UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior

Departamento de Teoría de la Señal y Comunicaciones

Proyecto Final de Carrera

Ingeniería de Telecomunicación

Performance Analysis of the Contention Access Period in the slotted IEEE 802.15.4 for Wireless Body Sensor Networks

Autor: Manuel Aymerich de Franceschi

Tutor: Dr. Dña. Nadia Khaled

Leganés, Mayo 2009

TÍTULO: *Performance Analysis of the Contention Access Period in the slotted IEEE 802.15.4 for Wireless Body Sensor Networks.*

AUTOR: Manuel Aymerich de Franceschi {maymerich@tsc.uc3m.es}

TUTOR: Dr. Nadia Khaled {nkhaled@tsc.uc3m.es}

La defensa del presente Proyecto Fin de Carrera se realizó el día 21 de Mayo de 2009; siendo calificada por el siguiente tribunal:

PRESIDENTE: Prof. Dr. Jesús Cid Sueiro *Dpto. TSC, Escuela Politécnica Superior. UC3M.*

SECRETARIO Rocío Arroyo Valles *Dpto. TSC, Escuela Politécnica Superior. UC3M.*

VOCAL Dr. Joaquín Recas Piorno *Dpto. ACyA, Facultad de Informática. UCM.*

Habiendo obtenido la siguiente calificación:

CALIFICACIÓN:

Presidente **Secretario** **Vocal**

To my niece Sofía and my
nephews Dado, Ian, Lorenzo and Hugo.

To the MTA.

"Your task is not to foresee the future, but to enable it".

Citadelle (1948)

Antoine Saint-Exupéry

Acknowledgments

The finalization of this Master thesis puts an end to my university studies. I would like to take this opportunity to express my sincere gratitude to all those who have lent their support along these years.

First of all, I would like to express my gratitude and appreciation to my master thesis supervisor Nadia Khaled, for her exceptional guidance, advice, patience and everlasting effort in making me think as a scientist and not as a programmer. It has been a great pleasure to work with a person with such a great natural talent.

I would also like to thank my classmates who have been all along with me in this incredible educational journey. Many thanks to Ire, Ines, Lucas, Blanca, Laurita, David and Patricia for all the great moments we spent together in campus and outside the university campus. Special thanks to my good 'buddy' Ramón from whom I have learned and shared so many things in life; thank you Ramón for being such a great person, such a good friend of your friends and such a great engineer.

The countless hours spent studying at Pozuelo's Library were always pleasant thanks to the support of Guille, Manu DJ, Manu Z., Jesús B., Jesús M., the Rojo brothers, Santi L., Bea, Mer Soto and Luquitas (thank you for solving me those microwave engineering doubts!).

I am also really thankful to my close friends, who have followed up all my career work from the

very beginning with great encouragement. Their company has always been an enlightening inspiration. Thanks to Alvaro Aterido, Alvaro Domínguez, Mer, Nacho Lorenzo, Sonso, Alex Rico, Serafín, Alberto, Alex Fdez., Ian, Alfonso, Rodrigo, Gonzalo M., P.Carlos, P.Lorenzo, Paco, Hector, Javi Braci, Diego Navalpotro, Carlos Gimeno, Susana and Diego Cienfuegos

The finalization of my career would have been impossible without the help and love of my fiancé Pía. She is the author of the artistic portions of most of the figures in this thesis. More importantly, she is the rock upon which I stand. Many thanks, for your endless patience and encouragement throughout my studies. Thanks for waiting for the publication of each exam grade as if it was yours. This thesis is as much of your work as it is mine.

Lastly, my most emotional thanks goes to my family. They are the truly driving forces of my determination and achievement. Thanks to Grandmère. Thanks to my siblings María+Juan, Edu+Rosa, Pablo, Ana and Clara. Thanks to my niece Sofía and my nephews Dado, Ian, Lorenzo and Hugo, who have been the greatest gift of these years. Foremost, I would like to thank my parents, whom since I was a small child have taught me almost everything I know; your sacrifice is my strength and optimism.

Abstract

Wireless body sensor networks (WBSN) are a particular type of wireless sensor networks (WSN) that are becoming an important topic in the technological research community. Advances in the reduction of the power consumption and cost of these networks have led to solutions mature enough for their use in a broad range of applications such as sportsman or health monitoring.

The development of those applications has been stimulated by the finalization of the IEEE 802.15.4 standard, which defines the medium access control (MAC) and physical layer (PHY) for low-rate wireless personal area networks (LR-WPAN). One of the MAC schemes proposed is slotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). This project analyzes the performance of this MAC, based on a state-of-the-art analytical model for a star topology, which captures the behavior of the MAC using two Markov chain models; the per-node state model and the channel state model. More importantly, we extend this model to include acknowledged traffic. The impact of including acknowledgments is evaluated in terms of energy consumption, throughput and latency.

The performance predicted by the analytical model has been extensively verified with simulations using the ns-2 IEEE 802.15.4 contributed module. Throughput, energy consumption and latency analysis is performed. Additionally, we have simulated a statistical channel model describing the radio channel behavior around the human body to calculate the packet error rate (PER) found in a typical WBSN under the aforementioned standard. This PER is then introduced into our analytical model.

Contents

List of Figures

List of Tables

Acronyms

ACK Acknowledgement

BE Backoff Exponent

BI Beacon Interval

BO Beacon Order

BSN Body Sensor Network

BPSK Binary Phase-Shift Keying

CAP Contention Access Period

CCA Clear Channel Assessment

CDMA Code Division Multiple Access

CFP Contention Free Period

CONSER Collaborative Simulation for Education and Research

CSMA/CA Carrier Sense Multiple Access/Collision Avoidance

CSMA/CD Carrier Sense Multiple Access/Collision Detection

CW Contention Window

DARPA Defense Advanced Research Projects Agency

DoD U.S. Department of Defense

DSSS Direct Sequence Spread Spectrum

ECG Electrocardiogram

ED Energy Detection

FDMA Frequency Division Multiple Access

FCS Frame Check Sequence

FFD Full Function Device

GE General Electric

GTS Guaranteed Time Slot

ICSI International Computer Science Institute

ID Identifier

IEEE Institute of Electrical and Electronics Engineers

IFS Inter-frame Spacing

ISI USC Information Sciences Institute

ISM Industrial, Scientific and Medical

LBNL Lawrence Berkeley National Laboratory

LIFS Long Inter-frame Spacing

LOS Line-of-Sight

LQI link quality indication

LR-WPAN Low Rate-Wireless Personal Area Network

MAC Medium Access Control

MIT Massachusetts Institute of Technology

NB Number of Backoff (periods)

NAM Network Animator

NS Network Simulator

NSF National Science Foundation

O-QPSK Offset Quadrature Phase-Shift Keying

PAN Personal Area Network

PER Packet Error Rate

PHY Physical Layer

POS Personal Operating Space

QPSK Quadrature Phase Shift Keying

RF Radio Frequency

RFD Reduced Function Device

RSSI Received Signal Strength Indication

RX Receive or Receiver

SAMAN Simulation Augmented by Measurement And Analysis for Networks

SD Superframe Duration

SHR Synchronization Header

SIFS Short Inter-frame Spacing

SO Superframe Order

SpO2 Saturation of Peripheral Oxygen

TDMA Time Division Multiple Access

TRX Transceiver

TX Transmit or Transmitter

UC University of California

USC University of Southern California

VINT Virtual InterNetwork Testbed

WBSN Wireless Body Sensor Networks

WPAN Wireless Personal Area Network

WSN Wireless Sensor Network

Part I

Motivation and Objectives

CHAPTER 1

Motivation and Objectives

1.1 Motivation

Wireless body sensor networks (WBSN) are a particular type of wireless sensor networks (WSN) that thanks to the development of innovative wearable, wireless and implantable biosensors have gained tremendous international interest in recent years. The applications of WBSN extend from in-vivo monitoring and intervention to everyday healthcare, as well as fitness, sport and security [1].

Dr. Leonard Fass, Director of Academic Relations at GE Healthcare, and one of the joint organizers of the BSN 2007 International Workshop, was interviewed at [1] reporting that:

> "*One of the greatest barriers to the adoption of emerging BSN technologies is the whether or not they can be integrated with existing systems, under common standards*".

The finalization of the IEEE 802.15.4 standard, which defines the medium access control (MAC) and physical layer (PHY) for sensor networks, has stimulated the development of general WSN applications. Even more, the IEEE 802.15.4 standard is poised to become the global standard for low data rate, low energy consumption WSN.

However, it remains to be analyzed and discussed whether or not the IEEE 802.15.4 standard meets the general WBSN application requirements. In particular, a deep analysis of the novel slotted access protocol featured in the contention access period (CAP) of its beacon enabled mode for its use in

WBSN remains to be performed.

1.2 Objectives

The main objective of this project is to analyze the IEEE 802.15.4 standard working under a WBSN application scheme. In WBSN applications, information is gathered from human body monitoring or the environment, and then forwarded to a coordinator under a star network topology. Most of the communication is uplink (nodes-to-coordinator), as opposed to downlink (coordinator-to-nodes). Because of this, we will concentrate our analysis on the uplink mode only. More precisely, we will analyze the access protocol featured in the contention access period (CAP) of its beacon enabled mode: slotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

Our study is based on the work of Ramachandran et al. [2] which develops a frame work based on two Markov chains; the node-state chain and channel-state chain. This thesis extends this Markov-chain-based analytical model of the IEEE 802.15.4 CAP for acknowledged traffic and under a WBSN channel.

1.3 Structure of this Thesis

This Thesis is organized into 8 chapters and IV well differentiated parts. After this initial introductory part, we present the second part dedicated to the state-of-the-art, organized into two chapters; the first one tackles the medium access control (MAC) design in WBSN and the second one provides an overview of the most relevant aspects of the IEEE 802.15.4 standard, and the underlying slotted CSMA/CA algorithm.

In Part II, in chapters 4 we present the analytical model of the IEEE 802.15.4 slotted CSMA/CA for WBSN. In chapter 5 we describe a statistical channel model describing the radio channel behavior around the human body to estimated the needed transmit power and the packet error rate (PER) found in a typical WBSN, to later include both on the analytical model. In Chapter 6, the ns-2 simulation environment is described.

On Part III, Chapter 7 presents in detail the results of the analytical model behavior in terms of throughput, energy consumption and latency. The validity of these results is verified extensively comparing with ns-2 simulation results. This Thesis finishes in Chapter 8, which includes the conclusions of this work an identifies topics for future research.

Part II

State-of-the-Art

CHAPTER 2

MAC Design in Wireless Body Sensor Networks

2.1 Introduction to Wireless Sensor Networks

The last decade has witnessed an astounding miniaturization and cost reduction in the semiconductor industry. Advances in microelectromechanical systems, low power and highly integrated digital electronics, tiny microprocessors, wireless communications, etc, have permitted to develop many kinds of versatile sensors and actuators [3] [4]. These sensors may collaborate to accomplish a common task such as environment monitoring, asset tracking, home automation, personal health monitoring, etc. In order to complete this work, they are configured to form *wireless sensor networks* (WSN). Typical applications of WSN are: military, environmental, health, home and commercial [5] [6]. Despite the diversity of these applications, the following requirements are true for all of them:

- low power
- low cost
- wireless
- self-configuring

Among these requirements, the low power consumption requirement is most important.

In WSN the nodes must be self-organized in such a manner that they can do cooperative processing to accomplish tasks that they cannot do individually. Because of this, sensor networks face many problems that do not arise in other types of networks. Limited energy budget, limited processing resources, decreased reliability, and a typically higher density and number of nodes than found in conventional networks, is just a small portion of the problems that have to be considered when developing protocols for WSN [5].

2.2 A Particular Instantiation of WSN: Wireless Body Sensor Networks

At the same time, a significant progress has been carried out in wearable and implantable biosensors [7]. Many of them are used for sensing and monitoring vital health parameters at hospitals, for performance monitoring in athletes, security, etc. Sophisticated WSN could be implemented with these nodes. But technically, since the nodes are either implanted or close to human bodies, it would be more accurate to name this type of WSN as *Wireless Body Sensor Networks* (WBSN).

More precisely, the term body sensor networks (BSN) was coined by Prof. Guang-Zhong Yang of Imperial College in 2002. Since then, BSN technology has gained tremendous international interest from researchers both in academia and industry. Proof of this is that since 2004, an International Workshop on BSN has been held at some of the most prestigious Universities in the world: Imperial College in London UK (2004, 2005), MIT in Boston USA (2006), RWTH Aachen University in Aachen Germany (2007), and Chinese University in Hong Kong China (2008). In 2009 the International Workshop on BSN[1] will be held in UC Berkeley USA.

By now, it should be clear that while WSN technology for large scale environment monitoring continues to expand, it does not necessarily tackle the challenges associated with human body monitoring. The human body is a complicated environment that responds to and interacts with its external surroundings, but is in a way separated and self contained. Human body monitoring using wireless sensors may be achieved by attaching these sensors to the body surface as well as implanting them on tissues. In essence, the human body environment is not only smaller scale, but also requires a different type and frequency monitoring, with appreciation of different challenges than those faced by WSN. Therefore there are many aspects in WSN technology development that are not ideally suited to monitoring the body and its environment. This is what has led to the development of WBSN [8].

[1] For more information go to http://bsn2009.org/

Then, the discussion about WBSN is not just a question of terminology, but about a new concept, a different type of wireless networks that require a particular design approach. The ultimate proof of that, is the recent launch (at the end of 2007) of the Task Group 6 within the IEEE 802.15 Working Group, which will propose suitable PHY and MAC standards for body area networks.

2.2.1 Differences between wide-scale WSN and WBSN

Although the challenges faced by WBSN are in many ways similar to WSN, the reader should not forget that there are intrinsic differences between the two, which require special attention. In table 2.1, we present a summary of the differences between WSN and WBSN [8].

Some of the key research focuses of the WBSN community includes the latest technological developments of:

- Hardware considerations: low power RF transceiver, energy scavenging, battery technology, miniaturization, system integration, process and cost of manufacturing.
- Standards and light-weight communication protocols.
- WBSN architecture and platforms.
- Wearable and implantable sensor integration and development platforms.
- Quality of service, trust and security issues.
- Data inference, knowledge discovery, and prediction.
- Context awareness and multi-sensor data fusion.
- Clinical applications of WBSN.
- Biocompatibility and materials.
- Integration with ambient sensing with applications in smart dwellings, and home monitoring.

2.2.2 Topology of a WBSN

A WBSN usually consists in implantable or wearable biosensors, such as glucose sensors, blood pressure and oxygen saturation (SpO2) sensors, temperature, electrocardiogram sensor (ECG) and

Table 2.1: Differences between wide-scale WSN and WBSN

Challenges	WSN	WBSN
Scale	Environment being monitored (m/Km)	Human body parts (mm/cm)
Node Number	Greater number of nodes required for accurate, wide area coverage	Fewer, more accurate sensor nodes required (limited space)
Node Function	Multiple sensors, each perform dedicated tasks	Single sensors, each perform multiple tasks
Node Accuracy	Large node number compensates for accuracy and allows result validation	Limited node number with each required to be robust for accurate
Node Size	Small size preferable but not a major limitation in many cases	Pervasive monitoring and need for miniaturization
Dynamics	Exposed to weather, noise, and asynchrony	Exposed to predictable environment but motion artifacts is a challenge
Event Detection	Early adverse event detection desirable	Early adverse events detection vital
Variability	Likely to have fixed or static structure	Biological variation and complexity
Data Protection	Lower level wireless data transfer security required	High level wireless data transfer security required
Power Supply	Accessible and likely to be changed more easily	Inaccessible and difficult to replace in implanted setting
Power Demand	Greater	Lower
Energy Scavenging	Solar, wind	Motion and thermal (body heat)
Access	Easily replaceable or disposable	Difficult and requires biodegradability
Wireless Technology	Bluetooth, Zigbee, WLAN, GPRS	Low power wireless required, with more challenging MAC protocols
Data Transfer	Loss of data is likely to be compensated by number of sensors used	Loss of data more significant, and may require additional measures to ensure QoS

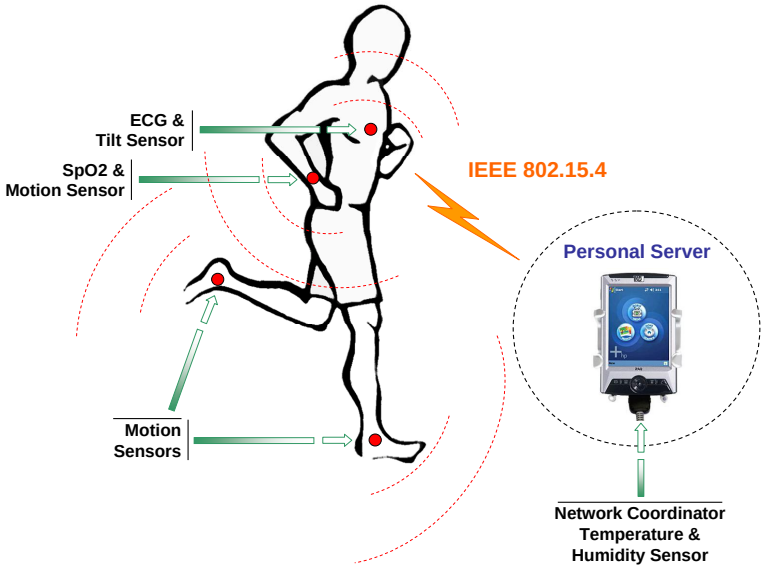

Figure 2.1: *Illustration of a WBSN*

even ingestible camera pills [8]. These sensors continuously monitor vital signs and report data to a powerful external device, such as a PDA, cell phone or bedside monitor station.

WBSN application designs suggests the use of a single hop star topology as shown in figures 2.1 and 2.2. The main advantage of using such topology is that an external coordinator can be used with access to rechargeable power supply. All devices will be connected to a single central controller often referred to as coordinator or master. The peripheral nodes are called slaves. Typically, slaves can only communicate with the coordinator. Therefore, communication between slaves requires passing all data through the coordinator.

The advantages of this topology are:

- Simplicity.
- Low power consumption.
- Low latency.
- Centralized system.

The disadvantages are:

- Dedicated central node.
- Single point of failure.

- Poor scalability.

- Inefficient slave-to-slave communication.

Using WBSN enables us to reduce the cost of health care systems, make ubiquitous mobile monitoring possible and give the patient involved much more freedom and comfort.

2.3 Medium Access Control in WBSN

2.3.1 Introduction

Medium access control (MAC) protocols have been created to help nodes coordinate when accessing the limited and shared resources of the channel. Because of this, an efficient MAC protocol possesses the greatest capability to decrease the energy consumption of the transceiver since it directly controls its operation [9].

While traditional MAC protocols are designed to maximize packet throughput, reduce latency and provide fairness, protocol design for WBSN focuses on minimizing energy consumption and latency. As mentioned in the introduction of this chapter, limited energy resources provide the primary constraint for WBSN. However, this must not overshadow another key objective: latency. Since WBSN are envisioned to handle critical scenarios where data retrieval time is crucial, good effort should be put into reducing the service time of critical frames. So when designing MAC protocols for a WBSN we must focus on reducing unnecessary energy expenditure due to wireless communication as well as insuring that the delay-sensitive data are serviced within the application constraints.

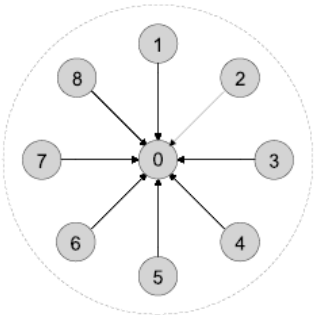

Figure 2.2: *Star Network Topology*

2.3.2 Traditional MAC Design

MAC protocols have been extensively studied in traditional areas of wireless communications. We find two types of MAC protocols :

- *Schedule based*: TDMA, FDMA and CDMA. Their basic idea is to avoid interference by scheduling nodes onto different sub-channels that are differentiated in time, frequency or by orthogonal codes. Since these sub-channels do not interfere with each other, MAC protocols in this group are largely collision-free.

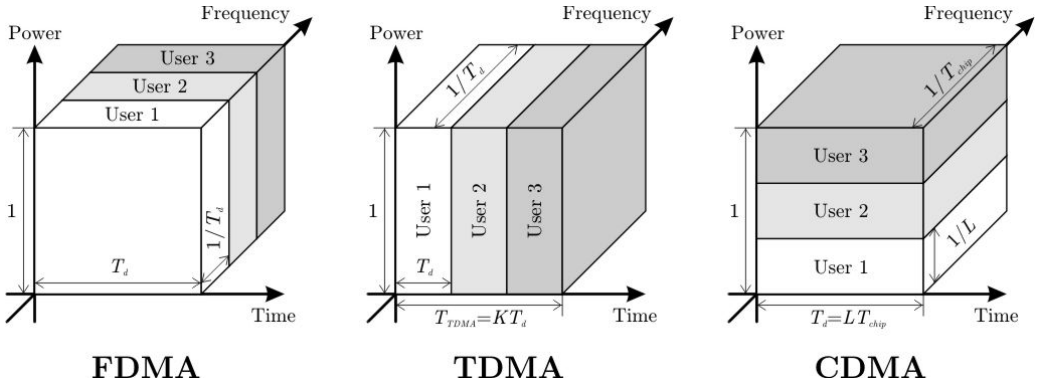

Figure 2.3: *Scheduled based access techniques*

- *Contention based*: CSMA, CSMA/CD, CSMA/CA. Rather than pre-allocate transmissions, nodes compete for a shared channel, resulting in probabilistic coordination. Collision happens during the contention procedure in such systems.

But sensor networks are different from traditional wireless networks in many ways. General wireless MAC protocols attempt to provide high throughput, low latency, fairness, and mobility management, but often have little or no consideration for energy conservation. Sensor network MAC protocols, however, must provide the best performance at the smallest amount of energy consumption due to the limited energy resources available to each sensor node. Most nodes in sensor networks are normally battery powered, and it is often very difficult or impossible to change their batteries. On the other hand, WBSN nodes are often deployed around the human body with little pre-planning. Therefore they must organize themselves into a communication network. And finally, most traffic in the network is triggered by sensing events, and it can be extremely bursty.

MAC protocols for sensor networks will be based on the same basic principles as traditional systems. However, the main difference will be on the choice of the particular MAC mechanisms for each specific

application. The contention parameters and contention procedures will be optimized for that particular scenario. In the case of WBSN, the MAC parameters need to be optimized for low-power, low latency, periodic traffic, with possibility of non-period high priority bursts (due to critical health alterations).

2.3.3 MAC Design in WBSN

The most common approach to reduce energy consumption in WBSN involves duty cycling the sensor node hardware between high power active states and low power sleep states. Sensor nodes cannot function in the network while asleep. But putting the sensor node to sleep when idle, i.e. no data to be transmitted and not expecting to receive any traffic, can dramatically increase a sensor's lifetime.

Another important consideration is the type of traffic generated on a sensor network. WBSN traffic differs from the communication patterns existing in other networks. Human monitoring with a sensor means periodically sending data to a central entity for collection and analysis. These devices individually produce traffic at periodic rates with small payloads. Exceptionally, some non-periodic high priority traffic may appear at some points of the communication, triggered by abnormalities or hazardous events in human health. In all cases, the traffic pattern will be mostly uplink from the sensors to the central node.

The basic trade-offs we must consider while designing a MAC protocol for a WBSN are the following [9]:

- *Collision Avoidance*: is the basic task of all MAC protocols. Collisions are not always completely avoided in regular contention-based operation, but should be as infrequent as possible.
- *Energy efficiency*: prolonging the lifetime of each node is a critical issue.
- *Scalability and adaptivity*: the topology of a WBSN may change in time. Sensors might be added or removed. A MAC protocol should adapt to those changes in a nice fashion.
- *Latency*: refers to the delay between the time when a sender has a packet to send and the time the packet is successfully received by the receiver. Normally, WBSN applications are critical in terms of latency. However, network speed is typically orders of magnitude faster than the speed of a physical magnitudes present on a WBSN.

The reader should note, that each one of the attributes above described define the characteristics of a MAC protocol.

2.3.4 Energy Efficiency in MAC Protocols

A complete energy management scheme must consider all sources of energy consumption, not just the radio. MAC energy control must be integrated with control of the CPU and other components. When trying to design an energy efficient MAC protocol, we must analyze closely the major causes of energy waste [10]:

- *Collision*: the sharing of a single communication channel by multiple nodes induces the phenomenon of packet collision. Follow-on retransmissions consume energy. All MAC protocols try to avoid collisions one way or another. Collision is a major problem in contention protocols, but is generally not a problem in scheduled protocols.
- *Idle listening*: it happens when the radio is listening to the channel to receive possible data. Most sensor networks are designed to operate over long time, and the nodes will be in idle state for long time. In such cases, idle listening is a dominant factor of radio energy consumption.
- *Overhearing*: this occurs when a node receives packets that are destined to other nodes. Overhearing unnecessary traffic can be a dominant factor of energy waste when traffic load is heavy and node density is high.
- *Packet overhead*: sending, receiving, and listening for control packets consumes energy. Since control packets do not directly convey data, they also reduce the effective throughput.

A MAC protocol achieves energy savings by controlling the radio to avoid or reduce energy waste from the above sources. Turning off the radio when it is not needed is an important strategy for energy conservation.

2.4 MAC Scheduled Protocols

In schedule protocols, duty cycling between active and sleep states is automatic. Because of this, TDMA has attracted the attention of sensor network protocol designers. The TDMA operating mode is well known; the channel is divided into N subchannels called time slots. Within each slot, only one node is allowed to transmit. The N slots comprise a frame, which repeats cyclically. Due to simplicity, TDMA is typically used in wireless scenarios with a star topology scheme, such as the one of WBSN. Nodes communicate only with the coordinator. And therefore there is no direct, slave-to-slave communication between nodes, without the intervention of the coordinator node.

The greatest advantage of TDMA in WBSN schemes, is energy efficiency. TDMA directly supports low-duty-cycle operations on nodes. However, conventional TDMA has some disadvantages that limit its use in sensor networks. TDMA protocols have limited scalability and adaptability to changes on number of nodes. When new nodes join in or old nodes leave the network, the master must readjust frame length or slot allocation. Finally, another important disadvantage is that TDMA requires fine-grained time synchronization.

2.4.1 Energy Conservation in Scheduled Protocols

As seen, TDMA is an attractive solution for WBSN applications because of its energy efficiency. Since slots are pre-allocated to individual nodes at initialization, they are collision-free. There is no energy wasted on collisions due to channel contention. Another advantage is that in TDMA a node only needs to turn on its radio during the slot that it is assigned to transmit or receive. Thus, low duty cycles are directly supported. Finally, overhearing can be easily avoided by turning off the radio during the slots that correspond to other nodes.

Sophisticated WBSN MAC protocols may be based in TDMA, with additional features that complement its benefits and fit the particular requirements of WBSN.

2.5 MAC Contention Based Protocols

In contention protocols, a common channel is shared by all nodes, and it is allocated on-demand. Thus, a contention mechanism is needed to decide which node has the right to access the channel at any moment.

Contention protocols have several advantages compared to scheduled protocols. First, because contention protocols allocate resources on-demand, they can scale more easily across changes in node density. Second, they can be more flexible as topologies change. Finally, contention protocols do not require fine-grained time synchronizations as in TDMA protocols.

The major disadvantage of these type of protocols are its inefficient usage of energy. It normally has all the sources of energy waste we discussed in section 2.3.4: nodes listen at all times and collisions and contention for the media can waste energy. Overcoming this disadvantage is required if contention-based protocols are to be applied to long-lived sensor networks.

2.5.1 Energy Conservation in Contention Protocols

Additional techniques need to be proposed to improve energy consumption in conventional contention-based protocols. Normally, the basic approach followed consists in putting the radio into sleep state when it is not needed. Following a similar path as TDMA, sleep/wake scheduling structures improve energy consumption. However, each particular implementation depends on the application requirements of the scenario.

2.6 Choice of IEEE 802.15.4

The success of WBSN will depend on the success of the standardization efforts to unify the technology and avoid incompatible protocols that will limit the penetration index of this technology in commercial applications.

Therefore, for our analysis, we must carefully choose from all the available MACs the one which offers the most promising potencial. Some of the MACs available in the scientific literature are: S-MAC [11], T-MAC [12], DSMAC [13], B-MAC [14], G-MAC [15], IEEE 802.15.4 [16], etc. They all share many similarities in their mechanisms. Essentially, these mechanisms are designed to enable a trade-off amongst the parameters of throughput, collision avoidance, energy, scalability and latency. Thus, no matter which MAC is chosen, a gain in performance of one dimension means a loss in another. A particular choice would depend on the specific circumstances of deployment scenarios and applications.

Among all, the IEEE 802.15.4 MAC is the most promising for wireless sensor networks. It is well-layered and provides a combination of link management mechanisms that can be enabled selectively depending on the user configuration. This means that it places adaptability tools at the hands of the user. It also has a comprehensive specification addressing basic deployment requirements such as network configuration, management and security services to guarantee data confidentiality and integrity. Its beacon enabled mode allows cycling between high power active states and low power sleep states, while it also includes an optional inactive period to further increase the sensor's life-time. From the perspective of adoption, IEEE 802.15.4 standard is being rapidly adopted and commercialized, leading to mature rapid prototyping products.

The above mentioned reasons have made us decided to choose and analyze the IEEE 802.15.4 for WBSN scenarios applications. The details of this standard are given in the next chapter.

CHAPTER 3

Overview of IEEE 802.15.4 LR-WPAN

3.1 Introduction

A *low rate wireless personal area network* (LR-WPAN) is, as defined in the standard, a simple, low-cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements. The main objectives of an LR-WPAN are ease of installation, reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol. LR-WPANs support simple devices that consume minimal power and typically operate in the *Personal Operating Space* (POS) of 10 meters or less.

The IEEE 802.15.4 standard [16], defines the physical layer (PHY) and medium access control sublayer (MAC) specifications for these LR-WPANs. There are 14 PHY primitives and 35 MAC primitives defined in IEEE 802.15.4. This total number of primitives is only about one third the number of primitives defined in Bluetooth for example, which makes IEEE 802.15.4 very suitable for simple devices with limited memory and computational capacity. The following sections describe the operation mode of this protocol as well as the most important parameters to correctly configure it for WBSN application requirements.

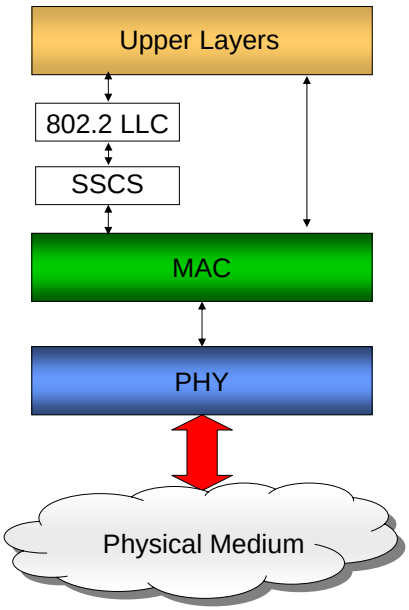

Figure 3.1: *IEEE 802.15.4 Architecture*

3.2 Network components

Two different device types can participate in an IEEE 802.15.4 network; a *full-function device* (FFD) and a *reduced-function device* (RFD). The FFD can operate in three modes:

1. *The Personal Area Network (PAN) Coordinator*: the principal controller of the PAN. This device identifies its own network as well as its configurations, to which other devices may be associated.

2. *The Coordinator*: provides synchronization services through the transmission of beacons. This device should be associated to a PAN Coordinator and does not create its own network.

3. *The End Device*: a device which does not implement the previous functionalities and should associate with a coordinator or PAN coordinator before interacting with the network.

FFDs have the capability to communicate with any device in a network within range of them, while RFDs are only able to directly communicate with FFDs. An RFD is intended for applications that are extremely simple, such as a light switch or a passive infrared sensor; they do not have the need to send large amounts of data and may only associate with a single FFD at a time. Consequently, the RFD can be implemented using minimal resources and memory capacity.

3.3 Network topology

Depending on the application requirements, an IEEE 802.15.4 LR-WPAN may operate in either of two topologies:

- star topology
- peer-to-peer topology

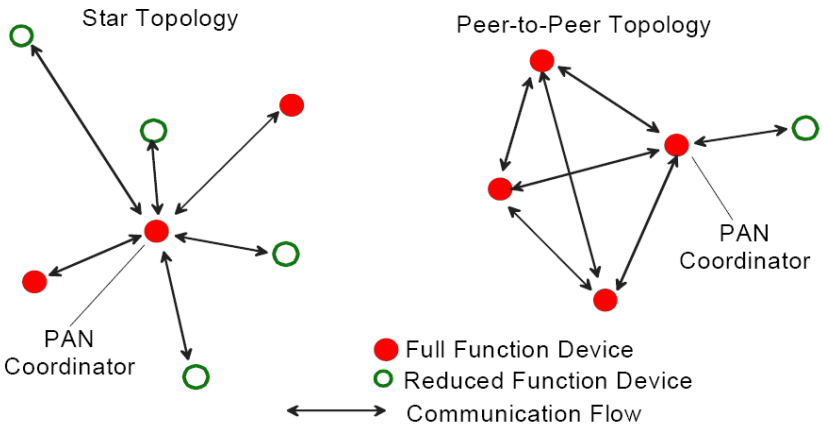

Figure 3.2: *Star and peer-to-peer topology*

Both are shown in figure 3.2. In the *star topology* the communication is established between devices and a single central controller PAN coordinator. A device typically has some associated application and is normally the initiation point for network communications. The PAN coordinator is the primary controller of the PAN. This means that even though this PAN coordinator might have a specific application, it is also used to initiate, terminate, or route communication around the network. It might often be mains powered, while the devices will most likely be battery powered. Applications for WBSN such as personal health care, will usually be deployed using this communication scheme.

The *peer-to-peer topology* also has a PAN coordinator; however, it differs from the star topology in that any device may communicate with any other device as long as they are in range of one another. Peer-to-peer topology allows more complex network formations to be implemented, such as mesh networking topology. Applications such as industrial control and monitoring, wireless sensor networks, asset and inventory tracking, intelligent agriculture, and security would benefit from such a network topology.

3.4 PHY Layer

The PHY layer specification dictates how IEEE 802.15.4 devices may communicate with each other over the wireless channel. It allows for the use of three frequency bands with varying data rates. The bit rates are 20 Kb/s in the European 868 MHz band (868-868.6 MHz), 40 Kb/s in the North American 915 MHz band (902-928 MHz), and 250 Kb/s in the worldwide 2.45 GHz band (2.4-2.4835 GHz). The physical data rates achieved for each frequency are summarized on table 3.1.

Table 3.1: IEEE 802.15.4 Physical Data Rates

Frequency Band	Bit Rate	Symbol Rate
868-868.6 MHz	20 Kb/s	20 Ks/s
902-928 MHz	40 Kb/s	40 Ks/s
2.4-2.4835 GHz	250 Kb/s	62.5 Ks/s

There is a single channel between 868 and 868.6 MHz with a data rate of 20 kbps, 10 channels between 902 and 928 MHz with 40 kbps, and 16 channels between 2.4 and 2.4835 GHz each supporting a data rate of 250 kbps. All of these frequency bands are based on the Direct Sequence Spread Spectrum (DSSS) spreading technique. The 865 MHz and the 915 MHz radios map each data symbol onto a 15-chip PN sequence, followed by binary phase-shift keying (BPSK) for chip modulation. The 2.45 GHz ISM radio band, on the other hand, maps each 4 bits of information onto a 32 chip PN sequence followed by offset orthogonal phase shift keying (O-QPSK).

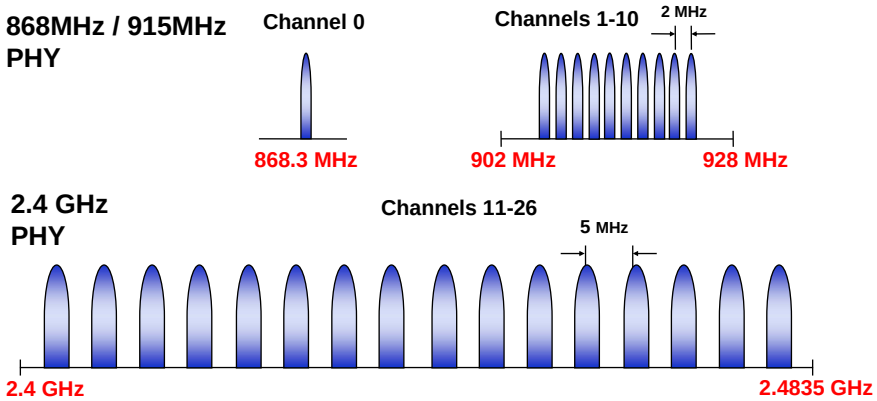

Figure 3.3: *Operating frequencies and bands*

The PHY IEEE 802.15.4 layer is responsible for:

- *Activation and deactivation of the radio transceiver:* The radio transceiver may operate in one of three states: transmitting, receiving or sleeping. Upon request of the MAC sub-layer, the radio is turned ON or OFF. The turnaround time from transmitting to receiving and vice versa should be no more than 12 symbol periods, according to the standard (each symbol corresponds to 4 bits).

- *Energy Detection* (ED): Estimation of the received signal power within the bandwidth of an IEEE 802.15.4 channel. This task does not make any signal identification or decoding on the channel. The energy detection time should be equal to 8 symbol periods. This measurement is typically used by the Network Layer as a part of channel selection algorithm or for the purpose of Clear Channel Assessment (CCA), to determine if the channel is busy or idle.

- *Link Quality Indication* (LQI): Measurement of the Strength/Quality of a received packet. It measures the quality of a received signal. This measurement may be implemented using receiver ED, a signal to noise estimation or a combination of both techniques.

- *Clear Channel Assessment* (CCA): Evaluation of the medium activity state: busy or idle. The CCA is performed in three operational modes:

  1. Energy Detection mode: the CCA reports a busy medium if the detected energy is above the ED threshold.
  2. Carrier Sense mode: the CCA reports a busy medium only is it detects a signal with the modulation and the spreading characteristics of IEEE 802.15.4 and which may be higher or lower than the ED threshold.
  3. Carrier Sense with Energy Detection mode: this is a combination of the aforementioned techniques.

  The CCA reports that the medium is busy only if it detects a signal with the modulation and the spreading characteristics of IEEE 802.15.4 and with energy above the ED threshold.

- *Channel Frequency Selection:* The IEEE 802.15.4 defines 27 different wireless channels. Each network can support only part of the channel set. Hence, the physical layer should be able to tune its transceiver into a specific channel when requested by a higher layer.

A PHY packet includes a 5-byte *synchronization header* (SHR) which allows devices to synchronize with the bit stream which forms the message.

3.5 MAC Layer

The basic services provided by the MAC are beacon generation and synchronization. It supports node association and disassociation, optional device security, manages channel access via CSMA/CA, maintains *Guaranteed Time Slot* (GTS) allocation, provides message validation and message acknowledgment. It may work in either beacon enabled mode or non-beacon enabled mode.

3.5.1 MAC Frame Structure

The frame structures of IEEE 802.15.4 have been designed to keep the complexity to a minimum while at the same time making them sufficiently robust for transmission on a noisy channel. It defines four frame structures:

- A beacon frame, used by a coordinator to transmit beacons.
- A data frame, used for all transfers of data.
- An acknowledgment frame, used for confirming successful frame reception.
- A MAC command frame, used for handling all MAC peer entity control transfers.

For complete details of this MAC frames refer to [16]. The total description of the fields on these frames is out of the scope of this thesis.

3.5.2 Operational Modes

As we aforementioned, the MAC protocol supports two operational modes:

- *Non beacon-enabled mode*: In this mode, a network node can send data to the coordinator at will using unslotted CSMA/CA. However, to receive data from the coordinator the node must power up and poll the coordinator. To achieve the required node lifetime, the polling frequency must be pre-determined by power reserves and expected data quantity. The advantage of non-beacon mode is that the node's receiver does not have to regularly power-up to receive the beacon. The disadvantage is that the coordinator cannot communicate at will with the node but must wait to be invited by the node to communicate.
- *Beacon-enabled mode*: Communication is controlled by the network coordinator, which transmits regular beacons for device synchronization and network association control. The network coordinator defines the start and end of a superframe by transmitting a periodic beacon. The length

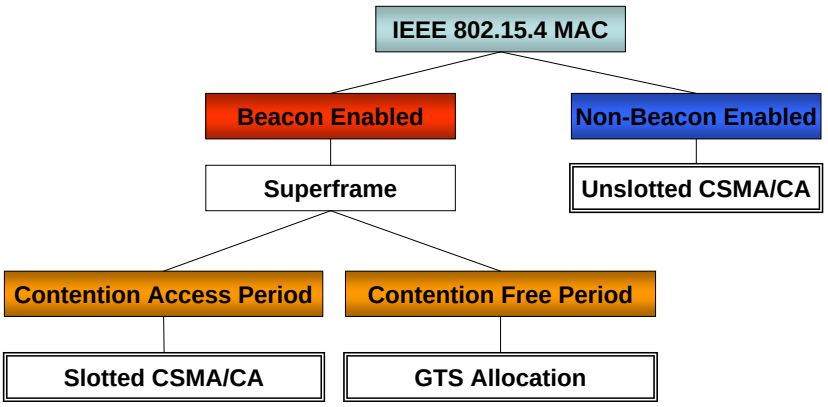

Figure 3.4: *IEEE 802.15.4 Operational Modes*

of the beacon period and hence the duty cycle of the system can be defined by the user between certain limits as specified in the standard. The advantage of this mode is that the coordinator can communicate at will with the nodes. The disadvantage is that the nodes must wake up to receive the beacon. However this mode of operation is still considered as low-power since nodes can put their radio to sleep in between beacons when they don't have data packets to transmit.

All devices must go through an initial association phase in order to become part of a PAN. To support self-configuration, IEEE 802.15.4 embeds association and disassociation functions in its MAC sublayer. Various configurations are also done during the association procedure, such as selecting a channel and an identifier (ID) for the PAN, determining whether beacon enabled mode or non-beacon enabled mode to be used, choosing the beacon order and superframe order in beacon enabled mode, and assigning a 16-bit short address for a device.

3.5.3 The Beacon-Enabled Mode: A Detailed Description

In beacon-enabled mode, beacon frames are periodically sent by the PAN coordinator every *Beacon Interval* (BI) to identify its PAN, to synchronize devices that are associated with it, and to describe the superframe structure (see Figure 3.5), comprising an active period and, optionally, an inactive period.

During the active period communication takes place, and during the inactive period, devices may turn off their transceivers in order to conserve energy.

The active portion of the superframe structure, is divided into 16 equally-spaced slots. Each one of this slots is further decomposed into smaller slots of length $320\mu s$ called "backoff periods" or "backoff

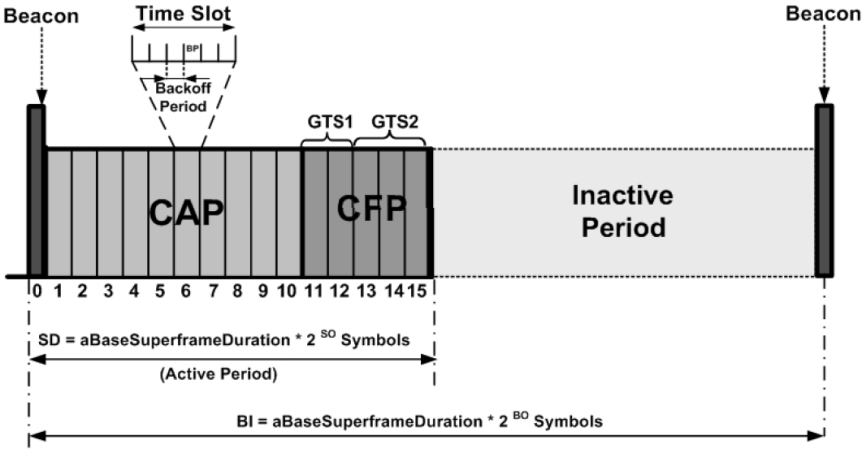

Figure 3.5: *IEEE 802.15.4 Superframe structure*

slots", as shown in Figure 3.5. These backoff periods are the time unit of this mode of operation. Furthermore, there are three well differentiated parts in the active portion of the superframe:

1. the *Beacon*: the beacon frame is transmitted at the start of slot 0. It contains the information on the addressing fields, the superframe specification, the GTS fields, the pending address fields and other PAN related information.

2. the *Contention Access Period* (CAP): Immediately following the beacon is the CAP. During this period, devices may communicate using a slotted CSMA/CA mechanism. This is similar to unslotted CSMA/CA, except that the back-off periods are aligned with slot boundaries, meaning that the devices are contending for the right to transmit over entire slots. The CAP must contain at least nine active period slots but may take up all 16.

3. the *Contention Free Period* (CFP): Following the CAP is an optional CFP, which may last up to seven active period slots. In the CFP, devices are allocated GTS slots by the PAN coordinator. During a GTS a device has exclusive access to the channel and does not perform CSMA/CA. Inside a GTS, a device may either transmit data to or receive data from its PAN coordinator, but not both. The length of a GTS must be an integer multiple of an active period slot. All GTSs must be contiguous in the CFP and are located at the end of the superframe active period. A device may disable its transceiver during a GTS designated for another device in order to conserve energy.

The relative size of each of the active and inactive periods is determined by the values of the

macBeaconOrder (BCO[1]), the *macSuperframeOrder* (SFO[1]), the *Beacon Interval* (BI) and the *Active Superframe Duration* (SD) as follows:

$$BI = (aBaseSuperframeDuration \times 2^{BCO}) \tag{3.1}$$

$$\text{where } 0 \leq BO \leq 14$$

$$SD = (aBaseSuperframeDuration \times 2^{SFO}) \tag{3.2}$$

$$\text{where } 0 \leq SO \leq 14$$

The BI defines the time in between two consecutive beacon frames. The SD defines the active portion in the BI. On the other hand it can be checked, that the length of the base super frame duration is equal to *aBaseSuperDrameDuration*=960 symbols or 15.36 ms (assuming we are working in the 2.4 GHz band). Low duty cycles can be configured by setting small values of the SFO as compared to BCO, resulting in greater sleep (inactive) periods. This feature is particularly interesting for WBSN applications, where energy consumption and network lifetime are main concerns. Additionally, the GTS mechanism is an attractive feature for time-sensitive WSNs, since it is possible to guarantee end-to-end message delay bounds.

3.6 Functional Overview in WBSN

3.6.1 Data Transmission

There are three different types of data transmission in the IEEE 802.15.4 standard:

1. Transmission from a device to the coordinator
2. Transmission from the coordinator to the device
3. Transmission between any two devices.

However in the considered WBSN, only device to coordinator transmissions will take place. Since we will typically have sensing devices and a single central node in a star topology scheme, all the data traffic will be uplink, and only the first transmission type will hold.

[1]We denote the beacon and superframe orders BCO and SFO instead of BO and SO as is done in the standard, to differentiate from Backoff, which we denote as BO

Steps followed by a transmission from a device to a coordinator n a beacon-enabled PAN (this sequence is summarized in Figure 3.6):

1. The device first listens to the network beacon.
2. When the beacon is found, the device synchronizes to the superframe structure.
3. The device transmits its data frame, using slotted CSMA/CA, to the coordinator.
4. On its turn, it will transmit the data to the coordinator.
5. The coordinator may acknowledge the successful reception of the data by transmitting an optional acknowledgment frame. Acknowledgment frames are transmitted without using the slotted CSMA/CA mechanism to access the channel.

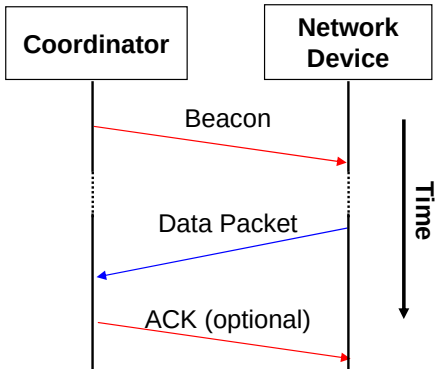

Figure 3.6: *Communication from a device to a coordinator in a beacon-enabled PAN*

3.6.2 Slotted CSMA/CA Mechanism

We have already discussed, how the MAC of the IEEE 802.15.4 is based on either slotted or unslotted CSMA/CA, depending on the network operation mode: beacon-enabled or non beacon-enabled modes, respectively. In WBSN we will exclusively work under the beacon-enabled mode, therefore we will now describe in detail the slotted CSMA/CA contention access procedure.

The CSMA/CA mechanism is based on backoff periods with a duration of 20 symbols (or 320 μs). Three variables are used to schedule medium access:

- *Number of Backoffs* (NB): representing the number of failed attempts to access the medium.

- *Contention Window* (CW): representing the number of backoff periods that the channel must be sensed before starting transmission.

- *Backoff Exponent* (BE): enabling the computation of the number of waiting backoff slots before attempting to access the medium during a given backoff stage.

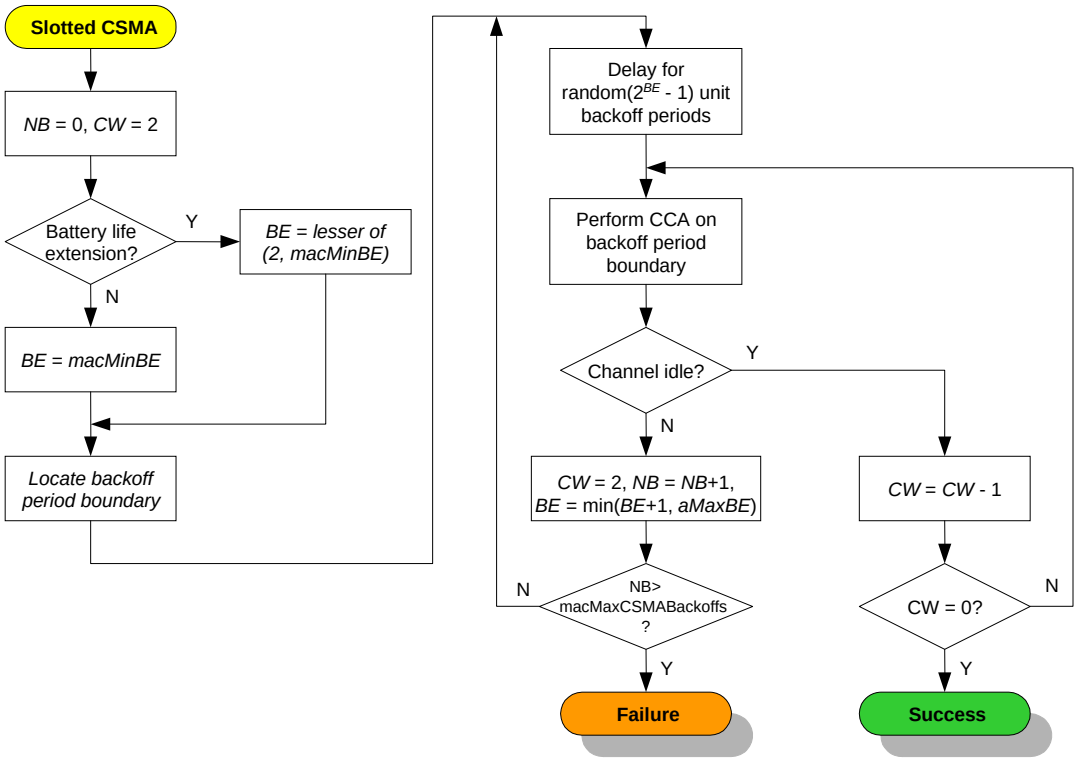

Figure 3.7: *Slotted CSMA/CA Mechanism Flow Chart*

Figure 3.7 depicts a flowchart describing the slotted version of the CSMA/CA mechanism. The algorithm works as follows; all the nodes are synchronized and transmissions can begin only at the boundaries of the backoff slots. Then a node which has a packet ready for transmission follows the next procedure:

- Step 1. *Initialization of the algorithm variables*: $NB = 0$ and $CW = 2$. If *macBattLifeExt* is set true, BE is initialized to $min(2, macMinBE)$, otherwise it is initialized to *macMinBE=3*.

- Step 2. *Backoff procedure*: the node backsoff for a random number of backoff slots, chosen uniformly between 0 and $2^{BE} - 1$, before sensing the channel. This random backoff serves to reduce the probability of collisions among contending nodes.

- Step 3. *Clear Channel Assessment*: After the backoff counter expires, the node verifies if the medium is idle or not performing a CCA. If this CCA reported an idle channel, CW is decremented by 1 and if $CW = 0$, then the message is transmitted. If $CW \neq 0$, the algorithm jumps again to Step 3 with the new value of CW. If the CCA reported a busy channel, then BE and NB are incremented by 1 and the algorithm starts again in Step 2. This mechanism is repeated until either BE equals the parameter *aMaxBE* (which has a default value of 5), at which point it is frozen at *aMaxBE*, or until a certain maximum number of permitted random backoff stages is reached, at which point an access failure is declared to the upper layer. The maximum number of permitted random backoff stages is determined by the parameter *macMaxCSMABackoffs*, which has a default value of 5.

- Step 4. *Acknowledgment* (optional): If the acknowledged mode is activated, the successful transmission is followed by an acknowledgement packet (ACK) from the receiver of length 11 bytes. This acknowledgment frame is transmitted without using the slotted CSMA/CA mechanism to access the channel. However, if the transmitting node does not receive the ACK packet within *MacAckWaitDuration*, it declares a collision or a packet loss. Then the same data packet is retransmitted with the initial backoff parameters. The maximum number of retransmissions collision or packet loss is limited to *aMaxFrameRetries*. If the number of retransmissions reaches *aMaxFrameRetries*, the packet is discarded.

3.6.3 Inter-Frame Spacing (IFS)

The IFS is an idle communication period that is needed for supporting the MAC sub-layer to process data received by the physical layer. To allow this, all transmitted frames are followed by an IFS period as shown in Figure 3.8. If the transmission requires an ACK, the IFS will follow the ACK frame. The length of the IFS period depends on the size of the transmitted frame: a long inter-frame spacing (LIFS) or short inter-frame spacing (SIFS). The selection of the IFS is based on the IEEE 802.15.4 *aMaxSIFSFrameSize* parameter. For $frameSize \leq aMaxSIFSFrameSize$ SIFS is used, while for $frameSize > aMaxSIFSFrameSize$ LIFS is used.

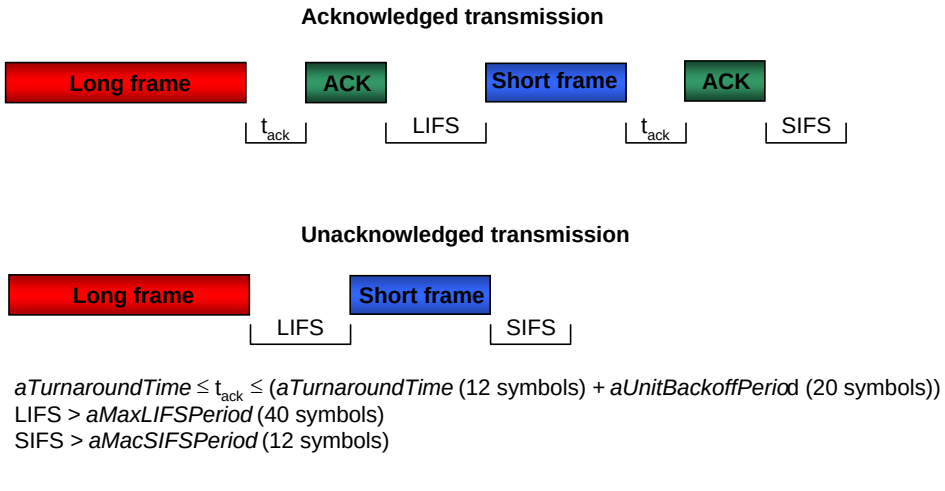

Figure 3.8: *Inter-frame Spacings*

Part III

Analytical model development and Simulation Tools

CHAPTER 4

Analytical Model for the Acknowledged CAP of IEEE 802.15.4

4.1 Introduction

Since the ratification of the IEEE 802.15.4 MAC, much interest has been put on evaluating its throughput and energy performance. Specially, the slotted CSMA/CA access protocol featured in the CAP of its beacon-enabled mode, has attracted the attention of many researchers. Many preliminary simulation studies were conducted [17] [18], and several accurate analytical models have been introduced. These analytical models follow two main approaches.

The first approach is inspired from the seminal work of Bianchi on the analysis of the IEEE 802.11 [19], and is based on modeling the behavior of a sensing node using a Markov chain. The most relevant contributions along this approach are [20], which adapts Bianchis's model to the specific contention access procedure of the IEEE 802.15.4 CAP under unacknowledged saturated traffic conditions, [21] which corrects the analysis in [20] and extends it to include acknowledgment transmissions, and [22] which proposes a more complex model reflecting additionally the superframe structure under unsaturated traffic conditions. A final relevant contribution, upon which our work is based, is the work of Ramachandran et al. [2] which develops a framework based on two Markov chains; the node-state chain to determine the fraction of time the nodes spends in different states, and the channel-state chain to calculate the throughput of the network. Our choice is motivated by the accuracy of this model with respect to ns-2 simulations, as well as its simplicity and amenability for extension.

The second approach alternatively models the contention access mechanism as an embedded Markov renewal process and performs a fixed-point analysis to calculate the throughput of the network [23].

This thesis undertakes the evaluation of the CAP of IEEE 802.15.4 in WBSN scenarios. Typically, WBSN are not dense enough to ensure that there is sufficient redundancy in sensor deployment so as to avoid the use of MAC level acknowledgments. Therefore, in WBSN applications there is a need for ACKs, as a means to ensure reliable data transfer between the node and the coordinator. In this Chapter we extend the IEEE 802.15.4 CAP Markov-chain-based model in [2] to support acknowledged traffic; we also provide a comprehensive analysis in terms of throughput, energy efficiency and latency.

4.2 Model Assumptions

Our acknowledged model will be based on the same assumptions as [2]. However, particularizing the model to our extensions needs will result in the modification of some of these assumptions as well as the formulation of new ones. The following now summarizes all of them.

WBSN applications work under a one-hop star topology. Within this star topology, we assume a network of a fixed number M of sensing devices, with a common coordinator where all nodes are within carrier sensing range of each other. This ensures that an ongoing transmission will not be interrupted by other nodes (no hidden terminal problem).

Since we are only concerned about the MAC performance in the CAP of the standard, it will be assumed that superframe does not have a CFP nor an inactive period. Although having an inactive period allows the nodes to sleep periodically and conserve energy, it introduces undesirable delays in delay-critical monitoring applications such as WBSNs, particularly at higher beacon orders. Therefore, in our analysis, we assume that the entire superframe duration is active; that is SFO = BCO.

In WBSN applications where information is gathered from human body monitoring or the environment, and then forwarded to the coordinator, most of the communication is uplink (nodes-to-coordinator), as opposed to downlink (coordinator-to-nodes). Consequently, we will concentrate our analysis on the uplink mode only. This will allow nodes to enter the sleep state depending on their own availability of data to transmit rather than having to stay awake for the entire active period.

In this first acknowledged CAP analysis, we will also assume that there are no packet losses, the only cause of transmission failures being possible collisions. MAC level ACKs will always be sent after each successful data transmission and will always arrive to the destination node. We will not consider however data packet retransmissions after a collision. In this scheme the coordinator notifies

the sensing node after a packet reception with an ACK. If the node does not receive an ACK, it assumes a collisions occurred, however retransmissions are not contemplated.

Data packets are assumed to be of fixed N-backoff slots duration and arrive at the nodes for transmission according to a Poisson arrival rate of λ packets per packet duration. The probability p that a node will get a packet to transmit at the next slot is $p = \lambda/N$. Moreover, this unsaturated traffic assumption implies that no buffering is considered at the nodes. New packets are not accepted for transmission ($p = 0$) when the node is currently transmitting, or, is attempting a transmission.
Notation: all probabilities associated with channel states have a superscript 'c' (e.g., p_i^c) and those associated with node states have a superscript 'n' (e.g., p_i^n).

4.3 Analytical Formulation

4.3.1 Approximations

Prior to the formulation of the analytical model, we introduce certain approximations:

- Approximation 1: The computation of the probability that the channel is sensed idle in a given slot is difficult. We will the approximate it with the *steady state probability* that the channel is idle, p_i^c. Thus, every node sees a probability p_i^c that the channel is idle in the first of the two slots after every random backoff. We do not assume that channel idleness is independent from one sensing slot to the next. However, it is reasonable to assume channel state independence for two slots separated by a backoff duration, particularly when packet lengths are small. This approximation allows us to model a single node independently of all others.

- Approximation 2: Channel throughput can be computed if the probability that an individual node begins transmission in any generic slot is known. However, computing this probability is no easy task. We therefore approximate it with the *steady state probability* that a node transmits, p_t^n . The channel thus sees a probability p_t^n that an individual node begins transmission in any generic slot, except when it is already transmitting.

- Approximation 3: The IEEE 802.15.4 standard specifies that the number of slots a node has to wait at each random backoff stage should be drawn from an uniform distribution. For the sake of analytical tractability, we replace the uniform distribution with a geometric distribution of the same mean, so that the backoff algorithm is memoryless.

4.3.2 Node state model

The behavior of an individual node is modeled by means of a Markov chain as shown on Figure 4.1. Initially the node is in the $IDLE$ state until it receives a packet to be transmitted in a backoff slot, which occurs with the already define probability p . Then, it moves on to the state BO_1 to perform the first backoff stage of the protocol. The node spends a random number of backoff slots chosen uniformly between 0 and $(2^{BE} - 1)$ (it is supposed that the *BatteryExtension* is turned off which results on a initial value of $BE = macMinBE = 3$). In our model, the uniform random variable which describes the number of backoff slots the node spends in BO_1 is replaced by X_1, an equivalent geometrical distribution defined by $P[X_1 = k] = (1 - p_1^n)^k p_1^n$ for $k = 0, 1, ...\infty$. The value of p_1^n is set to $p_1^n = 1/4.5$ so that this geometric distribution has the same mean as the uniform distribution defined in the standard (resulting in $E[X_1] = 3.5$).

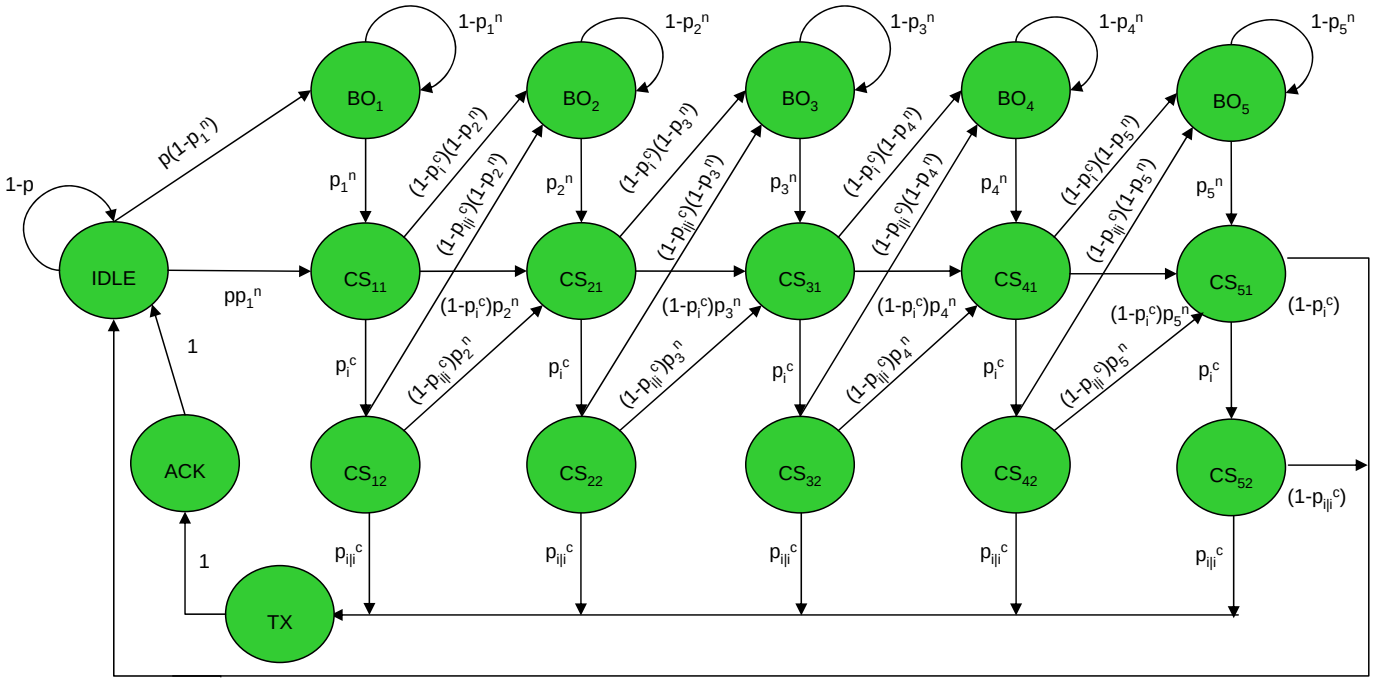

Figure 4.1: *Embedded Markov chain model for a sensing node*

The node reaches the CS_{11} after leaving BO_1, and to confirm that the channel is idle it performs the first CCA. If the channel is idle in next backoff slot, which occurs with probability p_i^c, the node then transits to the state CS_{12} which denotes the second CCA attempt. If the channel is again found idle for this second CCA, event characterized by probability $p_{i/i}^c$, the node would enter the TX state and start transmitting the packet[1].

[1]Note that the probability of finding the channel idle in this second backoff slot $p_{i/i}$ is not independent from p_i^c.

If however, the channel is found busy in the first CCA attempt at state CS_{11} , which occurs with probability $(1 - p_i^c)$, the node goes into the second backoff stage BO_2, increasing the value of the BE by 1, and repeats the same backoff and clear channel assessment procedure of the first stage. However the number of backoff slots spent in this BO_2 state is now represented by X_2, a random variable geometrically distributed according to $P[X_2] = (1 - p_2^n)^k p_2^n$, where $p_2^n = 1/8.5$, to remain consistent with the BE increment.

In general, we adopt the notation CS_{ij} , $1 \leq i \leq 5$, $1 \leq j \leq 2$, to denote the j^{th} carrier sensing backoff slot after the i^{th} random backoff stage, BO_i. Also to complete the notation of these stages it can proved that $p_3^n = p_4^n = p_5^n = 1/16.5$ since $BE = aMaxBE = 5$ for $BO_{3 \leq i \leq 5}$.

When the node is in the TX state, it spends N backoff slots in that state (since the length of a packet, in terms of number of backoff slots, is equal to N) and then transitions to the ACK state where it waits to receive the acknowledgment from the coordinator. The time spent on the ACK state, denoted by T_{ACK}^n, will be computed on the next section. From this ACK, the nodes finally goes back to the initial $IDLE$ state with probability 1.

The complete Markov chain for the sensing node is constructed starting in the IDLE state, and moving all the way around through the chain until either the ACK packet has been correctly received, the *MacAckWaitDuration* is consumed without receiving the ACK, or until the maximum number of backoff stages $macMaxCSMABackoffs = 5$ is reached and access channel failure is declared.

The *steady state occupancy* of this chain can be obtained by solving the Markov equation system in Equation (4.1). The notation $\pi(state_i)$ denotes the long term proportion of transitions into $state_i$:

$$
\begin{aligned}
\pi(idle) &= (1-p)\pi(idle) + \pi(ack) + (1-p_i^c)\pi(cs_{51}) + (1-p_{i/i}^c)\pi(cs_{52}) \\
\pi(bo_1) &= (1-p_1^n)[p\pi(idle) + \pi(bo_1)] \\
\pi(cs_{11}) &= p_1^n[p\pi(idle) + \pi(bo_1)] \\
\pi(cs_{12}) &= p_i^c\pi(cs_{11}) \\
\pi(bo_2) &= (1-p_2^n)[(1-p_i^c)\pi(cs_{11}) + (1-p_{i/i}^c)\pi(cs_{12}) + \pi(bo_2)] \\
\pi(cs_{21}) &= p_2^n[(1-p_i^c)\pi(cs_{11}) + (1-p_{i/i}^c)\pi(cs_{12}) + \pi(bo_2)] \\
\pi(cs_{22}) &= p_i^c\pi(cs_{21}) \\
\pi(bo_3) &= (1-p_3^n)[(1-p_i^c)\pi(cs_{21}) + (1-p_{i/i}^c)\pi(cs_{22}) + \pi(bo_3)] \\
\pi(cs_{31}) &= p_3^n[(1-p_i^c)\pi(cs_{21}) + (1-p_{i/i}^c)\pi(cs_{22}) + \pi(bo_3)] \\
\pi(cs_{32}) &= p_i^c\pi(cs_{31}) \\
\pi(bo_4) &= (1-p_4^n)[(1-p_i^c)\pi(cs_{31}) + (1-p_{i/i}^c)\pi(cs_{32}) + \pi(bo_4)] \\
\pi(cs_{41}) &= p_4^n[(1-p_i^c)\pi(cs_{31}) + (1-p_{i/i}^c)\pi(cs_{32}) + \pi(bo_4)] \\
\pi(cs_{42}) &= p_i^c\pi(cs_{41}) \\
\pi(bo_5) &= (1-p_5^n)[(1-p_i^c)\pi(cs_{41}) + (1-p_{i/i}^c)\pi(cs_{42}) + \pi(bo_5)] \\
\pi(cs_{51}) &= p_5^n[(1-p_i^c)\pi(cs_{41}) + (1-p_{i/i}^c)\pi(cs_{42}) + \pi(bo_5)] \\
\pi(cs_{52}) &= p_i^c\pi(cs_{51}) \\
\pi(ack) &= \pi(tx)
\end{aligned}
\tag{4.1}
$$

And:

$$
\pi(idle) + \pi(tx) + \pi(ack) + \sum_{i=1}^{5}[\pi(bo_i) + \pi(cs_{i1}) + \pi(cs_{i2})] = 1 \tag{4.2}
$$

The probability $p_{i/i}^c$ that the channel is idle at the next backoff slot given that it is idle at the current backoff slot, can be computed by:

$$
p_i^c = p_{i/i}^c\, p_i^c + p_{i/b}^c(1-p_i^c) \tag{4.3}
$$

where $p_{i/b}^c$ is the probability that the channel is idle at the next backoff slot given that is busy at the current backoff slot. The channel might be busy due to two reason:

1. A data packet is being transmitted.

2. An ACK is being transmitted.

We will then note $p^c_{b_{data}}$ as the probability that the channel is busy due to a data packet, and $p^c_{b_{ack}}$ as the probability that the channel is busy due to an ACK. Then we can write:

$$\begin{aligned} p^c_{i/b} &= \frac{p^c_{i,b}}{p^c_b} = \frac{p^c_{i,b_{data}} + p^c_{i,b_{ack}}}{p^c_b} \\ &= \frac{p^c_{i/b_{data}} p^c_{b_{data}} + p^c_{i/b_{ack}} p^c_{b_{ack}}}{p^c_b} \\ &= \frac{\frac{1}{N} p^c_{b_{data}} + \frac{1}{2} p^c_{b_{ack}}}{(1 - p^c_i)} \end{aligned} \tag{4.4}$$

where $p^c_{i/b_{data}}$ is equal to 1/N, N being the length of the packet in terms of number of backoff slots, and $p^c_{i/b_{ack}}$ is equal to 1/2, since the ACK packet size is $L_{ack} = 11bytes$, a minimum of two backoff slots. To calculate $p^c_{b_{ack}}$ and $p^c_{b_{data}}$ the channel state model should be first introduced.

4.3.3 Channel state model

The behavior of the channel state will be also modeled using a Markov chain as shown in Figure 4.2. Initially the channel is in the $(IDLE, IDLE)$ state; it will remain in that state if none of the nodes begins transmission, which will occur with a probability α. On the other hand, when exactly one node begins transmission and all the others refrain, the channel progresses to the $SUCCESS$ state, which represents a successful transmission. This happens with probability β. Finally, if more than one node begins transmission simultaneously, a collision takes place, being modeled as if the channel goes from the $(IDLE, IDLE)$ state to the $FAILURE$ state, which occurs with probability $\delta = 1 - \alpha - \beta$.

When the channel is in the $SUCCESS$ state, it spends N backoff slots in that state, since the length of all packets is assumed to be N backoff slots. In the same way and since there is no collision detection mechanism, the channel remains in the $FAILURE$ state for the entire packet transmission time, that is N backoff slots.

At the end of the transmission, successful or not, the channel returns to the $(IDLE, IDLE)$ state through an intermediate $(BUSY, IDLE)$ state. It is on this $(BUSY, IDLE)$ where the node will wait for the coordinator ACK packet, which will arrive only if the previous state was $SUCCESS$. Of course, if the previous state was $FAILURE$, the coordinator never received the data packet and the sensing node will consume completely the *MacAckWaitDuration* waiting for an ACK that will never arrive. In the next section we will explain in detail how the time spent in the $(BUSY, IDLE)$ state, which we will denote as $T^c_{B,I}$, is computed. Besides that, the process to determine the probabilistic values of α and β is continued in section 4.3.5.

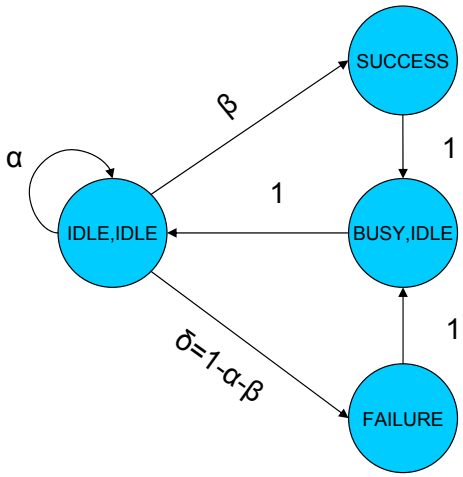

Figure 4.2: *Embedded Markov chain model for the Channel State*

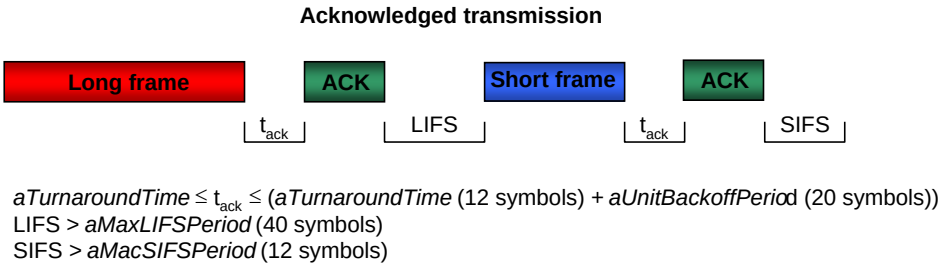

Figure 4.3: *Inter-frame Spacings for Acknowledged Transmission*

The computation of the time the channel is idle waiting for the ACK, could be computed at the $SUCCESS$ and $FAILURE$ states independently rather than together in the $(BUSY, IDLE)$ state. However, aiming to keep the channel notation of [2], we have decided to choose the option described in the above paragraph.

4.3.4 Time spent in the *ACK* node state and *(BUSY,IDLE)* channel state

In Section 3.6.3 we already discussed about the inter-frame spacings before and after the ACK reception. Figure 4.3 presents the graphical representation of this timings in the acknowledged reception mode.

According to the IEEE 802.15.4 standard, the transmission of an acknowledgment frame shall

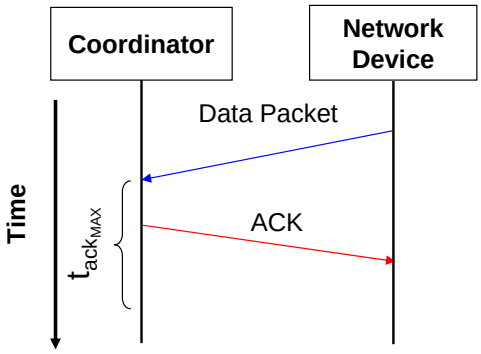

Figure 4.4: *ACK reception scheme*

commence between *aTurnaroundTime* and *(aTurnaroundTime + aUnitBackoffPeriod)* symbols after the reception of the last symbol of the data or MAC command frame. The constant *aTurnaroundTime* is equal to 12 symbols.

We will then denote the maximum and minimum idle periods that will precede the ACK transmission by the coordinator after a correct data reception as $t_{ack_{MAX}}$ and $t_{ack_{MIN}}$ respectively, while the size of the acknowledgment will be denoted as $L_{ACK} = 11bytes$. A simple scheme of this procedure is shown in Figure 4.4. The maximum and minimum values of t_{ack} are the following:

$$\begin{aligned} t_{ack_{MAX}} &= aTurnaroundTime + aUnitBackoffPeriod = 12 + 20 = 32symbols \quad (4.5) \\ &= 1.6\ backoffSlots \\ t_{ack_{MIN}} &= aTurnaroundTime = 12symbols = 0.6\ backoffSlots \quad (4.6) \end{aligned}$$

That is:

$$0.6 \leq t_{ack} \leq 1.6 \quad (4.7)$$

Calculating the time spent in the ACK state of the node Markov chain, T^n_{ACK}

To obtain the exact expression of the time spent in the ACK state of the node Markov chain model, T^n_{ACK}, we will analyze two scenarios as depicted in Figure 4.5: successful transmission and failure transmission. Additionally some assumptions will be drawn.

For the first case, it will be assumed that if a data packet transmission was successful, the processing time of the frame at the coordinator can be considered negligible. This means that in this situation the coordinator will respond with the ACK after $t_{ack_{MIN}}$. Following a successful transmission, the node

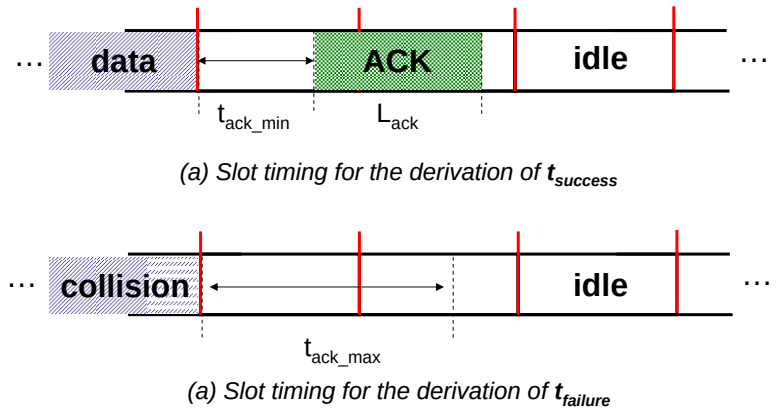

Figure 4.5: *Slot timing for the derivation of* T^n_{ACK} *and* $T^c_{B,I}$

will wait in the ACK state a total time of:

$$\lceil T^n_{ACK_{success}} \rceil = \lceil (t_{ack_{MIN}} + L_{ACK}) \rceil = \lceil (0.6 + 1.1) \rceil = \lceil 1.7 backoff\ slots \rceil = 2 backoff\ slots \quad (4.8)$$

However, if there has been a collision while transmitting the data packet (see Figure 4.5), the coordinator will never send the ACK, so the node will wait in the ACK state a total time of:

$$\lceil T^n_{ACK_{failure}} \rceil = \lceil t_{ack_{MAX}} \rceil = \lceil 1.6 backoff slots \rceil = 2 backoff\ slots \quad (4.9)$$

This two events, data transmission success and data transmission failure, lead to the same result:

$$T^n_{ACK_{success}} = T^n_{ACK_{failure}} = 2 \quad (4.10)$$

Then, we conclude that in all cases we achieve a $T^n_{ACK} = 2$, which means that the node will always remain in the ACK state for a total of 2 backoff slots.

Calculating the time spent in the $(BUSY, IDLE)$ state of the channel Markov chain, $T^c_{B,I}$

We are now in place to evaluate the time spent in the $(BUSY, IDLE)$ state of the channel Markov chain, $T^c_{B,I}$. Proceeding as before, we will analyze the time spent in this state when we have a successful transmission and a failure transmission. After a successful transmission the channel will remain in the $(BUSY, IDLE)$ state for:

$$\lceil T^n_{B,I_{success}} \rceil = \lceil (t_{ack_{MIN}} + L_{ACK} + idle) \rceil = \lceil (0.6 + 1.1 + 1) \rceil = \lceil 2.7 backoff\ slots \rceil = 3 backoff\ slots \quad (4.11)$$

Please note the extra idle backoff slot in the computation, corresponding to the idle backoff slot which follows a busy backoff slot, and that is computed as part of the definition of this state.

Following a collision, or what is the same a failure transmission, the channel will remain in the $(BUSY, IDLE)$ state for just one backoff slot $T^n_{B,I_{failure}} = 1$. This computation is a bit more tricky, but is well understood by looking at Figure 4.5. After a collision, the channel remains idle for $t_{ack_{MAX}} = 1.6$ backoff slots. Then at the next backoff boundary, the next backoff slot that follows will be clear for node transmissions, since the channel has been idle for two consecutive backoff slots (a node who performed the CCA procedure in this two backoff slots could start transmitting). Then, this situation will be broken down in the following way; the computation of the first idle backoff slot will correspond to the $(BUSY, IDLE)$ state and the computation of the second idle backoff slot will correspond to the $(IDLE, IDLE)$ state.

Now that we have obtained the number of backoff slots that the channel remains in the $(BUSY, IDLE)$ state for a successful and failure transmission ($T^n_{B,I_{success}}$ and $T^n_{B,I_{failure}}$ respectively), we can compute the overall time spent in this state. It can be checked that a successful an a failure transmission occur with probability of β and δ respectively. Then finally:

$$T^n_{B,I} = T^n_{B,I_{success}} \frac{\beta}{\beta + \delta} + T^n_{B,I_{failure}} \frac{\delta}{\beta + \delta} \tag{4.12}$$

All the obtained results in the section are coherent with the IFS defined in the standard. On the other hand, all the analytical assumptions made in this section will be validated with simulation results.

4.3.5 Evaluating the remaining node state model probabilities

Now that we have introduced the channel state model, we are ready to calculate p^c_{ack}. An ACK will be transmitted only if there has been a successful transmission previously. This again occurs with a probability defined as β, coming from an $(IDLE, IDLE)$ channel state. Then, using joint probabilities we can write:

$$p^c_{b_{ack}} = \beta \, p^c_i \, p^c_{i/i} \tag{4.13}$$

And using equation Equation (4.3) we obtain:

$$p^c_{i/i} = \frac{p^c_i - p^c_{i/b}(1 - p^c_i)}{p^c_i} \tag{4.14}$$

With all the defined probabilities, the probability that any node would begin transmission in a generic backoff slot, $p^n_t = p^c_{b_{data}}$ can be computed. By Approximation 2, the probability that a node transmits

in a generic backoff slot is equal to the *steady-state* probability that the node is in one of the states where it is sensing the channel for a second consecutive backoff slot (i.e.,$\bigcup_{i=1}^{5} cs_{i2}$) multiplied by $p^c_{i/i}$. Note that $\pi(cs_{i2})$ denotes the steady-state proportion of transitions into state cs_{i2}. To obtain the long-term proportion of time that the chain is in $\bigcup_{i=1}^{5} cs_{i2}$, we need to account for the time spent in each state. By doing this we obtain:

$$p^n_t = \left(\frac{\sum_{i=1}^{5} \pi(cs_{i2})}{\pi(idle) + N\pi(tx) + T^n_{ACK}\pi(ack) + \sum_{i=1}^{5}\sum_{j=1}^{5} \pi(cs_{ij}) + \sum_{i=1}^{5} \pi(bo_i)} \right) p^c_{i/i} \tag{4.15}$$

4.3.6 Evaluating the remaining node channel model probabilities

Knowing the probability p^n_t that an individual station transmits in a generic backoff slot, we can also calculate the transition probabilities α and β of the Markov chain model for the channel state. We defined α as the probability of none of the nodes beginning transmission, that is $\alpha = (1 - p^n_{t/ii})^M$ where M is the number of sensing nodes, excluding the coordinator. The probability that any node begins transmission, given that the channel has been idle for two consecutive backoff slots, is denoted by $p^n_{t/ii}$ and computed as follows:

$$p^n_{t/ii} = \frac{p^n_t}{p^c_{ii}} = \frac{p^n_t}{p^c_{i/i} p^c_i} \tag{4.16}$$

On the other hand, β can be computed as $\beta = M p^n_{t/ii}(1 - p^n_{t/ii})^{M-1}$. The Markov chain for the channel, as shown in Figure 4.2, can be solved to determine the probability that the channel remains idle for two consecutive backoff slots, p^c_{ii}. We first define π^c_{ii}, π^c_{bi}, π^c_f and π^c_s as the long term proportions of transitions into states $(IDLE, IDLE)$, $(BUSY, IDLE)$, $FAILURE$ and $SUCCESS$ respectively. Then, the state balance equations corresponding to the Markov chain for the channel are:

$$\begin{aligned} \pi^c_{ii} &= \alpha\pi^c_{ii} + \pi^c_{bi} \\ \pi^c_s &= \beta\pi^c_{ii} \\ \pi^c_f &= (1 - \alpha - \beta)\pi^c_{ii} \\ \pi^c_{bi} &= 1 - \pi^c_{ii} - \pi^c_s - \pi^c_f \end{aligned} \tag{4.17}$$

which can be solved to obtain:

$$\begin{aligned} \pi_{ii}^c &= \frac{1}{3-2\alpha} \\ \pi_s^c &= \frac{\beta}{3-2\alpha} \\ \pi_f^c &= \frac{\delta}{3-2\alpha} \\ \pi_{bi}^c &= \frac{1-\alpha}{3-2\alpha} \end{aligned} \tag{4.18}$$

And finally we calculate the probability that the channel is idle for two consecutive backoff slots, p_{ii}^c:

$$p_{ii}^c = \frac{\pi_{ii}^c}{\pi_{ii}^c + T_{B,I}^c \pi_{bi}^c + N\pi_s^c + N\pi_f^c} = \frac{1}{1 + T_{B,I}^c(1-\alpha) + N(\beta+\delta)} \tag{4.19}$$

With this, the probability that the channel is idle at any generic backoff slot, p_i^c , can be obtained as follows:

$$p_i^c = \frac{p_{ii}^c}{p_{i/i}^c} \tag{4.20}$$

The inclusion of β, p_i^c and $p_{i/i}^c$ through Equations (4.13) and (4.15) in the calculation of $p_{i/b}^c$ in Equation (4.4), couples the modeling of the channel states and the node states . In (4.14) we have a non linear equation for $p_{i/i}^c$ written as a function of p_i^c and as function of β through $p_{i/b}^c$. On the other hand, β is computed as a function of $p_{i/i}^c$ and p_i^c through $p_{t/ii}^n$. Finally p_i^c in Equation (4.20) is a function of $p_{i/i}^c$ and a function of β through p_{ii}^c. Then this system of nonlinear equations is consistent and can be solved following numerical approximation techniques.

4.4 Performance Metrics

4.4.1 Aggregate throughput

The aggregate channel throughput S is defined as the fraction of time spent in successful transmissions, given by the *steady state probability* of being in the *SUCCESS* state in Figure 4.2. It can be derived as:

$$S = \frac{N\pi_s^c}{\pi_{ii}^c + T_{B,I}^c \pi_{bi}^c + N\pi_s^c + N\pi_f^c} = \frac{N\beta}{1 + T_{B,I}^c(1-\alpha) + N(\beta+\delta)} \tag{4.21}$$

4.4.2 Average power consumption per node

To determine the average power consumption of the sensing nodes, we will first need to identify the different states of a radio. We will consider the three general radio states as shown in Figure 4.6:

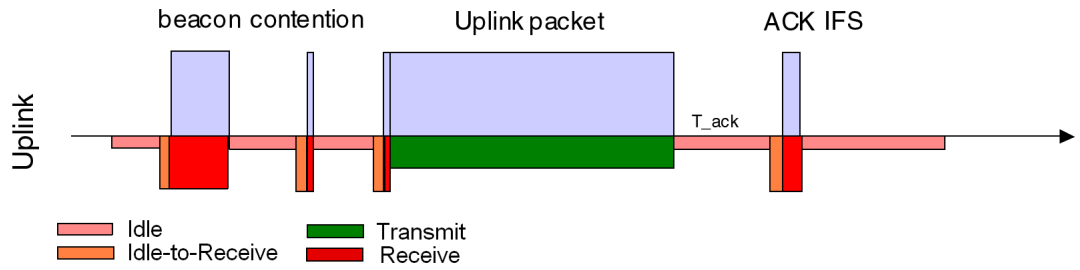

Figure 4.6: *Uplink Radio cycling*

1. *Idle*: The radio is ready to receive a command to switch to Transmit or Receive state.
2. *Transmit*: The radio is transmitting.
3. *Receive*: The radio is receiving.

Initially he radio will stay in the Idle state until it needs to receive a beacon or perform a CCA; at that time, it transitions to the Receive state. If a beacon reception was received, the radio returns back to the Idle state after receiving the beacon. If CCA was requested, after two CCA backoff slots, it either transitions to the Transmit state if the channel is found to be idle or back to the Idle state if the channel is busy. In other words, the radio is in:

- *Idle state:* When it either has no packet to transmit or when it has one and is backing off (corresponding to the $IDLE$ and BO_i states of Figure 4.1).
- *Receive state:* when it is doing carrier sensing (corresponding to the CS_{ij} states of Figure 4.1) or receiving a beacon.
- *Transmit state:* when it is transmitting.

Let the beacon duration be n_{beacon} backoff slots. The frequency of beacon reception is $f_{beacon} = 1/BI$, where BI. The fraction of time spent in receiving beacons is thus $p^n_{beacon} = n_{beacon}/BI$. We assume that part of the time spent by a node in $IDLE$ state is used to receive the beacons. In WBSN applications, the only time a sensing node is receiving data is during the beacon and ACK reception. While it may be possible to explicitly define a node state for this beacon receive duration, we make a simplifying assumption that the beacon reception occurs during the node's IDLE state and adjust the power consumption budget accordingly. This adjustment is necessary since the radio's Receive state power expenditure is normally several orders of magnitude higher than its Idle state power dissipation.

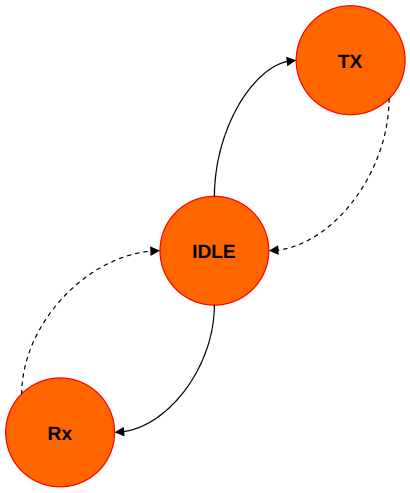

Figure 4.7: *Transitions on the Energy States*

A similar observation holds for the radio's Idle-to-Receive transition. We assume that the time required for this transition is budgeted off the node's IDLE state, but that the power consumed during this transition is on the order of the radio's Receive state power. With the description above stated, the average power expenditure of any node, Y_{av}, can be expressed as follows:

$$Y_{av} = (p^n_{idle} - p^n_{beacon} + p^n_{bo} - p^n_{ir})Y_{idle} + (p^n_{cs} + p^n_{ir} + p^n_{beacon} + p^n_{ack})Y_{rx} + p^n_{tx}Y_{tx} \tag{4.22}$$

where Y_{idle}, Y_{rx} and Y_{tx} are the power expenditures corresponding to the radio's Idle, Receive and Transmit states respectively. The parameter p^n_{ir} denotes the fraction of time spent in switching the radio from Idle to Receive state. This transition happens whenever the backoff counter reads 1 and once every beacon interval for beacon interval. In each of these occasions, the radio spends 192 μs, or equivalently, 0.6 backoff slots. Finally, the parameters p^n_{idle} , p^n_{bo}, p^n_{cs}, p^n_{tx} and p^n_{ack} denote the fractions of time spent by a node in IDLE, backoff (any BOi_i), carrier sense (any CS_{ij}), transmit (TX) and acknowledgment (ACK) states respectively of Figure 4.1 and are given by:

$$
\begin{aligned}
p^n_{idle} &= \frac{\pi(idle)}{1+\pi(tx)(N-1)+\pi(ack)(T^n_{ACK}-1)} \\
p^n_{bo} &= \frac{\sum_{i=1}^{5}\pi(bo_i)}{1+\pi(tx)(N-1)+\pi(ack)(T^n_{ACK}-1)} \\
p^n_{cs} &= \frac{\sum_{i=1}^{5}\sum_{j=1}^{2}\pi(cs_{ij})}{1+\pi(tx)(N-1)+\pi(ack)(T^n_{ACK}-1)} \\
p^n_{tx} &= \frac{N\pi(tx)}{1+\pi(tx)(N-1)+\pi(ack)(T^n_{ACK}-1)} \\
p^n_{ack} &= \frac{2\pi(ack)}{1+\pi(tx)(N-1)+\pi(ack)(T^n_{ACK}-1)}
\end{aligned}
\tag{4.23}
$$

Note that the denominator of all the previous equations should strictly be:

$$
\pi(idle)+\sum_{i=1}^{5}\pi(bo_i)+\sum_{i=1}^{5}\sum_{j=1}^{2}\pi(cs_{ij})+N\pi(tx)+T^n_{ACK}\pi(ack) \tag{4.24}
$$

since any node spends N backoff slots when in the transmit state, T^n_{ACK} backoff slots when in the ACK state and 1 backoff slot in all other states. However, $\pi(idle)+\sum_{i=1}^{5}\pi(bo_i)+\sum_{i=1}^{5}\sum_{j=1}^{2}\pi(cs_{ij})+\pi(tx)+\pi(ack)=1$, and therefore the equation denominator can be simplified to $1+\pi(tx)(N-1)+\pi(ack)(T^n_{ACK}-1)$ as shown.

4.4.3 Performance metric: per node bytes-per-Joule capacity

A metric that combines per-node throughput and energy consumption, is the bytes-per-Joule capacity. We use a normalized version which is defined as follows:

$$
\eta = \frac{(S/M)(250\times 10^3/8)}{Y_{av}} \tag{4.25}
$$

where S is the overall throughput and M is the number of sensing nodes. The throughput seen by each user is therefore S/M. The factor $(250\times 10^3/8)$ is due to the fact that the channel capacity is 250 Kbps in the 2.4 GHz ISM band, or equivalently, $(250\times 10^3/8)$ bytes per second.

4.4.4 MAC Latency

The MAC latency is defined as the average duration from the instant a packet becomes available for transmission to the end of its successful transmission. It is equal to the average duration between two successful transmissions minus the average time spent in the IDLE state. That is:

$$L = \frac{NM(1 - p_{idle}^{n})}{S} \tag{4.26}$$

CHAPTER 5

Analysis of a High Path Loss WBSN

5.1 Introduction

The human body is a very lossy medium. As one might think, the WBSN radio signals experience a higher path loss compared to a free space propagation. Even more, transmissions over an arbitrary distance near the human body are not always possible. In this Chapter we will analyze how is the behavior of a WBSN pathloss. Then, we will introduce in the analytical model defined on the previous chapter, the packet loss rate, when the reliability of the communications is not guarantee (LQI too low at reception).

5.2 Path Loss Model for the Human Body

The channel model for propagation along the human body was studied by E. Reusens et al. in [24] and by A. Fort et al. in [25]. In this two papers they proposed the use of a lognormal model distribution to determine the node's communication range. They use the following semi-empirical formula for the path loss computation:

$$P_{dB} = P_{0,dB} + 10nlog(d/d_0) \tag{5.1}$$

where P_{dB} represents the pathloss in decibels at a distance d, $P_{0,dB}$ represents the path loss at a reference distance d_0 and n is the path loss exponent. This formula is the standard pathloss law

parameter	value LOS	value NLOS
d_0	10 cm	10 cm
$P_{0,dB}$	35.7 dB	48.8 dB
σ	6.2 dB	5.0 dB
n	3.38	5.9

Table 5.1: Parameter values for the Shadowing model

used in many indoor wireless and some body-area propagation studies. The special case where $n = 2$ corresponds to free-space propagation. For other multipath scenarios such as WBSN, n is varied empirically to match the measured data.

However, the model showed in Equation (5.1) only represents the mean path loss. In practice, there will be variations with respect to the nominal value. Then [24] and [25] make the common assumption that the distribution around this mean path loss is a zero-mean Gaussian random variable with variance σ, which is called *shadowing*. It is crucial to account the value of σ, often called *shadowing deviation*, in order to provide a certain reliability of communication. If we do so, the total path loss becomes a random variable given by:

$$PL = P_{dB} + P_s = P_{dB} + t\sigma \tag{5.2}$$

where $P_s = t\sigma$ is the shadowing component. This shadowing extra margin has to be added, according to the reliability required from the system. The value of t can be calculated according to the formula [24]:

$$t = \sqrt{2} erfc^{-1}[2(1-p)] \tag{5.3}$$

where $erfc^{-1}()$ is the inverse of the standard cumulative error function, and p is the percentage of reliability required in the system. Table 5.2 shows the parameter values of the fitted path loss models calculated for two different channels, according to Equation (5.2), and the variation σ of the individual measurements around the model. The first channel, presented in [24], is located along the front of the torso with LOS. The second channel is measured around the torso in [25], resulting in NLOS propagation.

5.3 Inclusion of the Error Probability on the Analytical Model

Operating under the circumstances of a high pathloss body channel, the Section 4.2 assumption about no package losses is not valid anymore. The fading of the channel will reduce dramatically the LQI of certain received packets, so that they are dropped by the receiving MAC. Then, we should find a way to introduce in our analytical model the packet loss rate, which we will denote by P_e. We will try to do so, minimizing the impact of the updates in the already fixed node and channel Markov chains as far as possible.

Since our analysis is concentrated only on the uplink, possible ACK losses will not be taken into account. It will still be assumed that after a correct transmission, the sensing nodes always receive the ACK from the coordinator. We may justified this point by the fact that since the coordinator is main powered, it can increase the transmitted power to avoid channel shadowing.

5.4 Changes in the Analytical Model

5.4.1 Markov Chains

The channel state will be modeled using the same Markov chain as before, but including the probability P_e as shown in Figure 5.1. It works the same way as before. The only difference is that now the channel progresses form the $(IDLE, IDLE)$ into the $SUCCESS$ state only if one node begins transmission and all the others refrain (probability β), and the packet is not lost in the channel (probability $(1 - P_e)$). If the transmitted packet is lost (probability βP_e) or a packet collision takes place (probability $\delta = 1 - \alpha - \beta$), the channel goes to the $FAILURE$ state.

The state balance equations for the new channel Markov chain are:

$$\begin{aligned} \pi_{ii}^c &= \alpha\pi_{ii}^c + \pi_{bi}^c \\ \pi_s^c &= \beta(1 - P_e)\pi_{ii}^c \\ \pi_f^c &= (\beta P_e + \delta)\pi_{ii}^c \\ \pi_{bi}^c &= 1 - \pi_{ii}^c - \pi_s^c - \pi_f^c \end{aligned} \tag{5.4}$$

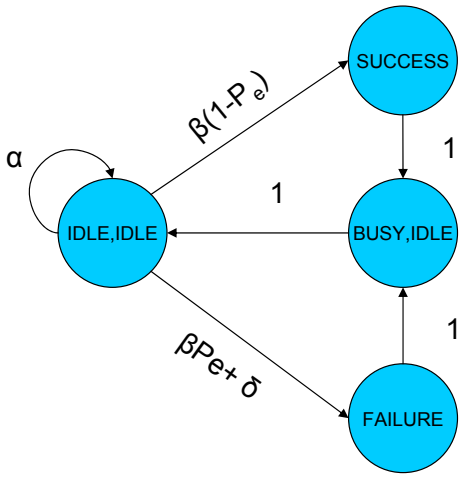

Figure 5.1: *Channel State Model including* P_e

which can be solved to obtain:

$$\begin{aligned} \pi_{ii}^c &= \frac{1}{3-2\alpha} \\ \pi_s^c &= \frac{\beta(1-P_e)}{3-2\alpha} \\ \pi_f^c &= \frac{\beta P_e + \delta}{3-2\alpha} \\ \pi_{bi}^c &= \frac{1-\alpha}{3-2\alpha} \end{aligned} \tag{5.5}$$

And the aggregate throughput S is calculated according to the first part of equation (4.21), with the obtained long term proportions.

The node state Markov chain figure remains the same.

5.4.2 Recalculation of the time spent in the (BUSY,IDLE) state

The number of backoff slots the channel remains in the $FAILURE$ state is the same as in the previous analysis. If there is a collision or the packet is lost the channel remains in the $FAILURE$ state for the entire packet transmission time, that is N backoff slots. The time spent in the $(BUSY, IDLE)$ state, $T_{B,I}^c$, needs to be recalculated however. Following the same process as in section 4.3.4 with the new state transitions probabilities, it can be show that the expression of Equation (5.6) is obtained.

$$T_{B,I}^n = T_{B,I_{success}}^n \frac{\beta(1-P_e)}{\beta+\delta} + T_{B,I_{failure}}^n \frac{\delta}{\beta+\delta} \tag{5.6}$$

With all the updated values of the model parameters we will calculate the performance metrics of the network exactly like we did in the previous chapter.

CHAPTER 6

Simulation environment: *ns-2*

6.1 Introduction to *ns-2*

ns-2 [26] is the second version of an open source discrete event simulator targeted at networking research. The *ns* project began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. In 1995 *ns* development was supported by the DARPA[1] through the VINT[2] project, Xerox PARC[3], LBNL[4], and UC Berkeley. Currently *ns-2* is developed in collaboration between a number of different researchers and institutions, including SAMAN[5], CONSER[6], and ICSI [7]. However, *ns* has always included substantial contributions from other researchers and it is currently maintained by volunteers.

ns-2 is developed in C++ and provides a simulation interface through OTcl, an object-oriented dialect of Tcl [27]. A network animator, Nam [28], provides packet-level animation for design and debugging of network protocols.

[1] Defense Advanced Research Projects Agency, central research and development office for the U.S. Department of Defense (DoD)

[2] Virtual InterNetwork Testbed, developed at USC/ISI

[3] Palo Alto Research Center, Inc.

[4] Lawrence Berkeley National Laboratory

[5] Simulation Augmented by Measurement And Analysis for Networks, supported by DARPA

[6] Collaborative Simulation for Education and Research, supported by the National Science Foundation (NSF) in USC/ISI

[7] International Computer Science Institute, affiliated to UC Berkeley

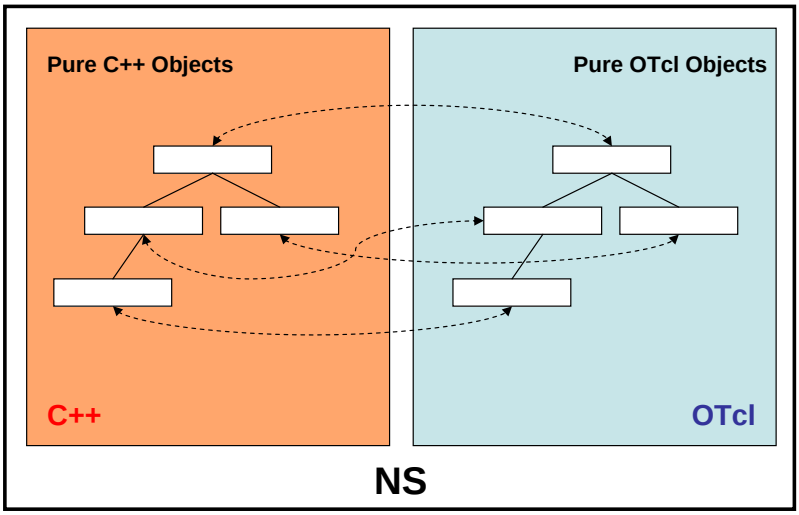

Figure 6.1: *ns split-language programming*

ns-2 provides substantial support for wired and wireless networks and is very popular among the scientific community. Different levels of configuration are present in *ns-2* due to its open source nature, including the capability of creating custom applications and protocols as well as modifying parameters at different layers. This *ns-2* property, which makes it extensible, is perhaps what has made it so popular [29].

Regarding sensor networks, *ns-2* is the most used WSN simulator, and this status has also encouraged further popularity, as developers would prefer to compare their work to results from the same simulator [29]. However, the learning curve for *ns-2* is steep and debugging is difficult due to the dual C++/OTcl nature of the simulator.

6.2 *ns-2* Architecture: C++ and OTcl duality

ns is written in C++, with an OTcl interpreter as a frontend interface to create and control the simulation environment itself, including the selection of output data. The simulator supports a class hierarchy in C++, and a similar class hierarchy within the OTcl interpreter. The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy [30].

ns uses two languages because the simulator has two different kinds of things it needs to do. On the one hand, detailed simulations of protocols requires a systems programming language which can

efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important.

On the other hand, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important.

ns meets both of these needs with two languages, C++ and OTcl. C++ is fast to run but slower to change, making it suitable for detailed protocol implementation. OTcl runs much slower but can be changed very quickly (and interactively), making it ideal for simulation configuration. *ns* links them both and makes objects and variables appear on both languages.

6.3 *ns-2* Simulation and Results Postprocessing

The *ns-2* simulator is initialized using a OTcl interface. This interface configures the simulation. A simulation script would generally begin calling various methods to create nodes, topologies, etc. When a new simulation object is created in OTcl, the initialization procedure performs the following operations:

1. Initialize the packet format (calls create_packetformat).
2. Create a scheduler.
3. Create a 'null agent' (a discard sink used in various places).

From this point on and since ns-2 is an event-driven simulator, the scheduler runs by selecting the next earliest event, executing it to completion, and returning to execute the next event. The simulator is single-threaded, and only has one event in execution at any given time.

The simulation generates two separate output files, an output trace file (*.tr) and a Nam trace file (*.nam). The trace file (*.tr) will contain information on the various events which occurred, details of node behavior, packet transmissions and receptions, communication layer, packet drops, etc. Analyzing this trace file using programs like gawk [31], can help us determine the performance of the network. On the other hand, the .nam file contains information on the topology, packet traces and other events. Its syntax is used by the Nam for visual representation of the simulated scenario. A complete flow diagram for the running simulation can be seen in Figure 6.2.

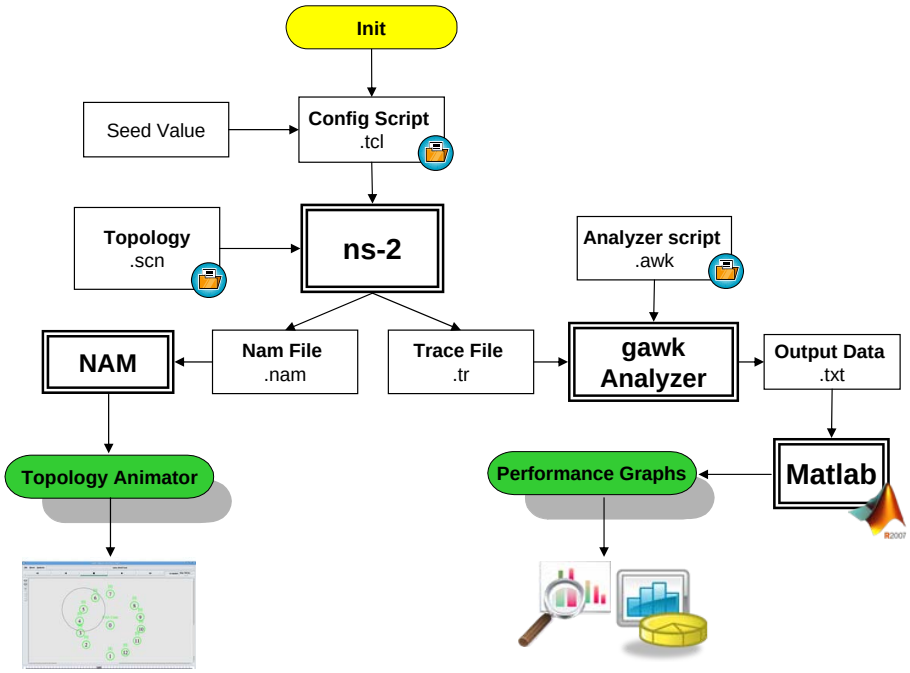

Figure 6.2: *Flow Diagram for Running Simulation in ns-2*

6.4 Basic ns-2 modules involved in WSN simulation

To perform this thesis, *ns-2* latest release (ns-2.33) was employed. It includes a contributed module named WPAM (Wireless Personal Area Networks), implementing partially IEEE 802.15.4. But before introducing the WPAN module, we will first describe some other general ns-2 modules needed to interact and simulate a WSN environment.

6.4.1 Mobile Node

The ns-2 wireless model essentially consists of the MobileNode object at the core, with additional supporting features that allows simulations of wireless networks. The MobileNode is a split object. The C++ class MobileNode is derived from parent class Node, receiving from it typical node features such as a node identifier, address and port, control functions, etc. MobileNode on its own implements the mobility features of the node including movement, periodic position updates, maintaining topology boundary, etc. The complete implementation can be found in $\sim ns2-33/common/mobilenode.\{cc, h\}$. The plumbing of network components within MobileNode itself (like classifiers, LL, MAC, Channel, etc.) are implemented in Otcl. Let's have a look at the Tcl API for the MobileNode:

```
$ns_ node-config -addressingType <usually flat or hierarchical used for wireless topologies>
  -llType <LinkLayer>
  -macType <MAC type like Mac/802_15_4>
  -propType <Propagation model like Propagation/TwoRayGround>
  -ifqType <interface queue type like Queue/DropTail/PriQueue>
  -ifqLen <interface queue length like 50>
  -phyType <network inteface type like Phy/WirelessPhy>
  -antType <antenna type like Antenna/OmniAntenna>
  -channelType <Channel type like Channel/WirelessChannel>
  -topoInstance <the topography instance>
  -energyModel <EnergyModel type>
    -initialEnergy <specified in Joules>
    -rxPower <specified in W>
    -txPower <specified in W>
  -agentTrace <tracing at agent level turned ON or OFF>
  -routerTrace <tracing at router level turned ON or OFF>
  -macTrace <tracing at mac level turned ON or OFF>
  -movementTrace <mobilenode movement logging turned ON or OFF>
```

It configures a mobile node with all the given values of MAC, channel, topography, propagation model, with wired routing turned on or off and tracing turned on or off at different levels (router, MAC, agent). A MobileNode is initialized as follows:

```
for { set j 0 } { $j < $opt(numnodes)} {incr j} {
  set node_($j) [ $ns_ node ]
  $node_($i) random-motion 0 ;# disable random motion
}
```

The above procedure creates a MobileNode object, creates the network stack consisting of a link layer, interface queue, MAC layer, and a network interface with an antenna, uses the defined propaga-

tion model, interconnects these components and connects the stack to the channel. The MobileNode then looks like the schematic in Figure 6.3.

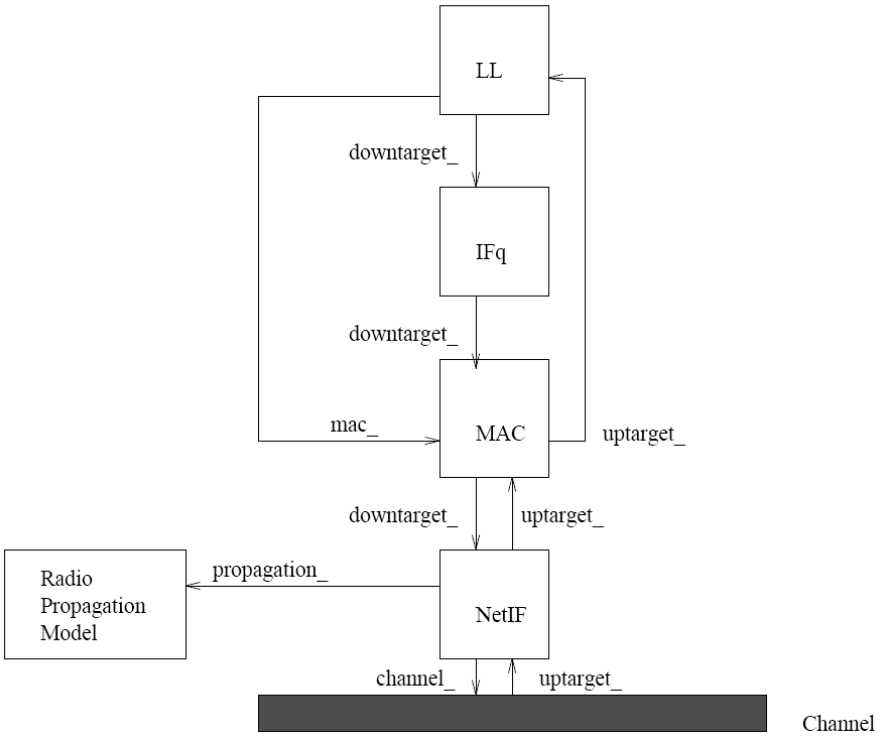

Figure 6.3: *Schematic of MobileNode in ns-2*

6.4.2 Radio Propagation Model

There are three propagation models in *ns*, which are the free space model, two-ray ground reflection model and the shadowing model. Their implementation can be found in $\sim ns{-}2.33/propagation.\{cc, h\}$, $\sim ns - 2.33/tworayground.\{cc, h\}$ and $\sim ns - 2.33/shadowing.\{cc, h\}$. These models are used to predict the received signal power of each packet. At the physical layer of each wireless node, there is a receiving threshold. When a packet is received, if its signal power is below the receiving threshold, it is marked as error and dropped by the MAC layer.

Free Space Model

The free space propagation model assumes the ideal propagation condition that there is only one clear line-of-sight (LOS) path between the transmitter and receiver. H. T. Friis presented the following

equation to calculate the received signal power in free space at distance d from the transmitter [32],

$$P_r(d) = \frac{P_t G_t G_r}{L} \cdot \left(\frac{\lambda}{4\pi d}\right)^2 \tag{6.1}$$

where P_t is the transmitted signal power. G_t and G_r are the antenna gains of the transmitter and the receiver respectively. L ($L \geq 1$) is the system loss, and λ is the wavelength.

The free space model basically represents the communication range as a circle around the transmitter. If a receiver is within the circle, it receives all packets. Otherwise, it loses all packets. The OTcl interface for utilizing a propagation model is the node-config command. One way to use it here is: `$ns_ node-config -propType Propagation/FreeSpace`

Two-ray ground reflection model

A single line-of-sight path between two mobile nodes is seldom the only means of propagation. The two-ray ground reflection model considers both the direct path and a ground reflection path. It is shown [33] that this model gives more accurate prediction at a long distance than the free space model. The received power at distance d is predicted by:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \tag{6.2}$$

where h_t and h_r are the heights of the transmit and receive antennas respectively. Equation (6.2) shows a faster power loss than Equation (6.1) as distance increases. However, The two-ray model does not give a good result for a short distance due to the oscillation caused by the constructive and destructive combination of the two rays. Instead, the free space model is still used when d is small.

Therefore, a cross-over distance d_c is calculated in this model. When $d < d_c$, Equation (6.1) is used. When $d > d_c$, Equation (6.2) is used. At the cross-over distance, Equation (6.1) and (6.2) give the same result. So d_c can be calculated as:

$$d_c = (4\pi h_t h_r)/\lambda \tag{6.3}$$

Similarly, the OTcl interface for utilizing the two-ray ground reflection model is as follows:
`$ns_ node-config -propType Propagation/TwoRayGround`

Shadowing model

ns-2 implements the shadowing model described in section 5.2. The user may select the values of the path loss exponent n and the shadowing deviation σ in dB according to the simulated environment.

The OTcl interface available to set this variables is the node-config command. One way to use it is as follows:

```
Propagation/Shadowing set pathlossExp_ 3.38 ;# path loss exponent
Propagation/Shadowing set std_db_ 6.2 ;# shadowing deviation (dB)
Propagation/Shadowing set dist0_ 0.1 ;# reference distance (m)
Propagation/Shadowing set seed_ 0 ;# seed for RNGç

$ns_ node-config -propType Propagation/Shadowing
```

The user may set an appropriate value of the receiving threshold in the network interface with the command:

```
Phy/WirelessPhy set RXThresh_ <value>
```

It is important to emphasize that in this model, $P_{0,dB}$ is not a variable of the model, but it is calculated according to the Friis equation defined in (6.1). It is common to select $G_t = G_r = 1$ and include in L all additional losses so that the desired $P_{0,dB}$ is obtained. The term $(4\pi d_0/\lambda)^2$ is normally referred as $L_{fs}(d_0)$ or free-space losses. Then we can rewrite:

$$P_{0,dB} = \frac{P_t G_t G_r}{L} \cdot \frac{1}{L_{fs}(d_0)} \tag{6.4}$$

6.4.3 Energy Model

The Energy Model, as implemented in ns-2, is only a node attribute. The energy model represents level of energy in a mobile host. It has an initial value which is the level of energy the node has at the beginning of the simulation. This is known as `initialEnergy_`. It also has a given energy usage for every packet it transmits and receives. These are called `txPower_` and `rxPower_`. The files where the energy model is defined are $\sim ns/energymodel.\{c,h\}$. Other functions methods described in this section may be found in $\sim ns-2.33/wireless-phy.cc$, $\sim ns-2.33/cmu-trace.cc$, $\sim ns-2.33/tcl/lib/ns-lib.tcl$, $\sim ns-2.33/tcl/lib/ns-node.tcl$ and $\sim ns-2.33/ns-mobilenode.tcl$.

The C++ EnergyModel Class

The basic energy model is defined by class EnergyModel. Some of the most important parameters of this class are shown below:

```
class EnergyModel : public TclObject
public:

EnergyModel(MobileNode* n, double energy, ...)

inline double energy() const { return energy_; }
inline double initialenergy() const { return initialenergy_; }
inline void setenergy(double e) { energy_ = e; }
...

virtual void DecrTxEnergy(double txtime, double P_tx);
virtual void DecrRcvEnergy(double rcvtime, double P_rcv);
virtual void DecrIdleEnergy(double idletime, double P_idle);
virtual void DecrSleepEnergy(double sleeptime, double P_sleep);
;
```

As seen from the EnergyModel Class definition above, there is a class variable `energy_` which represents the level of energy in the node at any given time. There are four class methods which are used to decrease the energy level of the node for every packet transmitted, every packet received, and during the Idle and Sleep time. `P_tx`, `P_rcv`, `P_idle` and `P_sleep` are respectively the transmitting, receiving, idle and sleep powers required by the node's interface. When the energy level at the node goes down to zero, no more packets can be received or transmitted by the node.

The OTcl interface

As we already saw, the energy model is a node attribute, defined by the following node configuration APIs:

```
$ns_ node-config
  -energyModel $energymodel \
  -rxPower $p_rx \ Watts
  -txPower $p_tx \ Watts
  -idlePower $p_idle\ Watts
  -sleepPower $p_sleep \ Watts
  -initialEnergy $initialenergy \Joules
```

6.5 IEEE 802.15.4 in *ns-2*

As we anticipated at the beginning of section 6.4, the ns-2 WPAN module implements the IEEE 802.15.4 standard partially. The WPAN C++ source code contains about 12000 lines. It is based on the release of Jianliang Zheng [34] [17] from USC and includes the updates of Iyappan Ramachandran [2] from University of Washington. It supports:

- Beacon enabled (Slotted CSMA/CA) and non-beacon enabled (pure CSMA/CA) modes.
- Star and Peer-to-Peer topologies.

However, we said that the protocol is partially implemented because it does not support the Contention Free Period, which is a major drawback when simulating IEEE 802.15.4. This simulation limitation will not affect our work, since we will only analyze the contention access period of its beacon enabled mode.

It is important to emphasize how the WPAN module documentation is very limited. At the same time tools to describe simulation scenarios and analyze simulation traces have to be written independently in scripting languages. This lack of documentation and generalized analysis tools may lead to different people measuring different values for the same metric names.

6.5.1 C++ WPAN CSMA/CA procedure

In this section we will describe the fundamental *ns-2* code flow (based on the release notes from [2]) when an outgoing or an incoming packet is received by the MAC, following the CSMA/CA pro-

cedure. We do so because it is probably the most important and the most critical aspect of the IEEE 802.15.4 implementation, and should be well understood. The complete code can be found in $\sim ns-2.33/wpan/p802\_15\_4mac.\{cc,h\}$.

Outgoing Packet:

1. The upper layer hands down a packet to MAC by calling `Mac802_15_4::recv(Packet *p, Handler *h)`.

2. `recv()` then calls `Mac802_15_4::mcps_data_request( )`. `mcps_data_request()` works by using a variable called step, which is initialized to 0. Every time the function needs to pass control to a different function it increments step so that when the control returns to it, it will know where to proceed next.

3. For direct transmission `mcps_data_request()` then calls `Max802_15_4::csmacaBegin(pktType)` after incrementing step to 1.

4. `csmacaBegin` in turn calls `CsmaCA802_15_4::start()`. `start()` calculates the random backoff time, determines if it can proceed and starts the `macBackoffTimer` with the backoff time determined.

5. On expiry of the timer, `CsmaCA802_15_4::backoffHandler()` is called. `backoffHandler` then turns on the receiver (through `Phy802_15_4::PLME_SET_TRX_STATE_request()`) and requests a CCA by calling `Phy802_15_4::PLME_CCA_request()`.

6. `PLME_CCA_request()` synchronizes the CCA calling `Phy802_15_4::CCAHandler` which searches if channel is idle at the end of the backoff slot. Its finding is then reported by calling `Mac802_15_4::PLME_CCA_confirm()`.

7. If the channel is found busy, `CsmaCA802_15_4::start()` is called to go to the next backoff stage. If the channel is found to be idle and if CW$\neq$0, `CsmaCA802_15_4::CCA_confirm()` called by `PLME_CCA_confirm()` decrements CW and in turn calls `backoffHandler` to perform CCA again. If CW=0, it calls `Mac802_15_4::csmacaCallback()`.

8. `csmacaCallback()` subsequently returns control to `Mac802_15_4::mcps_data_request()`, which enables the transmitter by calling `Phy802_15_4::PLME_SET_TRX STATE_request()` after incrementing step to 2.

9. `Mac802_15_4::PLME_SET_TRX_STATE_confirm()` then passes the data to `Mac802_15_4::txBcn CmdDataHandler()`, which uses `Mac802_15_4::senDown()` to give the data to `Phy802_15_4::recv (Packet *p, Handler *h)`

10. recv() then calls Phy802_15_4::PD_DATA_request(), which in turn uses `WirelessPhy::sendDown()` (`Phy802_15_4` is a sub-class of `WirelessPhy`) to decrement energy and transmit the data to `Channel::recv()`.

Incoming Packet:

1. `Channel::recv()` gives one copy of the received packet to each node using `WirelessChannel::sendUp()`, which subsequently passes the packet to `Phy802_15_4::recv()` after propagation delay.

2. `Phy802_15_4::recv()` uses `WirelessPhy::sendUp()` to decrement energy and indicates packet reception to MAC using `Phy802_15_4::PD_DATA_indication()`. `Phy802_15_4::recvOverHandler()` involved here drops packets not intended for the node. `PD_DATA_indication()` then calls `Mac802_15_4:: recv(Packet *p, Handler *h)`.

3. `recv()` drops the packet if there is a collision or calls `Mac802_15_4::recvData()` if there is no collision. `recvData()` then calls `Mac802_15_4::MCPA_DATA_indication()`, which passes the data to the upper layer.

6.5.2 WPAN Tcl Interface commands

The WPAN simulation scenarios are implemented using Tcl scripts that comprise commands and parameters for the simulator initialization, node creation and configuration. The most important ones are the following:

- `\$node sscs startPANCoord <txBeacon=1> <beaconOrder=3> <SuperframeOrder=3>.` This command can be used to start a new PAN, and the corresponding node will serve as the PAN coordinator. If some parameters are omitted, the default values shown will be assumed.

  - Examples:

    * `$node_(0) sscs startPANCoord`
    * `$node_(0) sscs startPANCoord 1 2 2`

- `\$node sscs startDevice <isFFD=1> <assoPermit=1> <txBeacon=0> <beaconOrder=3> <SuperframeOrder=3>.` This Command can be used to start a device or coordinator. If some parameters are omitted, the default values shown will be assumed.
  - Examples:
    - `$node_(0) sscs startDevice 0 //device`
    - `$node_(0) sscs startDevice //coor., non-beacon`
    - `$node_(0) sscs startDevice 1 1 1 //coor., beacon enabled`
- `\$node sscs startBeacon <beaconOrder = 3> <SuperframeOrder = 3>.` Start to transmit beacons if originally in non-beacon mode, or change the beacon order and superframe order if originally in beacon mode.
- `\$node sscs stopBeacon` Stop the transmission of beacons.
- `Mac/802_15_4 wpanCmd ack4data [on/off]` MAC level acknowledgement for upper layer packets. Default: on

6.5.3 WPAN Trace Format

The WPAN module uses two types of output traces:

1. Wireless Event Trace.
2. Node's energy update trace after a sent event.

The event trace is similar (but not equal) to what the ns manual [30] calls "Old Wireless Trace Format". The node's energy update trace is logged into the trace file right after a coordinator sent event. Let's see it with an example (the meaning of each particular trace field of the traces can be seen at Table 6.1 and Table 6.2.):

```
s 156.511552027 _5_ MAC  --- 851 exp 94 [0 0 5 800] [energy 994.478060 ei 5.409 es
    0.000 et 0.009 er 0.104]
N -t 156.511552 -n 4 -e 994.477962
N -t 156.511552 -n 6 -e 994.478009
N -t 156.511552 -n 3 -e 994.477998
N -t 156.511552 -n 0 -e 994.478160
N -t 156.511552 -n 7 -e 994.478027
```

```
N -t 156.511552 -n 2 -e 994.478110
N -t 156.511552 -n 8 -e 994.478001
N -t 156.511552 -n 1 -e 994.478097
N -t 156.511552 -n 9 -e 994.477891
N -t 156.511552 -n 12 -e 994.477964
N -t 156.511552 -n 10 -e 994.478035
N -t 156.511552 -n 11 -e 994.477928
s 156.515072000 _0_ MAC  --- 851 ACK 5 [0 5 0 0] [energy 994.478160 ei 5.411 es 0.000
     et 0.014 er 0.097]
N -t 156.515072 -n 2 -e 994.478086
N -t 156.515072 -n 5 -e 994.477917
N -t 156.515072 -n 1 -e 994.478074
N -t 156.515072 -n 7 -e 994.478004
N -t 156.515072 -n 8 -e 994.477978
N -t 156.515072 -n 11 -e 994.477904
N -t 156.515072 -n 6 -e 994.477986
N -t 156.515072 -n 12 -e 994.477941
N -t 156.515072 -n 3 -e 994.477974
N -t 156.515072 -n 9 -e 994.477867
N -t 156.515072 -n 10 -e 994.478012
N -t 156.515072 -n 4 -e 994.477938
D 156.515262326 _5_ IFQ  --- 853 exp 87 [0 0 5 800] [energy 994.477917 ei 5.409 es
    0.000 et 0.009 er 0.104]
r 156.515424027 _5_ MAC  --- 851 ACK 5 [0 5 0 0] [energy 994.477917 ei 5.409 es 0.000
     et 0.009 er 0.104]
r 156.516064000 _0_ MAC  --- 851 exp 87 [0 0 5 800] [energy 994.478136 ei 5.411 es
    0.000 et 0.014 er 0.097]
```

In the above sequence we have the following events: 1) Node 5 sends a Poisson data packet to the coordinator (Node 0) with id 851. 2) The coordinator replies to this data packet with id 851 with an ACK destined to node 5, but before the packet reception has even been logged into the trace file. 3) A new packet (with id 853) ready to transmit arrives to the MAC layer of node 5 but it is dropped by the queue since node 5 is currently busy waiting for the ACK from the coordinator. 4) Node 5 receives the ACK from the coordinator. 6) The reception of the data packet with id 851 at the coordinator is finally logged into the trace file. On this whole process, after each packet sent recorded, an energy update trace is logged in the trace file right after.

Table 6.1: Wireless Event Trace format

Field	Type
Event	s: Sent r: Receive d: Drop
Time	double
Node Id	int
Trace Name	string
Reason	string
Event Id	int
Packet Type	string
Packet Size	int
Time To Send Data	hexadecimal
Destination MAC Address	hexadecimal
Source MAC Address	hexadecimal
Type	hexadecimal
Remaining Energy	double
Energy Consumed in Idle	double
Energy Consumed in Shutdown	double
Energy Consumed in Tx	double
Energy Consumed in Rx	double

The sequence described in the above paragraph seems very confusing. How can the coordinator send an ACK before it has even received the data packet? To answer this question must explain how the trace hook works in *ns-2*.

When a packet is received by the coordinator MAC, it is not logged into the trace file immediately. The packet is logged into the trace file when the packet is passed from the MAC to upper layer. When a packet is received by the MAC, an ACK is first sent (if the reception is successful) and then the packet is passed up. So although the packet is received first, it appears later in the log file. Another way to clear the confusion is to treat the time that appears in the trace file as packet 'passing-up' time rather than packet arriving time.

Table 6.2: Node's Energy Update Trace

Field	Type
Time -t	double
Node Id -n	int
Remaining Energy -e	double

6.6 Performance Metrics calculations in ns-2

The performance metrics of simulated network under test, as were defined Section 4.4, have to be extracted from the *ns-2* output trace in order to make a valid comparison between both the analytical model and the simulation results. In this section, we will describe how this is done.

6.6.1 Aggregate throughput

The total number of received bytes by the coordinator of the network can be easily extracted from the output trace. We just have to make sure to add we add 13 bytes to the each received data payload, corresponding to the *ns* header. Then the aggregate throughput is:

$$S = \frac{TotalReceivedBytes \times 8}{SimulationTime \times BitRate} \tag{6.5}$$

where the Bit Rate=250 Kbps for the 2.45 GHz band.

6.6.2 Probability of accessing the channel

Similarly to throughput, we can calculate the probability of accessing the channel from the sent bytes and total simulation time as:

$$P_t = \frac{TotalSentBytes \times 8}{M \times SimulationTime \times BitRate \times N} \tag{6.6}$$

6.6.3 Average Power Expenditure

From the output trace, the initial energy (at the very beginning of the simulation) and the remaining energy (at the end of the simulation) can be obtained. Then the average power expenditure for a single node can be computes as:

$$Y_{av} = \frac{\sum_{i=1}^{M}(InitialEnergy_i - RemainingEnergy_i)}{M \times SimulationTime} \tag{6.7}$$

where M is the number of sensing nodes.

6.6.4 Bytes-Per-Joule Capacity

The bytes-per-Joule capacity is calculated from the obtained S and Y_{av} following the same formula as for the analytical model:

$$\eta = \frac{(S/M)(250 \times 10^3/8)}{Y_{av}} \tag{6.8}$$

6.6.5 Latency

As far as latency is concerned, we were unable to extract it from the *ns-2* output trace, since not enough information is available. The time instant when a packet becomes available for transmission is not recorded on the trace file.

Part IV

Results, Conclusions and Future Work

CHAPTER 7

Analytical Results and Simulation Validation

7.1 Considerations before solving the model analytical and simulating

7.1.1 Parameters for the analytical resolution and simulation validation

The analytical model presented in Chapter 4 will be solved using Matlab, and then the obtained results will be validated through comparing them with ns-2 simulation results. But prior to the resolution and later simulation, we should first define the value of certain parameters that will be used. The most important ones are described in Table 7.1.

The values of $aMaxBE$ and CW are fixed in the standard definition. The value of $macMaxCSMABackoffs$ and $aMinBE$ are the maximums from its available range ($[0,5]$ and $[0,3]$ respectively). The p^n_{beacon}

CSMA/CA parameters	$aMinBE = 3$	$aMaxBE = 5$
	$macMaxCSMABackoffs = 5$	$CW = 2$
	$BCO = 6$	$SFO = 6$
Analytical parameters	$p^n_{beacon} = 1/3072$	
Data Packet size	$N = L_{data} = 10 backoffslots$	$n_{beacon} = 2 backoffslots$
Number of sensing Nodes	$M = 12$	

Table 7.1: Parameters used to solve the model and perform simulations.

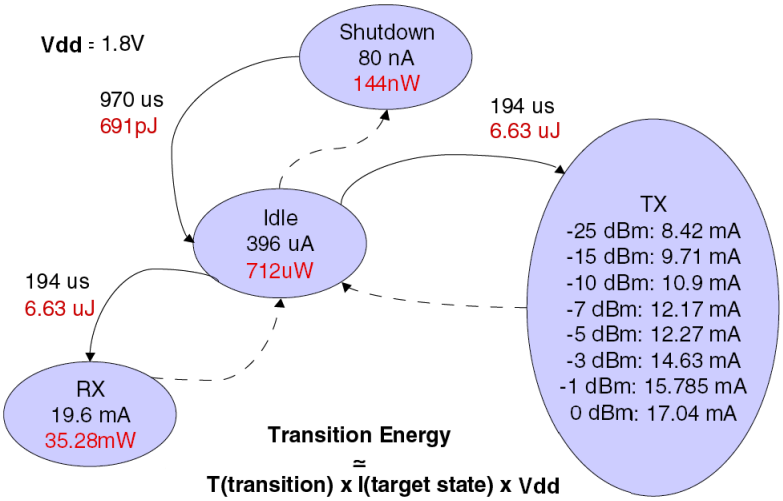

Figure 7.1: *Energy States and Transitions for the CC2420 transceiver*

	Max [dBm]	Min [dBm]
Sensitivity $S(R)$	-94	-90

Table 7.2: CC2420 sensitivity ratings

value corresponds to a $BCO = 6$ or what is the same, a $BI = 3072$ backoff slots. The size of the beacon and the data packet are set according to [2] so that we could compare both performances.

7.1.2 Energy State Values

The energy values of each energy state remain to be defined. For illustration purposes and just as it is done in [2] we will be considering the Chipcon IEEE 802.15.4 CC2420 RF transceiver [35] energy values. The CC2420 is very popular low-power chipset used in well tested sensors such as TelosB, MICAz or BSNnode. Detailed measurements of the power consumption in each state of this transceiver have been reported in [36] and are reproduced in Figure 7.1. For both the analytical resolution and ns-2 simulation, these energy parameters will be employed. For the ns-2 simulation, we will be using the CC2420 sensitivity, which is found in [35] and described in Table 7.2.

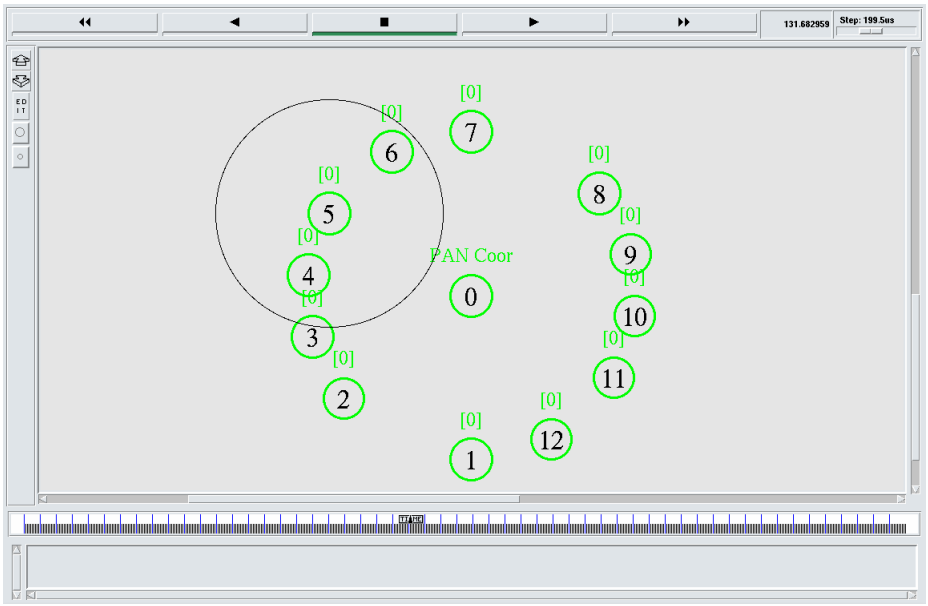

Figure 7.2: *ns-2 sensor localization as seen in Nam*

7.1.3 ns-2 Topology and Simulations Average

ns-2 expects that the user defines a topology in the OTcl script. Since in the analytical model we defined a set of M sensing devices with exactly the same configuration, for simulations we will locate the nodes uniformly distributed around a circle with the coordinator in the center. The radius of the circle will be initially set to $r = 1m$. This will emulate as if the sensors were within a body distance from the coordinator, as shown in the Nam screen shot of Figure 7.2. As an input to the Tcl script, we will also defined the Poisson inter arrival rate λ for traffic generation. For each traffic input we will perform 20 ns-2 simulations, to later obtain the output mean for all these realizations. By doing this, we will guarantee that the obtained results are valid and are not biased by only one realization. This process was automated with a bash script.

7.1.4 ns-2 overhearing problem

When dealing with energy simulations, we found an ns-2 bug that up-to-date remains as an unsolved issue. For some reason out of our knowledge, the sensing nodes waste energy overhearing packets from other nodes addressed to the coordinator, even when they are explicitly configured to only receive during the beacon time as well the CCA time. An example of the problem can be seen in Figure 7.3. If we break down the average energy consumption into the energy that is consumed on each state

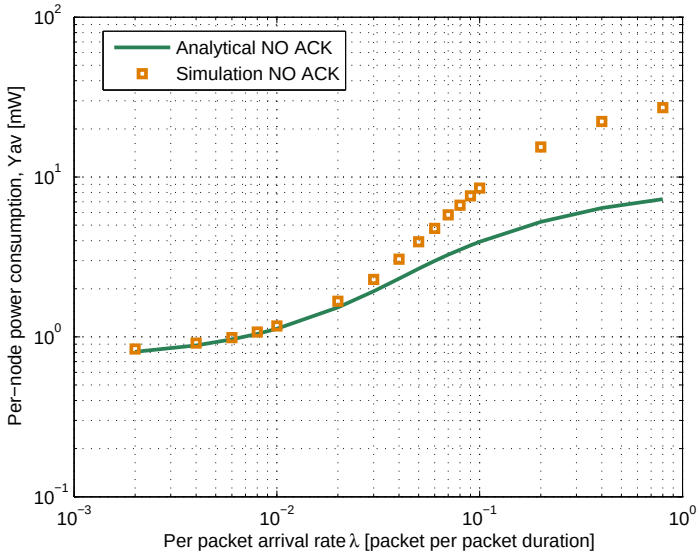

Figure 7.3: *Per node power consumption*

(as shown in Figure 7.4), we rapidly identify the problem. For the transmitting state, the analytical and simulation results for energy consumptions match. But if we look closely, we can also see how the simulation energy consumption for the receiving state goes above the analytical prediction. This implies a reduction of the idle energy consumption, since the nodes spend more time overhearing and less time idle. This difference is almost unnoticeable for low data rates, but should be taken into consideration beyond data rates of $\lambda = 0.02$. In any case, it is interesting to mention how the receiving energy consumption represents the major part of the total system energy consumption breakdown.

We put a great effort trying to solve this overhearing problem but due to the project time constraint and learning curve for ns-2,we were finally unable. ns-2 is an environment with a complicated interdependency between different classes and objects, which makes solving problems like this one a really hard task. Because of this, the rest of the energy figures that will be presented on the ensuing sections will not include a simulation validation.

7.2 Performance Comparison between Acknowledged and Unacknowledged traffic

In this section we present a set of results for the resolution of the acknowledged and unacknowledged analytical model without losses. The results are validated with simulation results. The ns-2 simulator

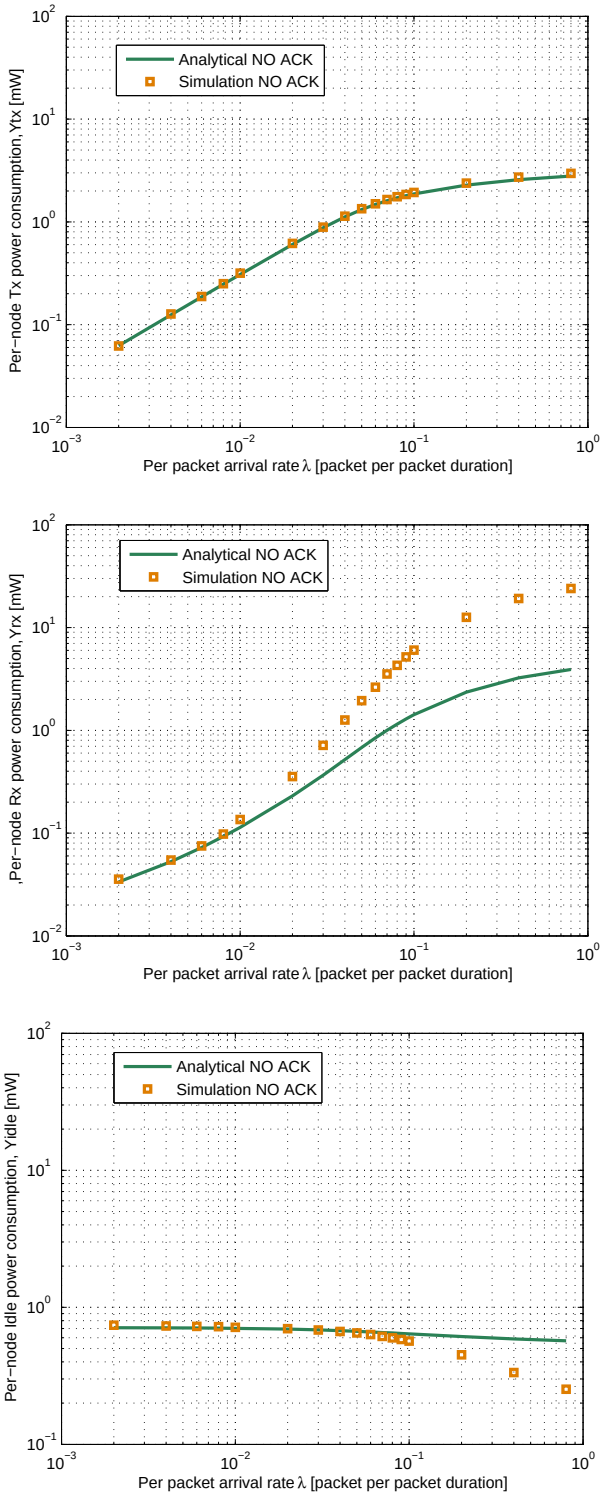

Figure 7.4: *Per node Tx, Rx & Idle power consumptions*

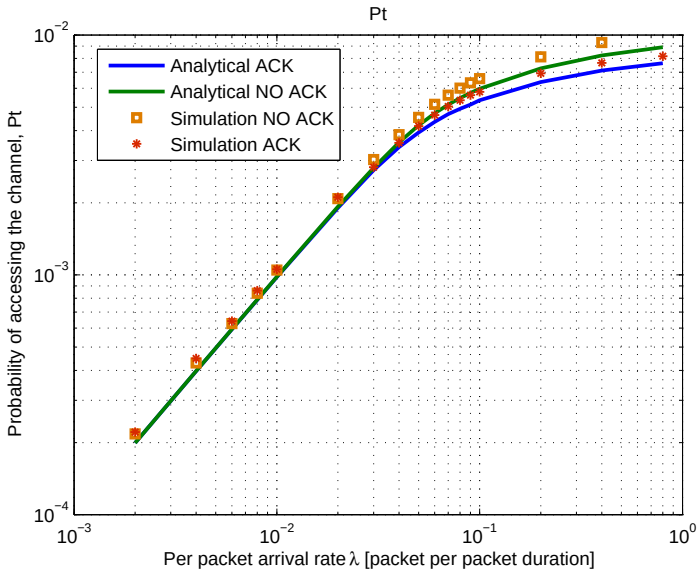

Figure 7.5: *Probability of accessing the channel*

worked under a two-ray ground reflection propagation model, to consider both the direct path and a ground reflection path. The transmission power was set to the maximum CC2420 rating , that is $0dBm$. Under these simulation settings, there are no packets losses due to the two-ray model channel conditions. Only collisions lead to transmission failures.

7.2.1 Probability of accessing the channel

In Figure 7.5 we can view the comparison in terms of probability of accessing the channel for the ACK and non-ACK scenarios. Their respective ns-2 validation is also plotted. We can observe the befitted accuracy of our analytical model compared with the simulation results. With this justification and along with the next to be presented Figure 7.6, we could state that the analytical approximations made to develop the analytical system are now validated.

7.2.2 Throughput

In Figure 7.6 we can see the comparison in terms of throughput of the ACK and non-ACK models. First of all, it can be seen how for both acknowledged and unacknowledged results, the maximum achieved throughput is lower than the maximum bit rate, due to collisions.

The plot reveals that no significance difference in terms of throughput is noticed between acknowledged and unacknowledged traffic up to a packet arrival rate of $\lambda = 0.02$ (or equivalently 5 kbps). This

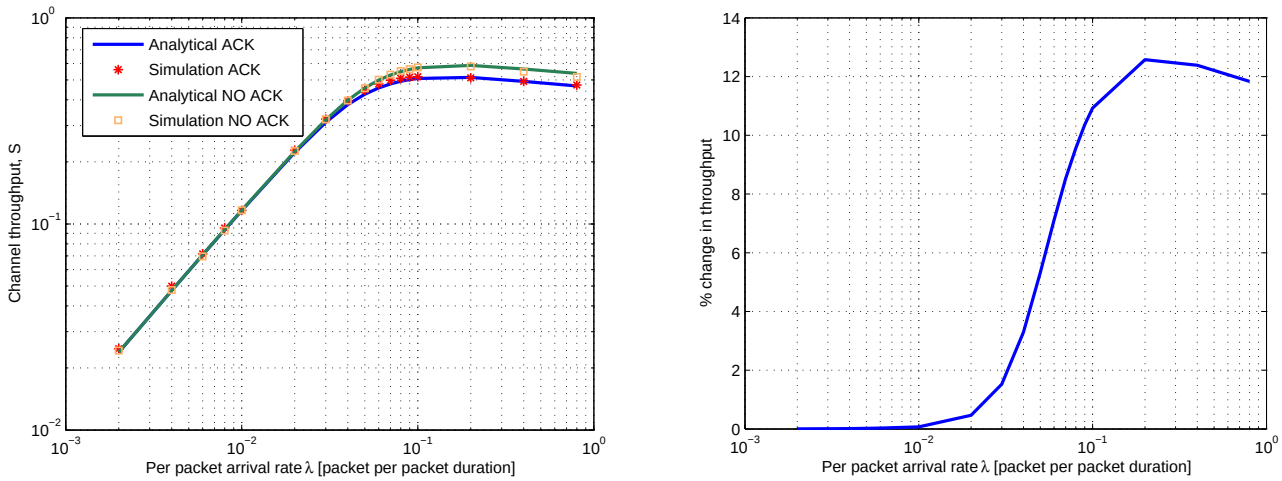

Figure 7.6: *Throughput comparison in ACK and non-ACK models*

is because for low packet arrival rates the nodes are mainly idle. For higher data rates, the ACK model throughput starts to decrease with respect to the non-ACK case almost linearly, reaching a significance maximum difference of 12.7% for $\lambda = 0.2$. This decrement is coherent, since as the network load increases, the channel becomes more and more congested. The transmission of the acknowledgments contributes increasing even more the network traffic, resulting in more channel congestion and reducing the overall throughput.

We can also distinguish how for traffic loads higher than $\lambda = 0.1$ that is 25 kbps, the overall throughput performance starts to decrease due to channel congestion, and consequently collisions. Therefore, to obtain the maximum network performance, we should always work below this threshold.

7.2.3 Power consumption

In Figures 7.7 and 7.8, we can contemplate the behavior of the MAC analytical models in terms of energy consumption. We can see how the inclusion of the ACK increases the power consumption in the sensing nodes. Extra energy is consumed waiting and receiving the ACK. This difference in the power consumption for both models increases very little but constantly with the packet arrival rate to reach a maximum difference of 7% for the channel maximum load. Naturally, as the power consumption increases for the acknowledged model, the bytes per joule capacity decreases. However, this decrement is not linear because it is adjusted by the channel throughput. The maximum difference is achieved for $\lambda = 0.06$ obtaining a difference of 15.8%. It should also be pointed out how there is a trade-off between

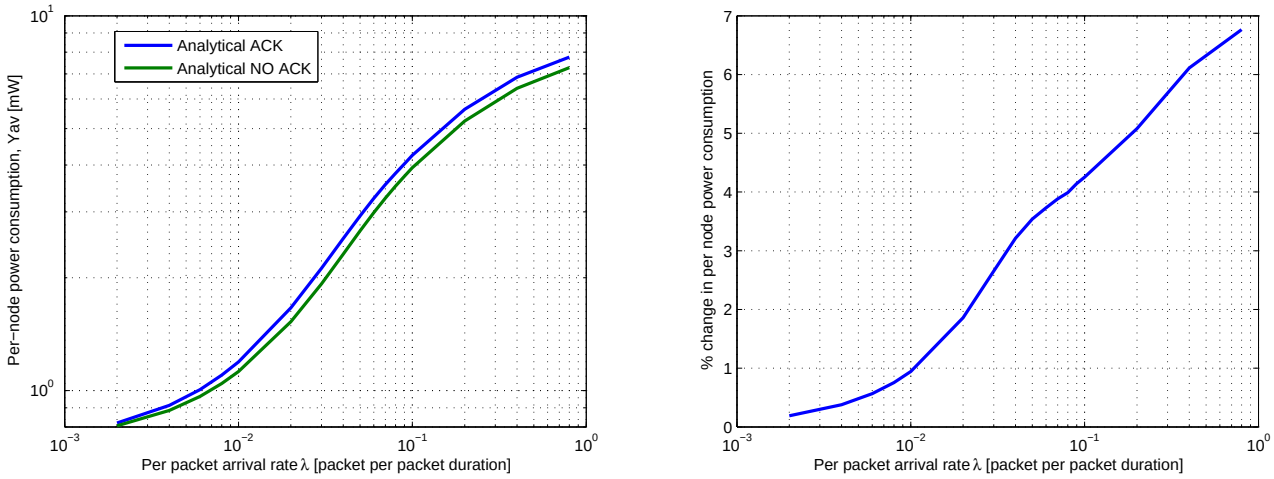

Figure 7.7: *Average per node power comparison in ACK and non-ACK models*

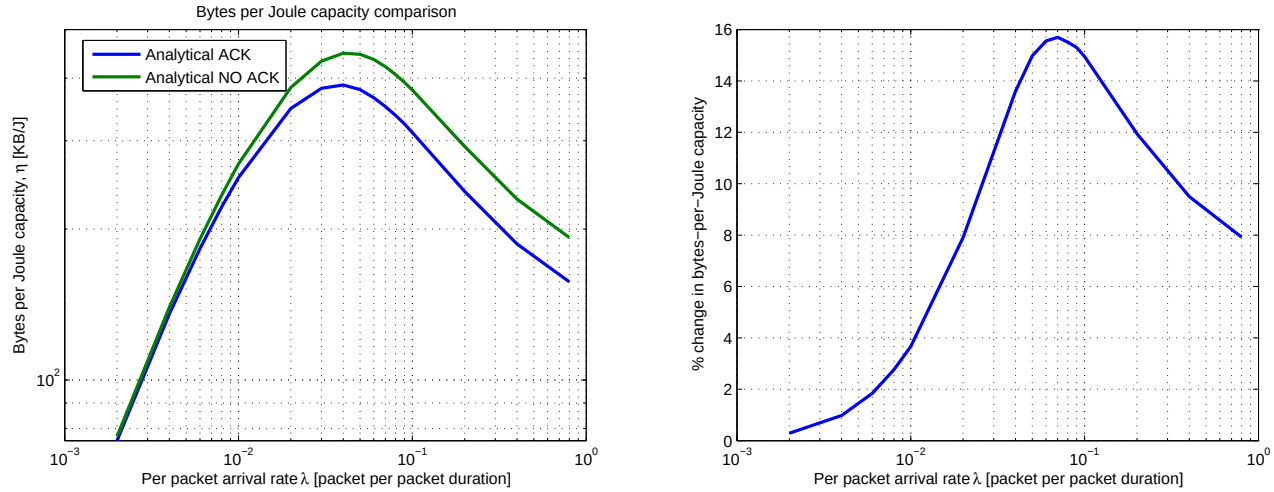

Figure 7.8: *Bytes per Joule capacity comparison in ACK and non-ACK models*

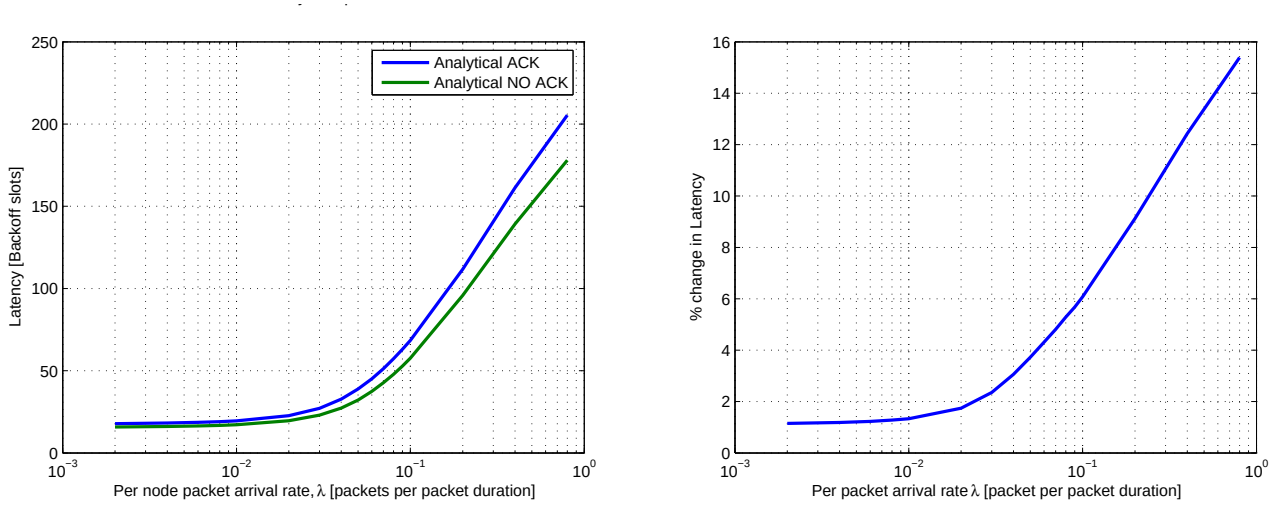

Figure 7.9: *Latency comparison in ACK and non-ACK models*

throughput and energy consumption. The best results are obtained for $\lambda = 0.04$ which translates into a data rate of 10 kbps, for which the highest bytes per joule capacity is obtained.

7.2.4 Latency

For both the ACK and non ACK models, latency follows a similar trend as seen in Figure 7.9. Latency is kept below 25 backoff slots up to a value of $\lambda = 0.03$, where it starts to increase dramatically due to channel congestion. With the inclusion of the ACK in the model, latency obviously increases. This means that the time frame between the instant a packet becomes available for transmission to the end of its successful transmission also increases. This increment gets larger monotonously as the packet arrival rate increases as seen on Figure 7.9. A maximum increment of 15.6% in terms of latency, is achieved for the ACK model.

7.3 Performance Results for a high path loss WBSN

In this section we present the performance results for acknowledged traffic working in a high pathloss channel model as described in section 5.2. In the OCTL simulation script, we will insert all the parameters that describe the channel (see Table 7.1). Additionally, the simulation receiving threshold `RXThresh_` will be set to the minimum CC2420 receiving sensitivity, that is:

$$\texttt{RXThresh\_}=S(R)_{min} = -90dBm.$$

In order to achieve the desired reference power $P_{0,dB}$ in ns-2 simulations, the system loss, L defined in Equation (6.4) needs to be fixed. For the LOS shadowing model is obtained as follows:

$$L_{LOS_{dB}} = P_{0,dB} - L_{fs,dB}(d_0) = 35.7dB - 20.05dB = 15.65dB$$

that is, $L_{LOS} = 36.76$ in natural units. Proceeding the same way, for the desired $P_{0,dB}$ NLOS shadowing model we obtain respectively:

$$L_{NLOS_{dB}} = P_{0,dB} - L_{fs,dB}(d_0) = 48.8dB - 20.05dB = 28.75dB$$

which is equal to, $L_{NLOS} = 749.89$ in natural units.

7.3.1 Performance in the LOS channel

To achieve a particular probability of losing a packet, P_e, the minimum transmitted power should be set accordingly, under the channel model characteristics. For the LOS channel, we will first compute the pathloss PL, for the given coordinator-to-node distance ($d = 1m$) and for a communication reliability of 95% ($P_e = 5\%$). Proceeding this way and according to Equations (5.1), (5.2) and (5.3) we derive:

$$\begin{aligned} PL(d = 1m) &= P_{dB} + P_s \\ &= P_{0,dB} + 10nlog(d/d_0) + \sqrt{2}erfc^{-1}(2P_e)\sigma \\ &= 35.7dB + 33.8dB + 10.2dB = 79.7dB \end{aligned}$$

Thus, the minimum transmitted power to achieve the desired communication reliability is:

$$\begin{aligned} P_{t_{min}}|_{dBm} &= P_{r_{min}}|_{dBm} + PL \\ &= S(R) + PL = -90dBm + 79.7dB = -10.3dBm \Rightarrow P_{t_{min}} = 0.093mW \simeq 0.1mW \end{aligned}$$

On Figure 7.10 we plot the analytical results for $Pe = 5\%$ and the equivalent simulation for $P_t = 0.1mW$, together with the analytical graphs when there are no packet losses in the channel.

By looking at the analytical plots, we observe an important tendency; there is a constant reduction in terms of throughput when we move from $P_e = 0\%$ to $P_e = 5\%$.This occurs, because the probability of losing packet in the analytical model is constant. However, as we reduce the transmitted power in simulation time, the throughput reduction is non constant. On the other hand, the analytical results for $P_e = 5\%$ are higher than the equivalent simulations results ($P_t = 0.1mW$) for data rates above

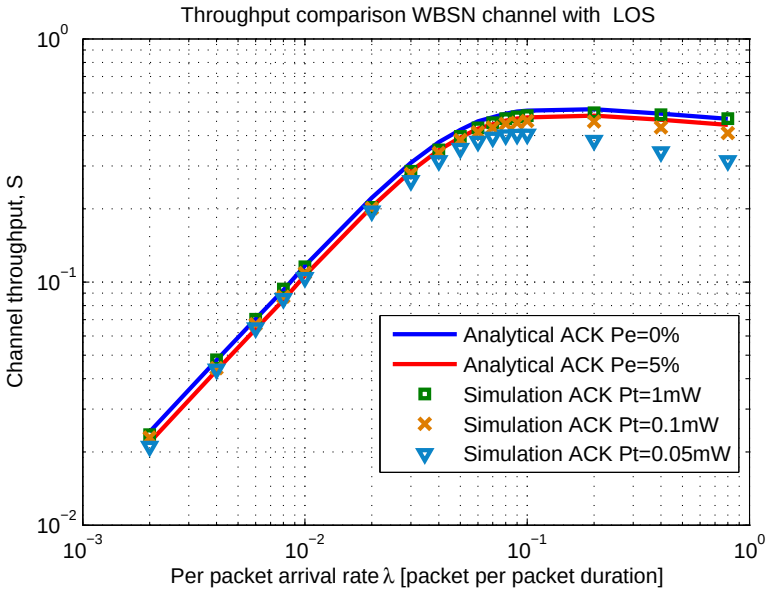

Figure 7.10: *Throughput comparison WBSN channel with LOS*

$\lambda = 0.1$. Both of this facts could mean that there are some important factors affecting the probability of losing packets which are not captured in the analytical model. The most important one is found to be the hidden node problem.

Hidden nodes in a wireless network environment, refer to nodes that are out of range of other nodes. In a star topology, each sensing node must be within communication range of the coordinator, but not necessarily within communication range of all the other sensing nodes. Therefore, it might happen as in Figure 7.11 , where nodes 1 and 5 are not within communication range of each other. Then the CSMA/CA procedure to communicate with the coordinator node will not work properly, since the radios of both sensing nodes cannot reach each other. This hidden node problem is not so severe for lower data rates, but becomes an important issue as congestion increases. A way to solve the problem is by means of increasing the transmitted power P_t. Let's calculate the power value for which the hidden node problem is avoided in our network configuration. It will be denoted as $P_{t_{NoHNP}}$. The minimum radius of coverage a node should have to avoid hidden nodes, is the distance to the node at the far edge of the topology; that is the distance to the node located diametrically opposed to him ($d = 2m$). Then:

$$PL(d = 2m) = 89.87dB \Rightarrow P_{t_{NoHNP}}|_{dBw} = -0.13dBmW \Rightarrow P_{t_{NoHNP}} = 0.97mW \simeq 1mW$$

In Figure 7.10, the $P_t = 1mW$ simulation plot is clearly not affected by the hidden node problem

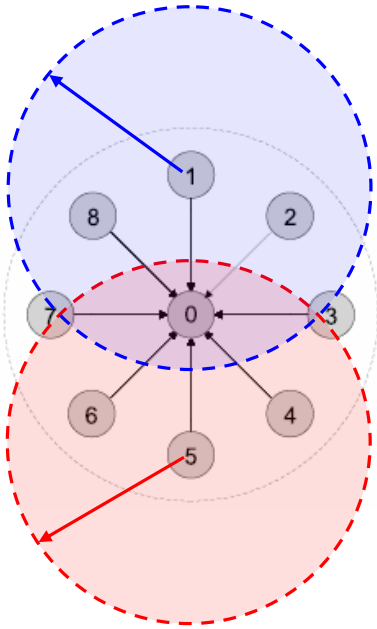

Figure 7.11: *Illustration of the hidden terminal problem*

and matches the analytical results without losses. To further highlight the effect of the hidden terminal problem, the $P_t = 0.05mW$ throughput simulation results have been plotted, where it can be observed how there is a higher impact of the hidden terminal problem as the traffic load increases.

To further observe the behavior of the energy consumed in this LOS channel, Figure 7.12 shows the per-node average energy consumed for the transmitted powers of 1mW and 0.1mW, which are very illustrative examples since they are almost unaffected by the hidden node problem It can be observed how a reduction of the transmitted power by a factor of 10, is not translated in the same reduction in the overall consumed power. This is because most of the power consumed by the node is invested in the receiving state rather than in the transmitting state.

Finally in Figure 7.13, we can observe how the latency increases, when the probability of losing a packet also increases. However this increment is almost negligible.

7.3.2 Performance in the NLOS channel

In the NLOS channel introduced in Section 5.2, we observe a higher path loss due to diffraction around the human body and absorption of a large amount of radiation by the body. Because of this, it can be easily computed that if we keep the topology as previously (distance to the coordinator $d = 1m$), in this NLOS model, communication between the nodes and the coordinator is unfeasible for

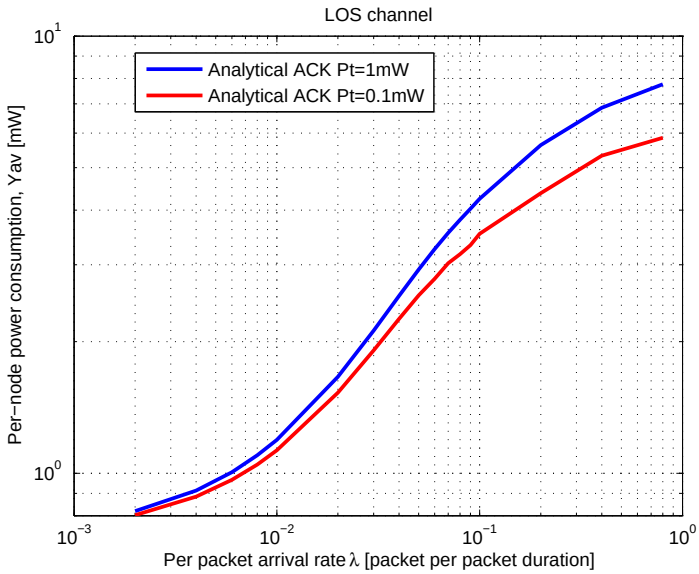

Figure 7.12: *Per-node power consumption in LOS channel*

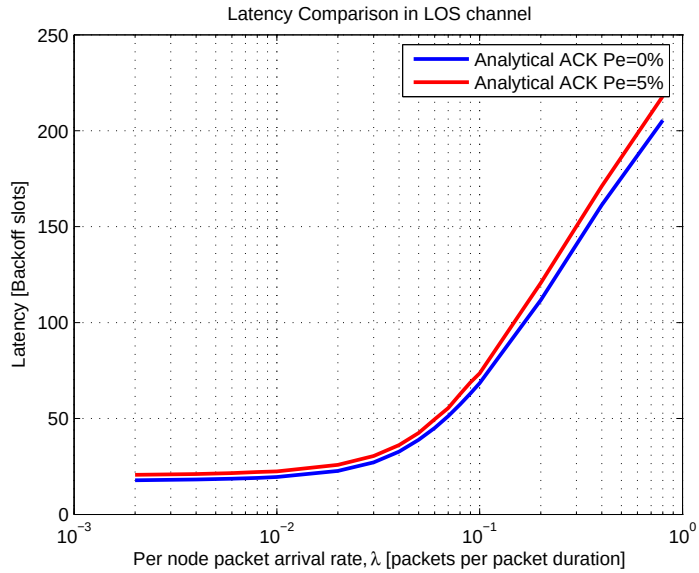

Figure 7.13: *Latency in LOS channel*

the given transmitting powers of the CC2420 transceiver.

A NLOS channel could for example model a situation where the sensing nodes are located in the back of the torso of a person, while the coordinator is located in the front torso. Then, in this type of channels, the distance between the sensing nodes and the coordinator is normally smaller than in the previous LOS channel. When simulating this scheme, we will therefore locate the nodes closer to the coordinator. A good illustrative example could be positioning the sensing nodes around a circle with radius 0.3m, and the coordinator in the middle. Under this working topology lets compute again the pathloss PL for $d = 0.3m$ and a communication reliability of 95% ($P_e = 5\%$):

$$\begin{aligned} PL(d = 0.3) = P_{dB} + P_s & \\ &= P_{0,dB} + 10nlog(d/d_0) + \sqrt{2}erfc^{-1}(2P_e)\sigma \\ &= 48.8dB + 28.15dB + 8.22dB = 85.17dB \end{aligned}$$

The minimum transmitted power to reach the coordinator and to achieve the desired communication reliability is calculated as:

$$\begin{aligned} P_{t_{min}}|_{dBm} &= P_{r_{min}}|_{dBm} + PL \\ &= S(R) + PL = -90dBm + 85.17dB = -4.83dBm \Rightarrow P_{t_{min}} = 0.32mW \end{aligned}$$

We will also compute the expected transmitted power for which the hidden node problem is avoided ($d = 0.6m$):

$$PL(d = 0.6m) = 102.93dB \Rightarrow P_{t_{NoHNP}}|_{dBw} = 12.93dBmW \Rightarrow P_{t_{NoHNP}} = 19.64mW$$

It can be observed that the obtained $P_{t_{NoHNP}}$, is actually higher than the maximum power available for transmission at the CC2420 transceiver. Then, for the NLOS channel modeling, the hidden node problem can't be solved by means of increasing the transmitted power (at least for the CC2420 radio). In Figure 7.14, we plot the throughput results for the NLOS channel.

It can be seen how the impact of the hidden node problem, which is not contemplated in our analytical model, is much higher in this second scenario. Once the data rate reaches $\lambda = 0.03$, the simulation results decrease dramatically compared to the analytical results, because of the described hidden node problem. Then we will really need to reduce the traffic load to mitigate the problem. For low data rates, the analytical and simulation results match up.

Since the analytical and simulation results differ so much, there is no point in continuing the analytical prediction for the rest of the performance metrics of the network. It is most important to

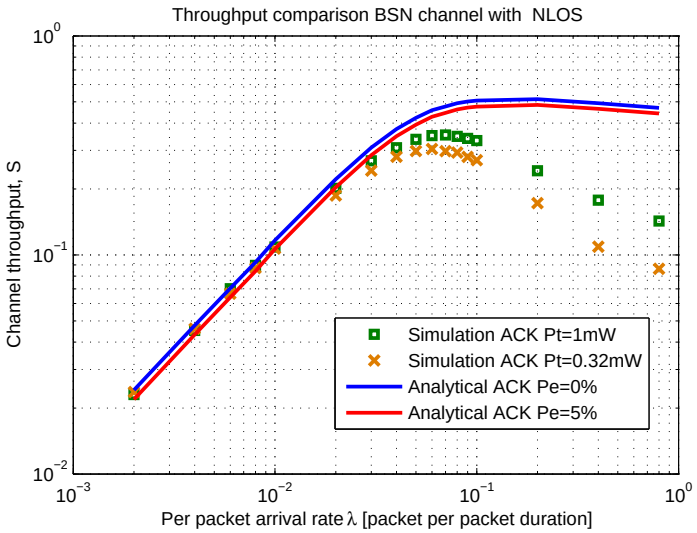

Figure 7.14: *Throughput comparison WBSN channel with NLOS*

understand that the hidden node problem has a great impact in this type of channels, and that our models does not capture its behavior.

CHAPTER 8

Conclusions and Future Work

8.1 Conclusions

This thesis is focused on determining the performance of the CAP protocol of the IEEE 802.15.4 standard in WBSN. A detailed study of the standard as well as the ns-2 simulator has been conducted. More importantly, we have extended the Ramachandran et al. [2] analytical model of the slotted CSMA/CA procedure in the CAP of the IEEE 802.15.4 standard to acknowledged traffic. Several assumptions have been made to simplify the analysis. However the validity of the analytical model has been demonstrated comparing with simulation results. Furthermore, for the purpose of conducting near realistic simulations, the Chipcon CC2420 IEEE 802.15.4 transceiver energy parameters have been used.

The results of the analytical model resolution have been then employed to predict throughput, energy consumption and latency. It is important to keep in mind that even though this results are correct, they are analytical upper bounds, and in real scenarios a lower performance should be expected. Based on this results we have also suggested some simple design guidelines in terms of desirable network traffic loads, related to the use of acknowledgments.

Finally, the channel model for propagation along the human body was studied. We have seen how the energy consumption in WBSN networks experiences a high path loss. This uncovers one of the main problems of using IEEE 802.15.4 in a human body environment, that is the hidden node problem.

The hidden node problem has a great impact in WBSN performance, both in terms of throughput and energy efficiency. We have found that there is no effective way to avoid this problem without increasing the transmitted power; one way to mitigate its effect however consists in reducing the aggregated traffic load.

8.2 Future Work

While performing ns-2 simulations, we have found an unsolved bug dealing with the fact that nodes overhear each other. This problem drives to errors when computing the average energy consumption in the network. We were unable to solve it due to the time constraint of this project. For future research, it would be desirable to take the time to debug the code and solve this implementation problem.

Our simulation results have indicated that the network performance is very much affected by the hidden terminal problem. This important factor is not taken into account in our analytical model. A possible future line of research would be trying to include in the model, the possibility of hidden nodes within the network.

The IEEE 802.15.4 GTS implementation is particularly effective for application that have timing constraints. This is really interesting for time critical monitoring WBSN applications, where data packets should not contend for the channel. However its performance is not investigated in this thesis. Parameters such as transfer reliability or energy consumption could be researched in the future.

Our results show that for high path loss NLOS human body propagation, star topologies do not lead to reasonable energy consumptions. The sensing node needs to be really close to the coordinator, otherwise the communication is unfeasible. A multi-hop topology strategy could be a future solution to this problem and should be investigated.

Even though we used some accurate human body propagation models, other sophisticated models available in the literature could be also implemented to perform different evaluations and contrast studies. In this project we have supposed and ideal disposition of the sensing nodes, but a more realistic WBSN configuration could also be made.

Bibliography

[1] BSN community web page at the Department of Computing, Imperial College London, UK. `http://www.bsn-web.org/`.

[2] I. Ramachandran, A.K. Das, and S. Roy. Analysis of the contention access period of IEEE 802.15.4 MAC. *ACM New York, NY, USA*, 2007.

[3] CS Raghavendra, K.M. Sivalingam, and T. Znati. *Wireless sensor networks.* Springer, 2004.

[4] H. Karl and A. Willig. *Protocols and architectures for wireless sensor networks.* Wiley, 2005.

[5] L.C. Zhong. *A unified data-link energy model for wireless sensor networks.* PhD thesis, University of California, Berkeley, 2004.

[6] K. Sohraby, D. Minoli, and T. Znati. *Wireless sensor networks: technology, protocols, and applications.* Wiley-Interscience, 2007.

[7] H. Li and J. Tan. An Ultra-low-power Medium Access Control Protocol for Body Sensor Network. *Conf Proc IEEE Eng Med Biol Soc*, 3:2451–4, 2005.

[8] G.Z. Yang. *Body sensor networks.* Springer, 2006.

[9] W. Ye and J. Heidemann. Medium Access Control in Wireless Sensor Networks. Technical report, USC, 2003.

[10] I. Lamprinos, A. Prentza, E. Sakka, and D. Koutsouris. Energy-efficient MAC Protocol for Patient Personal Area Networks. *Conf Proc IEEE Eng Med Biol Soc*, 4:3799, 2005.

[11] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking (ToN)*, 12(3):493–506, 2004.

[12] T. Van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180. ACM New York, NY, USA, 2003.

[13] P. Lin, C. Qiao, and X. Wang. Medium access control with a dynamic duty cycle for sensor networks. In *2004 IEEE Wireless Communications and Networking Conference. WCNC*, volume 3, 2004.

[14] J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *2nd ACM Conf. on Embedded Networked Sensor Systems. SenSys*, pages 95–107, 2004.

[15] M.I. Brownfield, K. Mehrjoo, A.S. Fayez, and N. Davis. Wireless sensor network energy-adaptive MAC protocol. In *IEEE Consumer Communications and Networking Conference*, volume 2, pages 778–782, 2006.

[16] IEEE 802.15.4-2006 standard, Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications.

[17] J. Zheng and M.J. Lee. A comprehensive performance study of IEEE 802.15. 4. *Sensor network operations*, pages 218–237, 2006.

[18] A. Koubaa, M. Alves, and E. Tovar. A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15. 4 wireless sensor networks. *IEEE WFCS*, pages 183–192, 2006.

[19] G. Bianchi. Performance analysis of the IEEE 802. 11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000.

[20] S. Pollin, M. Ergen, S.C. Ergen, B. Bougard, L. Van Der Perre, F. Catthoor, I. Moerman, A. Bahai, and P. Varaiya. Performance Analysis of Slotted Carrier Sense IEEE 802.15. 4 Medium Access Layer. In *IEEE Global Telecommunications Conference. GLOBECOM'06*, pages 1–6, 2006.

[21] RK Patro, M. Raina, V. Ganapathy, M. Shamaiah, and C. Thejaswi. Analysis and improvement of contention access protocol in IEEE 802.15. 4 star network. In *IEEE Mobile Adhoc and Sensor Systems, MASS'07.*, pages 1–8, 2007.

[22] CY Jung, HY Hwang, DK Sung, and GU Hwang. Enhanced Markov Chain Model and Throughput Analysis of the Slotted CSMA/CA for IEEE 802.15. 4 Under Unsaturated Traffic Conditions. *IEEE Transactions on Vehicular Technology*, 58(1):473–478, 2009.

[23] X. Ling, Y. Cheng, J.W. Mark, and X.S. Shen. A renewal theory based analytical model for the contention access period of IEEE 802.15. 4 MAC. *IEEE Transactions on Wireless Communications*, 7(6), 2008.

[24] E. Reusens, W. Joseph, G. Vermeeren, and L. Martens. On-Body Measurements and Characterization of Wireless Communication Channel for Arm and Torso of Human. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, page 264. Springer Verlag, 2007.

[25] A. Fort, J. Ryckaert, C. Desset, P. De Doncker, P. Wambacq, and L. Van Biesen. Ultra-wideband channel model for communication around the human body. *IEEE Journal on Selected Areas in Communications*, 24(4):927, 2006.

[26] USC Information Sciences Institute, Marina del Rey, CA. Network Simulator ns-2. `http://www.isi.edu/nsnam/ns`.

[27] B.B. Welch. *Practical programming in Tcl and Tk*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2000.

[28] USC Information Sciences Institute, Marina del Rey, CA. Network Animator Nam. `http://www.isi.edu/nsnam/nam`.

[29] D. Curren. A survey of simulation in sensor networks. *project report (CS580), University of Binghamton*, 2005.

[30] K. Fall and K. Varadhan. The ns Manual (formerly ns Notes and Documentation). *The VINT Project*, 1, 2002.

[31] D.B. Close, A.D. Robbins, P.H. Rubin, and R. Stallman. *The GAWK Manual*. Free Software Foundation, 1989.

[32] HT Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, 1946.

[33] T.S. Rappaport. *Wireless communications*. Prentice Hall PTR Upper Saddle River, NJ, 2002.

[34] J. Zheng and MJ Lee. Will IEEE 802.15. 4 make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard. *IEEE Communications Magazine*, 42(6):140–146, 2004.

[35] CC2420 Datasheet. 2.4 GHz IEEE 802.15.4 RF Transceiver. *ChipCon INC*, 2004.

[36] B. Bougard, D.C. Daly, A. Chandrakasan, and W. Dehaene. Energy Efficiency of the IEEE 802.15. 4 Standard in Dense Wireless Microsensor Networks: Modeling and Improvement Perspectives. In *Design, Automation and Test in Europe*, pages 196–201. IEEE Computer Society, 2005.