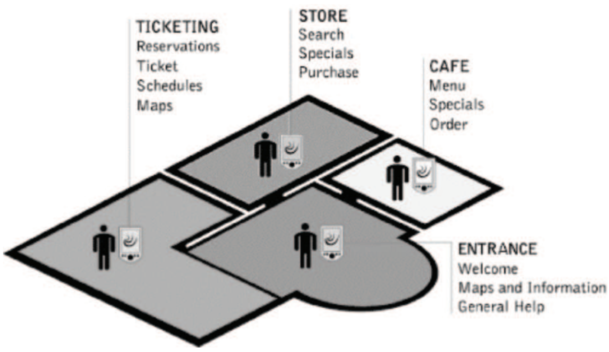


*Figure 6: Example of PanGO utitlity, using a system implemented on a mobile PDA [PAN01]*

**Operation**

IATS registers and monitors the movement of people and asset by evaluating the combination between RFID event-driven location and RSSI based Wi-Fi location. The values are sent to the control platform that possesses a database allowing the user to treat the location information for each device. The system demands that the tag must be in coverage of at least 3 Access Points with a minimum power of 32nW.

| | |
|---|---|
| **Range** | 50 m |
| **Precision** | Room |
| **Technology** | Wi-Fi and RFID |
| **Purpose** | People and Asset Tracking |
| **Commercial** | Yes |

*Table 6: PanGO IATS features*

The system currently provides the ability to track the assets in customized Maps. Also allows individual tracking of a specific asset, displaying also its characteristic in an intuitive way. PANGO also allows alarms to the system, such as noticing the system manager when an asset is in a determined place for too much time, or when it enters a restricted area.

## 2.7. Aeroscout

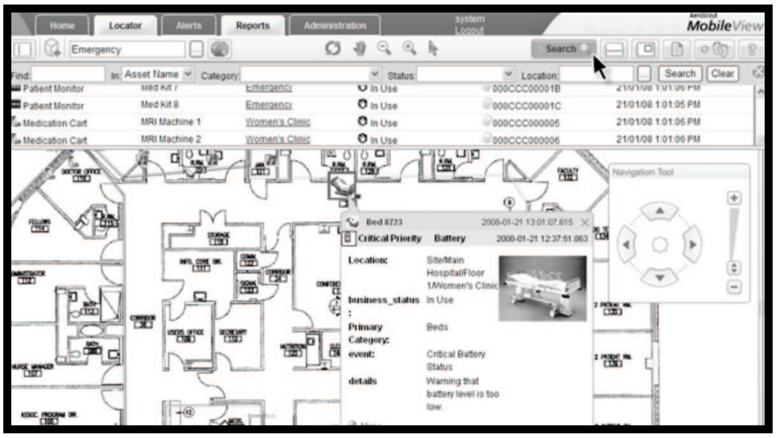


*Figure 7: Mobile View Application PrintScreen [AER01]*

**Overview**

AeroScout uses a standard Wi-Fi network as a single wireless architecture providing asset tracking, sensing and monitoring, inventory management and auto-ID [AER01]. This system is a commercial solution and allows a company to track inventory in a yard, a manufacturer can try to streamline production processes, or a hospital can plan to track and improve medical equipment usage.

Aeroscout is used, for example, in Healthcare to strengthen the equipment management and utilization capabilities. Also, it offers increased safety, patient throughput and unified data for analysis and reporting through Wi-Fi and RFID, providing a single source of data and reduced expenses. Aeroscout can also be applied in other areas to improve business logistics that can improve the efficiency of a process, providing a Return Of Investment (ROI) to companies that install this system. This ROI can easily justify the system implementation costs.

**Operation**

AeroScout's Wi-Fi-based Active RFID tags or standard Wi-Fi devices send a brief signal at a regular interval, adding status or sensor data if appropriate. This signal is sent to standard wireless Access Points or to AeroScout Location Receivers, without any infrastructure changes needed, and is fed to a processing engine. The engine uses signal strength and/or time of arrival algorithms to determine location coordinates, and sends location and status data to AeroScout MobileView, captured in Figure 7. The MobileView system merges RTLS and sensor data with information from other technologies and pre-existing asset data, to provide a valuable context. The MobileView interface provides a single platform for all visibility data, to display maps, enable searches, automate alerts, manage assets, work with third party applications, etc.

| | |
|---|---|
| **Range** | n.d. |
| **Precision** | Room |
| **Technology** | Wi-Fi, RFID and others. |
| **Purpose** | Healthcare and Industrial applications using location |
| **Commercial** | Yes |

*Table 7: Aeroscout's features*

## 2.8. 60 GHz OFDM

**Overview**

60 GHz OFDM is an Indoor Wireless Location System protocol based on micro-waves (60 GHz) [WIN01]. The prototype was designed to provide a low cost implementation by Humboldt University Berlin in a partnership with Innovations for High Performance microelectronics (IHP), in Frankfurt.

In this solution modulation techniques such as Orthogonal Frequency Division Multiplexing (OFDM) are applied to the transmitted signals turning this system into a high performance system.

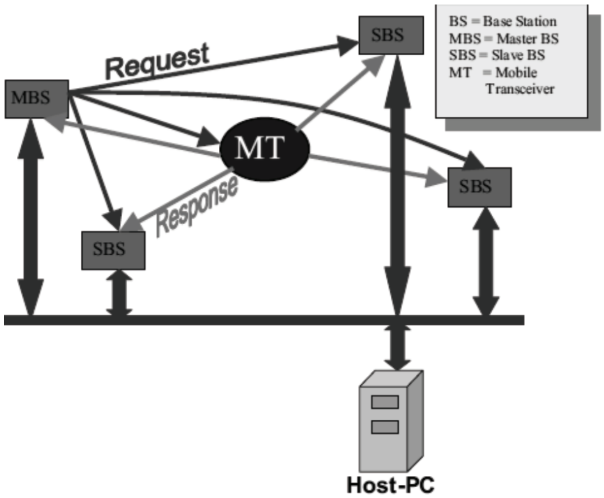


*Figure 8: 60GHz OFDM Location Protocol [WIN01]*

**Operation**

The system is composed by a control PC where the calculations and the information registry are made and by several slaves fixed on the implementation site. The location is based on Different Times of Arrival (DTOA).

As it can be seen in Figure 8, the Master Base Station (MBS) sends a request with a Pseudo Noise (PN) sequence that is recognized by the network. The Mobile Transceiver (MT) and the Slave Base Station (SBS) capture this signal. Each MT, by receiving the PN sequence will produce a response time. When the Base Station (BS) captures the response transmitted by the MT, it calculates the time difference between the request and the response requests and sends this value directly to the PC. The PC knowing the position of every BS can calculate the approximate position of the MT device.

| | |
|---|---|
| **Range** | 10 m |
| **Precision** | 1 m |
| **Technology** | 60GHz Microwaves |
| **Purpose** | Fine Location |
| **Commercial** | No |

*Table 8: 60GHz OFDM features*

## 2.9. Active Badge Location System

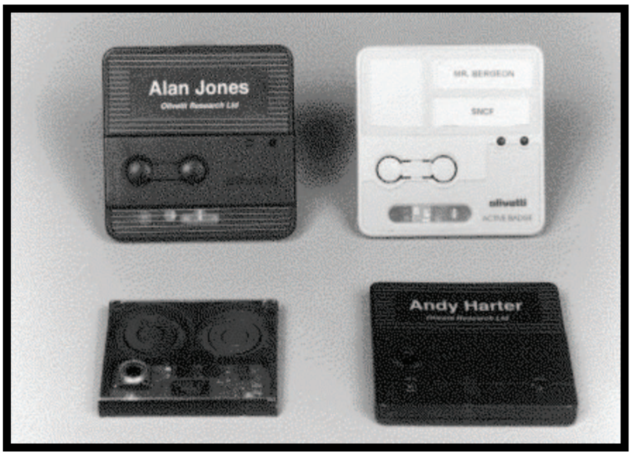


*Figure 9: Photo of an Active Badge [FRA01]*

**Overview**

Active Badge (AB) was one of the pioneer systems concerning wireless location. Some of the papers concerning this article date from 1989 [WAN01]. Although this system is limited in its performance when compared to the other referred systems, AB was an innovative system in its time. The Active Badge Location System is an ILS based on infrared technology. The prototype created by Olivetti Research Ltd., in Cambridge, is composed by Infra Red (IR) solid state sensors installed on the walls and ceilings of a room, floor or building, creating a sensor network encompassing small portable devices called Active Badges, that send IR codes to a Network Server that consists on a group of workstations. The AB shown in Figure 9, does not calculate distances or interprets measurable characteristics of the transmitted signals, the location consists only on identifying the receivers that capture the signals in a determined AB. Each workstation can manage a maximum number of four Active Badges. These terminals communicate between themselves through an Ethernet network, the same that unites them to the database.

**Operation**

Each AB sends periodically a signal whose is frequency is modulated with the badge ID. To minimize power consumption, the transmission periods are large, usually 15 seconds [FRA01]. The sensors receive the signal and create elements in a database that associates the ID to a position and time. Every workstation tests if each sensor possesses information to be distributed or collected. This data is processed by the system to create a history of the current badges in the system. The system efficiency depends on the number on collisions in the system between adjacent badges.

To be located, every AB has to be near the sensors that will detect it. As ABs are based on IR technology, if a sensor is placed for example in a pocket it probably won't be detected by the system. The signals do no cross walls, which minimizes the existence of ambiguous paths to the transmissions. IR sensors have low power consumption and have a low cost when compared to other similar technologies.

| | |
|---|---|
| **Range** | 5 m |
| **Precision** | Checkpoints |
| **Technology** | IR |
| **Purpose** | Locate people by room / checkpoints |
| **Commercial** | No |

*Table 9: Active Badge System*

# 2.10.Zebra

*Figure 10: Some Zebra Implemented Projects[ZEB01]*

**Overview**

The Zebra system uses a wide range of scalable RTLS technologies to generate accurate information about the physical location and status of assets (Figure 10). Zebra can apply to ISO/IEC 24730, Cisco CCX Wi-Fi, precision GPS and UWB Technologies and provides a combination of hardware and software solutions that can work together to provide, for example, an asset tracking visibility from high resolution X-Y-Z coordinate positioning to basic presence detection [ZEB01].

Configuration tools allow to easily program various RTLS parameters into asset tags, exciters and infrastructure products and the Middleware offers a rich set of software development tools for integrating the hardware, middleware applications and software applications with third-party applications.

**Operation**

Zebra encompasses a set of Tags, Exciters and Infrastructures to receive tag transmissions and forward the information to the Visibility Server Software, the application that manages the whole location process. This software combines accurate position calculation with database, configuration and system management functions.

As zebra is relatively recent commercial solution it wasn't possible to obtain more details about the system architecture.

| | |
|---|---|
| **Range** | variable |
| **Precision** | variable |
| **Technology** | ISO/IEC 24730, Cisco CCX Wi-Fi, precision GPS and UWB Technologies |
| **Purpose** | Indoor and Outdoor Location and Presence detection |
| **Commercial** | Yes |

*Table 10: Zebra features*

# 2.11.nanoLOC

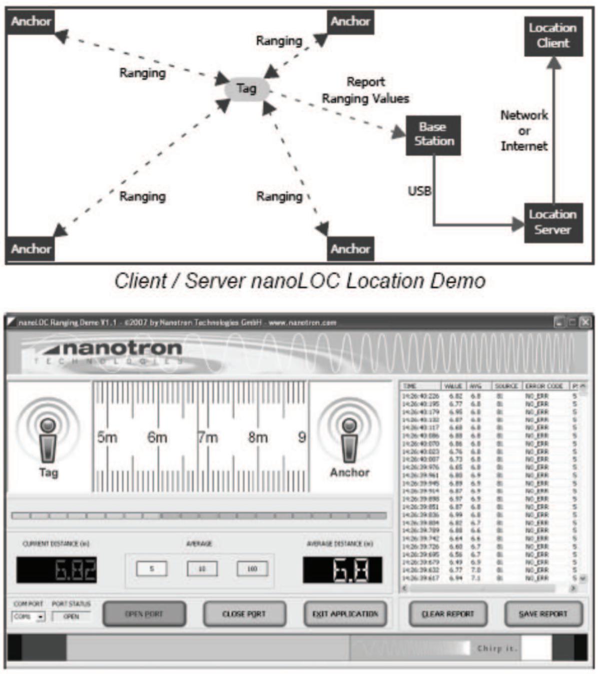


*Figure 11: Client-Server nano-LOC Location Demo [NAN01]*

**Overview**

Nanoloc is a system developed by Nanotron that combines high speed data transmission with location information at about 1 meter accuracy with the guarantee of high signal robustness and low energy consumption. This system provides robust wireless technology for RTLS and Location-Aware WSN solutions for applications like Medical Applications, Logistics and Healthcare. The system can be delivered in form of a development kit with a set of easy-to-use tools for developing a RTLS like the ones shown in Figure 11.

**Operation**

The 2.4 GHz ISM nanoLOC TRX Transceiver combines wireless communication and location in one solution. This module can measure the link distance between two nodes with low-cost crystal oscillators as they communicate using a technology called SDS-TWR - Symmetrical Double Sided - Two Way Ranging.

Data rates are selectable from 2 Mbps to 125 kbps.

Due to the chip's unique chirp pulse, adjustment of the antenna is not critical. This significantly simplifies the system's installation and maintenance.

| | |
|---|---|
| **Range** | 20 |
| **Precision** | 1 m |
| **Technology** | IEEE 802.15.4/Zigbee adapted |
| **Purpose:** | Monitoring the course of a person/object in a room. |
| **Commercial Solution** | Yes |

*Table 11: NanoLOC features*

# 2.12. Lopes

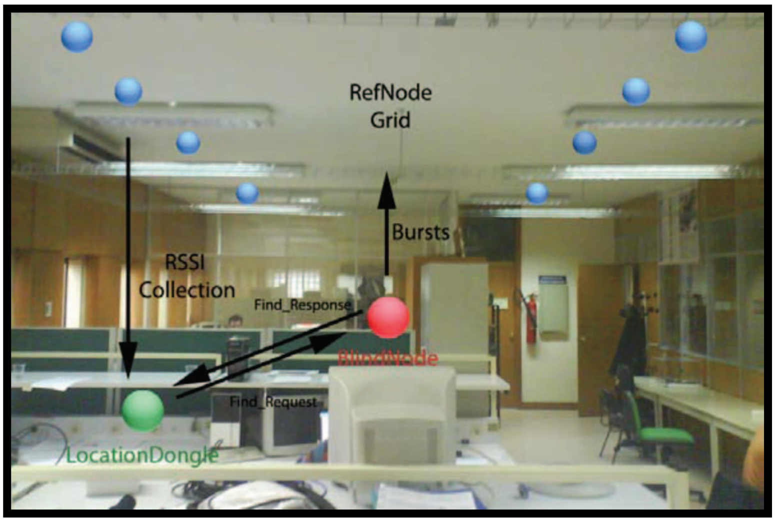


*Figure 12: Lopes System Communications scheme [JOL01]*

**Overview**

LOcalização de PESsoas (LOPES) is an IEEE 802.15.4/Zigbee based Indoor Location System that is being developed at University of Aveiro. The project was financed by TeleSAL and is being developed by Telecommunications Institute of Aveiro, Micro I/O, HFA and Maisis.

The system was initially designed to monitor a visitor's course through a museum. By doing so, the Museum's Manager could study the organization of the museum and then, p. e. reorganize the positioning of the pieces so that only the interesting ones are visited. It would also be possible to treat statistically the number of visits for each peace, and determine the most probable way that a visitor would follow.

A demo version of the actual system is currently in exhibition at Fábrica da Ciência Viva, Aveiro.

**Operation**

In the LOPES project, the location is based on the Received Signal Strength Indication (RSSI) value of a burst emitted by a TAG, the mobile device being located. The IEEE 802.15.4/Zigbee signatures of the burst packets are collected by the RefNodes on the ceiling of the room and forwarded to a master device as displayed in Figure 12, that is called Master Gateway Unit (MGU), and are then sent via RS232 to a Java Application running on a computer [JAF01]. The collected RSSI signatures are then inserted in a previously calibrated Neural Network that estimates, through non-linear calculations, the location of the current active TAGs in the system [JOL01]. The Neural Network is calibrated and adjusted to each room in which the system is placed.

| | |
|---|---|
| **Range** | 4m from the REF mesh |
| **Precision** | < 1 m |
| **Technology** | IEEE 802.15.4/Zigbee |
| **Purpose** | Monitoring the course of a person/object in a room. |
| **Commercial** | No |

*Table 12: LOPES features*

## 2.13. Conclusion

Some of the most representative Indoor Location Systems were approached on this survey. Indoor Wireless Location is an emerging technology that is currently under research and development in a broad range of areas. Many of the studied solutions are based on Wi-Fi Networks. Wi-Fi is currently the most diffused type of wireless network. The application of this kind of system to these networks is therefore cheaper because it avoids the acquisition of new hardware. Though, Wi-Fi Networks have a very reduced development potential for this kind of systems especially due to its physical properties.

Wi-Fi modules also present very high power consumption rates, when compared to many other wireless protocols, which reduces the autonomy of TAGs. Most location technologies combine with Wi-Fi networks to RFID to reduce consumption and to improve reliability of the system [MAN01]. Other types of networks are emerging like IEEE 802.15.4 / Zigbee [ZIG01] that are able to provide minor consumptions and better accuracy to location systems [NOH01].

IEEE 802.15.4 was designed for control applications like domestic applications as controlling lights or windows to industrial applications like monitoring remote sensors. On IEEE 802.15.4, the main characteristics are Low Power, Low data rate (for control applications), short range, low cost fully functional network protocol transceivers, small memory footprint and simplicity.

The next chapter is a short overview on IEEE 802.15.4.

# Chapter 3

# Overview on IEEE 802.15.4

In the last decade, the proliferation of the wireless communication networks has reached unprecedented values. Various types of wireless networks were studied and developed to many different purposes, encompassing the most diverse characteristics. The evolution and specialization of wireless technologies became possible to expand the number (and type) of applications enabled through wireless communications by avoiding the installation of cables for example, enabling wider and more flexible systems. Due to this evolution, the control applications based on wireless sensor networks started to spread [KOU01]. An example of an emerging Wireless Network Protocol that is focused on control applications is IEEE 802.15.4.

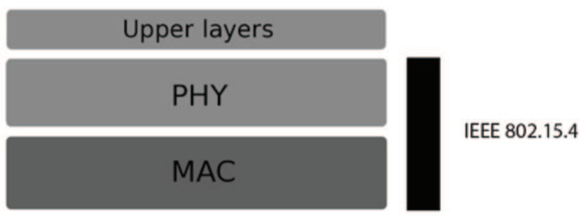


*Figure 13: IEEE 802.15.4 Stack*

The IEEE 802.15.4 protocol is divided in two different layers, represented in Figure 13. The PHYsical layer (PHY) and the Medium Access Control (MAC). All the upper layers are defined in other protocols that complement IEEE 802.15.4's functionalities, such as Zigbee, from the Zigbee Alliance or MiWi, from Microchip. These protocols have different features that fit diverse purposes and applications. These features will be approached further in the article.

## 3.1. The PHY layer

In IEEE 802.15.4, the physical layer is responsible for specifying the radio channel in use and the modulation and spreading techniques. The IEEE 802.15.4 operates in three frequency bands: 2.4 GHz, 868 MHz and 915 MHz. There is a single channel between 868 and 868.6 MHz, 10 channels between 902 and 928 MHz separated by leaps of 2 MHz, and 16 channels between 2.4 and 2.4835 GHz in leaps of 5 MHz [ANI01].

The scheme on Figure 14 illustrates the channel distribution in the RF spectrum.

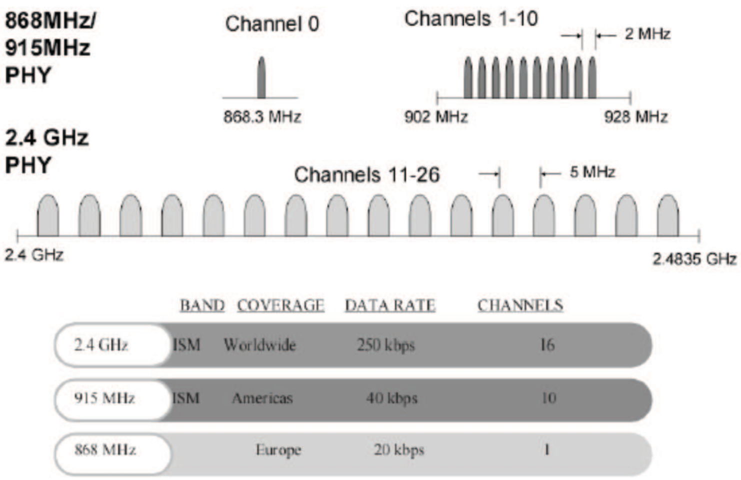


*Figure 14: Available channels for transmission specified in the PHY layer*

The data rates are 250 kbps (at 2.4 GHz), 40 kbps (at 915 MHZ) and 20 kbps (at 868 MHz). Lower frequencies are more suitable for longer transmission ranges due to lower propagation losses, though only one of the channels is completely immune to Wi-Fi, the channel 26 (0x1A). The bands were chosen based on Direct Sequence Spread Spectrum (DSSS) spreading technique. The protocol implements a dynamic channel selection algorithm, for scanning the supported channels by detecting the energy rates in each channel.

## 3.1.1. PHY Layer Features

The physical layer is responsible for the (de)activation of the transceiver. A radio transceiver has 3 available states: *transmitting*, *receiving* or *sleeping*. Upon the request of the MAC sub-layer, the radio can be turned ON or OFF. The turnaround time from transmitting to receiving and vice versa should be no more than the time for transmitting 48 bits.

As for Channel Frequency Selection, IEEE 802.15.4 defines 27 different wireless channels from which one can be chosen to host a PAN. So, an upper layer must be able to specify the channel to be used.

When considering data transmission the Physical Layer is responsible for the physical transmission of the data. The transceiver must be capable of a transmission power of 1 mW, with a center frequency tolerance of 40 ppm. The packet error rate must be < 1 % with a receiver sensitivity of -85 dBm @ 2.4 GHz band and -92 dBm @ 868/915 MHz band.

IEEE 802.15.4 provides Energy Detection (ED) as the transceiver must be able to calculate an estimate of the RSSI of a given channel. This task does not make any signal identification or decoding on the channel. The ED time should be equal to the time of transmitting 32 bits.

The Link Quality Indicator (LQI) measurement characterizes the Strength or Quality of a received packet. It measures the quality of a received signal on a link. This measurement may be implemented using receiver ED, a signal to noise estimation or a combination of both techniques [ANI01]. Upper layers can use the LQI value, and is very useful in many applications, such as Location Systems based on RSSI.

Clear Channel Assessment (CCA) allows the transceiver to detect if the medium is available for a transmission. Without this algorithm it would be impossible to communicate through wireless sensors because signals from different transceivers emitting at the same time would collide. There are two possible modes for detection. The first is the Energy Detection mode that reports that the medium is busy if the detected energy is above a threshold. The other mode is the Carrier Sense mode that detects that the medium is ccupied only if a signal with the modulation and the spreading characteristics of IEEE 802.15.4 is detected no matter if it is higher or lower than the ED threshold. Though IEEE 802.15.4 allows a combination of both the referred techniques. CCA reports that the medium is busy only when it detects a signal with the modulation and the spreading characteristics of IEEE 802.15.4 and with energy above the ED threshold.

## 3.1.2. The PHY Packet Field

The PHY layer has, as every other layer, an associated header with specified fields. A PHY packet is composed by a maximum of 133 bits.

Here is a brief description of the PHY Packet Structure, represented in Figure 15.

The Preamble consists on 32 bits that synchronize the packet. The Start of Packet Delimiter (SPD) is an 8 bit sequence that indicate the end of the Preamble. The PHY header consists on 8 bits that indicate the length of the PHY Service Data Unit (PSDU) whose size is ≤ 127 bytes.


*Figure 15: IEEE 802.15.4 PHY frame*

## 3.2. The MAC layer

The IEEE 802.15.4 MAC layer is the layer responsible for establishing an interface between the physical layer and the upper protocol layers.

The MAC layer enables two operational modes that can be selected by the coordinator. One is the *Beacon-enabled* mode: the coordinator to synchronize attached devices and to identify the PAN periodically generates beacons. The other is *Non Beacon-enabled* mode: in non beacon-enabled mode, the devices can simply send their data by using unslotted CSMA-CA. There is no use of a superframe structure in this mode.

This layer is also responsible for generating network beacons if the device is a coordinator, synchronizing the beacons, supporting PAN association and disassociation, supporting device security, applying the CSMA-CA mechanism for channel access, handling and maintaining the Guaranteed Time Slot (GTS) mechanism and for providing a reliable link between two peer MAC entities.

MAC implementation on IEEE 802.15.4 has its basis on the following directives: the system must be extremely low-cost, ease of implementation; the data transfer must be reliable with very low power consumption.

The MAC layer specifies two device classes: the Full Function Device (FFD) and the Reduced Function Device (RFD). The FFDs are able to operate in every network topology. They are also capable of being Network Coordinators and communicating to any other device of its network. As for RFDs, they are limited to a star topology. A RFD cannot be turned into a network coordinator, only talks to FFD and must be easily implemented.

All the devices in a network, whether FFD or RFD have a certified IEEE address, and can be specified a configurable short address to be used in the PAN.

## 3.2.1. Network Topologies

There are three available network topologies. The main two are Star and Mesh topology, though, Mesh derives in two different topologies: a point-to-point and a cluster tree technology. The upper protocols are able to define different types of network topologies.

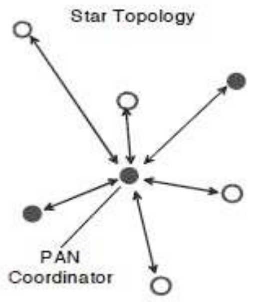


*Figure 16: Star Topology Scheme*

In a Star topology Every RFD or FFD connects directly to the PAN Coordinator and does not establish connections with other devices as shown in Figure 16. In a Mesh topology there are two topologies that derive from this base connections model displayed in Figure 17. These are the point-to-point and the Cluster Tree topology.

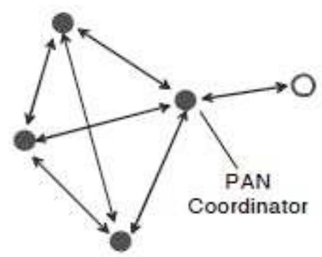


*Figure 17: Mesh topology Scheme*

Point-to-point is only available for FFD. FFD communicate to each other, despite of the common alternative connections between them. Cluster Tree, is also only available for FFD: Devices communicate through a tree mechanism, greatly reducing the number of connections between devices in the network.

## 3.2.2. MAC Packet Data Structure

The MAC Packages are devided into three main parts, the MAC Header (MHR), the MAC Service Data Unit (MSDU) (referred in the following Tables as “MAC Payload”) and the MAC Footer (MFR).

There are 4 types of MAC Data Frames. The 4 types of frames are described by Table 13 to Table 16.

**General MAC Frame**

*Table 13: General MAC Frame Size specifications*

| **Octets** | 2 | 1 | 0/2 | 0/2/8 | 0/2 | 0/2/8 | variable | 2 |
|---|---|---|---|---|---|---|---|---|
| **Field** | Frame control | Sequence Number | Dest. PAN ID | Dest. Add | Source PAN ID | Source Add | Frame Payload | Frame Check Sequence |
| | MAC header | | | | | | MAC Payload | MAC footer |

**Beacon Frame**

*Table 14: Beacon Frame Size Specifications*

| **Octets** | 2 | 1 | 4 or 10 | 2 | variable | variable | variable | 2 |
|---|---|---|---|---|---|---|---|---|
| **Field** | Frame control | Beacon Sequence Number | Source address info. | Superframe specification | GTS | Pending address | Beacon Payload | Frame Check Sequence |
| | MAC Header | | | MAC Payload | | | | MAC footer |

**Data Frame**

*Table 15: Data Frame Size Specifications*

| Octets | 2 | 1 | 4 to 20 | Variable | 2 |
|---|---|---|---|---|---|
| Field | Frame control | Data Sequence Number | Add. Info. | Data Payload | Frame Check Sequence |
| | MAC header | | | MAC payload | MAC Footer |

**Acknowledgement Frame**

*Table 16: Acknowledgement Frame Size Specifications*

| Octets | 2 | 1 | 2 |
|---|---|---|---|
| Field | Frame control | Data Sequence Number | Frame Check Sequence |
| | MAC header | | MAC Footer |

**MAC Command Frame**

*Table 17: Command Frame Size Specifications*

| Octets | 2 | 1 | 4 to 20 | 1 | variable | 2 |
|---|---|---|---|---|---|---|
| Field | Frame control | Data Sequence Number | Add. Info. | Command Type | Command Payload | Frame Check sequence |
| | MAC header | | | MAC payload | | MAC footer |

The size of the MAC packages varies according to the type of package and data to be transmitted.

## 3.2.3. Functional description: Channel Access

The IEEE 802.15.4 MAC layer describes two channel access mechanisms, the Contention Free and the Contention Based. The Contention Based allows the device to access the channel in a distributed way, using a CSMA-CA backoff algorithm. The Contention Free access is entirely controlled by the PAN coordinator using Guaranteed Time Slots (GTS).

**The Superframe Structure**

On a PAN, the coordinator can optionally bound its channel time using the superframe structure represented in Figure 18. A superframe is bounded by the transmission of a beacon frame and can have an active portion and an inactive portion.

The MAC layer has to ensure the integrity of the superframe timing. This can be done by operations similar to compensating the clock drift error.

If a PAN chooses not to employ a superframe structure, its coordinator should not transmit beacons, except upon receipt of a beacon request command. All transmissions, except the acknowledgement frames or any other data frames that quickly follows the acknowledgement of a data request command, should use the CSMA-CA algorithm to transmit. In this case, GTS will not be permitted.

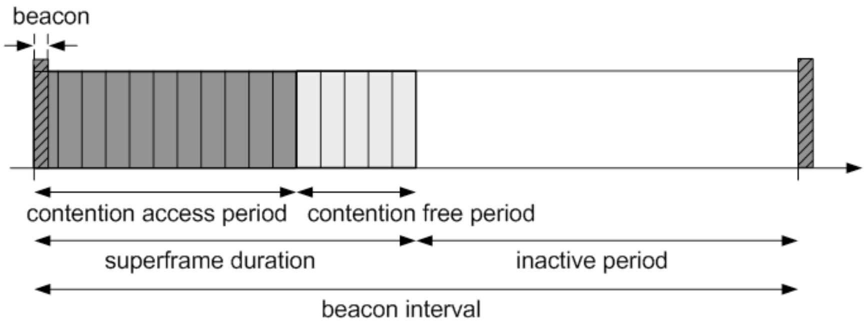


*Figure 18: Superframe Structure Scheme*

The Contention Access Period (CAP) starts immediately after the beacon and must be complete before the beginning of the Contention Free Period (CFP). If the CFP is zero length, the CAP must be complete at the end of the active portion of the superframe. A device transmitting within the CAP must ensure that its transaction is complete. MAC command frames must always be transmitted in the CAP. The CFP must start immediately at the end of the CAP, and it must end inside the active portion of the superframe. If the PAN coordinator has allocated any GTS, they must be located within the CFP and occupy contiguous slots. The CFP grows or shrinks depending on the total length of all combined GTSs.

**Superframe timing**

When a PAN is beacon enabled, a coordinator that is not the PAN coordinator must maintain the timing of the beacons. The relative timing between incoming and outgoing beacons can be software defined and is structured as we can see in Figure 19.

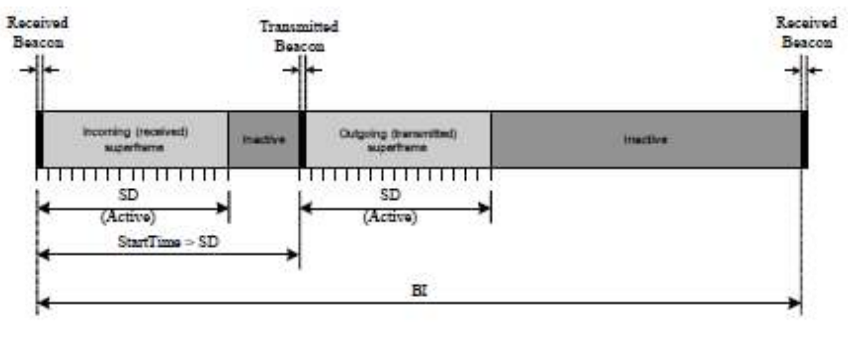


*Figure 19: Example of Synchronization between incoming and outgoing messages*

**InterFrame Spacing (IFS)**

The MAC layer needs a finite amount of time to process the data received by the PHY layer. Two consecutive frames must be separated by at least one IFS period. The time of the IFS period depends on the size of the frame that has just been transmitted.

**The CSMA-CA algorithm**

The Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) algorithm can be used to transmit data that can be, for example, quickly transmitted following the acknowledgement of a data request command. If the PAN is beacon enabled, this algorithm cannot be used. The MAC layer uses a slotted version of CSMA-CA of transmissions in the CAP of the superframe. If the PAN is not beacon enabled, or if for some reason a beacon could not be located on a beacon enabled PAN, the MAC sublayer uses the Unslotted CSMA-CA. In both cases the algorithm is implemented using time units called backoff periods.

The backoff period boundaries must be aligned with the superframe slot boundaries of the PAN coordinator. The first backoff period for every device must be aligned with the start of the beacon transmission. In slotted CSMA-CA the MAC layer must ensure that the PHY starts all of its transmissions on the boundary of a backoff period. In a PAN, the backoff period of a device is not related to the backoff period of any other device in the PAN.

Each device has to sustain three variables for each transmission attempt:

- Backoff Exponent (BE), a relation that determines how many backoff periods a devise must wait before attempting to assess a channel.
- NB: The number of times that CSMA-CA was required to backoff.
- Contention Window Length (CW), the number of contention periods that a channel must be free of activity before the transmission starts.

Although the receiver of the device is enabled during the CCA analysis portion of the algorithm, the device may discard any frames received during this time.

## 3.3. Zigbee vs MiWi

Zigbee and Miwi are both Protocol Stacks based on IEEE 801.15.4. Zigbee is a worldwide Public Standard for large, robust, highly secured and low power networks. This protocol allows Advanced Encription Standards (AES) and a set of network topologies (star, tree and mesh) to support a wide variety of applications. This standard makes interoperability possible, certifying that the devices that use the Zigbee specification are compatible with each other. Zigbee can also co-exist with other wireless technologies such as Bluetooth, Wi-Fi and GSM.

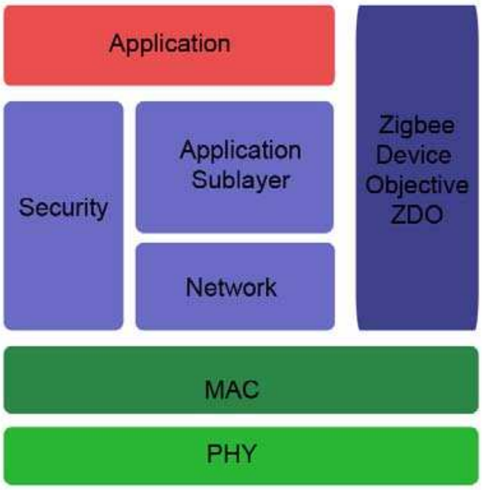


*Figure 20: Zigbee Protocol Stack*

Zigbee is nowadays a standard that has been available and matured over the years. The entity that regulates and checks the protocol compliance is the "Zigbee Alliance". The Zigbee Protocol Stack can be seen in Figure 20. The MAC and PHY layers in the Figures refer to IEEE 802.15.4. The other layers consist on the main Zigbee Layers specified by the Zigbee stack.

The Zigbee Alliance conducted a survey in Japan and found that only 9% of the networks require the usage of more that 100 nodes in a PAN. This study also concluded that the early adopters of the system are using the MiWi stack [MOR01].

MiWi is a Proprietary Wireless Standard also based on IEEE 802.15.4 that was designed and maintained by Microchip.

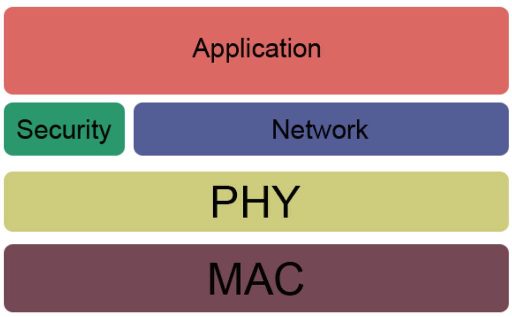


*Figure 21: MiWi Protocol Stack*

The MiWi stack in Figure 21 was specially designed for Small Proprietary Networks, with a maximum of 1024 nodes and 4 hops per package. It has a small footprint, a low overhead and it does not require Fees or certification unlike Zigbee. This protocol was designed to be simple and cheap providing a low cost development and implementation. Its simplicity provides better performance and better control over the device behaviour.

## 3.4. Conclusion

IEEE 802.15.4 is a standard having a great development potential.

Wireless Control is in great development and the variety of solutions provided by this type of systems has many application possibilities.

LOPES system is a project that uses IEEE 802.15.4 to provide Wireless Location. This project will be approached in the following chapter.

# Chapter 4

# Overview on LOPES Project

LOPES – LOcalização de PESsoas is an IEEE 802.15.4 / Zigbee based Wireless Location System developed by Instituto de Telecomunicações de Aveiro, allied to three companies: MicroI/O, Maisys e HFA. Currently, LOPES enmpasses two implemented demonstrators, one in Telecommunications Institute of Aveiro and the other in “Fábrica da Ciência Viva”, a science museum established in Aveiro. The location system was designed for calculating the estimate position of a given mobile sensor belonging to the network in study. The initial objective was to achieve a system with a precision minor to 1,5 m. The time for location of one sensor must be the shortest possible. The location system must receive as an input the RSSI and the ID of a given mobile node. The location system outputs the XY coordinates and the ID of the sensor and the data must be sent through a TCP socket to an intended PC in the Network.

The system deployed in “Fábrica da Ciênica Viva” is characterized by a location precision of, approximately, 1,5 m and a time granularity of 2 seconds per location. The system is in continuous improvement and development. A general description of the LOPES system follows.

## 4.1. Hardware

In the development of sensors for wireless networks, the main goal is to produce small, autonomous and cheap devices since low cost and small size are crucial features when considering mobile and embebbed systems.

The Texas Instruments CC2431ZDK platform from Texas Instrument, developed by Chipcon [CHI01], was chosen to be used in this project because it fulfils the requirements for the target application. The kit encompasses a proprietary Zigbee Stack and a set of sample applications and manuals that explain its usage [CHI02]. Table 18 describes the three main components of the development used to implement the FTT-L Protocol. These components are briefly discussed in the following subsections.

| **Device** | **Description** |
|---|---|
| **CC2430/31** | *SoC - System on Chip* solution used in low power sensor networks.<br>Includes a microcontroller, a radio transceiver and embebbed location system functionalities. |
| **SOC_BB** | Battery fed docking station for CC2430/31 |
| **SmartRF04EB** | Development Board |

*Table 18: Sensors used in Lopes project.*

*Figure 22: Chipcon Development Kit for Texas Instruments CC2430 content [CHI03]*

## 4.1.1. SoC CC2430/31

The SoC CC2430/31 is composed by a CC2420 Radio transceiver, a Low Power uC 8051 and wireless programming support [CHI03]. The 8051 MCU is a Low Power micro-controller that runs the Z-Stack. The CC2430/31 can be configured to communicate with other network elements and includes a configurable ADC with configurable resolutions and 2 USART's supporting different serial protocols.

| | CC2430/31 | Obs. |
|---|---|---|
| **Flash Memory** | 128k bytes | |
| **Consumption** | 10.5 mA | Normal |
| | < 190μA | Sleep Mode |
| **Range** | 200m | Line of sight |
| **Indicators** | 1 LEDs | |

*Table 19: Some SoC CC2430/31 Technical features*

### 4.1.2. SmartRF04EB

The SmartRF04EB shown in Figure 23 is a board that provides a serial interface between the SoC CC2430/31 and the PC. By inserting a module into this board it is possible to debug and program it using a serial port from a connected PC.

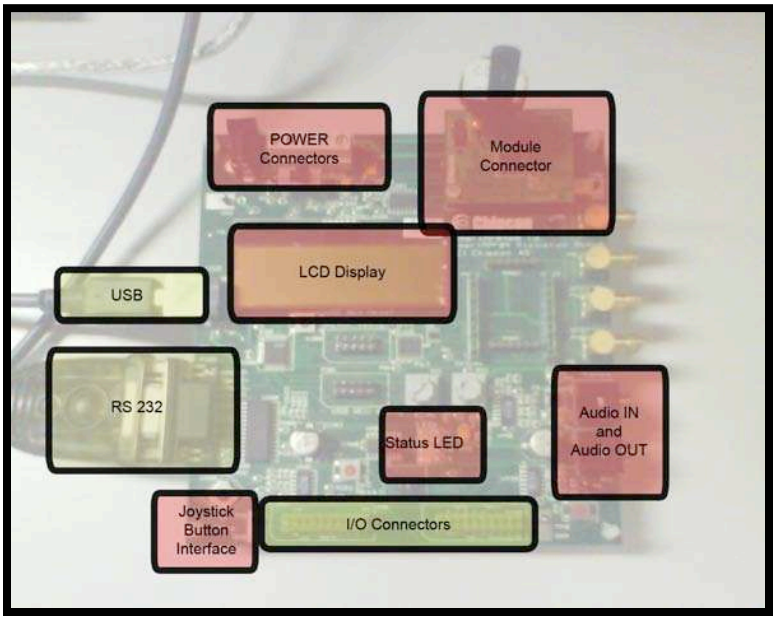


*Figure 23: SmartRF04EB most important interfaces*

As depicted in Figure 23, the SmartRF04EB provides various interfaces, namely an USB and RS232 digital I/Os, status LEDs and an LCD Display to inform about the state of the module. Besides it also includes an audio IN and an Audio OUT for transmitting audio applications.

## 4.2. Software

The LOPES project high-level software is based on a Java platform. The software/firmware compilation, debugging and installation to the sensors was conducted using the USB Port, with IAR Embebbed Workbench 8051 from IAR Systems.

### 4.2.1. The Z-Stack

The Texas Instruments, is a Zigbee Proprietary Protocol Stack made for CC2430/31 sensors. It supports several platform types and is available for download at the Texas Instruments web page [TEX01]. Z-Stack implements all the required layers as well as the corresponding interfaces so that the programmer can only be concerned the system application. As it will be seen further ahead, the Z-Stack allows a wide range of possible operations and applications, though it's complexity has repercussions in the performance level.

### 4.2.2. Java Engine

The LOPES application is based on a set of Java based applications. The two more important applications are the SerialForwarder and the Graphical User Interface application [JOL01]. The SerialForwarder, shown in Figure 24, is a crucial application to forward the data collected by the networks MGU - Master Gateway Unit, through the Serial Port of the PC to the Java Application that is being executed at the system's Host.

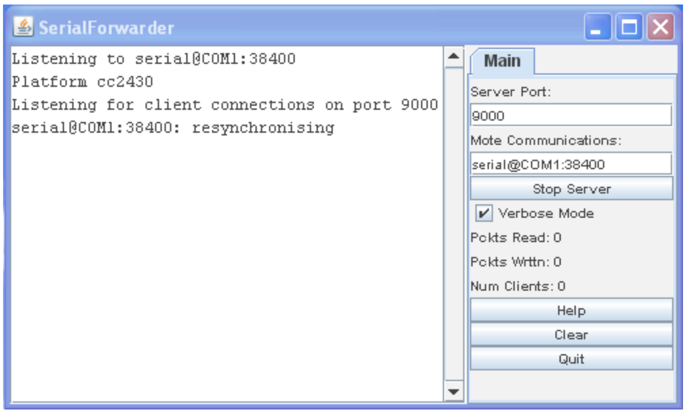


*Figure 24: SerialForwarder Graphical User Interface*

The SerialForwarder creates a TCP server in a user-defined port and sends all the received data from the sensor network to all the TCP clients that are connected.

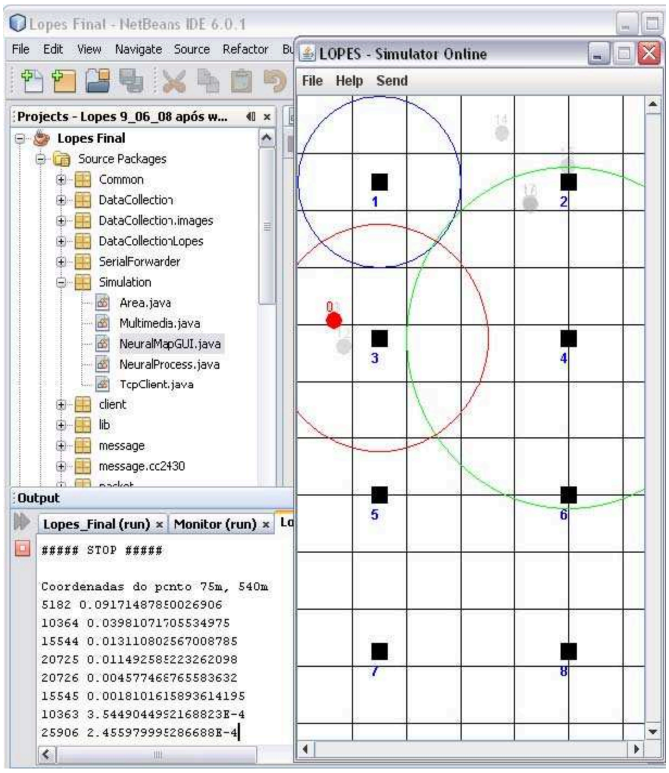


*Figure 25: The LOPES current Graphical User Interface*

The system includes a Java Interface that displays the approximate position of the mobile nodes that are being located. This interface can be seen in Figure 25 where a ceiling perspective is displayed. The eight black squares in the “LOPES – Simulator Online” window represent the 8 RefNodes. They represent a grid of Reference Nodes distributed on the Ceiling of the Laboratory. The three circles around the squares represent the RefNodes that have received the largest RSSI values from the mobile nodes spread through the room so called BlindNodes. The smallest circle represents the Greater RSSI value. The dot in the same window represents the estimate location of a BlindNode. The grey dots represent the previous locations determined by the system.

When the location system determines that the BlindNode is within a determined area, it sends a TCP/IP message to another computer in the same Ethernet Network that is running the MultimediaClient application. This computer receives this message that starts a pre-defined video sequence in the corresponding display.

## 4.3. The Demonstrators

The LOPES Project has deployed two demonstrators: one in Telecomunications Institute of Aveiro, Portugal and the other in “Fábrica da Ciência Viva” also in Aveiro. The following subsections describe them.

### 4.3.1. "Instituto de Telecomunicações", Aveiro

The first version of the LOPES demonstrator was installed at the RF Laboratory of the Telecommunications Institute in Aveiro. Figure 32 shows a scheme of the package exchange system.

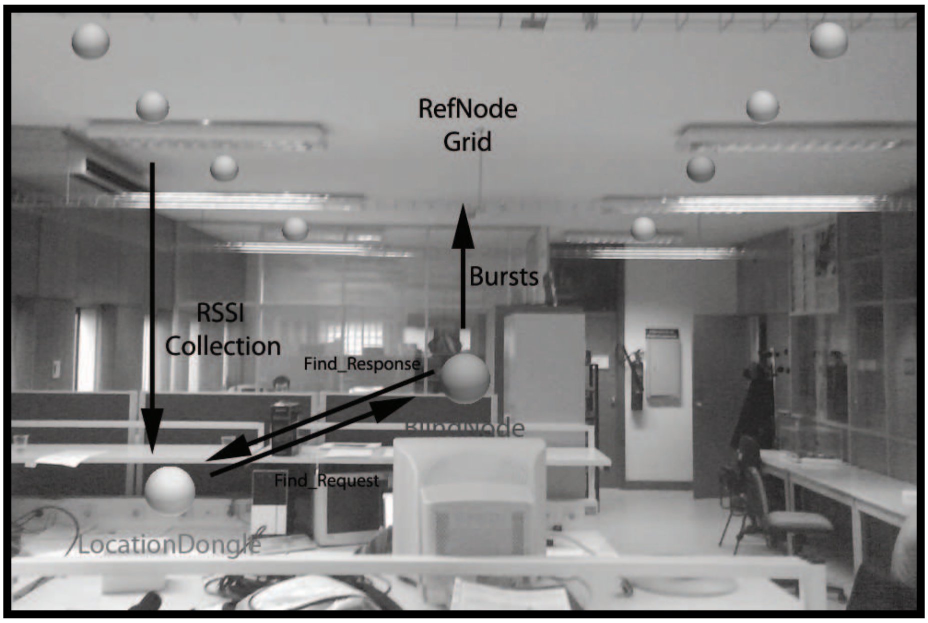


*Figure 26: Simplified Scheme of the package exchange applied to the demonstrators*

A computer server installed in one of the desks is directly connected to the LocationDongle represented by the green sphere on Figure 26. The red sphere, and the 8 blue spheres represent the RefNode Grid applied to the ceiling of the room represent a Mobile node, the BlindNode. Two other computers are placed in the room in a position that can be configured on the system, that will display to the visitor a video as the visitor with a BlindNode attached approaches the PC's area. In the year 2008, a first version of the LOPES demonstrator was deployed at Aveiro's Telecommunications Institute, more precisely at the RF Laboratory. The demonstrator implemented a MAC application, a Location System and a Multimedia System for demonstration. In the wireless sensor networks, all the necessary RSSI messages to calculate the node's position are exchanged between all the devices and the LocationDongle forwards the collected values to the Location Manager and Application Engine of the Sensor Network

In this version of LOPES system, whenever necessary to send a message to the wireless sensor network the TCP server is listening from messages come from the TCP client and forwards them to the RS-232 Port. Whenever a message comes that is addressed to the application, the TCP server transmits them through the TCP client.

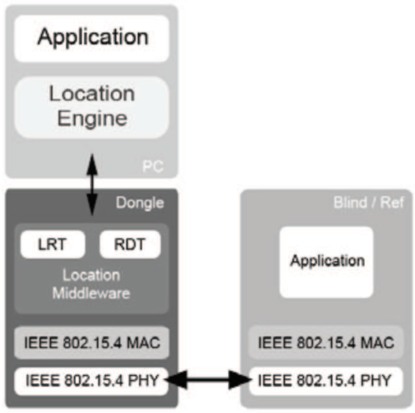


*Figure 27: Lopes System Architecture*

The middleware is responsible for managing the exchange of messages between the Application and the Sensor. Whenever the middleware has a set of messages from a given mobile sensor, it groups the values of RSSI by a defined order and sends that information to the location module. The location module receives RSSI values and the Sensor's ID and calculates the corresponding position. The result is sent to the upper layers, which are responsible for storing the position and displaying the data. The application layer is also responsible for sending the list of the nodes to be located the middleware. The interaction between all these elements is schematized in Figure 27.

### 4.3.2. "Fábrica da Ciência Viva" – Aveiro

The second demonstrator of the LOPES system was deployed in "Fábrica da Ciência Viva" in Aveiro, a Scientific Museum that provides cultural and scientific activities for people of different ages. The demonstrator consists on a "Poirot" Hat that a person wears which possesses an embebbed BlindNode. When a visitor puts the hat and initiates the location process. When a visitor enters a pre-defined area, a local display starts a video that presents an enigma that the visitor must to solve. This enigma leads the visitor to an exhibition object placed in the room "Mãos na Massa", in Figure 28.

*Figure 28: "Mãos na Massa" Exposition during the deployment of the LOPES system*

This demonstrator was implemented by placing a network of Reference Nodes in the ceiling of the room as documented in Figure 29. This network has a structure similar to the first demonstrator, except for the distances between the reference devices and the existing obstacles in the room.

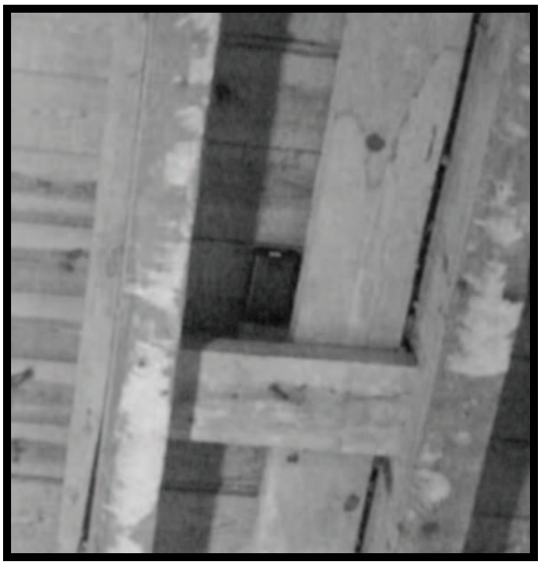

*Figure 29: RefNode on the ceiling of the "Mãos na Massa" exposition.*

The LOPES system calculates the position of the mobile node based on a Neural Network that determines through non-linear calculations the current position of the device having by reference the RSSI values of the messages sent from the BlindNodes to the RefNodes. As the values received by the Neural Network depend on the location of the obstacles and on the distance between the devices it obviously had to be re-calibrated to fit the new scenario. Figure 30 represents the Interface allowing the user to collect the RSSI signatures to input on the Neural Network. The blue dots are the Points where the RSSI values were measured. The system had to be adapted to the new equipment, meaning a whole new Chipcon CC2430 Kit and new device addresses of the devices as well as Servers for the system.

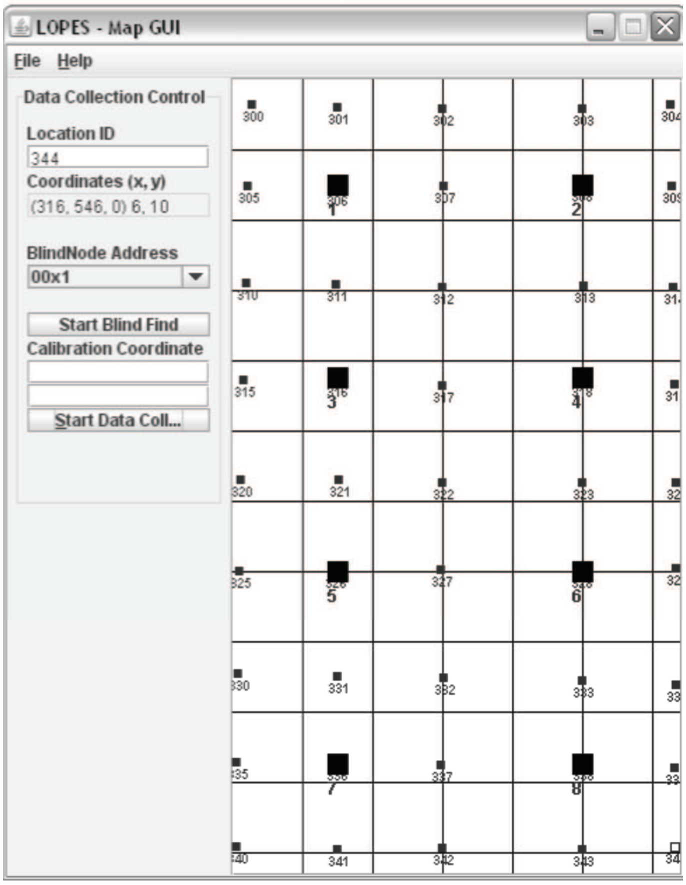


*Figure 30: Interface for Calibrating the Neural Network.*

The deployment of this demonstrator was positive to evaluate the adaptability of the system to different scenarios and to acknowledge, in a practical way, the virtues and weaknesses of the current system. The LOPES system is still in exhibition and can be experimented by anyone who visits “Fábrica da Ciência Viva” in Aveiro, Portugal.

## 4.4. Z-Stack Sample Location Protocol

In Z-Stack includes many sample applications for programmers to use as reference and to understand how to program each sensor modules. Figure 31 represents the message exchange scheme in the Z-Stack location protocol, by Texas Instruments. In this version, the BlindNode is the device that runs the CC2431 location engine. The location engine estimates its position by using the coordinates of all responding reference nodes within the radio range (maximum of 16) [CHI03] with the average RSSI of the received messages. When a RefNode ears a broadcast message sent from the BlindNodes, the device will estimate its approximate location, and then send the coordinates to the LocationDongle.

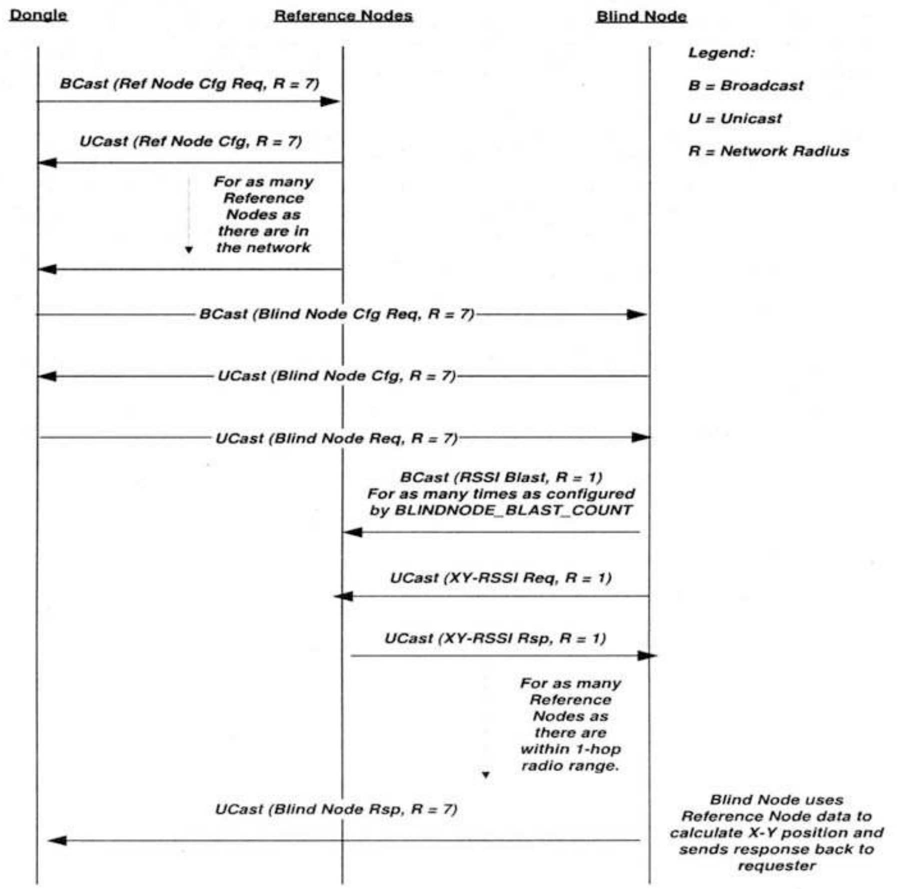


*Figure 31: Sequence of events Diagram in Texas Instruments Location Protocol*

There are two operation modes available for use in the Z-Stack Location profile: The Polled Mode and the Auto Mode. In Polled Mode any node in the Zigbee network can request the position of the BlindNode. The RefNode only acquires the BlindNode data when is requested using the BlindNode Request command. In Auto Mode, the process for acquiring the Reference Nodes is the same, except this mode allows the BlindNode to periodically collect the Reference Data automatically, or manually, by pushing a button for example. The BlindNode Response is sent by default to the Coordinator. However this destination can be configured, as the short address can be changed. The programmer can also set and request configuration parameters for the RefNodes and BlindNodes.

Figure 31 shows the position acquisition initiated by the LocationDongle. Afterwards, the BlindNode transmits 5 broadcast messages and requests the average RSSI and X-Y position. After the position of the BlindNode is calculated, the BlindNode position data is sent back to the Location Dongle, which is usually connected to a PC working as a server and receives the X-Y Coordinates to display them in a monitor.

This Location Stack combined with other applications can provide also further functionalities like network topology changing by software.

Each device runs a set of functions to provide adaptability to program other sample applications that are provided for the same kit, like simple home automation applications. These implemented stack functions increase the possible applications of the system, but at the same time slow the location process for the location sample application.

## 4.5. LOPES – The First Implemented Protocol

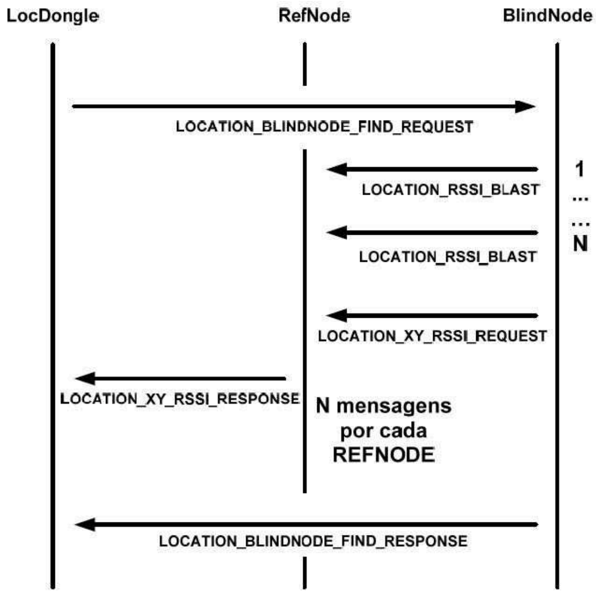


*Figure 32: Exchanged Messages in the sensor Network*

This first version of the LOPES demonstrator dates from the early 2008 and and is in continuous development at Instituto de Telecomunicações in Aveiro. In this version, when a BlindNode initializes or when it receives a request from the LocDongle, called LOCATION_BLIND_NODE_FIND_REQUEST, it transmits 25 Zigbee Broadcast Messages with a zero payload (LOCATION_RSSI_BLAST) with a period of 10ms. After the 25$^{th}$ Blast, the BlindNode sends a broadcast message to initiate the location process (LOCATION_XY_RSSI_REQUEST). Each Reference Node receives the broadcasted message to initiate the location process and creates a frame containing the received the RSSI Values and the BlindNode ID. This frame, called LOCATION_XY_RSSI_RESPONSE is sent by unicast to the LocDongle, which in its turn sends it to the Location Manager (LM). This package implies that the BlindNode must be always in reach of the LocationDongle. In this implementation, if the BlindNode gets out of reach to the system, the system will stop the process waiting for the LOCATION_XY_RSSI_RESPONSE message. This normally interrupts permanently the location process.

The LM gathers the information of the RefNodes and creates a vector of RSSI values. Then it uses an algorithm based on Neural Networks to determine the location coordinates and delivers this information to the Application.

The Figure 32 is a scheme of the packet exchange of the system installed at IT, which was implemented by Texas Instruments and has suffered some light structural changes. This was the first protocol to be used in the project since it was already built for the Chipcon CC2430 Kit. One improvement was the fact that 10 Blast by TAG were used and are sufficient to provide a reliable average RSSI to feed the NeuralNetwork. This reduced significantly the location time.

The Neural Network designed to calculate the location of the BlindNodes was also implemented differently from the one implemented in the original system. The new neural network was implemented by software in the Location Server. In the original system the CC2430 encompasses an embebbed processor that is capable of estimating the BlindNode's location and send the final coordinates through the serial port the final location coordinates. The last message shown, LOCATION_BLINDNODE_FIND_RESPONSE triggers the middleware location process.

## 4.6. Conclusion

The LOPES development was crucial to understand the requirements and compromises of a Real Time Indoor Wireless Location System.

The firss implementations of the LOPES project were devised without reflecting medium access aspects, which reduced their efficiency and number of locations per unit of time.

The FTT-L is a new upper-MAC protocol that has the purpose of optimizing the medium usage and accelerating the location process, providing the expansion of this project to real-time applications or to quickly locate a wider number of sensors. In this protocol the names of the components will change, and the location process will suffer severe changes. These changes will be explained further in the document.

# Chapter 5

# Flexible Time Trigger for Location

The Flexible Time Triggered Protocol for Location is the result of a study on the management of the Medium Access Control inspired by the Flexible Time Trigger (FTT) paradigm. FTT-L is based on a centralized MAC approach whose purpose is to prevent the collision of simultaneous messages of the location protocol. This protocol was proposed by Fonseca and Bartolomeu [JAF01] by specifying the basic structure and operation for the protocol.

The collision avoidance is reached through the scheduling of the messages involved in the location process, defining the specific instants in which modules, TAGs or RUs, will conduct the transmissions. The Location Manager (LM) controls all the process, being that the Master Gateway Unit (MGU) will coordinate all the messages in the system in a consistent way.

In this protocol the names of the elements are also different. The equivalent for LocDongle is the Master Gateway Unit, the RefNodes will now be called as RU - Reference Units and the BlindNodes, the mobile nodes, are the equivalent to the FTT-L's TAGs.

## 5.1. The Flexible Time Trigger paradigm

This protocol is based on the FTT paradigm in which to control the MAC, a new type of message devised, establishing a clear order of transmission between all the PAN components. This new message is called Trigger Message (TM). Flexible Time Trigger paradigm is a communications model designed for real-time systems, which has spread through the scientific community and has been target of several studies, tests and implementations. Some derivations of the original FTT are, for example the FTT-CAN [LAL01] and the FTT-Ethernet [PED02] to CAN and Ethernet, respectively.

FTT uses a synchronous master-slave architecture that is implemented in various nodes of the communications network. The Master of the system is responsible for managing and coordinating the communications between all the devices in the system. As such, the scheduling decisions are provided by the Master and are broadcasted to the Network using a specific periodic control message here designated by TM.

By using this message, the typical overhead of a Master-Slave communications can be greatly reduced, since a single TM can activate the transmission of multiple Slaves. The TM also provides the ability to filter a specific group of nodes that are targeted to a given transmission.

Given that the scheduling is centralized in one unique device, the system becomes more Flexible (given the designation to the protocol). Changes on the message properties can be performed in only one master node and distributed to the other nodes by the TM. The Master can hold the necessary information to adapt real-time traffic to the available medium. This allows the Slaves of the system to be unaware of a particular scheduling policy. Each slave follows strictly the schedule provided by the TM.

An example of the application of FTT-CAN is CAMBADA - Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture [LAU01], the Robotic Football Team from the University of Aveiro. The low-level sensing/actuation system is implemented through a set of micro-controllers interconnected by a CAN network, a fieldbus typically used in distributed embedded systems. The CAN network needs to be complemented with a higher-level transmission control protocol to enhance its real-time performance, composability and fault-tolerance [VSI01].

The FTT paradigm was developed at University of Aveiro, more precisely at Electronics, Telecommunications and Informatics Department (DETI) by José Alberto Fonseca and Luis Almeida. Paulo Pedreiras was responsible for evaluating the scheduling policies and developed the first implementation of FTT – Ethernet.

Joaquim Ferreira also made some contributions to the protocol, concerning dependability and fault tolerance by Master and Bus replication [SIL01]. More recently new approaches to the FTT protocol are in progress, especially on what concerns the QoS mechanism [PED01] and Switched Ethernet, given the name Flexible Time Trigger – Switched Ethernet (FTT-SE) [PED02]. Currently the paradigm is being experimentally adapted also to IEEE 802.15.4 networks to improve the performance of localization systems, as discussed in the following sections.

## 5.2. FTT-L

Flexible Time Trigger for Location (FTT-L) is a MAC protocol aiming at reducing the time between two consecutive locations and optimizing the medium usage, synchronizing, ordering and managing the packet transmission between all the devices contained in the network in a simple, effective and structured way. There are three fundamental elements in a location network: the Master Gateway Unit (MGU), the Reference Units (RU), and the TAGs. The MGU controls the complete location process and collects all the data received from thee RUs, which are typically fixed in the ceiling of the room forming a mesh network. This mesh collects the Received Signal Strength Indication (RSSI) values from the received packets from TAGs or so called blasts. This centralized MAC is coordinated through the main piece of all the protocol: the Trigger Message (TM).

### 5.2.1. The Trigger Message

The TM is the vehicle that the Location Manager (LM) uses to communicate to the Zigbee Network and to start two different tasks: starting the Blasts (the packets that are broadcasted by the TAGs to provide RSSI samples to the REFs) and collecting the RSSI values from the REFs. This way the MGU can clearly synchronize all the elements in the network. The TM is built by the MGU that is in direct reach of the Location Manager. This way, the application layer is able to define immediately the Location Cycle's (LC) properties as the LCs are occurring. The Location Manager can also define properties such as number of TAGs and REFs in the network. We also can define devices to be located in the current location round, providing a selective location. This way a subset of elements in the network can be located with a simple command of the MGU.

By changing the Offset field, which will be explained further, the application can also define the granularity of the network, adjusting the transmission delays between different types of messages in the system. These features allow the application to manage the granularity of the location as well as select the TAGs to Localize.

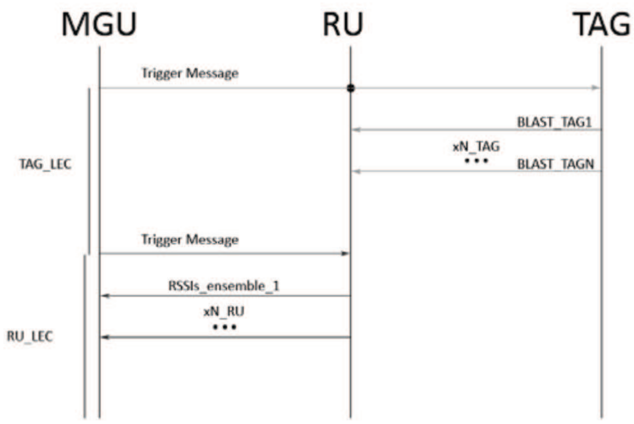


*Figure 33: FTT-L Communications Diagram*

In order to optimize the medium usage, the FTT-L controls the transmitting instant through Location Elementary Cycles (LEC) as shown in Figure 33. The TM triggers LECs in a synchronous way. The TM specified in Figure 34 defines the instant in which devices access the medium to transmit. Figure 34 consists on a simplified scheme of the Trigger Message Structure.

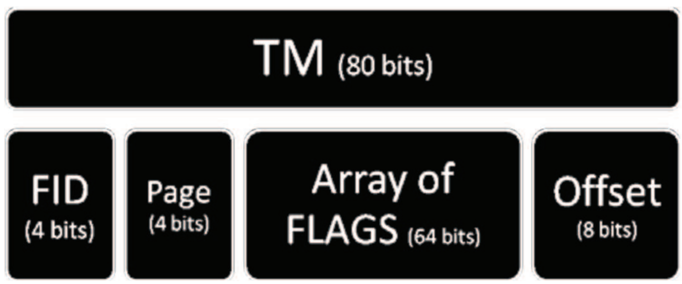


*Figure 34: Simplified structure of a TM.*

**FID**

This field is the Frame Identifier; its purpose is to identify the type of message to be transmitted. This field allows the system do have different types of TM for different purposes.

FID - 4 bits

MASK = 0xF0000000000000000000

**PAGE**

This field determines the group of devices to which the TM is addressed. In FTT-L's impelentation, if this field takes the value 0x0, the message is destined to TAGs. As for value 0x1 the message is destined to RUs.

PAGE - 4 bits

MASK = 0x0F000000000000000000

**Array of FLAGS**

The bits to 1 represent indexes of the devices that are supposed to be located by the system. This field allows the user to select up to 64 devices per TM.

Each device verifies if the TM's flag that corresponds to the device's index in set to 1, if NOT the device will not transmit until the next TM arrives, where the device verifies this property again.

ARRAY - 64 bits

MASK = 0x00FFFFFFFFFFFFFFFF00

**Offset**

The Offset field determines the system granularity. This field allows the system to adapt its granularity by software, providing a better adjustment to the type of packages to be transmitted. The granularity of the system is equal to the system Offset multiplied by 100us. When the Offset is 1 the granularity is 1ms.

Offset - 8 bits

MASK = 0x000000000000000000FF

It is important to notice that the element that controls all the process is the MGU. The MGU is the device that is responsible for sending all the TMs. This device is not Battery dependent and as it is not mobile, it is always in range of the system and connected to a fixed power source. These features allow the process to continue even if one of the TAGs gets out of range of the system.

*Equation 1: Transmission Offset Calculation Formula*

$$d_i^p = 10 * Offset(w) * \sum_{k=1}^{i} Flag(k)$$

The TMs are defined as $TM_j^p$ where j is the index of the sent message in the system and p is the page field of the TM. Considering the device correspondent to Page p and Flag I, designed by $U_{i,p}$. A TAG or RU is allowed to transmit only if the corresponding flag in the Array of Flags is ON. In that case, its transmission is delayed for an amount of time given by the Offset field of the TM multiplied by 100us and by an integer k that represents the number of set flags from the flag 1 to the current flag. This way the delay $d_i^p$ is therefore a value in miliseconds given by Equation 1 and represents the instant in which the $U_{i,p}$ will access the medium and transmit it's message.

## 5.2.2. The TAG's Location Elementary Cycle

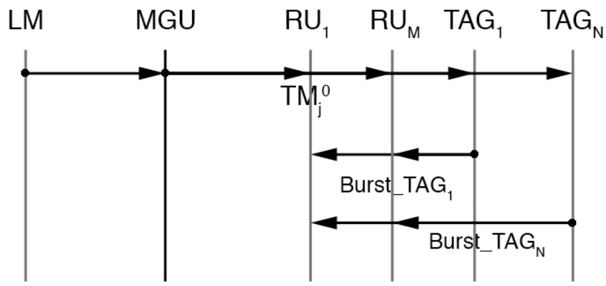


*Figure 35: TAG's LEC*

The TAG LEC is triggered by a TM with a page value of zero ($TM_j^0$). This message is broadcasted in the network by the MGU and received by all the nearby devices. After its decoding, the addressed devices will respond with a sequence (burst) of zero length payload broadcast messages known as Blasts, whose RSSIs are collected by the neighbour reference units. This process is shown in Figure 36.

The time reserved to the processing of the TM by the TAGs is called $t_o$. The time of guard between the bursts of consecutive transmitting TAGs is called $t_G$. Inside a burst we can determine $t_R$ that represents the time between consecutive blast messages inside the same burst. The duration of the transmission of a frame obviously depends on the size of the sent frame.

The TM defined above will be ignored by the RUs. Conversely, TAGs will use it to calculate each delay $d_i^p(w)$ and load the corresponding value into a local Timer that will expire in instant scheduled by the MGU for the TAG transmission.

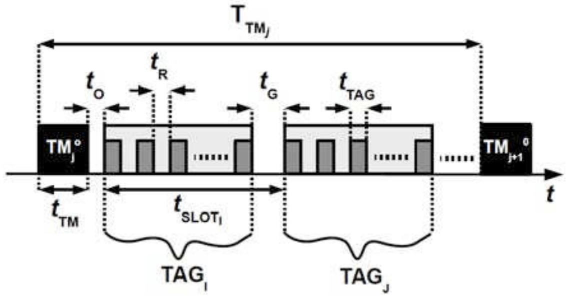


*Figure 36: TAG Location Elementary Cycle.*

During the LEC with a TM referring to the Page 0x0, each RU creates a structure that stores the RSSI values collected from each TAG, as schematized in Figure 35. In every RU, there will be stored 10 RSSIs per TAG per location round. When the burst ends the RU determines the average RSSI for each TAG and transmits these values to the MGU as will be explained ahead.

### 5.2.3. The RU's Location Elementary Cycle

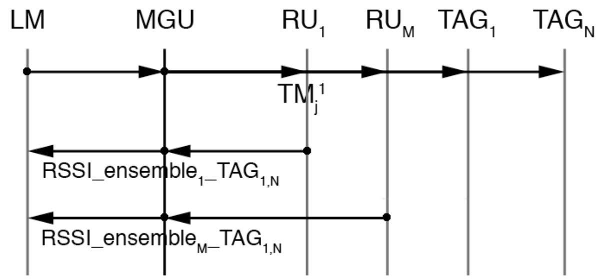


*Figure 37: RU Location Elementary Cycle*

As in the TAG LEC, the TM initiates the collection of average RSSIs stored in the RUs. This Trigger Message has a page of 0x01 ($TM_j^1$) and is called RU_Trigger_Message.

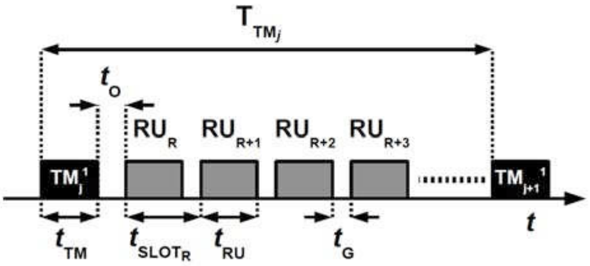


*Figure 38: RU LEC timeline*

In the same way as in the previous process, each RU will determine its delay to initiate a unicast transmission of the message RSSI_ensemble$_j$_TAGs. The messages are organized in time as shown in the diagram of Figure 38. This unicast is addressed to the MGU.

## 5.3. State Diagrams

In this section some simplified state diagrams illustrating the system operation mode of the localization devices will be presented. As the system consists on a distributed architecture, each element of the network processes its own information depending on the input signals received from the other network elements. These elements will be described in detail in the following subsections.

We will first approach the state/time diagram of the MGU, that is the node that is responsible for the whole system synchronization, followed by the TAG's and the RU's model.

### 5.3.1. MGU

The MGU is responsible for:

- Determining the active system devices.
- Sending the TMs.

- Defining the time granularity for each type of messages by adjusting the TM Offset.
- Gathering the RSSI signatures and sending them through the Serial Port

In Figure 39 a simplified state diagram for the MGU device is represented. The MGU is the coordinator element of the system. Its messages define the behaviour of the sebsequent messages on the system.

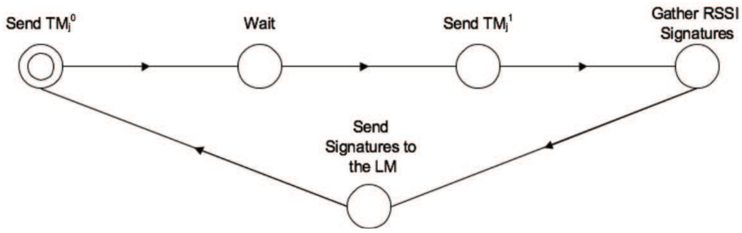


*Figure 39: MGU simplified state sequence diagram*

The MGU sends the first TM when it is turned on. Then, it waits the necessary time for the TAGs to transmit the Blast messages that will be collected by the Reference Units. When this time elapses, the MGU sends the TM2 that will start the transmission of RSSI_ensemble messages from RUs.

The application must provide to the MGU the information about the active devices in the network and the transmission offset. As the MGU has all the required information, the device builds the TM that defines the active flags in the Array of Flags and the Offset value in the Offset field. These values allow the system to calculate the LEC times, representing the delays between the TMs. The times must be big enough to allow the transmission of all the messages of the current LEC, which were directly or indirectly started by the sending of the last TM.

## 5.3.2. TAG

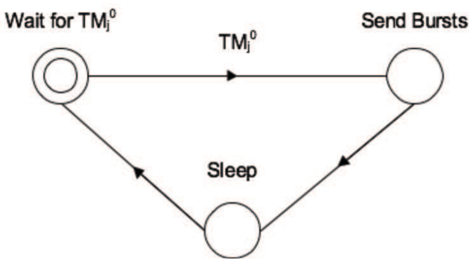


*Figure 40: TAG simplified state sequence diagram*

The TAG is responsible for:

- Interpreting the $TM_j^0$
- Evaluating the granularity defined on the TM.
- Sending the Bursts

Every time a TAG receives a $TM_j^0$, the device will answer to the request by sending 10 RSSI bursts with a delay that is calculated having by basis the index of the device in the system. Figure 40 shows a simplified state diagram for the TAG. TAG units are predefined to listen to the medium. As a TAG receives a $TM_j^0$, it immediately reads the content of the TM and evaluates its unique transmission offset, which is a function of the device index/position in the network. The Transmission_Offset and the Sleep_Offset are calculated. The device waits the Transmission_Offset time and sends the 10 Blast Messages in its system defined time slot. When the device finishes sending the blast messages, it can be set to sleep for the remaining LC. Since these devices are mobile, it is convenient to concern about power saving.

## 5.3.3. RU

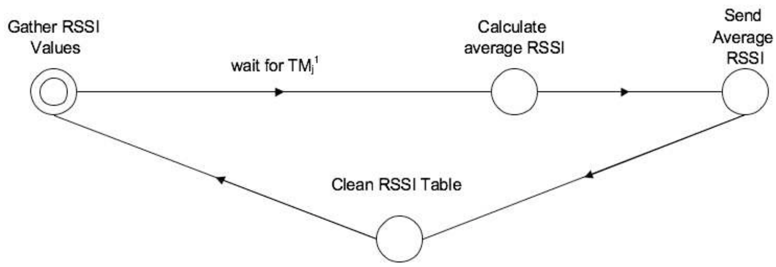


*Figure 41: RU simplified state sequence diagram.*

The RUs are responsible for:

- Gathering 10 RSSI blasts from each TAG
- Interpret the $TM_j^1$
- Calculate the average RSSIs
- Send the RSSI averages to the MGU (RSSI_ensemble messages)
- Clean the RSSI table.

Each RU gathers the RSSI_ensembleed from each TAG. By the time every RSSIs are collected, the MGU will send the $TM_j^1$ as specified in the state diagram in Figure 41. This message will give the order for the RUs to transmit the RSSI_ensemble messages containing the RSSI averages with a delay that is function of the device index on the system. The RU device is a fixed device, as the MGU, and has no specific power saving concerns. The device is predefined to listen to the medium and to collect all the received blast messages until it receives a $TM_j^1$. When this occurs, it builds an array with the RSSI averages associated to each device address and calculates its unique transmission offset, function of the Offset field of the $TM_j^1$. When the instant is reached, it sends the RSSI_ensemble message and clears the RSSI_ensemble Table containing the information about the collected Blast Messages.

# Chapter 6

# Coding on Chipcon's CC2430 Z-Stack

Each element of the network was coded based on an Event polling programming structure. Here are presented the base structures used to program the CC2431 modules on top of a Zigbee stack, more precisely on top of Texas Instruments Z-Stack.

As previously described, the Chipcon CC2430 development kit includes several sample applications designed for Home Automation and Control, such as light remote control. In this range of applications it is included an Indoor Location Sample Application, which differs from FTT-L in several aspects. For example, the number of exchange messages in the system to provide a single location is excessively high, because the tag encompasses a Localization Engine. Features like this slow down the location rate in a way that becomes difficult to support a Real Time Location.

The following subsection describes the coding of the most significant actions and events.

## 6.1. Sending a Zigbee frame

To send a message through the Zigbee network, the AF_DataRequest(…) function can be used. The AF_DataRequest function in Figure 42 receives as input 7 arguments. The most important one is the device source address, which is a structure that describes the endpoint. The third parameter is the Message byte array and is followed by the message size, in bytes. The following parameters are response and forwarding parameters, such as type of Acknowledge to the message, number of hops and transfer ID numbers.

```
(void)AF_DataRequest( &pkt->srcAddr, (endPointDesc_t *)&epDesc,
                      LOCATION_XY_RSSI_RESPONSE_CEM, 31,
                      rspDongleMsg, &transId, options, 1);
```

*Figure 42: AF_DataRequest function*

## 6.2. Scheduling an event

```
osal_start_timerEx( BlindNode_TaskID, BLINDNODE_BLAST_EVT, BLINDNODE_BLAST_DELAY );
```

*Figure 43: osal_start_timerEx function*

To schedule events or to provide a delay before initializing a given event, we can use the osal_start_timerEx function represented in Figure 43.

This function takes as arguments the TaskID, that will define if this operation is to be executed by the device in which the code is being executed. The second variable is the EVENT variable, which will activate a certain EVENT as soon as the timer reaches the value of the third variable, the DELAY.

## 6.3. Event polling

The programming is organized in an event polling structure. As each event flag is activated the correspondent routine is initialized. Figure 44 shows an example for event polling in the TAG case. In this fluxogram the device verifies each one of the system flags recursively. As a Flag is set, the device starts the correspondent routine to send a single blast message and depending on the number of blast messages that were already sent is schedules a new event or disables the flag that starts a new blast message. The device then exits the Blast Event and verifies all the other flags to be attended. When the same Flag.BlastEVT is verified again, the system restarts the Blast Event Process Again. This allows the system to attend to different events provided by distinguished system flags to be set.

Event polling was the most practical way found to apply the FTT-L algorithm. Any function can trigger an event with a previously defined timeout, and each event will be processed in a prioritized way. In the current implementation, every received message check event will be processed prior to the sending events.

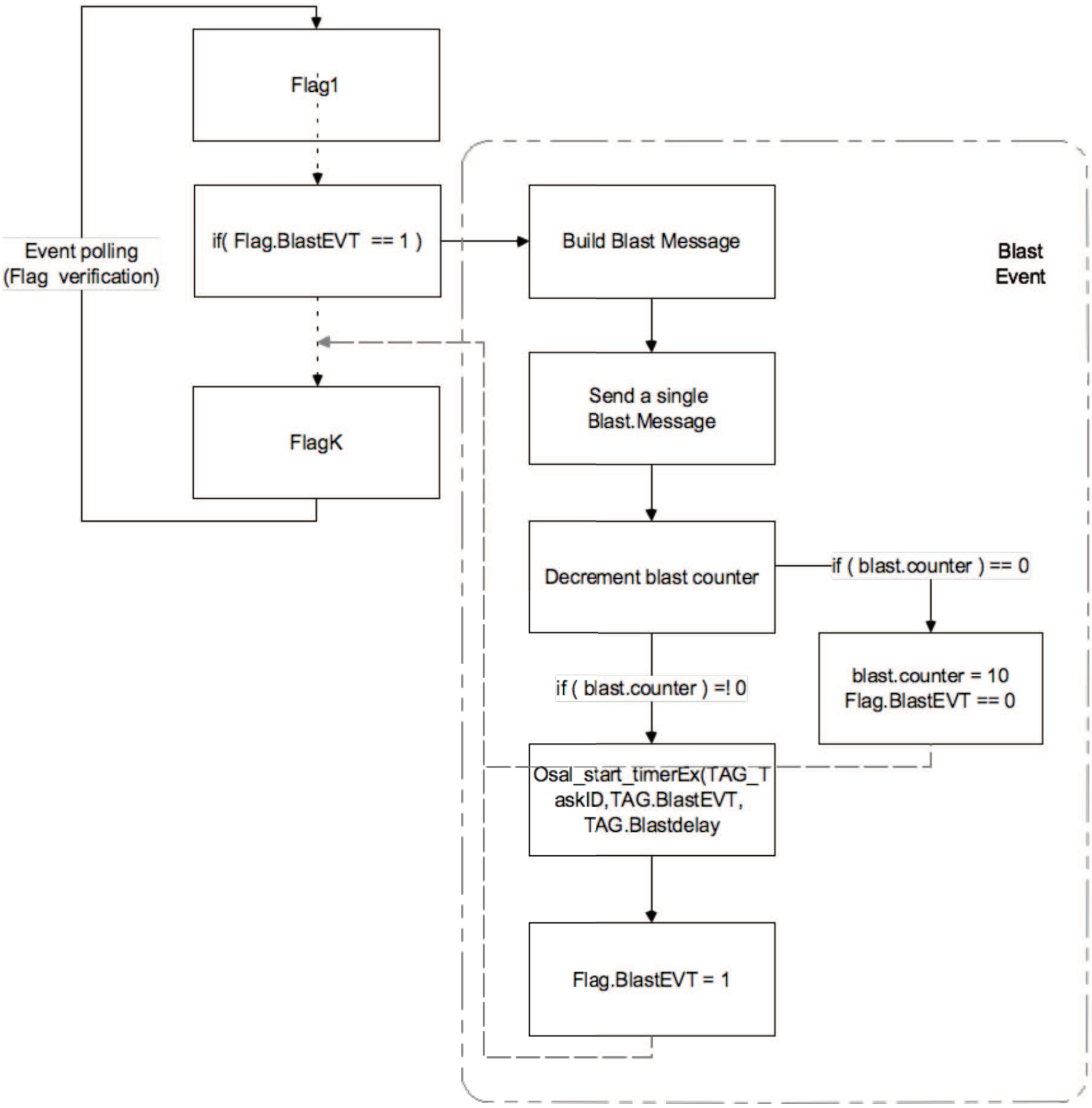


*Figure 44: Event polling structure example*

## 6.4. Processing a received message

```
static void processMSGCmd( afIncomingMSGPacket_t *pkt )
{
```

*Figure 45: processMSGCmd function*

The function responsible for processing the incoming messages is typically the processMSGCmd function, which filters the message content and initializes the correspondent task. Figure 45 depicts the processMSGCmd(…) header.

## 6.5. Filtering the TM

The algorithm for collecting the TM has a simple structure. Each byte of the message is sequentially collected to the corresponding variable, as defined in the TM specification. The first Byte includes the information about the FID (first 4 bits) and the Page (last 4 bits). The following 8 bytes carries the information about the Array of Flags. The last byte is the Offset field.

As for the Array of Flags, each bit of this field is a flag that indicates the index of the activated devices, TAGs or RUs in the system. Each byte of the Array of Flags is verified by means of a mask that is continuously shifted from the beginning to end.

# Chapter 7

# FTT-L performance assessment

This chapter presents an experimental analysis of the key parameters involved in the implementation of the FTT-L protocol.

In the original LOPES System, the minimum time between consecutive localizations is approximately. In many cases it takes more that two seconds. This value is acceptable for systems with objects that move very slowly, or are stored or static. Nevertheless, this rate is inadequate for most Real-Time Applications, where the system dynamic is more demanding.

## 7.1. Analytical Evaluation

The fist sub-chapter consists on an analytical evaluation of the FTT-L protocol performance initially approaching the one prior to the system implementation.

### 7.1.1. Original System Performance

As for many Location Protocols, the number of TAGs in the system contributes linearly to the period between consecutive locations.

For instance, in a scenario where 5 TAGs are being tracked ($N_{TAG}$ = 5), the time between consecutive locations of a TAG reaches 10 seconds, as it can be seen in Figure 46 (the dashed line plotted using Equation 2). Note that 10 seconds between consecutive locations for a system that only encompasses 5 TAGs is very high and can compromise many system applications. A system containing more that 10 TAGs would be difficult to manage with the original system specifications.

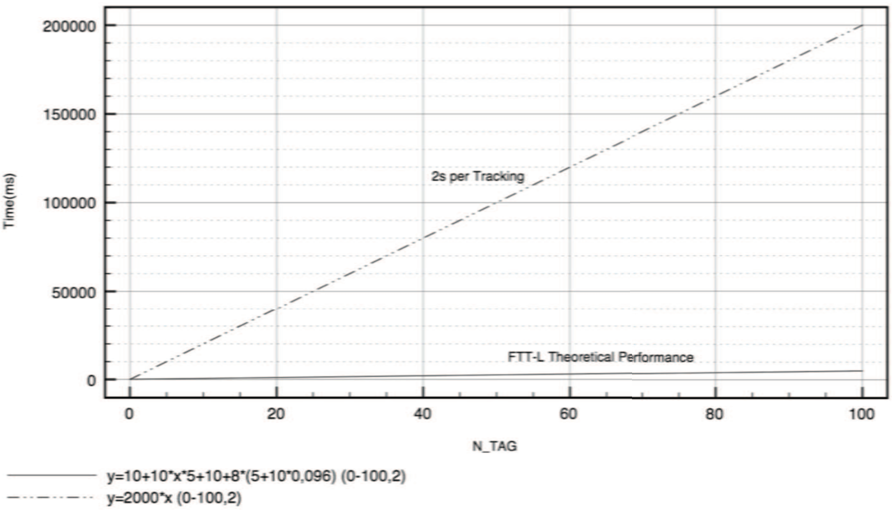


*Figure 46: FTT-L - Location System performance estimation*

## 7.1.2. FTT-L performance.

The continuous line in Figure 46 represents a plot of the theoretical performance of a FTT-L system implementation, obtained using Equation 3

By increasing the number of TAGs in the system, the time between two consecutive locations of the same TAG increases slowly when compared to the original approach. Considering the example of a given demonstrator with 8 RUs and 20 TAGs, respectively N_RU = 8 and $N_{TAG}$ = 20. Using Equation 2 we can determine a that the time between two locations of the same TAG is about 40 seconds assuming that the location time of single TAG is 2 seconds.$N_{BLAST}$ represents the number of blasts sent by each TAG in a TAG's LEC.

*Equation 2: Default Location Time*

$$y = 2 * N_{TAG}(s)$$

As for a theoretical location time using the FTT-L protocol in the same context, the location time was determined using Equation 3 to be 90.184 ms. Please refer to Figure 36 and Figure 37 for informations regarding the used variables.

*Equation 3: FTT-L Location Round Time Theoretical Formula*

$$t_{LR} = \sum_{i=1}^{N_{TAG}} (2 \times (t_{TM} + t_O) + t_{SLOT_T} + N_{RU} \times t_{SLOT_R})$$

where

$$t_{SLOT_T} = N_{BLAST} \times t_{TAG} + (N_{BLAST} - 1) \times t_R + t_G$$

and

$$t_{SLOT_R} = t_{RU} + t_G$$

This value was obtained assuming transmission times tested through practical experimentation described further in the document. The values are calculated assuming a transmission rate of 250 kbps and also assuming that the transmission of a packet sent by a RU or TAG is precise enough to be conducted in a pre-determined slot of 5ms ($t_{SLOT}$). It is also assumed that a TM can be sent successfully within a pre-determined 10ms slot ($t_o$). Each RU calculates the medium of the LQI values that were received by each TAG.

| Message | HEADER | Payload | Total | Time |
|---|---|---|---|---|
| $t_{TM}$ | 344 bits | 80 bits | 424 bits | 1,696 ms |
| $t_{TAG}$ | | 0 bits | 344 bits | 1,376 ms |
| RSSI_ensemble | | 32 bits[1] | 376 bits | 1,504 ms |

*Table 20: Relation between transmission times and Message size.*

For each additional TAG in the location round, each RU will have to send an additional average RSSI followed by the corresponding TAG ID. This will be 2 bytes for the RSSI medium value and 2 bytes for the TAG address, increasing the transmission of each RSSI_ensemble message time in 0,128 miliseconds at 250kbps.

In Table 20 the theoretical delays of transmitting a single packet are presented and change according to the type of message that is being sent, assuming a transmission bitrate of 250Kbps.By changing the parameters of the TM sent by the MGU, the delays between transmissions can be easily modified.

[1] [1] Considering the Location of a single TAG, this value varies geometrically with the number of TAGs to be located in the network.

Using FTT-L, a single TAG in an 8 RU system configuration can be detected in every 90.184 milliseconds. However transmission failures can occur and, happening for broadcasts, will impair the location round,

If failures occur for unicast transmissions, retransmissions will follow and the next slot must only begin after the worst-case contention delay in order to avoid collisions with the transmission of the current slot. Table 21 exposes the times from which the system implementation was idealized, even though later they were optimized throughout the implementation.

| **Variable** | **Designation** | **Value** |
|---|---|---|
| $t_{TM}$ | TM | 2ms |
| $t_{O}$ | Overhead time | 10ms |
| $t_{SLOT}$ | Slot time | 5ms |
| $t_{G}$ | Guard time | - |
| $t_{TAG}$ **or** $t_{REF}$ | Time to send a TAG/REF message | 2ms |
| $T_{TMj}$ | Time between two TMs with the same page field | - |
| $t_{R}$ | Time between two blast messages | Aprox. 3ms |

*Table 21: Time Slot approximate theoretical values*

A plot of this function up to 10 TAGs in the system is presented in Figure 47 for a system with 8 RUs.

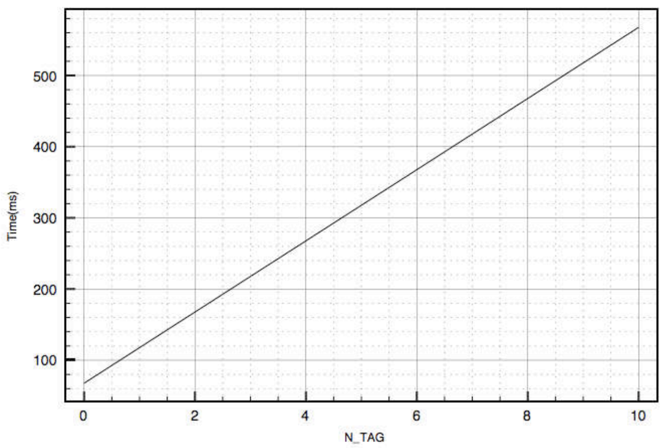


*Figure 47: FTT-L Experimental Performance*

As we can see in Figure 47 the LC time for locating a single TAG goes around 110 ms, as for locating 2 TAGs takes approximately 170 ms providing a detection rate of more than 5 locations per second. The LC time for locating 10 TAGs in an 8 RU system is 565 ms. This value indicates a location rate of approximately 2 locations per second. Figure 48 shows de differences in the performance of a system with an increasing number of RUs for fixed values of TAGs.

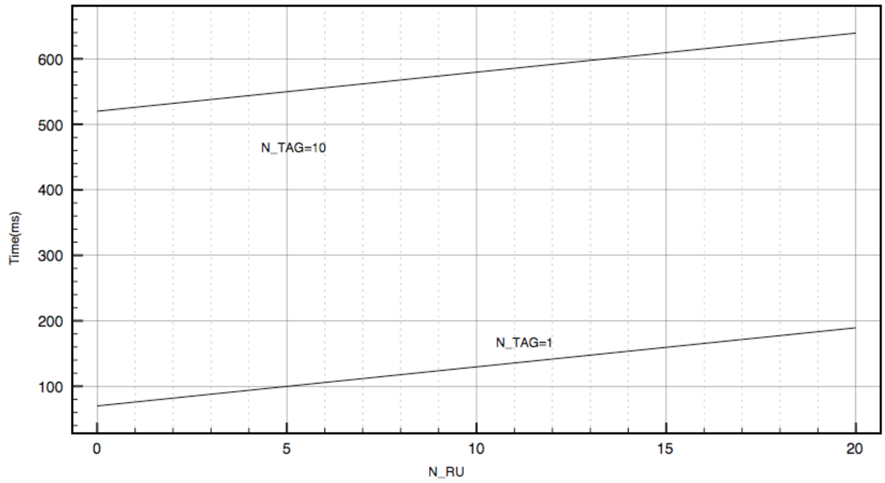


*Figure 48: System performance for 1 and 10 TAGs depending on $N_{RU}$*

In Figure 49 we can analyse the impact of increasing $N_{RU}$ from 8 to 20. This change provides a delay in the system of approximately 100 ms.

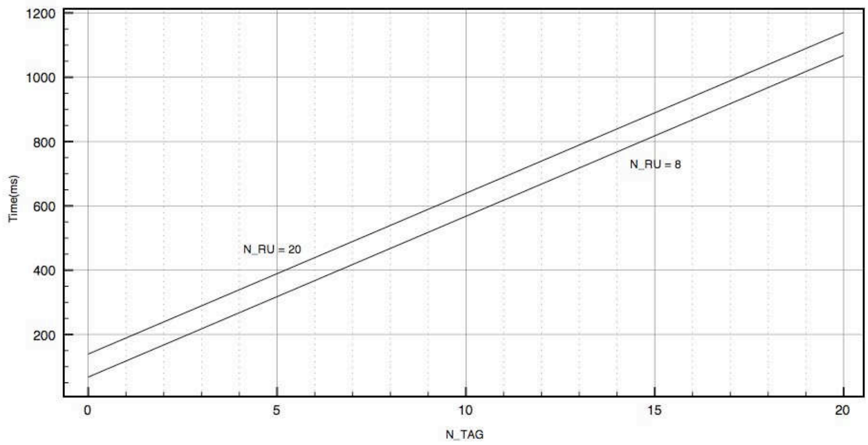


*Figure 49: System performance for a system with 8 or 20 RUs in function of $N_{TAG}$*

The performance changes by increasing the number of RUs are smaller than when we increase the number of TAGs in the system as we can see in Figure 50.

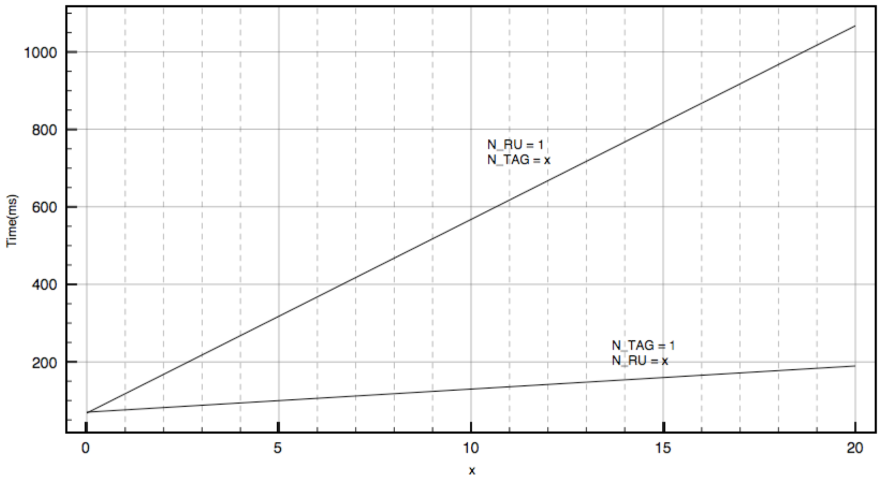


*Figure 50: Comparing between increasing number of TAGs and RUs*

Therefore, it is better to increase the number to TAGs than increasing the number of RUs in the system.

# 7.2. Evaluation Setup

Though, in this section we will analyse the time taken to execute different kinds of operations such as sending two consecutive messages or responding to an incoming message by sending another. To make this assessment several tests were conducted. The transmission calculation times were determined using a Texas Instruments 802.15.4 packet sniffer and were confirmed using a scope directly connected to the antenna interface or measuring the power consumption of the device.

## 7.2.1. Antenna interface

A oscilloscope was used to collect a picture of the behaviour of the antenna on the CC2430 module so we can get a glimpse of the effective medium occupation caused by the sending of a message. The Figure 53 was captured connecting directly the oscilloscope directly to the antenna interface through a 50 ohm adapted coaxial cable. As it can be seen it is clear the beginning and the end of the transmission, allowing to determine the transmission duration as well as the period.

## 7.2.2. Power Measurement

This interface allows to evaluate the timeliness of the protocol by monitoring the power consumption of the module. As such, a resistor connected in serial with the positive pole of the power source. Then the voltage on the terminals of the resistor, was monitored using an oscilloscope.

As it can be seen in Figure 51 the power consumption is maximum when the device listens to the medium. When a transmission occurs a sudden drop of the power consumed by the device is verified. By analyzing this phenomenon, the approximated duration of a frame, and the exact time between consecutively transmitted frames can be determined.

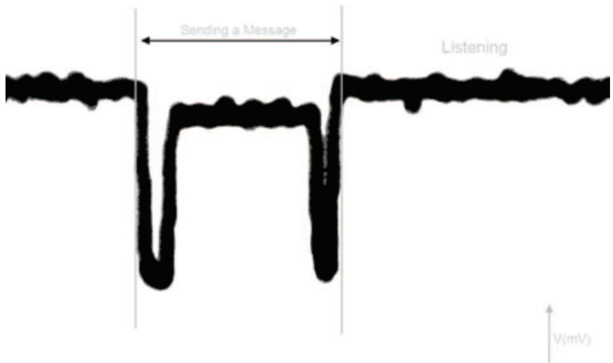


*Figure 51: Interpreting the Power Consumption of the CC2430 Module*

Most of the power consumed by the device is due to the listening ability. Therefore it is usually used a sleep function to set the device to a low power consumption state. By putting a device into sleep, the device looses the ability to receive the messages that are transmitted into the medium. The sleep function is commonly associated to a timer, usually called sleep timer, which is not deactivated by the sleep function, contrarily to the rest of the timers. Great part of the device functions are deactivated to achieve a hibernation state that will only be recovered when the Sleep Timer equals the time value set on the sleep function or set on some Network Polling definitions implemented on more complex stacks, such as the Z-Stack.

The FTT-L is a protocol that intends to optimize the usage of the medium, and so, the sleep mode implementation is not a priority in this phase of the project. Though the implementation in a more advanced state of the project could be useful to increase the battery lifetime of the mobile devices.

### 7.2.3. Sniffer

To determine the time between the transmitted packets in the system it was used an application provided by Texas Instruments called Texas Instruments 802.15.4 packet sniffer. This application downloads to a CC2430 EM module the necessary firmware for it to operate as a packet sniffer. The system provides the ability to collect the information contained in each packet and also the transmission order, transmission times and delays as shown in Figure 52.

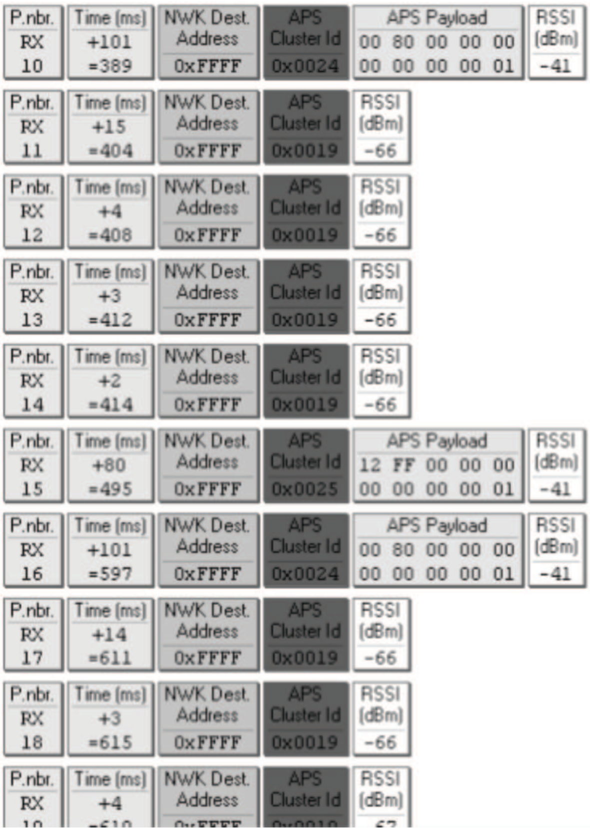


*Figure 52: Measuring the time between two consecutive messages sent by the same module.*

This application provided the experimental values necessary to evaluate the protocol performance.

## 7.3. Results

This topic is about measuring the time that a module takes to retransmit a certain message depending on how the message is called by the program.

### 7.3.1. MGU TM transmission delay.

In this case, two events were set. TM1_EVENT and TM2_EVENT were designed to send each one of the TMs. Each event initiates the transmission of a TM to the medium and schedules the next event to start in a software-defined timeout.

*Vpp: 129mV    Period: 6.73 ms*

*Figure 53: Capture of two consecutive Trigger Messages polled by the MGU*

The medium time that the system takes to build and send two TMs started by different events with the event delay/timeout set to 0 ms is 6,73 ms with a standard deviation of 0,77 ms (capture example in Figure 53).

Though TMs aren't messages to be re-sent in short intervals, this experiment shows that the Jitter for scheduling of a Trigger message does not surpass the 10ms timeslot that was defined in the theoretical objectives.

## 7.3.2. TAG Blast Retransmission Delay

When requesting the transmission of successive messages using the processing latency associated with event handling has no impact on the system retransmission time since consecutive requests are made without attending to other events. This way the retransmission delays decrease to half the time when compared to the event structure previously used to repeat the Blast messages. The Jitter between messages decreases, as the number of possible operations between the repeating of the same message is now deterministic, contrarily to the event-based algorithm.

For example, in the Z-Stack Location Sample, each BlindNode, the equivalent to the FTT-L protocol's TAG, sends each blast by polling the same event 10 times, like described in Figure 44.

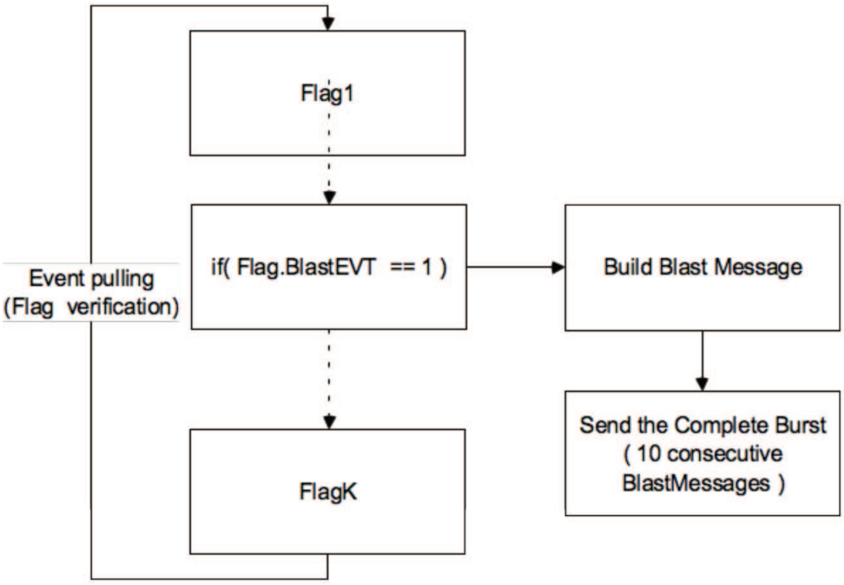


*Figure 54: A simple consecutive message transmission*

When sending messages consecutively as described in Figure 54, the average repeat time was reduced to 2,91 ms with a standard deviation of 0,89 ms. As the device exits the event, schedules the same event again and attends to other events that may have been activated in the meanwhile, the repeat time increases in a drastic way. These values can describe the time that a TAG takes to repeat a transmission when sending a Burst composed of Blasts.

## 7.4. TAG TM Response Time

It is also interesting to measure the time that a device takes to receive, process and answer to a given packet. This time is useful to estimate what is designated in the FTT-L as Overhead Time. As a TAG or RU device receives a TM, the device will process the received message fields and re-calculate variables like the transmission offset time. As soon as these variables are calculated, a transmission delay is calculated and a response message is sent. In the case of the TAG devices a blast is started, as for the RUs, the collected RSSI messages are sent to the MGU.

In this evaluation the offset field of the TM was set to zero so that te fastest response time is obtained. In this case, the time elapsed from the end of the TM and the end of the reply are measured, considering that it is requested in the TM incoming message event.

Results show that a device takes an average time of 7,78ms with a standard deviation of 0,71ms to answer to an incoming message.

In a second case, the implemented code answers a TM with the most simple and fastest Blast Sequence using an event polling structure as the one explained in Figure 44. The reception of a TM started the event TAG_BLAST_EVENT. As the event is terminated, the event itself schedules itself to be started on a certain delay, in this case TAG_BLAST_DELAY. The TAG_BLAST_DELAY was set to 0 as the repeating of the message is intending to be as fast as possible. The result is a medium retransmission delay of 5,13 ms with a standard deviation of 1,14 ms.

As the sending and reception of message is entirely controlled by the TMs, we know that as a TAG is sending it's own bursts, has no need for listening to the medium for other type of messages, so there is no need for listening to other type of events when a burst transmission is occurring. So, the most logical and practical way to send a given message is logically repeating directly the AF_DataRequest() function.

The result was a message response time of 14,94 with a standard deviation of 0,8ms. A capture example is on

## 7.5. Result Analysis

As we intend to optimize the medium usage, the program organization and the simplicity of the code can be very important when the programmer is concerned about tight deadlines and jitter. The bigger is the ambiguity of the code, the greater will be the Jitter of the packages and the bigger is the risk for a package to be lost. As the medium usage is being optimized, if a certain package is lost, this will compromise the whole location process.

To minimize the gravity of possible losses of packages we can apply to the system certain kinds of mechanisms that bypass errors that compromise the runnability of the system.

An example is the fact of each RU must calculate the medium of the received RSSI Blasts from each TAG. The RU must be able to calculate the medium of the received packages regardless of the number of received blasts. This way, considering that by definition, the number of blasts in a single location is set to 10 if a blast package is lost, the other 9 packages may still be considered. Otherwise their values would be lost and the effective location of the node would also be lost until the next round. Table 22 resumes the most relevant experimental results.

| Variable to be applied | Assumed (ms) | Experimental (ms) | |
|---|---|---|---|
| | | Avg. | Std. Dev. |
| $t_O$ | 8 | 6,404 | 0,71 |
| $t_R$ | 3 | 1,534 | 0,89 |
| $t_G$ | 2 | - | - |

*Table 22: Time Variable Analysis*

As a TM message takes an average time of 7,78 to be answered, we can assume that with a standard deviation of 0,71, the probability for it to be answered in a 10ms time slot is nearly 99,9% because the difference exceeds in three times the standard deviation from the medium.

## 7.6. Discussion

The values obtained by this analysis are useful to illustrate and estimate the response of a generic IEEE 802.15.4/Zigbee network.

A whole location cycle was implemented and tested using Texas Instruments IEEE802.15.4/Zigbee Modules, where it was observed that it is possible to implement a RTLS providing at least 6 locations per second for a single TAG in the system. Transmission delays and jitter were evaluated and are in line with the theoretical values presented. However since the FTT-L was designed to establish a controlled medium access and because the

# Chapter 8

# Conclusion

Wireless location services in interior spaces are experiencing an increasing demand, which fuels research and development in wireless communications and protocols as well as its wide adoption in multiple market segments. This thesis overviews current solutions for indoor location to conclude that the main commercially available ones are based on Wi-Fi or on hybrid approaches in which two different communication technologies are applied. These solutions pose strong limitations regarding the autonomy and cost of tags.

The LOPES system is a research project aiming at providing fine-grained localization in interior spaces. To this purpose, a demonstrator has been developed and is installed in the facilities of "Fábrica da Ciência Viva" in Aveiro, Portugal. This demonstrator employs ZigBee communications to obtain RSSI fingerprints, which are used to feed an Artificial Neural Network that computes the localization of mobile units. However, the performance bottleneck of this system (number of localizations per unit of time) was identified to be the medium access, which, for scenarios with multiple mobile tags, resulted in collisions and increased localization latency.

A recent work [JAF01] proposed a real-time protocol inspired in the Flexible Time Triggered paradigm, the FTT for Localization (FTT-L). Besides describing the FTT-L protocol in detail, this thesis analyses its implementation on top of ZigBee communications by providing indicative results regarding key performance parameters. Preliminary results show that it is feasible to localize 11 TAGs per second using ZigBee based FTT-L implementation supported on an infrastructure with 8 reference points (RUs). Besides, the values obtained in this analysis are also useful to illustrate the response of a generic ZigBee network in similar scenarios.

A line of future work is porting the FTT-L protocol to a lower-level platform were direct hardware access allows disabling the Carrier Sense Medium Access mechanism and reducing processing delays that result from handling events. This approach is expected to improve the efficiency of the protocol in using the medium as well as increasing the number of localizations per unit of time. Additionally, it would be important to enable multi-hopping localization through the use of gateways allowing, for example, roaming between different localization-enabled areas.

# Bibliography


[AER01]: "Real-time Visibility", Online available at: http://www.aeroscout.com accessed at 21/10/2008

[ANI01]: A. Koubâa, M. Alves, E. Tovar, "IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview", IPP Hurray, ISEP, TR-050702 v. 1.0 14/07/2005, download available at http://www.dei.isep.ipp.pt/~akoubaa/publication.htm

[BAR01]: P. Bartolomeu, J. A. Fonseca, F. Vasques, "On the Timeliness of MultiHop Non-Beaconed ZigBee Broadcast Communications", submitted to ETFA'2008.

[CHI01]: "Z-Stack", Chipcon, 2008

[CHI02]: "Z-Stack Location Profile User's guide", Chipcon, 2006

[CHI03]: "System-on-Chip Solution for 2.4 GHz IEEE 802.15.4 / Zigbee", Chipcon, 2006

[CLS01]: "Monitor and Optimize Your Business Processes", Cisco Location Solution, Cisco Systems, available at http://www.cisco.com/en/US/netsol/ns753/networking_solutions_package.html accessed at 20/10/2008

[CRI01]: H. Wang, et al., "The Research on Indoor Location System Based on Cricket" , Hit Shenzhen Graduate Sch., Shenzhen, ICMLC - International Conference on Machine Learning and Cybernetics, 2007, Publication Date: 19-22 Aug. 2007 Volume: 5, On page(s): 2662-2666, Hong Kong, Digital Object Identifier: 10.1109/ICMLC.2007.4370599

[CRI02]: The Cricket Indoor Location System, CSAIL – Computer Science and Artificial Intelligence Laboratory, MIT, http://cricket.csail.mit.edu/ accessed at 21/10/2008

[EKA01]: "Real-Time asset and people tracking." Online available at: http://www.ekahau.com, accessed at 21/10/2008

[ESH01]: eShepherd, Exavera Technologies, download available at http://www.exavera.com/healthcare/eshepherd.php accessed at 21/10/2008

[JAF01]: J. A. Fonseca, P. Bartolomeu, "A MAC protocol to manage communications in Localization Systems based on IEEE 802.15.4", 2007, Aveiro. Published on IECON 2008. 34th Annual Conference of IEEE, 10-13 Nov. 2008, On page(s): 2717-2723 ISSN: 1553-572XISBN: 978-1-4244-1767-4INSPEC Accession Number: 10415381Digital Object Identifier: 10.1109/IECON.2008.4758388

[JOL01]: J. Oliveira, P. Bartolomeu, "Projecto Lopes - Toolchain da Solução de Localização", 2008

[LAL01]: L. Almeida, P. Pedreiras, J. A. Fonseca - "The FTT-CAN Protocol: Why and How", IEEE Transactions on Industrial Electronics - special issue on Factory Communication Systems, Volume 49, Number 6, December 2002.

[LAU01]: N. Lau, L. Lopes, G. Corrente,"CAMBADA: Information Sharing and Team Coordination", Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open - ROBÓTICA'2008, April 2008, Aveiro, Portugal, pp. 27-32.

[MAN01]: S. Manapure, et al., "A Comparative study of RF-Based Indoor Location Sensing Systems", 2004

[MOR01]: V. Moretto, "Microchip Technology", February 2008, online available at http://www.tecnoimprese.it/user/File/Eventi/RF08-14feb_melchioni.pdf accessed at 20/02/2009

[NOH01]: A. Noh, et al., "Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing", 2008. SNPD apos;08. Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 6-8 August 2008 Pages:13 – 18, Digital Object Identifier 10.1109/SNPD.2008.125

[NAN01]: "nanoLOC - Robust, Reliable Communication Plus Precision Location for RTLS Solutions" online available http://www.nanotron.com/EN/PR_nl_TRX.php acessed at 25/04/2009

[PAN01]: PanGo Networks, http://www.pangonetworks.com/ accessed at 20/10/2008

[PED01]: P. Pedreiras, A. Luis, "The flexible time-triggered (FTT) paradigm: an approach to QoS management in distributed real-time systems", Parallel and Distributed Processing Symposium, 2003. Proceedings. 22-26 April 2003 Pages: 9 pp. Digital Object Identifier 10.1109/IPDPS.2003.1213243

[PED02]: P. Pedreiras, P. Gai, L. Almeida, G. C. Buttazzo, "FTT-Ethemet: a flexible real-time communication protocol that supports dynamic QoS management on Ethernet-based systems", IEEE Transactions on Industrial Informatics, Vol.: I, Aug. 2005.

[SIL01]: P. Silva, P. Bartolomeu, J. Fonseca " Implementing the FTT-L protocol with Zigbee" submitted to ICECS09 – IEEE International Conference on Electronics Circuits and Systems, Yasmine Hammamet, Tunísia

[TEX01]: Z-Stack - ZigBee Protocol Stack, Texas Instruments, download available at http://focus.ti.com/docs/toolsw/folders/print/z-stack.html, accessed at 10/03/2008

[UBI01]: "Platform for Precise Real-time Location." Online available at: http://www.ubisense.net, accessed at 21/10/2008

[VSI01]: V. Silva, J. A. Fonseca, R. Maia, U. Nunes, "Communications Requirements for Autonomous Mobile Robots: Analysis and Examples", FeT'2005- 6th IFAC International Conference on Fieldbus Systems and Their Applications, Universidade Autónoma de Puebla, Mexico, 14-15 November 2005.

[WAN01]: R. Want, A. Hopper, et al. "The active badge location system.", Olivetti Research Ltd. (ORL) Cambridge, England, http://web.media.mit.edu/~dmerrill/badge/Want92_ActiveBadge.pdf, 1992

[WIN01]: F. Winkler, E. Fischer, et al., “A 60 GHz OFDM Indoor Localization System Based on DTDOA”, 14th IST Mobile & Wireless Communications Summit, Dresden 19-23 June 2005, download available at http://www.eurasip.org/Proceedings/Ext/IST05/papers/361.pdf

[ZIG01]: ZigBee Alliance, “Wireless control that simply works”, http://www.zigbee.org accessed in 08/09/2008

[ZEB01]: “Zebra - RTLS Equipment Control”, online available at http://zes.zebra.com/products/application-software-solutions/terminal-automation/rtls-equipment-control.jsp