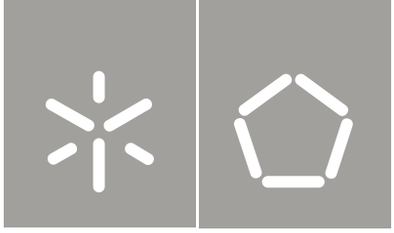
Universidade do Minho
Escola de Engenharia

Diogo Miguel Ferreira Taveira Gomes

Modeling and Experimental Performance Analysis of ZigBee/IEEE 802.15.4 for Wireless Body Area Networks

Outubro de 2012

Modeling and Experimental Performance Analysis of ZigBee/IEEE 802.15.4 for Wireless Body Area Networks
Diogo Miguel Ferreira Taveira Gomes
UMinho | 2012



Universidade do Minho
Escola de Engenharia

Diogo Miguel Ferreira Taveira Gomes

Modeling and Experimental Performance Analysis of ZigBee/IEEE 802.15.4 for Wireless Body Area Networks

Tese de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do
Professor Doutor José Augusto Afonso

Outubro de 2012

DECLARAÇÃO

Nome: Diogo Miguel Ferreira Taveira Gomes

Correio electrónico: diogomftgomes@gmail.com

TIm.: 917780207

Número do Bilhete de Identidade: 13253828

Título da dissertação: Modeling and Experimental Performance Analysis of ZigBee/IEEE 802.15.4 for Wireless Body Area Networks

Ano de conclusão:2012

Orientador:

José Augusto Afonso

Designação do Mestrado:

Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia de Comunicações

Escola: Engenharia

Departamento: Electrónica Industrial

Guimarães, ___/___/_____

Assinatura: _____

Acknowledgments

Firstly, I would like to express my deepest gratitude to my parents. Without their love and support, all of this would not possible.

I would like to thank my supervisor, Professor José Augusto Afonso, for the help, patience and guidance, which was essential for the accomplishment of this thesis.

I would also like to thank all my friends that, in one way or another, contributed for this work to be accomplished.

Abstract

The emerging field of wireless body area networks (WBAN) has the potential to play an important role in everyday life, and there are many industries such as health, sports and entertainment that can take advantage of these networks. The wireless monitoring of users' physical state, in indoor or outdoor environments, can bring benefits in several application scenarios; for example, it can increase patients' general well-being and reduce caregivers' workload by allowing continuous monitoring.

This dissertation identifies and analyzes key performance aspects of using the ZigBee and IEEE 802.15.4 protocols in WBAN applications. The main reason behind this work is because these protocols were designed primarily for wireless sensor networks (WSNs) but are also being used in WBAN applications, particularly in the healthcare area. The differences between WSN and WBANs are explained and are used to discuss the usage of the ZigBee and the IEEE 802.15.4 standards in WBANs.

The analysis performed in this work consists mainly in the execution of experimental tests with non-beacon enabled ZigBee/IEEE 802.15.4 networks, using widespread hardware and software platforms from Texas Instruments, regarding relevant quality of service (QoS) metrics (maximum throughput, delivery ratio and network delay), as well as the effects of multiple constraints, such as hidden nodes, clock drift and body interference in the network performance.

A clock drift model was proposed to estimate when two nodes will interfere with each other. This model was conceived due to the lack of support from the ZigBee to overcome this issue. A solution to overcome the clock drift and the hidden node problems was then designed. A parametric software delay model of ZigBee network devices was also defined and introduced into a simulator so that more accurate simulation results could be obtained. The proposed models were deemed valid since they were thoroughly tested and the predicted results were obtained.

Resumo

As redes de sensores sem fios de área corporal (WBAN) têm o potencial de desempenhar um papel importante no dia-a-dia. Hoje em dia há muitas indústrias, tais como na área da saúde, do desporto e do entretenimento, que podem tirar proveito dessas redes. A monitorização sem fios de sinais fisiológicos, tanto em ambientes fechados como ao ar livre, pode trazer benefícios em vários cenários de aplicação, tais como, aumentar o bem-estar de pacientes que são monitorizados e reduzir a carga de trabalho de médicos, permitindo a monitorização contínua.

Esta dissertação identifica e analisa aspetos chave do desempenho das redes ZigBee e IEEE 802.15.4, quando usadas em aplicações típicas das WBAN. A principal motivação para a realização deste trabalho reside no facto de que, apesar de terem sido projetados principalmente para redes de sensores sem fio (WSN), estes protocolos estão também a ser utilizados em aplicações características das WBAN, particularmente na área da saúde. As diferenças entre as WSN e as WBAN são destacadas e usadas para discutir o uso dos protocolos ZigBee e IEEE 802.15.4 nas WBAN.

A análise realizada neste trabalho consiste, principalmente, na execução de testes experimentais de redes ZigBee/IEEE 802.15.4 a funcionar no modo *non-beacon enabled*, usando as plataformas de *hardware* e *software* da Texas Instruments. A análise leva em consideração métricas relevantes (o máximo *goodput*, a taxa de entrega e o atraso da rede) de qualidade de serviço (QoS) e os efeitos de várias condicionantes, como os nós escondidos, o *clock drift* e a interferência do corpo humano no desempenho da rede.

Um modelo para o *clock drift* foi proposto para estimar quando dois dispositivos irão interferir um com o outro devido a este fenómeno. Este modelo foi concebido devido à falta de capacidade para o ZigBee superar este problema. Posteriormente foi concebida uma solução para ultrapassar os problemas associados ao *clock drift* e aos nós escondidos. Um modelo paramétrico de atrasos de *software* em dispositivos de redes ZigBee foi também definido e introduzido num simulador, de modo a que resultados de simulações mais precisos possam ser obtidos. Os modelos propostos foram considerados válidos dado que foram testados e os resultados previstos foram obtidos.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

ACK	Acknowledgment
AODV	Ad-hoc On-Demand Distance Vector
API	Application Programming Interface
APL	Application Layer
APS	Application Support Sublayer
APSDE	APS Data Entity
APSME	APS Management Entity
BCU	Body Control Unit
BE	Backoff Exponent
BI	Beacon Interval
BO	Beacon Order
BS	Base Station
BSN	Body Sensor Network
CAP	Contention Access Period
CCA	Clear Channel Assessment
CDMA	Code Division Multiple Access
CFP	Contention Free Period
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CSP	Command Strobe Processor
CW	Contention Window
DR	Delivery Ratio
ECG	Electrocardiography, Electrocardiogram
EEG	Electroencephalography, Electroencephalogram
EMG	Electromyography, Electromyogram
FDMA	Frequency Division Multiple Access
FFD	Full Function Device
FIFO	First In First Out
GT	Guard Time
GTS	Guaranteed Time Slot

HAL	Hardware Abstraction Layer
HNP	Hidden Node Problem
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial, Scientific, and Medical
ITU	International Telecommunications Union
ITU-R	ITU - Radiocommunication Sector
LQI	Link Quality Indicatior
MAC	Medium Access Control
MCPS	MAC Common Part Sublayer
MLME	MAC Layer Management Entity
NPDU	NWK PDU
MT	Monitor and Test
NB	Number of Backoffs
NWK	Network
OMNeT	Objective Modular Network Testbed
OSAL	Operating System Abstraction Layer
OSI	Open Systems Interconnection
PAN	Personal Area Network
PC	Personal Computer
PD	Personal Device
PDA	Personal Digital Assistant
PDU	Packet Data Unit
PER	Packet Error Ratio
PHY	Physical
PIB	PAN Information Base
PLME	PHY Layer Management Layer
PPDU	PHY Packet Data Unit
PSDU	PHY Service Data Unit
QoS	Quality of Service
RF	Radio Frequency
RFD	Reduced Function Device
RSSI	Received Signal Strength Indicator
RX	Receive

SAP	Service Access Point
SD	Superframe Duration
SO	Superframe Order
SoC	System on Chip
TDMA	Time Division Multiple Access
TX	Transmit
UART	Universal Asynchronous Receiver/Transmitter
UBP	Unit Backoff Period
WAN	Wide Area Network
WBAN	Wireless Body Area Network
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
ZC	ZigBee Coordinator
ZCL	ZigBee Cluster Library
ZDO	ZigBee Device Object
ZDP	ZigBee Device Profile
ZED	ZigBee End Device
ZR	ZigBee Route

Chapter 1

Introduction

1.1 Context

Recent advances in the development of wireless communication and sensors for monitoring physiological signals are instigating the research in the field of Wireless Body Area Networks (WBAN), also commonly known as Body Sensor Networks (BSN). A BSN consists of a group of sensor devices distributed over the human body using a wireless network to support communications. New sensors have been developed to monitor many kinds of physiological parameters with great value for healthcare, performance evaluations of sport athletes or even in the entertainment business. In healthcare monitoring systems, BSNs can be used to collect and send signals obtained from the electroencephalogram (EEG), electrocardiogram (ECG), electromyogram (EMG), oximetry and other physiological parameters such as temperature or blood pressure. BSN-based monitoring can provide benefits in the diagnosis and treatment of patients without constraining their normal activities. It allows the patient to move freely inside or outside the hospital environment while providing continuous monitoring, which can be very useful when an extended period of monitoring is required. For example, many cardiac diseases are associated with episodic abnormalities such as transient surges in blood pressure or arrhythmias [Lo05]. These transient abnormalities cannot always be detected using conventional monitoring equipment. BSNs have the potential to provide early detection and prevention of pathologies, replacing expensive therapies later on. BSNs may be used in the sports sector to monitor the respiration rate or the athlete's movements to optimize their performances. For example, swimming athletes synchronize

their movements and respiration rate, which can be improved by analysing the data to correct imprecisions. BSNs are also being extensively used in the entertainment industry where users interact with video games using their movements, which are acquired through kinetic sensors.

Every WBAN application usually has specific requirements and, due to this heterogeneity, a standard specification for the WBANs has not yet been published because to derive an all-in-one solution is very complex. The IEEE 802, an organization for the standardization of communication network protocols that proposed worldwide successful specifications such as the IEEE 802.11 standards, established the Task Group IEEE 802.15.6, or IEEE 802.15 TG6, for the standardization of WBANs. The main objective of the IEEE 802.15.6 is to define new Physical (PHY) and Medium Access Control (MAC) layers for WBAN. This standard aims to include a network solution for both medical/healthcare and other non-medical applications with different requirements by supporting short range, low-cost, ultra-low power, high reliability and the coexistence of several applications into the same BSN for wireless communications in and around the body [IEEE6-08].

The MAC is the core protocol of any shared medium communication network. Thus, a suitable MAC layer is fundamental to fulfill WBAN requirements. In the Open Systems Interconnection (OSI) model, the MAC belongs to the first layer above the PHY layer and is used to coordinate the access of the nodes to the network communication medium. Its fundamental task is to avoid collisions, which have negative impact to the network performance. For WBAN systems, it has also the important task of providing Quality of Service (QoS) support to the applications, by controlling metrics like throughput efficiency, latency, communication reliability and energy efficiency. The MAC is one of the key layers regarding energy consumption in a WBAN because, since it acts upon the PHY layer, it can set the state of the nodes radio transceiver, which usually is the component that has the highest energy consumption rate in a low power sensor device. To reduce the energy consumption, the transceiver's state may be switched to sleeping mode, reducing the amount of energy wasted during idle periods. Typically, wireless MAC protocols are divided in two groups: contention-based or random access; and contention-free or scheduled access protocols. In contention-based MAC protocols such as Carrier Sense Multiple Access-Collision Avoidance (CSMA-CA), the nodes perform the Clear Channel Assessment (CCA) function to sense the channel before transmitting the data, in order to prevent collisions. Contention-free MAC protocols usually use techniques such as Time Division Multiple Access (TDMA) where packets may be transmitted into a time slot allocated to a particular

sensor node. Other medium access techniques like Code Division Multiple Access (CDMA) or Frequency Division Multiple Access (FDMA) are not suitable in the context of the wireless sensor networks (WSN) due to limitations in frequency spectrum availability and computation capability [Gopalan10].

The IEEE 802.15.4 standard [IEEE4-03], particularly combined with the ZigBee protocol stack, is a widely adopted protocol in WSN applications, and is being used as an alternative for health care applications [Li09][López11]. The standard defines the PHY and MAC layers for low data rate, low power and low complexity short range radio frequency (RF) transmissions in a Wireless Personal Area Network (WPAN). The IEEE 802.15.4 was originally designed for WSN, in which, usually, most of the supported applications generate low traffic loads to the network. Typically, WSN applications generate traffic only when triggered by external events, e.g., an out of range event detected through sensors (e.g. temperature or humidity). On the other hand, some BSN applications are data-intensive, generating a considerable amount of traffic due to high sampling rate requirements from some sensors.

1.2 Motivations and Objectives

The IEEE 802.15.6 WBAN standard is still in a development phase, where it is receiving several contributions from different manufactures in order to define the new specification. Meanwhile, the ZigBee/IEEE 802.15.4 protocols already present several products in the market from multiple manufacturers and are currently being used in WBAN applications. Since the ZigBee/IEEE 802.15.4 standards were not originally developed taking into consideration the specificities of WBAN applications, further analysis and revisions of these protocols are necessary, making this the main motivation for the development of the dissertation.

The main objectives in this work are:

- To analyze the performance of non-beacon enabled ZigBee/IEEE 802.15.4 networks in the context of WBAN applications, through experimental and simulation evaluations of relevant QoS metrics (maximum throughput, network delivery ratio (DR) and data delay), and the effects of multiple constraints, namely, hidden-nodes, clock drift effects and body interference;

- To propose a solution to mitigate the hidden-node problem and clock drift effect in data-intensive WBANs with periodic traffic;
- To measure the software processing delay introduced by ZigBee/IEEE 802.15.4 devices and to define and integrate a parametric model that takes into account this delay into a simulator.

The experimental platform used to produce the results presented in this work was developed and tested using the ZigBee 2007 [ZigBee07] and IEEE 802.15.4-2006 implementations provided by Texas Instruments: the Z-Stack and the TIMAC, respectively. The hardware test platform is based on the CC2530 [TICC2530-10] System on Chip (SoC) integrated circuit (IC), which is also provided by Texas Instruments. This SoC includes a microcontroller and a transceiver compatible with the IEEE 802.15.4 standard, thus enabling the development of smaller sensor devices. The platform used in the simulations was the OMNeT++, which provides a simulation development environment based on discrete time events. It was used a software simulation model of the unslotted CSMA-CA of the IEEE 802.15.4 protocol implemented by Pedro Macedo in his master´s degree thesis [Macedo10].

1.3 Contributions

The main contributions of this work are:

- Experimental evaluation of the performance of ZigBee/IEEE 802.15.4 networks in the context of WBAN applications. Results for the maximum throughput in a ZigBee sensor device; and the DR and delay for networks composed by up to 5 sensor devices transmitting to the coordinator in star and 2-hop tree topologies; are provided.
- Experimental evaluation based on the Packet Error Ratio (PER) and Received Signal Strength Indicator (RSSI) using a fully wireless WBAN system, regarding the interference of the human body in the radio communications;
- The definition of a model to predict the effect of the clock drift in the performance of data-intensive WBANs with periodic traffic;
- The proposal and implementation of an application level algorithm to solve the hidden-node problem (HNP): the HNP Avoidance protocol;

- The definition of a parametric model to characterize the ZigBee/IEEE 802.15.4 software processing delay and its integration into a simulator in order to obtain more accurate simulation results.

1.4 Thesis Organization

This thesis is divided into five chapters, which are described as follows:

Chapter 2 provides an overview of WBANs regarding the communication architectures and technologies that were used, as well as a description of WSNs based on the ZigBee/IEEE 802.15.4 protocol, which includes these two protocols and other protocols of particular interest. This chapter also describes a kinetic monitoring system whose traffic parameters are used on the performance evaluations presented in this work.

Chapter 3 describes the configurations adopted in the experimental tests that were executed to evaluate the performance of ZigBee networks when supporting data-intensive BSN applications. An introduction to the hardware that was used and a brief explanation of the programming environment are given. A number of QoS metrics are considered, and a connection from a theoretical standpoint to a more practical analysis is established. A model for software delay is proposed, and the simulator where it was implemented is described. The clock drift effect and a method to measure it are explained. The hidden node problem is discussed alongside with a protocol developed to solve this issue. Finally, the evaluation setup to a set of experiments regarding the body interference in the radio communications is provided.

Chapter 4 presents the results obtained from the experimental component of this work, using the experimental evaluation scenarios and the proposed models detailed in the previous chapter. A series of graphs and tables are used to demonstrate the results from these experiments, which are commented and discussed

Finally, chapter 5 presents the conclusions and indicates possible lines of future work for this research topic.

Chapter 2

Wireless Monitoring Overview

This chapter provides some useful background information related to the topic presented in this work. An overview of wireless communications is given, covering body sensor networks, wireless sensor networks, and some protocols of particular interest, namely the IEEE 802.15.4 and the ZigBee protocols. This chapter also presents a body sensor network applied to motion capture, the posture monitoring system (PMS), whose traffic parameters were used to acquire the results presented in this work.

2.1 Wireless Communications

Wireless communications started in the late 19th century when the wireless telegraph was created. Since then, wireless communications have evolved drastically; however, the foundation for most communication systems is still present, where radio waves are used for the transmission of information. The radio spectrum and wireless systems standardization are managed worldwide by the International Telecommunications Union (ITU) Radiocommunication Sector (ITU-R). Additionally, different national and regional agencies may be responsible for further regulations. Radio spectrum has several licensed frequency bands allocated to different communication technologies, e.g., radionavigation or terrestrial mobile communications. A group of license free bands were assigned to industrial, scientific and medical (ISM) applications [Akyildiz02]. These bands, known as ISM bands, are listed in Table 2.1. The main advantage of using the ISM bands is that they are license free, unlike the other bands that are allocated to particular paid communication services. On the other hand,

many wireless technologies use the ISM bands, e.g., systems based on the IEEE 802.11 and IEEE 802.15.4 standards, Bluetooth and other private or research technologies. This may have a negative impact because, if several network technologies share the same physical medium and frequency band, the level of interference may increase significantly, which may cause the degradation in the performance of these networks. The IEEE 802.15.4 standard was the foundation for the development of this work and the 2.4 GHz frequency band is used, but unfortunately is also used by IEEE 802.11-based Wireless Local Area Networks (WLANs), making these networks susceptible to interference.

Table 2.1 - ISM bands.

Frequency range	Centre frequency
6.765 MHz - 6.795 MHz	6.780 MHz
13.553 MHz - 13.567 MHz	13.560 MHz
26.957 MHz - 27.283 MHz	27.120 MHz
40.660 MHz - 40.700 MHz	40.680 MHz
433.050 MHz - 434.790 MHz	433.920 MHz
902.000 MHz - 928.000 MHz	915.000 MHz
2.400 GHz - 2.500 GHz	2.450 GHz
5.725 GHz - 5.875 GHz	5.800 GHz
24.000 GHz - 24.250 GHz	24.125 GHz
61.000 GHz - 61.500 GHz	61.250 GHz
122.000 GHz - 123.000 GHz	122.500 GHz
244.000 GHz - 246.000 GHz	245.000 GHz

In order to communicate, wireless devices must agree on a communication protocol, which allows the exchange of messages by defining their meaning and structure. A communication protocol is usually very complex to implement, hence, it is organized in layers where each layer is designed to accomplish different functions. Through the service access points (SAPs), each layer uses services provided by lower layers and offers a set of services to the layers above it. Figure 2.1 represents the stack structure of the protocol used in this work. Vertical arrows symbolize the communication between layers on the same device, while each arrow connecting the same layer of different devices represent that layer's logical communication.

The physical layer (PHY) is responsible for the management of the radio hardware, e.g., to transmit and receive data and the signal modulation. The medium access control (MAC) layer manages the access of the network devices to the medium. The network (NWK) layer is introduced for routing frames through the network and, among other tasks, to create and maintain the network or to discover new routes. The application (APP) layer provides, for

instance, support for a multitude of applications in the device [ZigBee07].

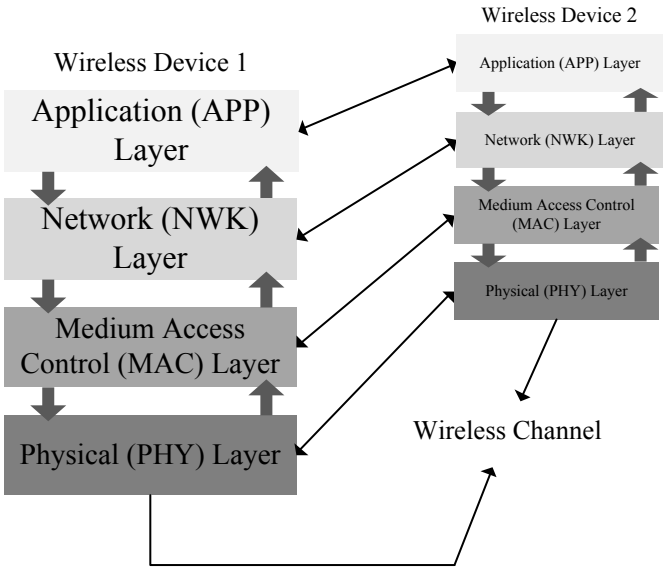


Figure 2.1 - Stack model of a wireless device.

Wireless communications are highly susceptible to interference due to the channel propagation characteristics, caused by phenomena such as large-scale fading and small-scale fading, and co-channel interference, which is caused by undesired transmissions on the same frequency channel by neighbor networks [Trigui09]. Large-scale fading is a consequence of the obstacles that affect the propagation of the radio waves, e.g., the walls in the interior of a building that may be separating network devices. Small-scale fading is caused by surfaces that reflect the radio waves. When these reflections occur, multiple copies of the transmitted signal may reach the receiver, which may result in a constructive or destructive interference because these copies of the signal experience different attenuation, delay and phase shift which, consequently, may cause unpredictable results. Other different sources of noise, for instance, radiofrequency interference generated by electric power transmission lines, may also introduce interference to the communications system [Mattos96].

The quality of service (QoS) is an important aspect in the context of the wireless communications. Several applications have different QoS requirements that must be assured by the wireless network. For instance, medical monitoring applications usually have low bandwidth requirements but are usually intolerant to high delays and data loss. On the other hand, file transfer applications can tolerate relatively high packet delays but are intolerant to data loss [Soomro06]. Different metrics are used to evaluate the QoS provided by the network, which is the case of the network delivery ratio (DR), the delay or the jitter. The

delivery ratio represents the percentage of successfully delivered packets in relation to the number of generated packets. The delay, or latency, represents the time period between the instants of generation and delivery of each packet. The jitter represents the variation of the delay and may me caused, for example, by an alternative route taken by a packet to reach its destination. Since the QoS must be guaranteed by the protocols used to support the network, every layer that constitutes the stack may have influence in the network response to the application requirements. For instance, in the lower layers, the modulation and codification techniques used may enhance the overall robustness against interferences of the transmitted signals, optimizing the QoS experienced in the applications. In the upper layers, MAC protocols, retransmissions and other error correction mechanisms may also influence the QoS provided to the applications.

Figure 2.2 shows the architecture of a wireless sensor device. The typical main components of wireless sensor devices are as follows: the memory, the microcontroller, sensors and/or actuators, the radio transceiver (and the respective antenna), the energy source (usually battery powered) and the interfaces between the components. The battery is one of the most important components and it has to be carefully chosen because it can affect the design and longevity of the device, in which the maximization of the latter is commonly desired [Vieira07]. Figure 2.2 illustrates a general architecture for wireless sensor devices because, usually, vendors may choose to implement their own architectures. At the implementation level, in system-in-package (SiP) wireless devices, the main components are available separately. On the contrary, in system-on-chip (SoC) devices memory, radio transceiver and the microcontroller components are integrated onto the same chip. SoC architectures tends to be the mostly accepted among vendors, aiming for the miniaturization of their wireless sensor devices [TICC2530-10] [Jennic10].

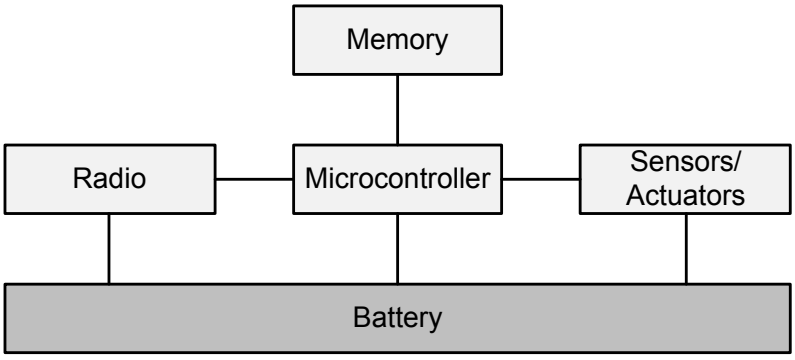


Figure 2.2 – Wireless sensor device typical architecture (main components)

In general, wireless sensor devices use crystal oscillators to derive time. Each device has its own oscillator, which means that different devices have different clock frequencies. The deviation of the real clock frequency with relation to the nominal clock rate is referred as clock drift. Clock drift is often expressed in parts per million (ppm), which means that after a million nominal frequency oscillations the real clock would have *n* ppm additional or missing oscillations. Air pressure, temperature or the electric supply voltage may cause short-term variations in the oscillator frequency, and the equipment's aging may cause long-term variations [Brzozowski09].

The hidden node and exposed node are inherent problems to carrier CSMA-based wireless network protocols. In carrier sense-based mechanism, before a device transmits a packet, it must always sense the wireless channel in order to avoid collisions with transmission from other devices in the network. This procedure is executed by the CSMA-CA mechanism of the IEEE 802.15.4 standard, one of the protocols under evaluation in this work. Figure 2.3 (a) illustrates a hidden node scenario in an IEEE 802.15.4 network between nodes A and C, where a circumference around the transmitting nodes represents their signal range *r*. These nodes are unable to sense each other's transmissions if they are separated by a distance $d > r$. Consider that a node B, in the range of both A and C, is receiving the transmission of A. If C wants to transmit a packet using a CSMA-CA protocol, the carrier sense procedure fails, since A is hidden from C. Therefore, C will start its transmission, causing a collision with the packet that is being transmitted by A at the receiver (B), which makes both packets to be lost. We refer to this situation as hidden-node problem (HNP), due to the degradation that may be introduced in the network performance when packets keep colliding due to the failure of the carrier sense mechanism. The exposed node problem is shown in Figure 2.3 (b) where the CSMA-CA algorithm in the device E, which wants to transmit data to D, reports a busy channel because the device F is transmitting to G. In this case, G cannot hear E and D cannot hear F, hence, the transmissions will never collide, and nevertheless, the CSMA-CA algorithm will block the transmission of device E.

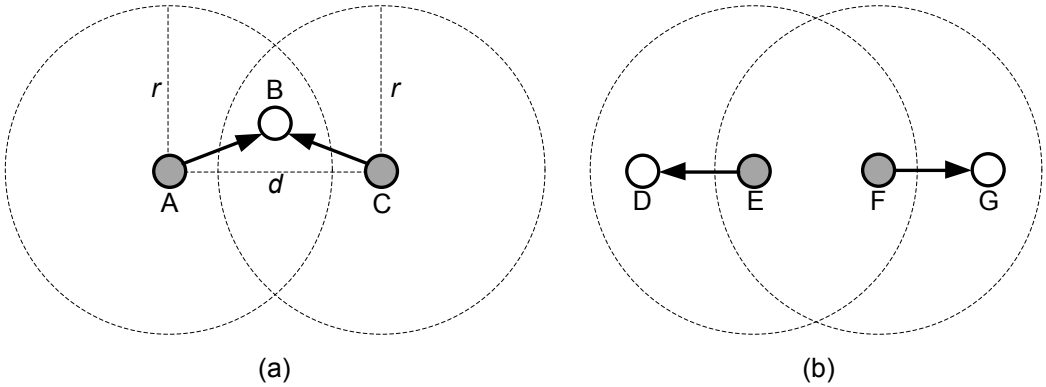


Figure 2.3 - Hidden-node (a) and exposed-node (b) scenarios.

The hidden node and the exposed node situation illustrated in Figure 2.3 are based on the limitation in the nodes radio signal range, which is caused by the free space path loss. However, there are other factors related to the spatial configuration and the propagation effects on the place where the nodes are located, such as fading or shadowing, which can also cause these situations.

2.2 Wireless Body Area Networks

2.2.1 Definition and Applications

A wireless body area network (WBAN) is a set of one or more body sensor networks (BSNs) used to monitor several parameters on, in or around the human body. WBAN applications include healthcare systems, athletic training, workspace safety, consumer electronics, secured authentication systems and safeguarding of uniformed personnel. A BSN consists in a group of sensors distributed over the human body, which are used to monitor several physiological signals and actions, with a wireless network to support communication. These monitored parameters may be stored in a personal device, e.g., a personal digital assistant (PDA) or smartphone, which collects the data from the sensor nodes and then transmit it to a personal computer (PC) or a datacenter for storage and for further analysis. The BSN topic is discussed in more detail in section 2.2.2.

In comparison to wired systems, WBAN healthcare monitoring systems aims to improve the way patients are cared for, providing a better quality of life and care for the patients. WBAN healthcare monitoring systems allows patients to move freely inside or outside the hospital environment without limiting their normal day-to-day activities while providing

continuous monitoring. This can be extremely useful as preventive care when a long period of monitoring is required for the detection of a particular disease as preventive care.

WBANs architectures may be classified into two categories: flat and multi-tier [Chin12]. In flat architectures, a BSN is composed of a single data-gathering unit that transmits the information to a PC or a personal server application running on a PDA. In multi-tier architectures, the BSN define the first tier (Tier-1-Comm). At a second tier (Tier-2-Comm), WBANs can be connected to Local Area Networks (LANs) and Wide Area Networks (WANs) through various wired and wireless communication technologies. At the last tier (Tier-3-Comm), the BSN may be accessed through computing devices, such as a PC or a PDA, by healthcare workers or the patient. Figure 2.4 shows the position of the WBANs in the realm of the wired/wireless communication networks and the multi-tiered BSN architecture.

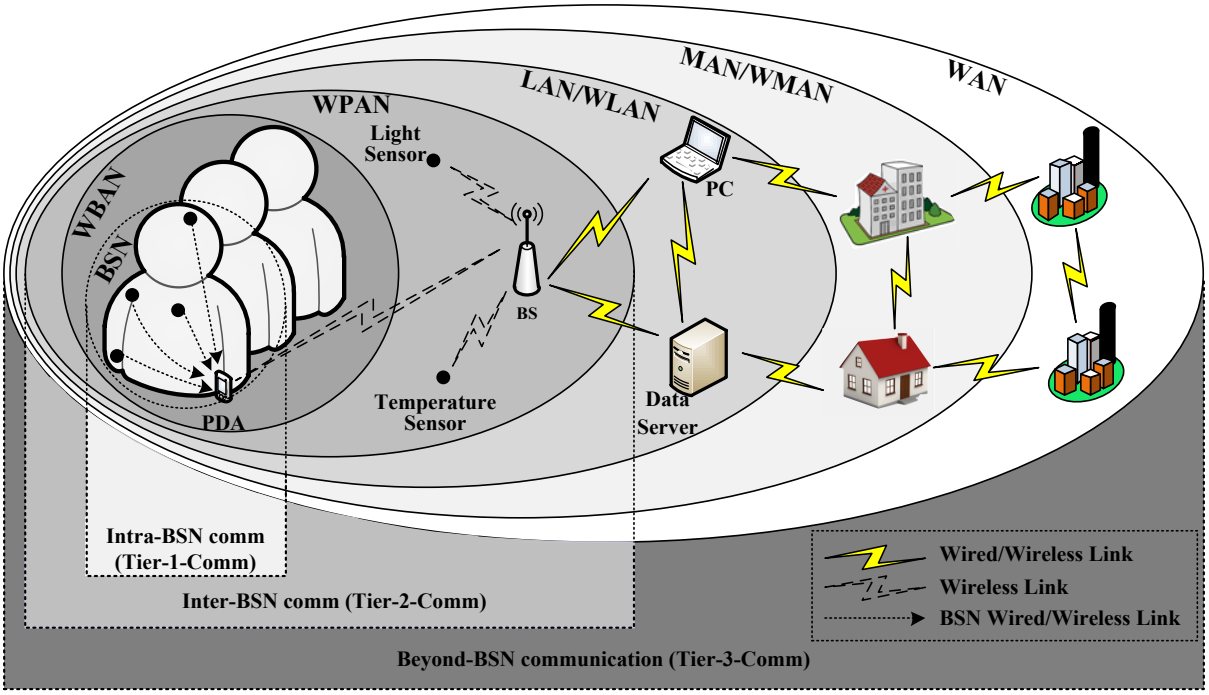


Figure 2.4 - Multi-tiered BSN architecture (adapted from [Ramli11]).

WBANs emerged from the existent wireless personal area networks (WPAN) technologies, which are used for short-range communications between wireless devices (≤ 10 m) [Ramli11]. Devices from a WPAN may vary in their capabilities, but typically have low processing and storage capabilities, and can be battery or mains-powered. WPANs may be used to obtain more information about a patient's living space through the measurement of several environment properties, such as luminosity, humidity, temperature or movement. This

may provide more specific information about the spatial context in which a patient is being monitored. The data in a WPAN are usually transmitted to a base station (BS), which may also be used to collect the data from a WBAN.

A WBAN may vary on the communication system architecture because it usually depends on particular contexts, more specifically, the location or the environment where the patients are being monitored. The data generated by the BSNs may be collected by a BS, which may interface with different communication infrastructures, such as private (e.g., Wi-Fi networks), public (e.g., the Internet or a mobile communication network) or ad-hoc networks. The following points describe three possible WBAN system architectures using different network communication infrastructures:

- A patient may be monitored in a hospital and a BS may collect the data generated by the BSN. The BS connects the BSN to the LAN implemented in the hospital, which connects devices such as PCs, PDAs or data servers. A PDA may also gather the BSN data and then transmit it directly to a PC or a data sever through the LAN. Finally, patients and physicians may access the information via PCs or PDAs in the LAN.
- A patient may be monitored at home or in an ambulance and the BSN uses a PDA or a BS to communicate with remote servers or physicians using a public network infrastructure such as the Internet, cellular communication networks or satellite communications.
- An ad-hoc WBAN may be created in the case of a major catastrophic event where a set of BSNs may be created to monitor injured people in an outdoor incident area. This ad-hoc network may use short-range devices that transmit and relay data among patient devices until reaching a caregiver's device.

2.2.2 Body Sensor Network

As we previously defined, a BSN is a group of sensor devices distributed in, on or around the human body that are used to monitor several physiological parameters and are capable of establishing a wireless communication network. Figure 2.5 shows an example of a BSN composed by several heterogeneous sensor devices strategically positioned in the human body transmitting the monitored information to a PDA using a wireless link. BSNs have gained much interest and have become an emerging technology in healthcare services. These services are used to monitor patients' vital signs while taking advantage of wireless

monitoring, which can provide some benefits in the diagnosis and treatment of patients without constraining their normal activities. Continuous monitoring is possible and can be of vital importance due to the ability to monitor for extended periods of time, making it possible to detect health abnormalities within a bigger time frame compared to periodic monitoring.

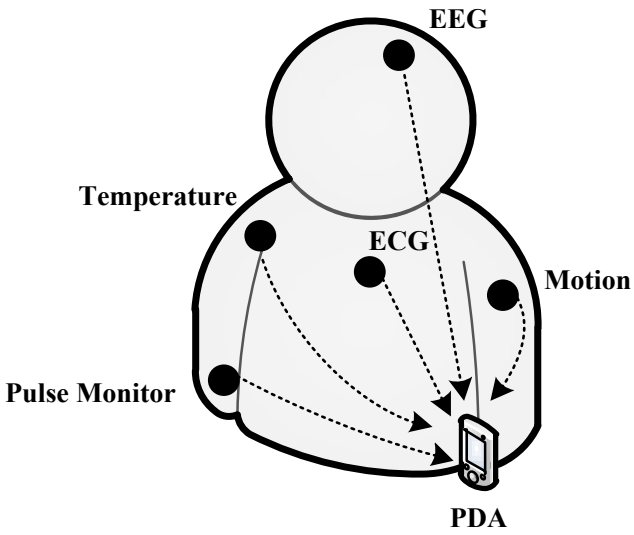


Figure 2.5 - Example of a BSN.

A set of possibilities for medical diagnosis and treatment applications, sensors and the role of the BSNs are described in Table 2.2.

Table 2.2 - BSN healthcare applications [Baraka12].

Field of Application	Sensors	Role of the BSN
Cardiovascular diseases	Pulse oximeter, heart rate sensor, and ECG sensor.	An internist can prepare treatment in advance as he receives the patient's monitored information related to the heart behavior.
Cancer	Nitric Oxide sensor.	A sensor can be placed in a cancer suspected area and an internist may start a proper treatment as soon the cancer is detected.
Diabetes	Biosensor gyroscope, insulin actuator.	If the sensor monitors a sudden drop of glucose, a signal can be sent to the insulin actuator in order to start the injection of insulin.
Post-operation care	Temperature sensor, blood pressure sensor, heart rate sensor, ECG.	A patient may be continuously monitored without restraining them to the bed, which improves the patient's quality of life.

Every BSN application usually has specific requirements and, due to this heterogeneity, a standard specification for the WBANs has not yet been published because deriving an all-in-

one solution is very complex. The task of developing a standard protocol for energy-efficient devices and WBAN applications is assigned to the Task Group IEEE 802.15.6. The IEEE 802.15.6 standard is discussed in section 2.2.2.4.

2.2.2.1 BSN Characteristics

As we previously mentioned, WBANs and BSNs emerged from WPANs, which are related to wireless sensor networks (WSNs). In spite of this relation, there are some differences between BSNs and WSNs, which are related to the applications requirements and characteristics. Protocols and algorithms designed for conventional WSNs may not be suitable for the BSNs due to these differences. Next, we describe some BSN requirements and characteristics:

- As seen in the last section, BSNs, unlike WSNs, are typically used in multi-tier systems.
- Sensor devices in WSNs usually have homogeneous requirements in terms of the data rate, power consumption and reliability of the network. On the other hand, sensor devices requirements in the realm of the BSNs often have heterogeneous network requirements.
- Sensor devices in BSNs are placed in strategic locations in the human body in order to correctly monitor a desired parameter and, consequently, provide an efficient way to capture data. Unlike BSNs, sensor devices of the WSNs may be randomly spread in an area to be monitored.
- The transmission power is an essential parameter in BSNs due to aspects related to the proximity of the sensor devices in the human body. These aspects include user health concerns, the interference between BSNs caused by the high transmission power and the human body as a propagation medium with great losses, which considerably attenuate the transmitted radio waves until they reach the receiver. In WSNs, there is a concern for reducing the transmission power for providing an extended lifetime for these sensor devices, thus saving one of the most precious resources in these networks: energy.
- Depending on the applications requirements, BSNs may be highly sensitive to the network latency because some latency-critical BSNs may generate alarm events that need an emergency response from the healthcare provider. On the other hand, conventional WSN applications do not usually have concerns regarding critical-

latency requirements and, because of this, mechanisms to save energy are used, where sensor devices may enter sleep mode and transmit data only when awake, even if a transmission event were set during the sleep mode.

- In order to monitor users' vital signs, BSN applications usually generate periodic traffic to the network due to the monitored signals characteristics, which are discussed in section 0. On the other hand, conventional WSN applications usually generate traffic when triggered by a particular event. This distinct behavior accentuates the difference between the data rates required by these two types of networks.
- The energy source is an important aspect both in BSNs and WSNs, where the sensor devices should operate for months or even years. Unlike WSNs, the battery replacement in the sensor devices may be more difficult in BSNs and a greater issue when they are implanted inside the human body.
- A BSN uses a reduced number of sensor devices when compared with the WSNs, due to the smaller coverage area. Conventional WSNs are used to monitor huge areas while BSNs only cover the area around the human body.
- BSNs sensor devices may move freely according the user movements, so the relative position between devices may change frequently. Thus, BSNs should be robust against the changes in the physical topology. On the other hand, in WSNs, sensor devices are usually static.
- BSN critical health applications are usually intolerant to data loss caused by a network failure, so the network reliability is of great importance, unlike in the conventional WSNs where a failure on a sensor device may be compensated by another sensor device.
- Security is of utmost importance in the BSNs. BSNs information must be protected from unauthorized users while being transferred and stored. Security requirements must comply with several needs such as confidentiality, integrity, availability and access control.

The previous discussed characteristics show that BSNs have unique requirements compared to the conventional WSNs, which are also discussed in section 2.3.

2.2.2.2 BSN Devices

In [Barakah12], three types of devices for the BSNs are considered: the sensor device, the actuator device and the personal device (PD).

The sensor device is a device that gathers the physiological data, processes it, if required, and reports it through a (wireless) communication system. Every BSN has at least one sensor device. The (wireless) sensor device is constituted by the sensor hardware, a power unit, a processor, memory and a transceiver, and typically has very limited computational and energy resources.

The actuator is a device carried by a patient and acts according to the information collected by the deployed sensor devices, where a proper treatment may be administrated to the patients immediately after the sensor devices detect some problem or when triggered by a doctor that has analyzed the collected data. An actuator device node can consist of a receiver or transceiver, a power unit, a processor and memory, where main component is the actuator hardware. For example, when a patient is being monitored for diabetes and a sensor detects a sudden drop of glucose, a signal can be sent to the actuator device in order to start an injection of insulin.

The PD is also known as body control unit (BCU), body gateway or a sink. The PD can be a dedicated unit or, in some implementations, a PDA or a smartphone. The PD's main function is to collect all the BSN information and relay it to the user or to an internist via an external gateway. The core components of this device are a power unit, a large processor, a large memory and a transceiver; hence computing and energy resources are considerably higher than in sensor and actuator nodes.

Another type of device, the base station (BS), is also considered in other systems [Shnayder05][Silva11]. This is a stationary device that collects the data packets directly from the sensor devices and transmits the information to a PC through a wired connection. Then, the PC may show the gathered information or relay it through an external gateway.

Energy Scavenging/Harvesting

Energy scavenging, or energy harvesting, refers to methods for sensor and actuator devices to obtain energy from their surrounding environment, since these devices have limited battery power. If a device can obtain energy from the environment where it is placed, it can become more autonomous or at least reduce human intervention. There are many different sources that can be used to obtain energy, for example: the sun, the wind, thermal sources and vibration. Regarding BSNs, the most convenient energy sources may be those provided by the human body, such as body heat and body vibration. However, current scavenging technics are only able to extract small amounts of energy from a source, where the scavenged energy is

proportional to the size of the scavenging device. Table 2.3 summarizes the power that could be harvested from different environmental sources.

Table 2.3 - Characteristics of various energy sources available in the environment and the harvested power [Fiorini08].

Energy Source	Source Characteristics	Harvested Power
Ambient Light Indoor Outdoor	 0.1 mW/cm^2 100 mW/cm^2	 10 μW/cm^2 10 mW/cm^2
Vibration/Motion Human Industrial	 0.5m@1Hz 1m/s^2@50Hz 1m@5Hz 10m/s^2@1kHz	 4 μW/cm^2 100 μW/cm^2
Thermal Energy Human Industrial	 20 mW/cm^2 100 mW/cm^2	 30 μW/cm^2 1 - 10 mW/cm^2

The sources characteristics vary with size, which is the case of the ambient light and thermal energy sources, or are based on movements and accelerations patterns, which can be mathematically approximated by sinusoidal functions of vibratory or motion sources. Considering a 1 cm^2 area for the harvesting device which is responsible to take the energy from the source and convert it to electrical power, the energy obtained may be used in systems with power consumption in the range of 10 μW – 10 mW [Fiorini08] for ambient light, vibrations/motion and thermal energy sources.

2.2.2.3 Physiological Signals

In a BSN, different physiological signals may be monitored simultaneously and these signals usually have heterogeneous network requirements. This is the case of a BSN monitoring different vital signs from postoperative patients, which may include ECG, blood pressure, heart rate and temperature. Table 2.4 shows the electrical characteristics of the vital signs usually monitored in emergency medical care.

Table 2.4 – Vital signals electrical characteristics [Gama09].

Vital Signal	Frequency Range (Hz)	Sampling rate (Hz)	Resolution (bit)	Data Rate (Kbit/s)
ECG (per lead)	0.01 … 60-250	120 - 500	16	4
Temperature	0 … 0.1-1	0.2 - 2	12	0.024
Oximetry	0 … 30	60	12	0.72
Blood Pressure	0 … 60	120	12	1.44
Respiration Rate	0.1 … 10	20	12	0.24
Heart Rate	0.4 … 5	12	12	0.12

The information obtained from these signals is sent in data packets to the base station in burst or in single packets, which is the case where there is no medical emergency monitoring situation. In urgent medical situations, where the patient's life is in danger, data packets should be transmitted continuously and in real-time [Gama09].

2.2.2.4 BSN Physical Considerations and Radio Technologies

BSNs' characteristics impose a series of challenges in the development of a suitable physical layer to support communications, thus, the following considerations must be taken:

- The level of transmission power for radio transceivers, as well as the reduction of the patients exposure to RF energy and the decrease of interference among adjacent BSNs;
- The BSN power consumption, which is highly related to the transmission power and the MAC protocol;
- The application data rate, radio-frequency modulation and wireless channel quality.
- The influence of the human body on the RF communication channel, which may affect the reliability of the communications and, consequently, the power consumption.

Human Body Interference on RF Communications

The path loss in a wireless channel is commonly represented through the empirical log-normal shadowing path loss model presented in equation 2.1, which is the received power in dB at a distance *d*. $P_{r(d0)}$ is the path loss at the reference distance d_0 in dB and $X_{\sigma,dB}$ is a zero-mean Gaussian random variable with standard deviation σ dB. For the wireless wave propagation, there is attenuation in transmission power at the rate d^{η}, where η is the path loss exponent, which is equal to two in free space and tends to be higher in indoor environments.

$$P_r(d) = P_r(d_0) + 10\,\eta\, log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma,dB} \qquad (2.1)$$

Since most of the BSN sensors are attached to the human body, several studies have been made in order to evaluate the interference of the human body on wireless communications. These studies include the analysis of static and dynamic BSNs with communication between line of sight (LOS) and non-line of sight (NLOS) sensor devices.

In [Uddin11], for communications between LOS transmitter and receiver devices attached in different body segments (arms, legs, torso, backs), it was found that η is between

three and four, but for communications between two NLOS devices, one placed in the torso and the other in the human back, it was found a η ranging from five to six. For device dynamic BSNs, it was also concluded that the path loss increases up to 5 dB for LOS and around 15-20 dB for NLOS in relation to static BSNs.

Human Body Communications

Human body communications (HBC) is a prospective communication technology that explores the possibility of using the human body as a signal propagation medium. At the moment, there are two solutions for HBC: electromagnetic coupling and electric field coupling (also known as body capacitive coupling - BCC). In the electromagnetic coupling solution, the human body is treated as a waveguide where the RF signal propagates through the body. In the electric field coupling solution, devices are placed on or near the body and the data is transmitted across the devices by near electric fields.

BCC is appellative for BSNs because, in contrast with electromagnetic coupling, BCC transceivers generate weak but still detectable electric fields that only extends outwardly a couple of centimeters from the surface of the skin, allowing the transmission of small amounts of information and enabling communications without interfering with other adjacent BSNs [Falck07].

IEEE 802.15.4

The IEEE 802.15.4 is focused in the work presented in this document and is discussed with detail in section 2.3.2.

IEEE 802.15.6

The IEEE 802.15.6 is the standard protocol that is being developed to address the specific requirement for the BSNs. This standard aims to develop the medium access control and physical layers for BSN. It focuses on functioning at relatively low frequencies (below one megahertz), aims for sort-range use, low cost, reliable wireless communications and especially ultra-low power [Bradai11].

The current IEEE 802.15.6 standard defines three physical layers: narrowband, UWB, and HBC layers. The selection of each PHY layer depends on the application requirements. In narrowband, it may operate in different bands, including the 2.4 GHz ISM band at 971.4 kb/s. In UWB, data rates range approximately from 0.4 Mbit/s up to 12.6 Mbit/s. HBC uses capacitive coupling and data rates may scale up to 2 Mbit/s [Batra11].

2.2.2.5 BSN Communication Architectures

Three different communication architectures may be defined for a BSN: wired, wireless and hybrid [Chen11].

Wired architectures are used to avoid the challenges of wirelessly interconnecting sensor devices. Existing schemes, such as MITHril [Pentland04] and SMART [Curtis08], use wired links to connect the sensor devices directly to a personal server (PS), i.e., a PDA. The PS will then relay the information wirelessly to a BS. However, wired systems compromise patient quality of life because they may be obligated to wear special suits or to live with wires attached to the body.

In wireless architectures, sensor devices may transmit the information following two different approaches: sensor devices communicate directly with BSs without a PS (e.g., CodeBlue [Shnayder05]); or, alternatively, sensors may communicate to a PS that then relays the information to a BS (e.g., WiMoCa [Farella08]).

In [Chen11], a different architecture is also considered, in which the BSN is divided in two levels. At the first level, sensor devices communicate through wires (hybrid approach) or wirelessly (forming a cluster) to a central processor device, in order to reduce the amount of raw data through the data fusion and to ultimately save energy. At the second level, the PS will then relay the information wirelessly to a BS.

The Posture Monitoring System

The evaluation scenarios proposed in this work uses traffic parameters extracted from a real implementation of a multi-user motion capture application that is based on several wireless sensor devices, each one containing multiple inertial and magnetic sensors: the wireless Posture Monitoring System (PMS). When one of these sensor devices is attached to an object, its orientation in 3D space can be obtained. Likewise, when several modules are attached to different segments of a user's body, the movements of the user can be tracked. The PMS is composed of three main components: the PC, the base station and multiple sensor devices that collect movement data, as represented in Figure 2.6. The PC is responsible to receive the data from sensors and compute the angles of the body segments being monitored, which are sent to a 3D model that displays the users' movements. It also provides a user interface to enable the configurations of several parameters of the BSN. The base station acts as the network coordinator and the sensor devices are the responsible for collecting posture

data from the respective body segments [Silva11].

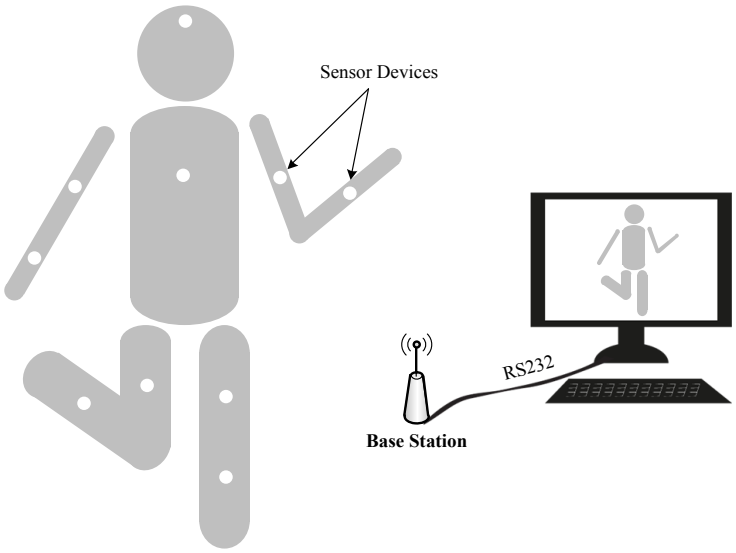


Figure 2.6 – Posture Monitoring System overview.

Table 2.5 presents a summary of the parameters used by the PMS. Each sensor device includes a set of six sensors, which enables the application on the PC to process the data and obtain the position of the body segment that is being monitored. These sensors are constituted by three accelerometers and three magnetometers, each of which generates 12 bits per sample. The set of data collected from these six sensors generate 72 bits of information for each body segment, which corresponds to 9 bytes. Each data packet generated by a sensor device contains at least one sample of each sensor and it also includes 2 bytes with information of the status of the battery.

Table 2.5 – Posture Monitoring System parameters.

Parameter	Designation	Value
Number of sensors per device.	N_s	6
Sensor accuracy.	Q_s	12 bit
Battery Accuracy.	Q_B	12 bit
Sampling Rate.	f_s	30 Hz

For the movement captured by the monitoring system to be smooth, a sampling rate of 30 Hz was chosen, because many motion capture applications typically require a frame rate of 30 fps. Other applications may require even higher sampling rates in order to track faster movements.

The main reason to use traffic parameters provided by the PMS application in this work is due to its data-intensive characteristics. The traffic generated by the PMS is highly intensive because each BSN device has multiple sensors which are sampled several times per second.

2.2.3 Quality of Service

In order to provide a pervasive, valuable and highly reliable assistance to any patient, health care monitoring systems should always provide quality of service support. QoS control mechanisms usually use traditional end-to-end QoS parameters, such as packet loss, delay, jitter, and available bandwidth, to characterize the performance of the network and to guarantee consistent service levels concerning application requirements. At the application level, QoS may also be regarded by guaranteeing the right number of sensors for monitoring the vital signals in accordance with the patient's emergency state [Gama09].

The importance of the collected information in a BSN is necessarily unique, especially in medical care monitoring applications, where the information may be classified as critical or non-critical. Critical information, such as a sudden clinical change in patient's health state, must be prioritized in relation to non-critical information. For instance, in patients with cardiac diseases, the heart activity information is more important than its body temperature information. The prioritization of the information may be assigned dynamically because several applications may consider the monitored information as non-critical, and, consequently, with low priority. If a sudden change in monitored data occurs, such a hypo or hyper-glycemia in a glucose monitoring system, a higher priority may be reassigned [Gama09]. In addition to measuring information from sensors, a BSN may also generate control or alarm triggered data. Under these circumstances, high priority level should be assigned to data packets carrying alarming notification and measurements, and to acknowledgements of correctly received packets. Also, a medium priority level should be assigned to scheduled transmissions of data packets and primary control packets (e.g. sensor configuration), and a low priority level should be given to periodic polling of nodes for checking the integrity of the network and secondary control packets [Lamprinos06].

An important parameter to take into account in a BSN is the availability of energy resources, in order to prevent energy failures. In the case of a lack of power, this may be achieved by controlling the consumed power in accordance with the patient clinical state. For

example, if a patient is in a normal state, the sampling rate of the sensors may be reduced. In a critical situation, energy consumption may be preserved for more important tasks [Gama09].

Since computing demands less energy than transmission, data may be compressed to reduce the number of transmitted packets and respective overhead, which reduces the overall energy spent in the data transmission. The packet length must be always considered, as it tends to increase linearly with the delay. Moreover, for efficiency reasons a large packet may be used for non-critical situations. In critical situations, the packet's length may be reduced in order to fulfill low delay QoS requirements [Gama09].

The QoS framework should be flexible so that it can be dynamically configured to suit application requirements without excessively increasing complexity or decreasing system performance. Real-time and critical BSNs may be both delay-sensitive and loss-sensitive, where loss or corruption of data due to an unreliable network may have severe consequences. Since sensor devices have limited memory, strong error detection and correction schemes, and efficient acknowledgment and retransmission mechanisms must be defined because there is little room to store and retry the transmission of unacknowledged data [Patel10].

2.3 Wireless Sensor Networks and Protocols

2.3.1 Definition and Applications

The field of wireless sensor networks (WSN) is growing and improving rapidly, allowing the creation of new communication services. Sensor networks are used to monitor and control various environmental parameters and information in industrial environments, houses, buildings, transportation systems, agricultural lands, wildlife areas, etc. A wireless sensor network consists in a set of sensor devices distributed over an area in order to monitor activity in real time. These devices may operate together to collect data such as temperature, humidity, acceleration, etc. Additionally, sensor devices may contain actuators, such as mechanical switches and piezoelectric actuators. With the advancements of the WSN technologies, wireless sensors can be smaller, battery powered and with capacity of self-organization. However, wireless sensors are limited in power, storage and processing capacity [Akyildiz02].

Some of the characteristics of WSN's are:

- Cooperation, where the network devices work together in order to achieve a common goal;
- Low data rate, where the network devices usually generate traffic when triggered by a particular event such as an out-of-range value detected by a sensor;
- Low network traffic, where nodes usually generate small length messages;
- Low processing loads in the network devices;
- Multi-hop communications;
- Low energy consumption;
- Ad-hoc operation;
- High density of network devices.

Several protocols have been developed to address the WSNs requirements [Suriyachai12][ZigBee07]. This work is based on the IEEE 802.15.4 and the ZigBee protocols, two standard-based communication protocols developed to support WSNs, which are described in the next sections.

2.3.2 The IEEE 802.15.4 Protocol

The IEEE 802.15.4 protocol was developed for low rate wireless personal area networks (LR-WPAN). The first version of this protocol was published in 2003 [IEEE4-03], but one revision [IEEE4-06] and three other modifications to the protocol [IEEE4-07][IEEE4-09I] [IEEE4-09II] were made since then. This protocol is used as the base of the ZigBee protocol.

2.3.2.1 IEEE 802.15.4 Protocol Overview

The IEEE 802.15.4 protocol specifies both the PHY and MAC layers for low power, low rate and low cost wireless network devices. Two different types of devices are allowed in the IEEE 802.15.4 protocol: full function devices (FFD) and reduced function devices (RFD). A FFD is usually mains powered and may operate as a personal area network (PAN) coordinator. A RFD is typically a battery powered end device. FFDs can communicate with other FFDs or RFDs, but an RFD can only talk with its FFD parent. FFDs implement the complete protocol set, which enables them become network coordinators. On the other hand, RFDs only implement part of the protocol, which enables its implementation on simpler devices. The protocol defines a set of designations for the different entities that may compose the network: the PAN coordinator, which is a FFD and is the principal controller of the PAN;

the coordinator, a FFD entity that controls and synchronize the cluster of its associated devices; the alternate PAN coordinator, which is a coordinator capable of replacing the PAN coordinator; and the device, which is any entity (FFD or RFD) that has an implementation of the IEEE 802.15.4 protocol.

Figure 2.7 illustrates the network topologies supported by the protocol: star and peer-to-peer. In both topologies there is a PAN coordinator that creates and controls the network. In the star topology, all the network devices are connected and transmit their packets only to the coordinator. On the other hand, in the peer-to-peer topology, network devices may communicate directly with any other device in its radio communication range. The standard provides an example of other network topology: the cluster tree topology, which is illustrated in Figure 2.8. Despite of the cluster tree topology being a peer-to-peer topology, the IEEE 802.15.4 protocol have no support for it because routing mechanisms are handled by a network layer, which the protocol does not provide.

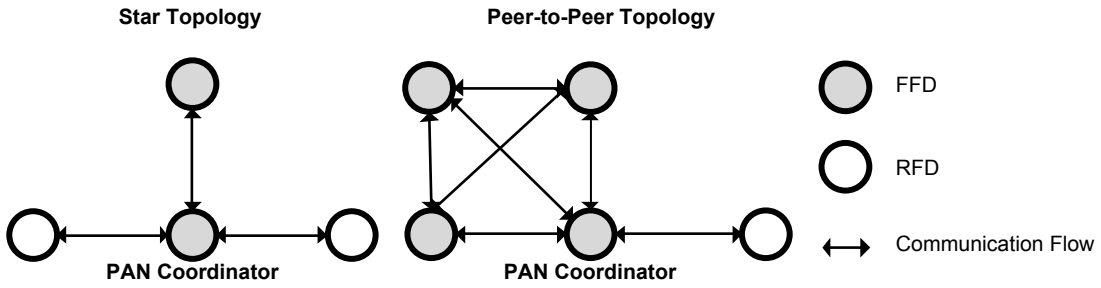


Figure 2.7 - Star and Peer-to-Peer topologies.

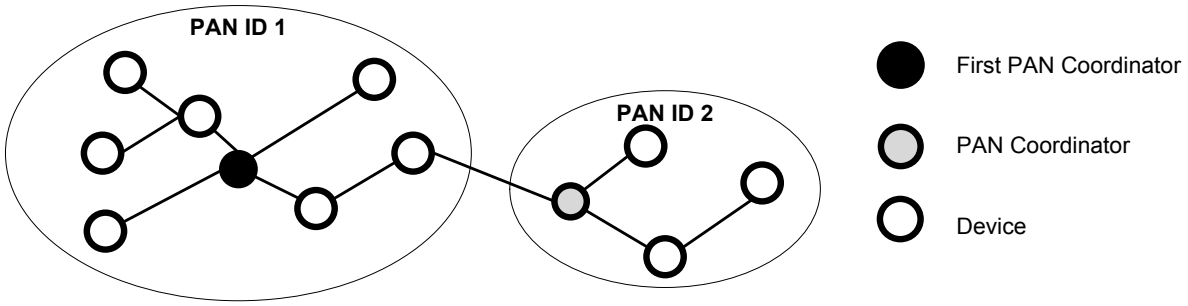


Figure 2.8 - Cluster tree network topology.

A PHY and a MAC layer constitute the IEEE 802.15.4 stack model, shown on Figure 2.9. The PHY layer contains functionalities provided by the radio transceiver. The MAC arbitrates the access of the devices to the medium. The PHY layer provides data services through the physical data SAP (PD-SAP) and management services through the physical layer

management entity SAP (PLME-SAP). The services provided by the PLME-SAP include channel frequency selection and clear channel assessment (CCA). Additionally, the PLME-SAP interfaces with the PHY layer PAN information database (PHY PIB), which maintains a set of objects used to configure its mechanisms. The MAC layer provides data and management services. The MAC data service, accessed through the MAC common part sub-layer SAP (MCPS-SAP), is the interface to transmit data. The MAC management service, accessed through the MAC layer management entity SAP (MLME-SAP), provides management functions. Similarly to the PLME-SAP, the MLME-SAP maintains the MAC information base, known as MAC PIB.

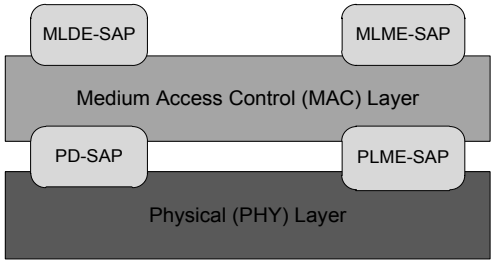


Figure 2.9 - IEEE 802.15.4 stack model.

IEEE 802.15.4 devices use the carrier sense multiple access collision avoidance (CSMA-CA) as the contention protocol to access the medium. The CSMA-CA has two different versions: the slotted CSMA-CA and the unslotted CSMA-CA. In non-beacon enabled networks, devices use the unslotted CSMA-CA mechanism and, in a beacon enabled network, the devices use the slotted version. The beacon enabled network uses a superframe structure, which is illustrated in Figure 2.10.

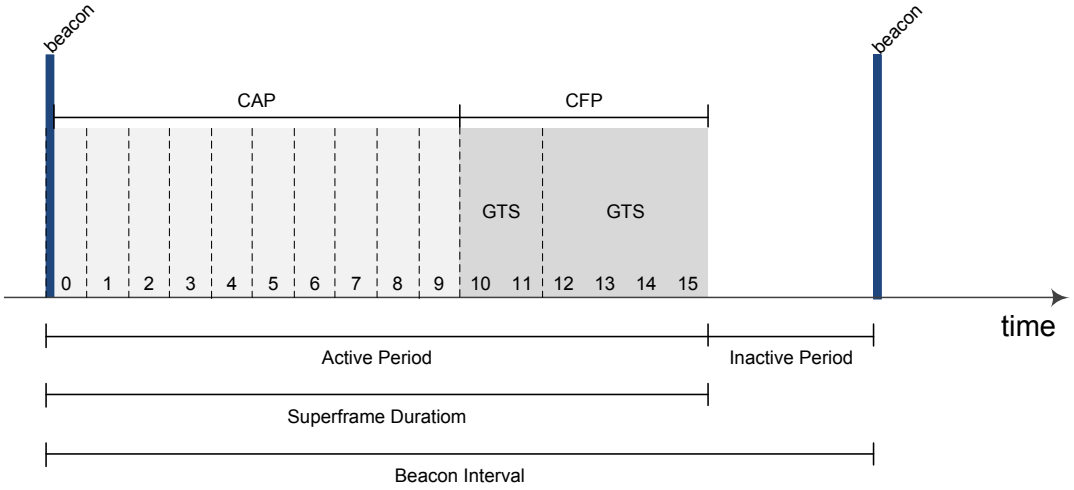


Figure 2.10 - IEEE 802.15.4 superframe structure.

A periodic beacon, transmitted by a network coordinator, delimits the superframe. The period between the beacons is known as Beacon Interval (BI). The superframe is divided into two different periods: the active period, which corresponds to the superframe duration (SD) and is used by the network devices to communicate, and the inactive period, in which all the communications are disabled. The active period, where the devices use the slotted CSMA-CA mechanism to communicate, is divided into 16 equal time slots. During the inactive period, the devices may enter in a sleep mode by switching off their radio transceivers in order to save energy. The SD and BI are given by equations 2.2 and 2.3, respectively.

$$SD = aBaseSuperframeDuration \times 2^{SO}\ symbols \quad (2.2)$$

$$BI = aBaseSuperframeDuration \times 2^{BO}\ symbols \quad (2.3)$$

The *aBaseSuperframeDuration* parameter, which is defined in the IEEE 802.15.4 standard as *aBaseSlotDuration* × *aNumSuperframeSlots*, corresponds to 15,36 ms. The SO (Superframe Order) and BO (Beacon Order) parameters are defined in the IEEE 802.15.4 standard as *macSuperframeOrder* and *macBeaconOrder*, respectively, and are related as follows: $0 \leq SO \leq BO \leq 14$. The SD and BI parameters are configurable and may be adjusted to the nodes traffic parameters.

For applications that require low latency or guaranteed bandwidth, the coordinator provides a scheme for allocation of dedicated slots. These slots are called guaranteed time slots (GTS) and form the contention-free period (CFP) in the superframe structure. The CFP is placed immediately after the contention access period (CAP), where the slotted CSMA-CA is used. The GTS scheme only allows a maximum of 7 GTS allocations. Once a GTS is allocated to a particular device in the CFP, the device may transmit its packets without any contention because no other device is allowed to transmit in that particular GTS.

The IEEE 802.15.4 protocol defines four different frame structures:

- Data frames, used to transfer data between devices;
- Acknowledgement frames, used to confirm the successful reception of a data frame;
- Beacon frames, used by the coordinator to synchronize devices and disseminate information;
- MAC control frames.

Three different models of data transmission are defined by the protocol. The first and the second models, both illustrated on Figure 2.11, indicate how to transfer information from the coordinator to the device and from the device to the coordinator, respectively. The third model is used to transfer data in peer-to-peer networks. In the star network topology the first and second models are used while in the peer-to-peer topology any of the three models can be used.

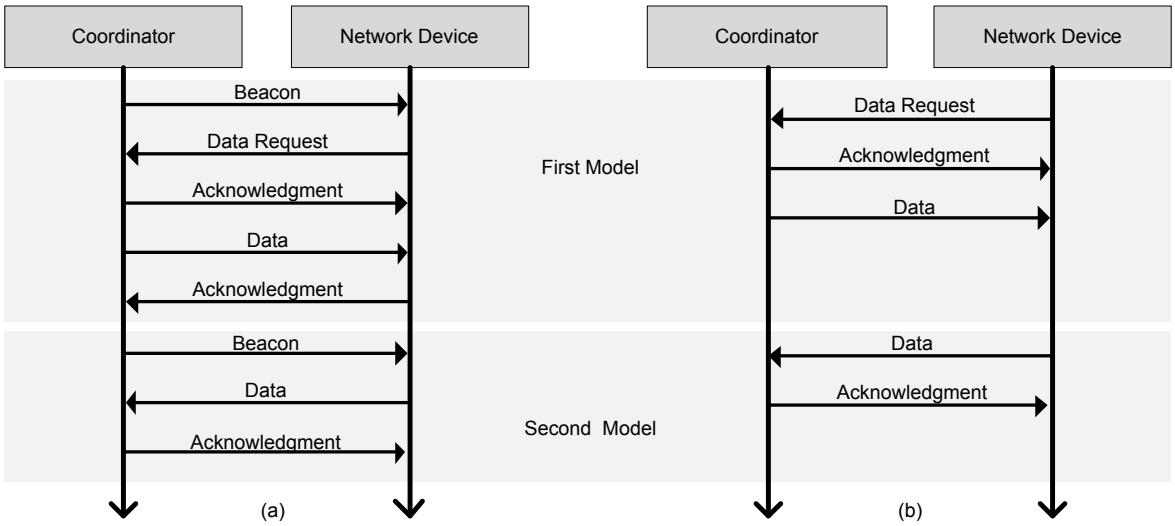


Figure 2.11 - IEEE 802.15.4 data transfer models in beacon enabled (a) and non-beacon enabled (b) networks.

To transfer data from a coordinator to a device in a beacon enabled network (Figure 2.11(a)), the coordinator indicates in the beacon that a packet is pending for that device. The device listens to the beacon and makes a data request to the coordinator. This request is made in CAP using slotted CSMA-CA. Both the coordinator and the device can perform the respective packet acknowledgments. If a device wants to transmit data to the coordinator, it has to hear the beacon, synchronize with the superframe and transmits the information using slotted CSMA-CA. If requested, the coordinator sends the acknowledgment.

The data transfer from a coordinator in a non-beacon enabled network mode (Figure 2.11(b)) depends on the destination device in which it has to do polling, asking whether there is data pending. If there is data available for the device, the coordinator sends an acknowledgment with this option and then sends the data. If requested, the device sends an acknowledgment to the coordinator. If there are no data pending, the coordinator indicates this to the device. All communication is achieved using unslotted CSMA-CA. When a device wants to transmit to the coordinator, it simply sends the data packet using the unslotted CSMA-CA, and the coordinator transmits an acknowledgment, if requested.

In a peer-to-peer topology, since the devices can communicate with all the others devices in the network, the protocol basically uses the unslotted CSMA-CA to avoid of having a more complex mechanism of synchronization between the devices.

2.3.2.2 Physical Layer

The PHY layer of the IEEE 802.15.4 protocol provides a set of frequency bands available for the communications. Depending on the protocol version, the modulation techniques and channel data rate capacity may vary within the different frequency bands. The PHY configurations of the first version of the protocol and the optional PHY options defined in the revision of 2006 are shown in Table 2.6. The original version specifies two physical layer options based on direct sequence spread spectrum (DSSS) technique. The first option works in the 868/915 MHz bands with data rates of 20 and 40 kbit/s, respectively, while the second option is in the 2.45GHz band with a data rate of 250 kbit/s. The revision of 2006 introduces the concept of channel pages, in which new optional physical configurations are defined, offering a tradeoff between complexity and data rate. In the channel page 0 are included the physical configurations of the 2003 version of the protocol. Channel pages 1 and 2 optional PHYs offer a data rate much higher than that of the 868/915 MHz BPSK PHY in the original version of the protocol, which provides for 20 kbit/s in the 868 MHz band and 40 kbit/s in the 915 MHz band. Channel pages 3 to 31 are reserved for future use.

Table 2.6 - IEEE 802.15.4 2006 [IEEE4-06] PHY configurations.

Channel Page(s)	Channel Number(s)	Optional	Frequency Band	Spreading Parameters		Data Parameters	
				Method	Modulation	Bit Rate (Kbps)	Symbol Rate (Ksymbol/s)
0	0	No	868 MHz	DSSS	BPSK	20	20
	1 - 10	No	915 MHz	DSSS	BPSK	40	40
	11 - 26	No	2450 MHz	DSSS	O-QPSK	250	62.5
1	0	Yes	868 MHz	PSSS	ASK	250	12.5
	1 - 10	Yes	915 MHz	PSSS	ASK	250	50
	11 - 26	…	…	…	…	…	…
2	0	Yes	868 MHz	DSSS	O-QPSK	100	26
	1 - 10	Yes	915 MHz	DSSS	O-QPSK	250	62.6
	11 - 26	…	…	…	…	…	…
3 -31	Reserved						

The center frequency (F_c) of each channel is obtained using the equations 2.4, 2.5 and 2.6, where k is the channel number.

$$F_c = 868.3\ MHz, \qquad k = 0. \tag{2.4}$$

$$F_c = 906 + 2(k-1)\ MHz, \qquad k = 1, 2, \ldots, 10. \tag{2.5}$$

$$F_c = 2405 + 5(k-11)\ MHz, \qquad k = 11, 12, \ldots, 26. \tag{2.6}$$

The PHY layer of the IEEE 802.15.4 protocol maintains the PHY PIB containing objects for its configuration, which that can be retrieved and updated using get and set primitives provided by the PLME-SAP. Some of the PHY PIB attributes are described in Table 2.7.

Table 2.7 - PHY PIB attributes.

Attribute	**Description**
phyCurrentChannel	The radio channel to use for all following transmissions and receptions.
phyTransmitPower	The transmit power and the tolerance.
phyCCAMode	The CCA mode to be used.
phyCurrentPage	This is the current PHY channel page. This is used in conjunction with *phyCurrentChannel* to uniquely identify the channel currently being used.

The PHY layer shall perform the clear channel assessment (CCA), a mechanism to verify if the channel is idle or occupied. The protocol specifies that, before transmitting, devices perform the CCA according to three different modes defined in the standard: CCA Mode 1, where the device considers a busy channel if the detected energy level is above a threshold value; CCA Mode 2, in which the device reports a busy channel if it detects any signal compliant with the IEEE 802.15.4 PHY, regardless of a threshold value; and CCA Mode 3, which combines both CCA Mode 1 and CCA mode 2, where the device reports a busy channel if it detects a signal compliant with the IEEE 802.15.4 PHY and it is above a threshold value. At least one of these modes should be implemented in the devices.

The IEEE 802.15.4 physical layer defines two constants of particular interest in this work: *aMaxPHYPacketSize* and *aTurnaroundTime*. The first one indicates the maximum length of the PHY service data unit (PSDU), which corresponds to 127 octets. The physical packet data unit (PPDU), which includes the PSDU and the PHY packet headers, may vary with the version of PHY in use. In the case of the 2.45 GHz band PHY layer, the PPDU maximum length is 133 octets. The second constant specifies the maximum time for the transceiver to change from transmit mode (TX) to receiver mode (RX) and vice-versa, which

is 192 μs.

2.3.2.3 Medium Access Control Layer

The MAC layer of the IEEE 802.15.4 protocol maintains the MAC PIB, which contains a set of objects for its configuration. A few MAC PIB attributes and constants are described in Table 2.8. The MAC PIB attributes can be retrieved and updated using get and set primitives provided by the MLME-SAP. In the CSMA-CA, a device maintains three variables, used in each transmission attempt: NB, CW and BE. NB is the number of times the CSMA-CA algorithm can backoff until it declares a channel access failure; CW, which is only used in the slotted CSMA-CA, is the contention window length and defines the number of times that CCA must declare a channel free of activity before the transmission can commence; BE is the backoff exponent, which is related to how many unit backoff periods (*aUnitBackoffPeriod*) a device shall wait before attempting to access the channel.

Table 2.8 - CSMA-CA attributes and constants [IEEE4-06].

Attribute	Values*	Description
aUnitBackoffPeriod	20 symbols (0.32 ms for 2450 MHz PHY)	The number of symbols forming the basic time period used by the CSMA-CA algorithm.
macMinBE	0 – 3 (default = 3)	The minimum value of the backoff exponent.
macMaxBE	3 – 8 (default = 5)	The maximum value of the backoff exponent.
macMaxCSMABackoffs	0 – 4 (default = 4)	The maximum number of backoffs the CSMA-CA algorithm will attempt before declaring a channel access failure.
macMaxFrameRetries	0 – 7 (default = 3)	The maximum number of retries allowed after a transmission failure.

**Range and default values used in the 2006 revision of the protocol.*

Figure 2.12 depicts the unslotted CSMA-CA algorithm defined in the 2006 version of the protocol. The CCA is performed at each iteration of the algorithm, and indicates if the channel is idle or not. Before performing CCA, this algorithm waits for a random interval between 0 and (2^{BE} - 1) unit backoff periods (*aUnitBackoffPeriod*), where BE takes the value of *macMinBE* at the beginning of the algorithm and increases at each iteration until it reaches *macMaxBE*[1]. If the CCA declares that the channel is idle, the algorithm ends with success status and the transmission may start; otherwise a new iteration is initiated. The algorithm

[1] In the 2003 version of the algorithm, the constant macMaxBE should be substituted by the aMaxBE parameter, which is a constant value and is equals to 5.

may perform CCA at most *macMaxCSMABackoffs* times. Once this value is exceeded, the algorithm declares a channel access failure.

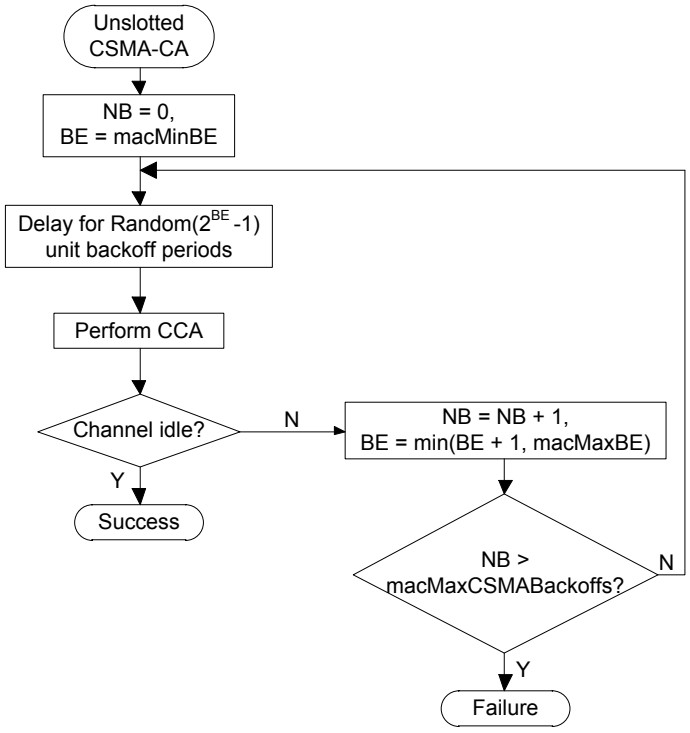


Figure 2.12 - IEEE 802.15.4 unslotted CSMA-CA [IEEE4-06].

Figure 2.13 shows the slotted CSMA-CA algorithm defined in the 2006 version of the protocol. In the slotted version, the backoff period boundaries of every device shall be aligned with the superframe slot boundaries of the PAN coordinator. The transmissions should start at the beginning of a backoff period. First, the MAC layer initializes NB, CW and BE, whose value depends if the battery life extension field is set or not. Then, it locates the beginning of the next backoff period boundary, delays for a random number between 0 and (2^{BE} - 1) unit backoff periods (*aUnitBackoffPeriod*) and performs the CCA in the current superframe. If the channel is considered idle, the algorithm will next perform the CCA as many times as CW indicates. After CW successful verifications of an idle channel, the CSMA-CA algorithm ends with a success status. On the other hand, if the channel is found busy, the values of NB and BE are updated. If NB exceeds *macMaxCSMABackoffs*, the algorithm declares a channel access failure, otherwise, the algorithm calculates a new random delay and the process repeats.

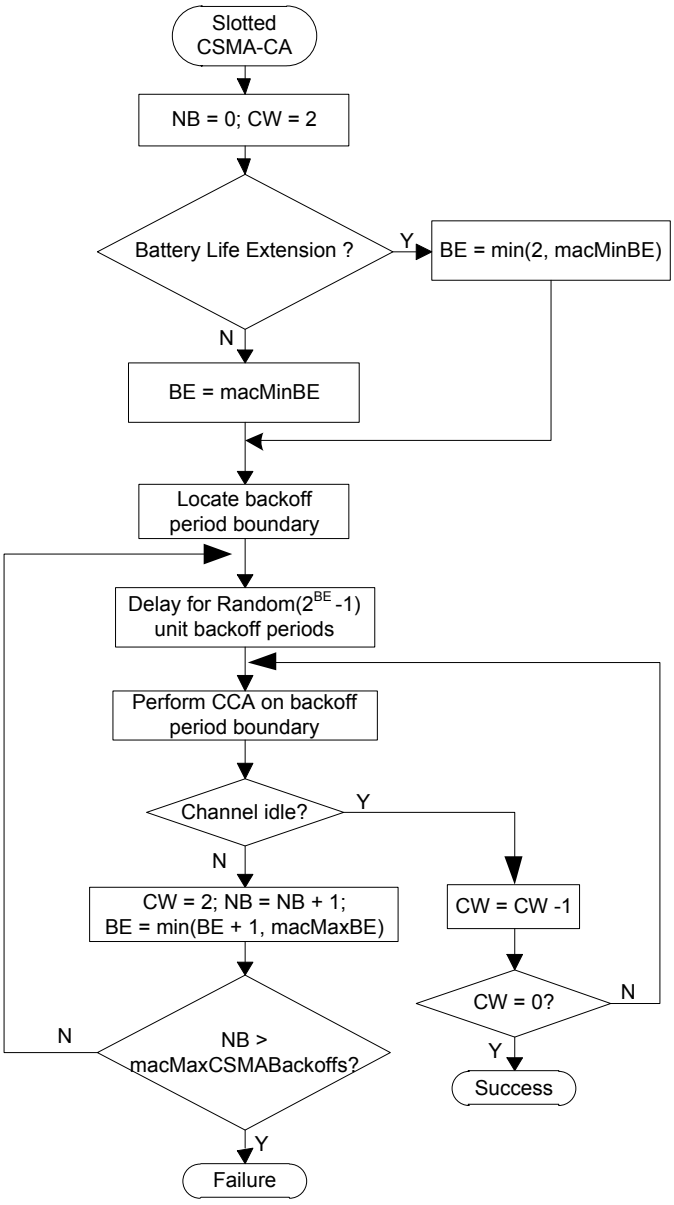


Figure 2.13 - IEEE 802.15.4 slotted CSMA-CA [IEEE4-06].

Although the CSMA-CA mechanism provides control in the access to the medium, it cannot guarantee that the messages are successfully delivered to the destination. Message loss may occur due to several factors, such as collisions, fading or interference. Furthermore, the CSMA-CA does not provide specific means to avoid the hidden node or the exposed node problems.

2.3.3 The ZigBee Protocol

ZigBee is a standard-based commercial protocol developed by the ZigBee Alliance, a non-profit association of companies, governmental regulatory groups and universities. It was

designed for low power devices used on wireless monitoring and control systems. Additionally, it was designed to support multi-application environments and interoperability between devices of different manufacturers.

The first version of the ZigBee protocol, ZigBee version 1.0 (ZigBee 2004) [ZigBee04], was released in December 2004. In December 2006 the second version was released, ZigBee 2006 [ZigBee06]. It was followed by the ZigBee 2007 specification [ZigBee07], which includes two stack profiles: ZigBee and ZigBee PRO. In this section, an overview of the ZigBee 2007 specification is given. This was the version used in the development of the work presented in this document. The ZigBee Alliance guarantees that ZigBee 2007 is compatible with the ZigBee 2006 version, but the compatibility with the 2004 version is not assured.

2.3.3.1 ZigBee Protocol Overview

Figure 2.14 shows the ZigBee stack model. Both the Security Services Provider and the ZigBee Device Object (ZDO) offer services to the Network (NWK) and Application (APL) layers. Users develop the application objects using the Application Framework and share Application Support Sublayer (APS) and ZDO services [ZigBee07]. The PHY and MAC layers of ZigBee 2007 are defined by the IEEE 802.15.4 2003 [IEEE4-03] standard.

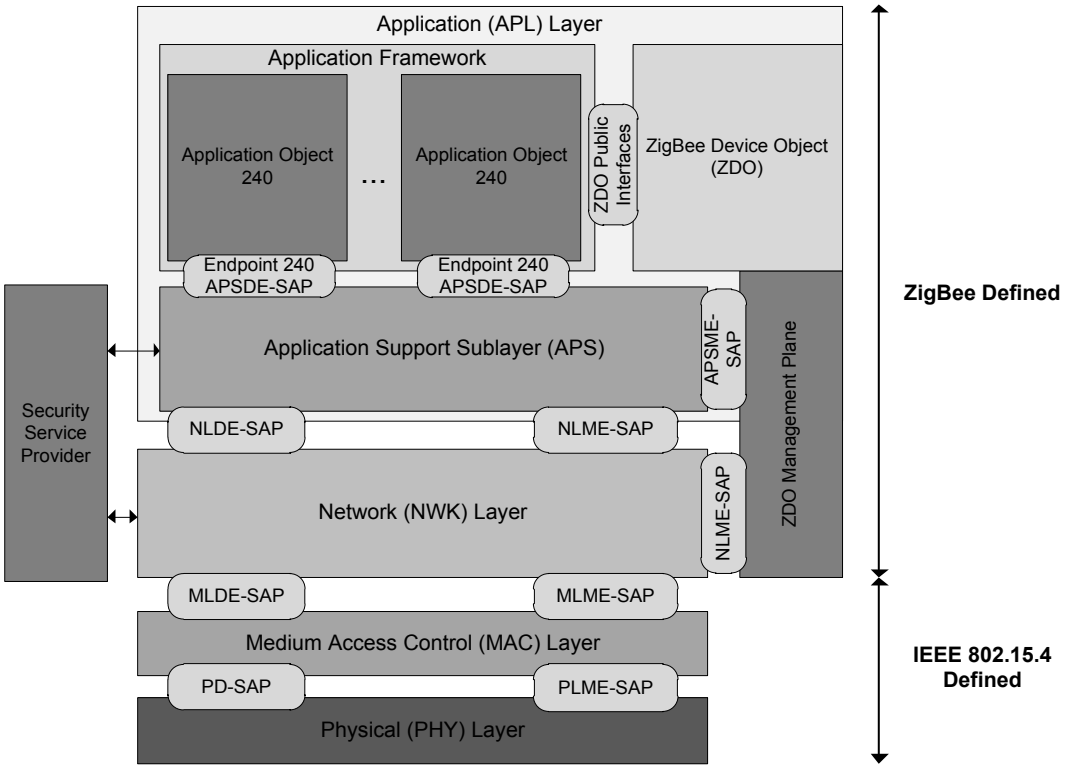


Figure 2.14 – ZigBee model [ZigBee07].

The ZigBee protocol defines the following device types: coordinator (ZC - ZigBee Coordinator), router (ZR - ZigBee Router) and end device (ZED - ZigBee End Device). Both ZCs and ZRs are FFDs, while the ZEDs are RFDs. The ZC is the equivalent to the IEEE 802.15.4 PAN coordinator. A ZR can work as an IEEE 802.15.4 coordinator and is capable of routing messages and accepting new device associations. ZEDs are always terminal network nodes because they cannot relay information from other nodes. They usually operate on battery power; therefore, energy conservation is crucial to assure longevity in their operation. To achieve this, ZEDs are endowed with the ability to sleep and can wake up only when a relevant event happens [Gislason08].

2.3.3.2 Network Layer

The network (NWK) is the lower layer defined in the ZigBee standard. The NWK layer provides a set of services offered via two entities: the NWK layer data entity (NLDE), whose services can be accessed through the NLDE-SAP, and the NWK layer management entity (NLME), available through the NLME-SAP. The NLDE is the entity responsible for the following data transmission services [ZigBee07]:

- Generate the network level PDU (NPDU) by adding the protocol overhead;
- Transmit a NPDU to a device that is either the final destination of the communication or the next step towards the final destination.

The NLME shall provide management services to allow an application to interact with the stack. The services provided are [ZigBee07]:

- Configure new devices, which include starting a device as a ZC or joining an existing network as a ZED or a ZR;
- Start a new network (ZC);
- Join, rejoin or remove devices to and from the network;
- Assign network addresses to new joining devices;
- Discover, record, and report information pertaining to the one-hop neighbors of a device;
- Discover and record paths throughout the network;
- Control when the receiver is activated and for how long;
- Use different routing mechanisms such as unicast, broadcast, multicast or many-to-one to efficiently exchange data in the network.

There are three general communication modes that are available: unicast, broadcast and multicast. Unicast is used to send a message to a single device, whereas broadcast messages are sent to all devices within a given radius. Multicast transmissions are used to send a message to devices that belong to a specific multicast group.

The ZigBee 2007 adopts four routing methods: tree, mesh, many-to-one and source-route. The ZigBee PRO does not support tree routing and is the only profile of the ZigBee 2007 protocol that supports many-to-one and source routing. Tree routing allow devices to relay messages without routing tables because network addresses are assigned through a special way using the Cskip algorithm. In mesh networks, routes are established using the Ad hoc On-Demand Distance Vector (AODV) routing protocol. The many-to-one routing is used in networks where most devices transmit data to a data concentrator node or gateway. The concentrator periodically broadcasts a single many-to-one routing request message to establish reverse routes on all devices. On the other hand, source routing may be used when a source device need to send data to multiple remote devices. In source routing a route record command is used, which is sent from the intended destination back to the source device, to record the path. The route record command appends the 16-bit address of each device on the route into the route record message payload and this information will be stored and used to send source-routed packets to the remote nodes.

A device NWK layer keeps the state of its neighbors to which it has an outgoing link by maintaining a transmission failure counter that is used to determine the link status. In case of a link failure, the NWK will proceed with the route maintenance protocol [ZigBee07].

2.3.3.3 Application Layer

The application (APL) layer is the upper layer of the ZigBee protocol and consists in the application support sublayer (APS), the application framework and the ZigBee device object (ZDO) [ZigBee07].

The application support sub-layer

The APS provides an interface between the network layer and both ZDO and manufactured application objects. This layer provides a set of general services offered via two entities: the APS data entity (APSDE), whose services can be accessed through the APSDE-SAP, and the APS management entity (APSME), available through the APSME-SAP. The APSDE is the entity responsible for the following data transmission services [ZigBee07]:

- Take an application PDU and generate an APS PDU by adding the protocol overhead;
- Transfer a message between bounded devices;
- Filter group-messages based on grouped application endpoints;
- Employ end-to-end retransmissions;
- Reject duplicated packets;
- Enable the fragmentation and the assembly of messages longer than allowed.

The APSME shall provide management services to allow an application to interact with the stack. The services provided by the APSME are [ZigBee07]:

- Match two devices based on their services and needs;
- Manage the APS information base (AIB) through set and get primitives;
- Authentication through secure keys;
- Manage network application groups by declaring network addresses shared by multiple devices and to add and remove devices from group.

Application Framework

The application framework contains up to 240 user-defined application objects, each one identified by endpoints 1 to 240, which allow to develop and to identify different applications into the same node. Endpoint 0 is used to address the ZDO, whereas endpoint 255 is used to address all active endpoints. Endpoints 241-254 are reserved for future use [ZigBee07].

Each ZigBee application is associated to a 16-bit identifier, the Profile ID, which identifies its profile. A profile is a domain of related applications and devices. The profiles are divided into two classes: public or private. Public profiles designate standard applications and devices in order to ensure interoperability among different equipment suppliers. The following public application profiles have been released: Home Automation, Building Automation, Remote Control, Smart Energy, Health Care, Input Device, Telecom Services, Retail Services and 3D Sync [ZigBee11]. Private profiles represent non-standard applications and devices, i.e., non-standard application or device developed by vendors or created for private use. Figure 2.15 illustrates the definition of a ZigBee profile, which defines an enumeration of device identifiers and cluster identifiers.

The device identifiers provide information related to an endpoint. It may indicate, for example, whether the endpoint is an on/off lamp or an on/off switch.

Clusters represent application objects identified by a 16-bit identifier. Similar to those used in some object-oriented programming languages, clusters are application objects composed of attributes and commands. Attributes can represent the state of the object's variables and commands represent functions on these variables. For example, if a cluster represents a lamp, then its attributes can represent the current state of the lamp and the commands may represent on or off functions. The ZigBee cluster library (ZCL) is a document maintained by the ZigBee Alliance that describes cluster functionality. Depending on the profile, clusters may be mandatory or optional [ZigBee07].

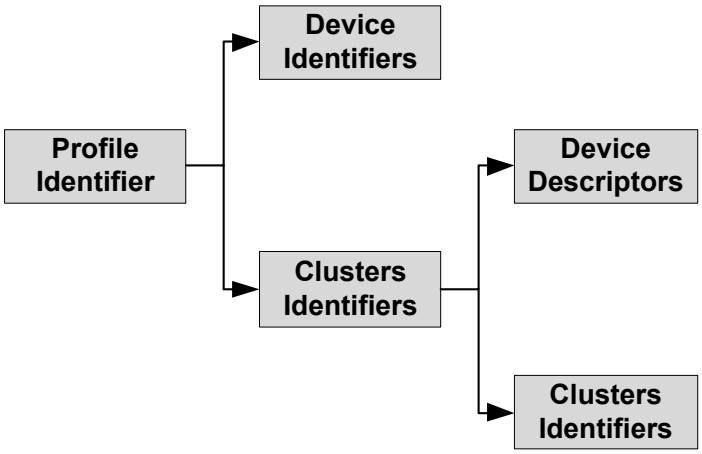


Figure 2.15 - Profile definition.

Each endpoint has a simple descriptor, which contains specific information about its profile identifier, device identifier and the supported clusters. Additionally, other four device descriptors are provided: the node descriptor, which provides information of the device capabilities, i.e., whether the device is a coordinator, a router or an end-device; the power descriptor, which informs if the device is battery powered and the current level of battery; the complex descriptor, which contains information of the manufacturer of the device; and the user descriptor, which contains a user defined string that may indicate the location of the device. The information provided by the descriptors can be used in network commissioning tools, in network services discovery and in binding services.

ZigBee Device Object

The ZDO is an application object running on the endpoint 0 in every network device. This application is responsible for managing the device on the network and provides an interface to the ZigBee device profile (ZDP). The ZDP is a special application profile that provides functionalities, such as discovering, configuring, and managing devices in the network and other network services. The ZDO is directly connected to the network layer and

controls when to form, join or leave the network, i.e., it is used as an interface to the network layer for applications. The ZDP allows the discovery of network devices, which may be used, for instance, when a node wishes to know information of other node. The ZDP also provides network services discovering, in the case where a device wishes to search for a particular service on the network, such as looking for a switch to control a lamp. Binding to a device on the network can also be acquired by the ZDP. This service allows an environment of transparency between applications running on different nodes. The ZDP can also provide mechanisms for network management. It contains a set of services with the purpose of obtaining information from other nodes; for example, know the routing table of a given node [ZigBee07].

2.3.3.4 ZigBee Versions Comparison

Table 2.9 compares some of the features provided by the various versions of the ZigBee protocol. The ZigBee 2004 stack is considered obsolete and is no longer in use. The three other stacks are very similar, which enables the compatibility between them. Although they have many common features, a particular stack may have more interest regarding a particular application.

Table 2.9 – ZigBee versions compared.

Feature	ZigBee 2004	ZigBee 2006	ZigBee 2007	ZigBee PRO
Frequency Agility	Yes	Yes	Yes	Improved
Fragmentation	No	No	Yes	Yes
Addressing	Hierarchical	Hierarchical	Hierarchical	Stochastic
Group Addressing	No	Yes	Yes	Yes
Routing	Tree;Mesh;	Tree;Mesh;	Tree;Mesh;	Mesh
Multicasting	No	No	No	Yes
Many-to-one routing	No	No	No	Yes
Source routing	No	No	No	Yes
Standard Security (AES 128 bit)	Yes	Yes	Yes	Yes
High Security	No	No	No	Yes

In the ZigBee 2004, 2006 and 2007 versions, the network coordinator selects the best available RF channel and PAN ID at startup time, while in the ZigBee PRO it is possible to detect channel failures due to channel interference and take measures to adopt a new operating RF channel and PAN ID (Frequency Agility).

Fragmentation is the ability to a device handle data transfers that are larger than the maximum payload size for a data frame.

In hierarchical addressing, addresses are assigned to devices based on tree schemes. In stochastic addressing, addresses are assigned randomly and a mechanism to avoid addresses conflicts is defined. The stochastic addressing mode is specified in the ZigBee PRO version, which increases the number of supported devices in the network.

When group addressing is used, devices can be assigned to groups and each group can be addressed with a single frame, thereby reducing network traffic for packets destined for groups.

Multicasting is a form of broadcast because it allows a device to transmit to many devices using a single packet. However, broadcast limits the packet transmission to the node circular radio range. With multicast, devices inserted into a multicast group may receive and relay the received packets to other devices in network that belongs to the same group.

Regarding the network security, all stacks provide AES encryption with 128 bits keys, but the ZigBee PRO provides the High Security mode. This mode requires Application Layer Link keys, peer-entity authentication and peer-to-peer key establishment using Master Keys.

2.4 Summary

Initially, this chapter presents a brief overview of the topic of wireless communications. Next, it presents some of the main characteristics of WBANs. Several WBAN and BSN architectures are addressed and reviewed. These can be used to the development of healthcare monitoring applications in different situations and environments, including indoor and outdoor monitoring. Then, it describes the ZigBee and IEEE 802.15.4 protocols, which are standard-based protocols used in the work developed and documented in this thesis. All protocol versions are addressed and the main differences between them are pointed out.

Chapter 3

Evaluation Setup and Models

The ZigBee protocol stack, which is built upon the 802.15.4 standard [IEEE4-03], is a widespread adopted protocol in WSN applications and is used as an alternative in healthcare applications. In [López11], the author presents a performance analysis based on simulation and field tests, using the ZigBee 2004 specification, for a vital signs monitoring application with data-intensive and delay-sensitive traffic requirements. The evaluation was performed for star and tree network topologies.

This chapter describes the setup of all the experimental tests that were performed to evaluate the performance of a WBAN using the ZigBee 2007 specification. It includes a theoretical evaluation that aims to predict the behavior of a ZigBee-based WBAN so it may help to detect anomalies in experimental results, which are provided in the next chapter. This chapter also describes the hardware and software platforms used to perform the evaluation of the communication protocols for the ZigBee specification in the context of WBANs. This was achieved using the Texas Instruments CC2530 development kit and the Texas Instruments ZigBee implementation, the Z-Stack. Various relevant QoS metrics were evaluated, namely the maximum throughput of the network, the network delivery ratio (DR) and the network delay.

A parametric model based on software delay was created to enable simulation results to be closer to those obtained in real experiments. An analysis of clock drift was also performed, which resulted in the creation of a model to predict its influence on the network behavior. In addition, an analysis of network performance in the presence of hidden-nodes was done, and an algorithm was developed to handle this situation, since a mechanism for this purpose is not

available on the IEEE 802.15.4 protocol. Finally an assessment of the effects of the human body in the performance of the network was made using devices from a WBAN system for posture monitoring, where measurements of received power and network delivery ratio at the coordinator were performed

3.1 Experimental Evaluation Platform

This section presents the hardware and software platforms used to obtain the results presented in this work, respectively, the CC2530 development kit and the ZigBee 2007 protocol software implementation, the Z-Stack. These platforms are both provided by Texas Instruments.

3.1.1 Texas Instruments CC2530 Development Kit

The hardware test platform is based on the CC2530 [TICC2530-10] System on Chip (SoC) integrated circuit (IC), which is also provided by Texas Instruments. This SoC integrates into the same chip a microcontroller and a transceiver compatible with the IEEE 802.15.4 standard, thus enabling the possibility of development of smaller sensor devices. The CC2530 operates in the 2.4 GHz frequency band and offers a data rate of 250 kbps, the maximum data rate defined by the IEEE 802.15.4. Figure 3.1 shows the main components of the development kit: the SmartRF05EB board (Figure 3.1 (a)), which provides several peripherals to the user, such as LCD, LEDs, UART, SPI, USB, joystick and buttons; and the CC2530EM module (Figure 3.1(b)), which contains the CC2530 chip (Figure 3.1(c)).

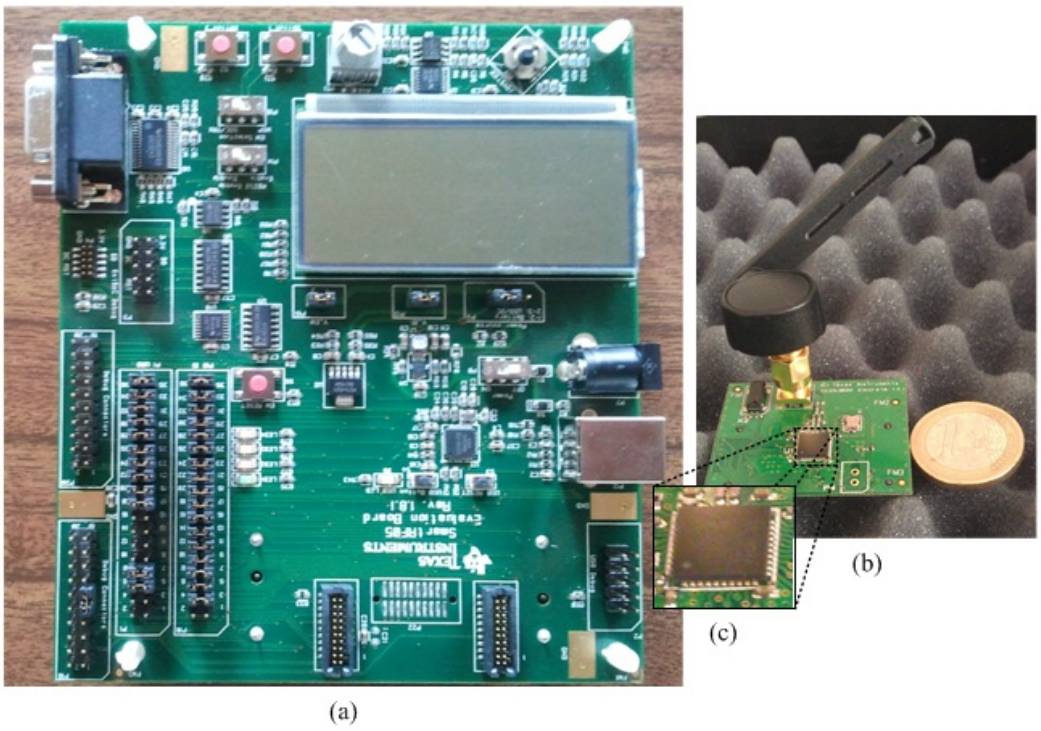


Figure 3.1 – Texas Instruments SmartRF05EB board (a), the CC2330EM module (b) and the SoC CC2530 unit (c).

The CC2530 [TICC2530-10] includes a set of functionalities that provides support for the IEEE 802.15.4 protocol, such as: automatic generation of the PHY preamble, automatic generation and verification of the packet 16-bit CRC, a CCA indicator for the last 8 symbols detected through the received signal strength indicator (RSSI) and automatic AES encryption/decryption.

The radio transceiver contains a processing core to automate procedures, in parallel with the microcontroller, enabling the possibility to process packets from or to the network while other packets may be received or transmitted by the radio. Furthermore, the radio core is capable of filtering and recognizing addresses in incoming packets so that it can reject packets not addressed to the device, decreasing the processing load in the microcontroller. The maximum output power of the transmitter can be programmable up to 4.5 dBm and the receiver sensibility is approximately -97 dBm.

The CC2530 includes a high-performance and low-power CPU core based on the 8051 microcontroller. Instructions execute faster than the standard 8051 because it is used one clock per instruction cycle instead of the 12 clocks per instruction cycle in the standard 8051. The version of the CC2530 used in this work includes 256 Kbyte of flash memory, 8 Kbyte of RAM and an extended 18 interrupts source. The CC2530 defines five different operating modes in order to save energy and to comply with the low-power requirements of the PAN

devices. Those operating modes are called active, idle, PM1, PM2 and PM3. The active mode is the normal operating mode and where most energy is consumed: up to 29 mA when transmitting at 1 dBm with the CPU idle. In contrast, the PM3 mode is where less energy is spent, typically consuming 0.4 μA when the CC2530 CPU core, the radio transceiver, and other components of this SoC are idle.

In addition, the CC2530 provides a set of peripherals, among which we emphasize:

- An IEEE 802.15.4 MAC timer (TIMER 2), a dedicated timer to be used by the IEEE 802.15.4 layer, and general purpose timers: one 16-bit timer (TIMER1) and two 8-bit timers (TIMER3 and TIMER4);
- A 32 kHz Sleep Timer that is used to set the period during which the system enters and exits a low-power sleep mode;
- Battery monitor and temperature sensor;
- A 12-bit ADC with eight channels and configurable resolution;
- Two USART's with support to UART and SPI modes.

3.1.2 Texas Instruments Programming Environment

The experimental platform used to produce the results presented in this thesis was developed and tested using the ZigBee and IEEE 802.15.4 stack implementations provided by Texas Instruments, a leading supplier of ZigBee products, the Z-Stack and TIMAC, respectively.

The Z-Stack version used in this work is the Z-Stack-CC2530-2.4.0-1.4.0 and it supports the two stack profiles of the ZigBee 2007 specification: ZigBee and ZigBee Pro. The Z-Stack-CC2530-2.4.0-1.4.0 is a combination of the ZigBee stack implementation version 2.4.0 and the IEEE 802.15.4 stack implementation version 1.4.0: the TIMAC-CC2530-1.4.0, which is also provided by Texas instruments. TIMAC-CC2530-1.4.0 features include support for the IEEE 802.15.4-2006. Some of the experiments provided in this work use only TIMAC, regardless of the Z-Stack, due to some limitations in implementation of the latter, such as the inability to support beacon-enabled networks. The standalone TIMAC version used in this work was the TIMAC-CC530-1.3.1. Figure 3.2 shows the stack architectures of the Z-Stack-CC2530-2.4.0-1.4.0 and the TIMAC-CC530-1.3.1.

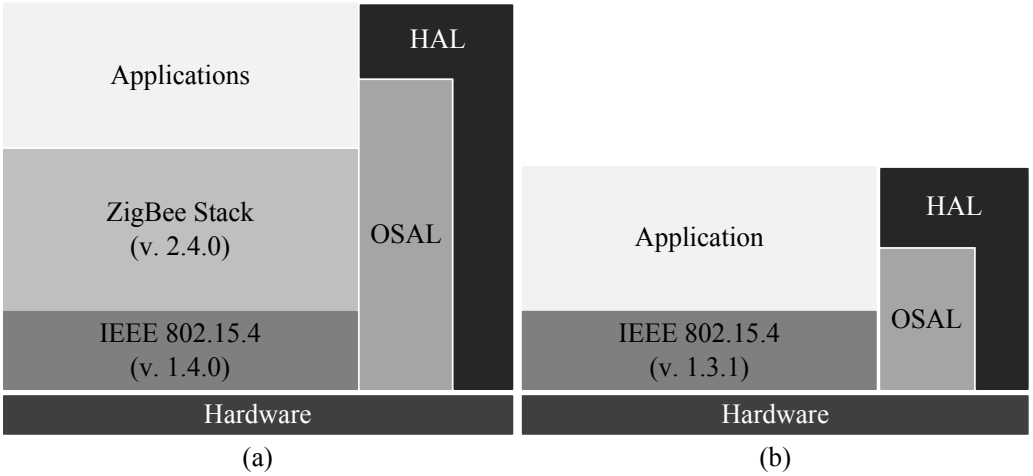


Figure 3.2- Z-Stack (a) and TIMAC (b) architectures.

The Z-Stack software is organized on the following components: OSAL (Operating System Abstraction Layer), HAL (Hardware Abstraction Layer), ZigBee and IEEE 802.15.4 Stack, Applications, and MT (Monitor and Test) interface. The TIMAC components are the OSAL, HAL, IEEE 802.15.4 and Application. The OSAL consists on the operating system and is used to control all the running tasks and to provide the API for communication and synchronization between tasks. The HAL provides a set of drivers to access all available peripherals. The ZigBee and IEEE 802.15.4 stack layers provide the implementation of the ZigBee 2007 protocol layers. The Application component refers to the set of applications running on the device. While Z-Stack has support for up to 240 applications, TIMAC only supports a single application. A device may be controlled by one of the Texas Instruments PC test tools, so the MT component provides an interface between these tools and the device.

The OSAL provides an API for communication and synchronization between tasks [TI_OSALAPI09]. It encapsulates all the system tasks to enable scaling, processing time, simplify the management of messages and events and improve the process of memory management. The OSAL task scheduler, executed in the *osal_run_system()* function, uses a task array (*taskArr[]*) in which all the system tasks must be inserted so they can be processed. In addition to the system task array, the OSAL defines an events array (*tasksEvents[]*) that contains information about all the events generated for the associated system tasks[1]. The events for a task indexed in *taskArr[t]* are found in *tasksEvents[t]*, which entries are 16-bit bitwise variables where each bit identifies a particular event. When a system task is processed, the processing period will end only when all the events for that task have been

[1] Events may contain associated messages/packets.

processed. The OSAL scheduler algorithm is shown in Figure 3.3. The index of a system task in the task array defines its level of priority where the task indexed in the position 0 is the highest priority task.

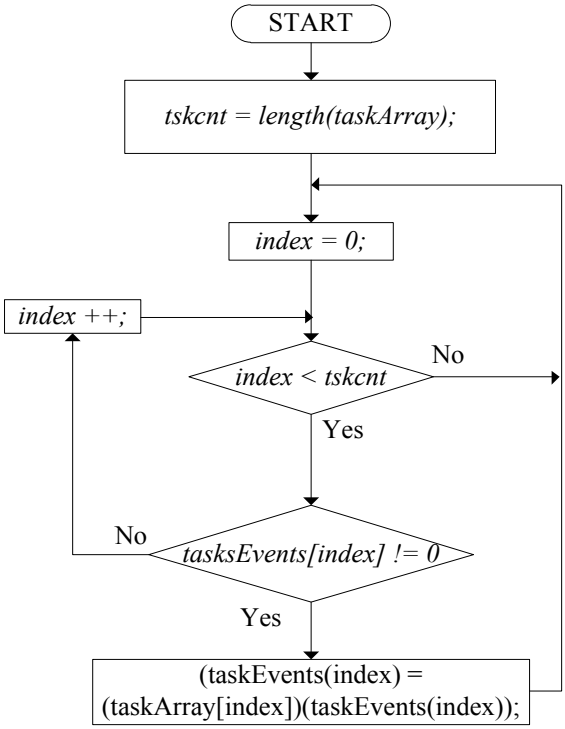


Figure 3.3- OSAL scheduler algorithm.

In Z-Stack, the system tasks and their priority levels are defined as follows, from the highest priority task (index 0 in the task array) to the lowest priority task:

- *taskArr[0]* - macEventLoop - task that manages all the events of the MAC layer;
- *taskArr[1]* - nwk_event_loop - task that manages all the events of the NWK layer;
- *taskArr[2]* - Hal_ProcessEvent - task that manages hardware interruptions;
- *taskArr[3]* - MT_ProcessEvent (optional) - manages events directed to an application entitled "Monitor and Testing", which can be used to network management via a computer with the help of tools provided by Texas Instruments, e.g., ZTOOL;
- *taskArr[4]* - APS_event_loop - task that manages all the events of the APS layer;
- *taskArr[5]* - APSF_ProcessEvent (optional) - task that manages all the events of the APS layer if fragmentation is used in the transmitted packets;
- *taskArr[6]* - ZDApp_event_loop - task that manages all the events of the ZDO;
- *taskArr[7]* - ZigBeeAppLevelTask1_ProcessEvent – task that manages all the events of a ZigBee applications registered as ZigBeeAppLevelTask1;

- *taskArr[n]* – At this point, up to 240 ZigBee applications can be registered in the system.

TIMAC excludes the nwk_event_loop, MT_ProcessEvent, APS_event_loop, APSF_ProcessEvent and ZDApp_event_loop tasks because they are exclusive from ZigBee. TIMAC supports only one application.

To create and register an application, both the initialization function and events processing function must be defined. Then, the initialization function must be registered in the system initialization function (*osalInitTasks()*), where all the initialization functions of the system tasks are registered, and the events processing function must be registered in *taskArr[]*. While the initialization function is used for the system to initialize all the task variables and parameters, the events processing function represents the task itself, because, when a new event is defined for a task, the respective events processing function is called by the OSAL scheduler.

The OSAL API [TI-OSALAPI09], as referred above, provides timer services. The timer service may be used to set a new event after a waiting period, through the *osal_start_timerEx()* and *osal_start_reload_timer()* API functions. These functions store the parameters in a list of timers, namely the timeout and the event identification to be triggered when the timer finishes. Due to hardware constraints, the timer mechanism is based on the system clock. Whenever the scheduling algorithm runs, the timers in the list using the *osalTimeUpdate()* function are updated. This function compares the timeout value of each timer registered in the list with the value of time elapsed since they were inserted.

As described above, the applications are registered in order to be managed by the OSAL. The stack provides an application level API that allows applications to transmit data packets. Next, we discuss Z-Stack and TIMAC application level transmission functions and events management.

The Z-Stack application framework API [TI-Z-SatckAPI09] provides a function to the application so it may transmit packets to the network: the *AF_DataRequest()*. This function takes as parameters the data to send (payload), the destination address of the packet, the target application in the receiver and some transmission settings. Among the transmission settings stands out the AF_ACK_REQUEST option. This allows the application to know if the packet was transmitted correctly to the destination. With this option, a message will be generated by the system for the application (*AF_DATA_CONFIRMATION_CMD*) indicating the status of

the transmission, i.e., whether the packet arrived at the destination or if the transmission has failed. The whole process of management of the transmission is done by the lower layers of ZigBee, which are implemented in other system tasks. If this option is set, an end-to-end acknowledgment is transmitted by the APS layer of the receiver to indicate that the package was successfully delivered to the destination. The value returned by this function indicates whether the request was accepted by the receiving system task, more specifically the APS_event_loop.

Packet transmissions in TIMAC, which are handled by the *nwk_event_loop* task in Z-Stack, are set in the application task through the *Mac_McpsDataReq()* function [TI_MACAPI09], which is provided by the MAC layer API.

Task events may be set either by the *osal_msg_send() and osal_set_event()* functions or through the timer functions. The system events, generated by the tasks of the lower layers of the stack, are identified as event 0x8000, leaving the remaining 15 bits available for user defined events. Then, system events are differentiated by a data message, available through the *osal_msg_receive()* function.

Z-Stack application level system messages include:

- KEY_CHANGE - Message generated when any button is pressed on the SmartRF05EB evaluation board. The message data indicate which button was pressed;
- AF_INCOMING_MSG_CMD (Z-Stack only) - Message generated when a data packet arrives from the network;
- AF_DATA_CONFIRMATION_CMD (Z-Stack only) - Message generated when there is an acknowledgment indication of a previous packet transmission. This message is generated whenever the application requires the transmission of a packet and its significance depends on the parameters used at the time of the request. If the AF_ACK_REQUEST option was set[1], the message data indicate if the packet was properly delivered to the destination or not. Otherwise it indicates if the packet was successfully or unsuccessfully delivered to the next hop in the network;
- ZDO_STATE_CHANGE (Z-Stack only) - Message generated when the status of the device changes in the network. The message data may indicate whether the device has

[1] It implies that, in addition to the MAC level acknowledgments exchanged in the relaying of a packet along all hops of the network, the receiver device confirm the reception of the packet by transmitting an APS level acknowledgment to the source.

just associated to the network or has lost its connection to the network.

TIMAC application level system messages, which are handled by the nwk_event_loop task in Z-Stack, include:

- MAC_MLME_ASSOCIATE_IND - Message generated when a device wants to join the network. This message is only received and processed by the network coordinator;
- MAC_MLME_ASSOCIATE_CNF - Message generated in the network devices indicating a successful join to an IEEE 802.15.4 network;
- MAC_MLME_START_CNF - Message generated in the network coordinator when the MAC layer successfully creates an IEEE 802.15.4 network;
- MAC_MCPS_DATA_CNF - Message generated indicating a previous transmission result. The message data defines whether the transmission was successful or not;
- MAC_MCPS_DATA_IND - Message generated when a data packet arrives from the network.

3.2 QoS Metrics Analysis

This section describes the setup of the experimental performance evaluation of BSN applications performed in this work, with particular emphasis on periodic traffic and data-intensive BSN scenarios using ZigBee and IEEE 802.15.4 networks. Three relevant QoS metrics are considered: maximum goodput, which represents the maximum application level throughput, delivery ratio (DR), which is the ratio of the number of successfully delivered packets to the number of packets generated by the source node application, and the end-to-end delay, which is the time elapsed since the packet is sent from the source node application layer until it reaches the destination node application layer.

The performance of ZigBee and IEEE 802.15.4 were evaluated in two different scenarios. In the first scenario, the maximum goodput supported by Z-Stack was measured and compared with a theoretical model. The main purpose was to evaluate the effect of the overhead introduced by both the protocol and the stack implementation in the throughput provided to the application. In the second scenario, the delivery ratio and the maximum and mean delays were measured in the scope of a motion capture application, in order to observe the behavior of the ZigBee and IEEE 802.15.4 protocols when subjected to data intensive

applications.

These scenarios were evaluated on both star and 2-hop tree topologies in a ZigBee network operating on channel 26. This channel was chosen due to the absence of interference from nearby Wi-Fi networks, verified using a spectrum analyzer. In the star topology, the end devices transmit the packets directly to the network coordinator. In the 2-hop tree topology, end devices transmit the packets to a router, which then relays the received packets to the network coordinator.

The IEEE 802.15.4 parameters and the respective values that were used in the experimental tests are specified in Table 3.1. Default values were used for the IEEE 802.15.4 varying parameters. The overhead introduced by all ZigBee layers in the evaluation scenarios accounts for a total of 264 bits. All tests finish after the coordinator has received 5000 packets from the end devices. The tests discussed in the QoS metrics analysis use the ZigBee Pro stack profile, but the same tests were performed using ZigBee stack and the results show no significant differences. The periodic ZigBee link status messages and IEEE 802.15.4 data requests commands were disabled. The tests were made in the absence of hidden nodes, since all network devices were in the radio range of each other.

Table 3.1- Parameters common to all experimental tests.

Parameter	**Value**
Maximum number of backoff periods that CSMA-CA shall execute until declares channel access failure. (*macMaxCSMABackoffs*).	4
Minimum value of the CSMA-CA backoff exponent. (*macMinBE*).	3
Maximum value of the CSMA-CA backoff exponent. (*macMaxBE*).	5
Number of symbols forming a unit backoff period (*aUnitBackoffPeriod*). A symbol corresponds to 16 μs.	20
Maximum number of retransmissions allowed by the 802.15.4 MAC layer after a transmission failure. (*aMaxFrameRetries*).	3
IEEE 802.15.4 Acknowledge frame size.	88 bits
IEEE 802.15.4 Acknowledge frame transmission period (T_{ACK}).	352 μs
Zigbee and IEEE 802.15.4 overhead.	264 bits
ZigBee 2007 profile.	ZigBee Pro
IEEE 802.15.4 channel.	26
IEEE 802.15.4 turnaround time (T_{TAT}).	192 μs
Addressing mode	unicast
Number of packets the network coordinator receives until the experiment ends.	5000

3.2.1 Maximum Goodput Analysis

This section presents a model to obtain the maximum theoretical goodput for the scenario of a single end device transmitting data to a coordinator in a non-beacon enabled ZigBee network. This model is used in the next chapter to evaluate the effect of the overhead introduced by the stack implementation in the throughput provided to the application, by comparing the model results with experimental results.

3.2.1.1 Maximum Goodput Model

The model presented in this section defines the maximum theoretical goodput as:

$$Goodput = \frac{Payload\ \ Length\ (bits)}{Average\ Transmission\ Period}. \quad (3.1)$$

The payload length represents the total length of the application level data, while the average transmission period is the average period needed to transmit a packet in a non-beacon enabled ZigBee network, discussed next.

Figure 3.4 represents the model that was used and the associated times for the transmission of a packet using the unslotted CSMA-CA algorithm of the IEEE 802.15.4 standard in a non-beacon enabled star network topology. The transmission period is constituted by a random backoff interval ($T_{Backoff}$), the transceiver turnaround time (T_{TA}) from RX to TX, the packet transmission time (T_{Packet}), a turnaround time from TX to RX and the ACK transmission time (T_{ACK}). The packet contains the payload and also the overhead introduced by the ZigBee stack. The CCA period, which is used to verify the channel status immediately after the backoff period, is not taken into account because the CC2530 maintains an updated CCA status function, indicating the status of the channel in the last 8 symbol period. For the 2-hop tree network topology, it is assumed that the transmission period is the double of the value obtained for the star topology, due to the packet being relayed from the router to the coordinator.

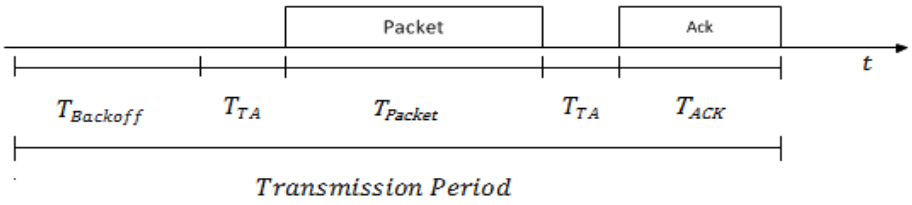


Figure 3.4- IEEE 802.15.4 associated times.

The turnaround time is defined in the IEEE 802.15.4 standard and corresponds to 192 µs. The ACK transmission time is 352 µs and the packet transmission time can be obtained through the following equation:

$$T_{Packet} = \frac{Packet\ Length\ (bits)}{Network\ Data\ Rate\ (bits/s)}, \quad (3.2)$$

where the IEEE 802.15.4 network data rate corresponds to 250 kbit/s [IEEE4-06] and the packet length corresponds to the total length of the transmitted packet, comprised by the payload length plus 264 bits due to the ZigBee protocol overheads.

The average transmission period is calculated using the mean backoff interval ($\overline{T_{Backoff}}$). The mean backoff period depends on the backoff exponent (BE) and its values are presented in Table 3.2. BE is equal to *macMinBE* (3) in the first iteration of the CSMA-CA algorithm, thus, assuming the end device guarantees access to the medium in the first iteration (which is true when there is a single transmitting device in the network and there are no sources of interference capable to introduce errors in the transmitted packets), the mean backoff interval used in the calculation of the maximum network goodput is 1.12 ms.

Table 3.2 – Mean backoff interval in the CSMA-CA.

BE	Backoff Interval (UBPs): from 0 to (2^{BE} - 1)	Mean Backoff Period (UBPs)	Mean Backoff Period (ms)
0	CSMA-CA disabled	---	---
1	0 to 1	0.5	0,16
2	0 to 3	1.5	0,48
3	0 to 7	3.5	1,12
4	0 to 15	7.5	2,4
5	0 to 31	15.5	4,96

Therefore, the average transmission period is:

$$Average\ transmission\ Period = \overline{T_{Backoff}} + T_{TAT} + T_{Packet} + T_{TAT} + T_{ACK}. \quad (3.3)$$

Figure 3.5 shows the results for the maximum theoretical goodput for the Star (Star - Theoretical) and the 2-hop tree (Tree - Theoretical) network topologies, obtained through the model. The network goodput increases as the packet's payload increases, but in the 2-hop tree topology it is half the value obtained from the star topology because the transmission period is twice the value. Clearly, the maximum network goodput is well below the network bit rate provided by the IEEE 802.15.4 (250 kbit/s). The maximum values, obtained for a 90 byte

payload, were 124.3 kbit/s and 62.2 kbit/s for the star and 2-hop tree topologies, respectively. This is due to the overheads introduced by the ZigBee and the IEEE 802.15.4 protocols ($T_{Backoff}$, T_{TA}, T_{ACK} and the packet headers and footers).

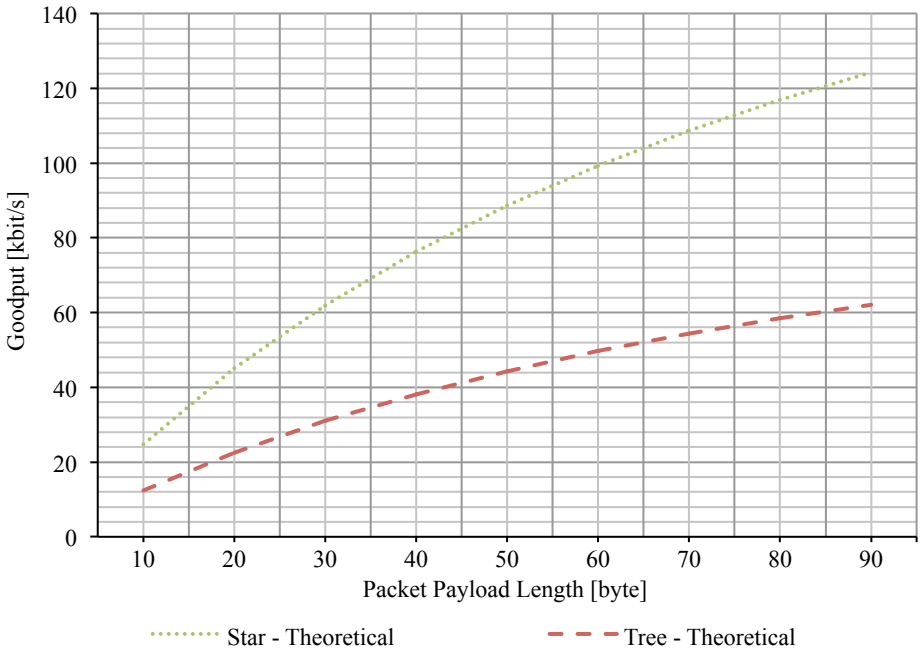


Figure 3.5- Maximum theoretical goodput for star and tree network topologies.

3.2.1.2 Experimental Evaluation Setup

The experimental evaluation scenario used to determine the maximum goodput is shown in Figure 3.6. A single end device transmits packets to the network coordinator. The coordinator receives the packets, calculates the result and then transmits it to a PC via RS-232, where it is presented. In the star topology the transmission is direct, while in the 2-hop tree topology the end device transmits packets to a network router (its parent), which relays the packets to the coordinator. In order to determine the maximum network goodput, the end device transmits packets in burst. For the experimental tests, two modes were implemented:

- **Mode 1** - the application layer generates and sends packets to the lower layer, one after another, as fast as it can. Since Z-Stack attributes higher priority to the tasks representing the lower layers of the ZigBee, the application will start the transmission of the next packet only when the previous packet transmission is fully processed by the software

stack[1].

- **Mode 2** - the application layer waits for the indication that the ACK has arrived before sending the next packet. Before the ACK reaches the application, all the intermediate ZigBee layers must process it, which may increase the time interval between packet generations.

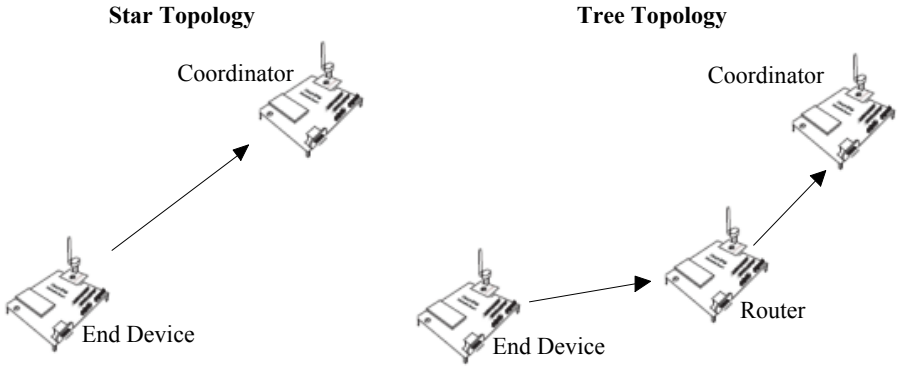


Figure 3.6 – Star and 2-hop tree experimental topologies.

3.2.2 Network Delivery Ratio and Delay Analysis

In this evaluation scenario, the delivery ratio and delay are analyzed in a contention environment where multiple end devices generate packets and send them to the coordinator simultaneously. These tests were performed with both Z-Stack and TIMAC to observe overall system behavior.

These experimental evaluations were performed in the scope of a BSN motion capture application where the end devices correspond to sensor nodes equipped with a set of magnetic and inertial sensors (section 2.2.2.5). Each sensor node transmits packets with the same amount of information, which is used to determine the 3D space orientation of the body segment where the device is attached. Two different traffic configurations used to test the network performance: mode A and mode B, as summarized in Table 3.3. In mode A, the packet length is 89 bytes and packets are transmitted in intervals of 200 milliseconds. In mode B, each packet has 62 bytes and the transmission period is set to 100 milliseconds. The packet length in mode A is larger because packets have to carry twice the number of motion capture

[1] After this, the packet is automatically transmitted by the radio of the CC2530. Then, the radio generates interruptions so the microcontroller may control the transmission mechanism. Consequently, this may increase the delay of the packet currently being processed in the ZigBee stack.

sensor samples. In both modes, the stack overhead is 33 bytes (264 bits).

Table 3.4 summarizes the network modes used to make the evaluation proposed in this work. Four different modes were used, namely, the Star_With_Ack and Star_Without_Ack modes, where the star network topology is tested with and without acknowledgements respectively. Similarly, in the Tree_With_Ack and Tree_Without_Ack modes, a 2-hop tree network topology is tested with and without the acknowledgment enabled, respectively.

Table 3.3 - Traffic operation modes used in the delivery ration and delay experiments

Traffic Mode		
Mode Designation	**Tx Period (ms)**	**Packet Length**
A	200	89
B	100	62

Table 3.4 - Network operation modes considered in the delivery ration and delay experiments

Network Mode			
Designation	**Topology**	**Number of Hops**	**ACK**
Star_With_Ack	Star	1	Yes
Tree_With_Ack	Tree	2	
Star_Without_Ack	Star	1	No
Tree_Without_Ack	Tree	2	

3.2.2.1 Delivery Ratio Analysis

The delivery ratio represents the number of successfully delivered packets divided by the number of packets generated by the source node application. Numerous problems may affect the network DR, such as: failure to access to the medium during the execution of the CSMA-CA protocol, packet collisions due to inability of the CSMA-CA protocol to detect transmissions from hidden nodes in the network; or errors in received packets caused by channel interference introduced by nearby IEEE 802.11-based networks.

Figure 3.7 represents the normalized network load for networks that carry the traffic generated by the sensor nodes in modes A and B. The normalized network load represents the ratio between the amount of traffic generated by the sensor nodes and the network data rate (250 kbit/s). The dashed line in red represents the normalized network load for a network transmitting in mode A, while the purple dashed line represents the load for a network transmitting in mode B. In both modes, the network load grows linearly with the number of sensor nodes in the network and was obtained through the following relation:

$$Normalized\ Throughput\ = \frac{Packet\ Length\ (bits) \times N}{Transmission\ Period\ (s) * 250 \times 10^3}, \tag{3.4}$$

where N represents the number of sensor nodes in the network.

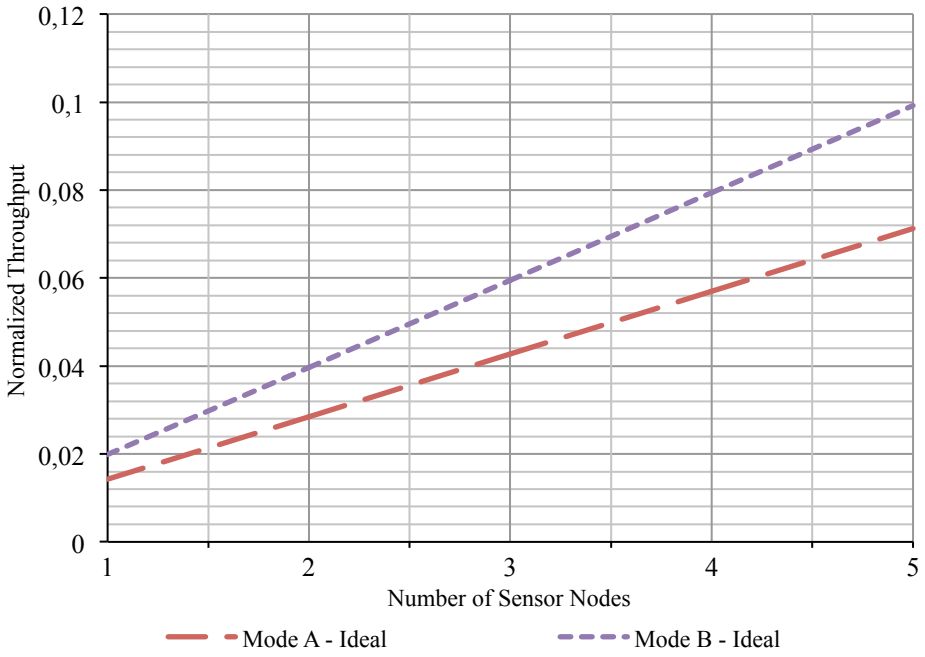


Figure 3.7 – Ideal normalized throughput for an increasing number of sensor nodes transmitting in modes A and B, in star and 2-hop tree topologies.

Observing the normalized network load in Figure 3.7, we may conclude that it is well below the network data rate, reaching only 7.1% and 9.9% for a total of five sensor nodes transmitting in mode A and mode B, respectively.

3.2.2.2 Delay Analysis

The end-to-end delay is the time that has elapsed since the moment when a packet it transmitted at the source application to the moment when it is received by the destination application. The following factors may influence the end-to-end delay:

- The delay introduced by the CSMA-CA mechanism of IEEE 802.15.4 MAC protocol;
- The delay introduced by MAC queuing mechanisms, which is manufacturer-dependent because the IEEE 802.15.4 standard leaves the queue buffer size definition to the vendors. In Z-Stack and TIMAC implementation, for the SoC CC2530, the MAC buffer size is set by default to 4 packets;
- The delay introduced by ZigBee software layers caused by the processing load and other existing buffering mechanisms.

This section presents an analysis of the delay at the MAC level, i.e., the delay introduced by the IEEE 802.15.4 MAC layer of the ZigBee protocol. This analysis aims to obtain the theoretical delay values in order to compare them with experimental results, so it may be possible to detect and correct abnormalities registered during the evaluation process, such as delays out of the theoretical range.

According to the model of the IEEE 802.15.4 transmission times described in section 3.2.1.1 (see Figure 3.4), the delay experienced by packets transmitted in a non-beacon enabled star network topology is constituted by the access delay induced by the CSMA-CA in the MAC layer ($T_{Backoff}$), the turnaround time (T_{TAT}), the packet transmission time (T_{Packet}) and the propagation time (considered negligible). If the acknowledgement is used[1], the minimum delay also includes a T_{TAT} and the acknowledgment packet transmission times (T_{ACK}).

The MAC attributes *aMaxFrameRetries, macMaxCSMABackoffs, macMinBe* and *macMaxBe* are set to their default values, presented on Table 3.1. T_{Packet} can be obtained through equation 3.2 and corresponds to 2.848 ms and 1.984 ms, for sensor nodes transmitting in modes A and B, respectively. Knowing this, the minimum delay that may be experienced by a packet in a star network topology correspond to a guaranteed access to the channel on the first transmission attempt at the end of a minimum backoff period ($T_{Backoff\_min}$ = 0 UBPs), with BE = 3 and NB = 0. This corresponds to a minimum delay of 3.040 ms and 2.176 ms, for end devices transmitting in modes A and B, respectively. Considering the acknowledgement, the minimum delay is 3.584 ms and 2.720 ms, for end devices transmitting in modes A and B, respectively. Table 3.5 shows how these values are determined.

Table 3.5 - Minimum delay experienced by a packet transmitted in mode A and mode B in a non-beacon enabled star network.

Transmission Attempts	Maximum Delay (ms)	
	Mode A	Mode B
1th Transmission Attempt	0	
+ 0 ms (1st CCA Succeeded ;NB=0;BE=3;$T_{Backoff\_muin}$=0UBPs)	0	
+ T_{TAT}	0.192	
+ T_{Packet}	3.040	2.176
* + T_{TAT}	3.232	2.368
* + T_{ACK}	3.584	2.720
TOTAL	3.040	2.176
**TOTAL*	3.584	2.720

** Considering the acknowledgment mechanism.*

[1] The receiver transmits the acknowledgment frame before the received data frame is delivered to the upper stack layer.

Table 3.6 shows the maximum delay that can be experienced by a packet transmitted in mode A and mode B in a non-beacon enabled IEEE 802.15.4 star network, considering the acknowledgment mechanism is enabled.

Table 3.6 – Maximum delay experienced by a packet transmitted in mode A and mode B in a non-beacon enabled star network.

Transmission Attempts	**Maximum Delay (ms)**	
	Mode A	**Mode B**
1st Transmission Attempt	0	
* + 2.240 ms (1st CCA failed;NB=0;BE=3;$T_{Backoff\_max}$=7 UBPs)	2.240	
* + 4.800 ms (2nd CCA failed;NB=1;BE=4;$T_{Backoff\_max}$=15 UBPs)	7.040	
* + 9.920 ms (3rd CCA failed;NB=2;BE=5;$T_{Backoff\_max}$=31 UBPs)	16.960	
* + 9.920 ms (4th CCA failed;NB=3;BE=5;$T_{Backoff\_max}$=31 UBPs)	26.880	
* + 9.920 ms (5th CCA failed;NB=4;BE=5;$T_{Backoff\_max}$=31 UBPs)	36.800	
* + T_{TAT}	36.992	
* + T_{Packet}	*39.840	*38.976
+ *macAckWaitDuration*	40.704	39.840
2nd Transmission Attempt (1st retry)	40.704	39.840
+ 2.240 ms (1st CCA failed;NB=0;BE=3;$T_{Backoff\_max}$=7 UBPs)		
+ 4.800 ms (2nd CCA failed;NB=1;BE=4;$T_{Backoff\_max}$=15 UBPs)		
+ 9.920 ms (3rd CCA failed;NB=2;BE=5;$T_{Backoff\_max}$=31 UBPs)		
+ 9.920 ms (4th CCA failed;NB=3;BE=5;$T_{Backoff\_max}$=31 UBPs)		
+ 9.920 ms (5th CCA failed;NB=4;BE=5;$T_{Backoff\_max}$=31 UBPs)		
+ T_{TAT}		
+ T_{Packet}		
+ *macAckWaitDuration*	81.408	79.680
3rd Transmission Attempt (2nd retry)	81.408	79.680
+ 2.240 ms (1st CCA failed;NB=0;BE=3;$T_{Backoff\_max}$=7 UBPs)		
+ 4.800 ms (2nd CCA failed;NB=1;BE=4;$T_{Backoff\_max}$=15 UBPs)		
+ 9.920 ms (3rd CCA failed;NB=2;BE=5;$T_{Backoff\_max}$=31 UBPs)		
+ 9.920 ms (4th CCA failed;NB=3;BE=5;$T_{Backoff\_max}$=31 UBPs)		
+ 9.920 ms (5th CCA failed;NB=4;BE=5;$T_{Backoff\_max}$=31 UBPs)		
+ T_{TAT}		
+ T_{Packet}		
+ *macAckWaitDuration*	122.112	119.520
4th Transmission Attempt (3rd retry)	122.112	119.520
+ 2.240 ms (1st CCA failed;NB=0;BE=3;$T_{Backoff\_max}$=7 UBPs)		
+ 4.800 ms (2nd CCA failed;NB=1;BE=4: $T_{Backoff\_max}$=15 UBPs)		
+ 9.920 ms (3rd CCA failed;NB=2;BE=5: $T_{Backoff\_max}$=31 UBPs)		
+ 9.920 ms (4th CCA failed;NB=3;BE=5: $T_{Backoff\_max}$=31 UBPs)		
+ 9.920 ms (5th CCA failed;NB=4;BE=5: $T_{Backoff\_max}$=31 UBPs)		
+ T_{TAT}		
+ T_{Packet}		
+ T_{TAT}		
+ T_{ACK}		
TOTAL	162.496	159.040

*non-acknowledged transmission.

In the limit, to a packet experience the maximum delay, first, the MAC layer must select the maximum possible number of backoff periods ($T_{Backoff\_max}$) in all the *macMaxCSMABackoffs* channel access attempts before the packet is transmitted; then, it must execute the maximum number of retransmissions attempts (*aMaxFrameRetries*) due to non-received acknowledgments[1]. The maximum delay is 162.496 ms for end devices transmitting in mode A and 159.040 ms for end devices transmitting in mode B. For a non-acknowledged transmission, the maximum delay is 39.840 ms and 38.976 ms with the end devices transmitting in mode A and mode B, respectively.

For the 2-hop tree network topology, it is considered that the minimum and maximum delay, with acknowledged and non-acknowledged transmissions, is twice as long as the results presented in Table 3.5 and Table 3.6. This is due to the packet being relayed from the router to the coordinator. The router makes use of the same model used by the end devices to transmit their packets.

3.2.2.3 Experimental Evaluation Setup

The DR and delay experimental tests were performed in a laboratory environment consisting in a set of end devices transmitting in modes A and B to the network coordinator. The coordinator collects the packets, measures the DR and the delay, and then transmits the results to a PC via RS-232, which presents the obtained results.

Figure 3.8 shows the topologies used to evaluate the performance of the networks. In the star topology, the end devices transmit the packets directly to the coordinator and in the 2-hop tree topology the end devices transmit to the router, which in turn relays the packets to the coordinator. The IEEE 802.15.4 standard does not define a network layer, so in TIMAC the router used for the 2-hop tree topology is simulated by using a peer-to-peer network where all end devices transmit the packets to a device, which then relays the packets to the coordinator.

A trigger signal controlled by the coordinator is used to generate a periodic interruption on the end devices according to the transmission period. The main objective of the trigger is to create a contention scenario where all the end devices try to access the medium at same time, which represents a worst-case scenario. For the delay test, an end device was designated

[1] The IEEE 802.15.4 standard [IEEE04-06] defines the *macAckWaitDuration* as the maximum period to wait for the arrival of an acknowledgment packet followed by a transmitted data frame, before a transmission failure is declared. This period correspond to 864 μs.

to be the reference for the measured values.

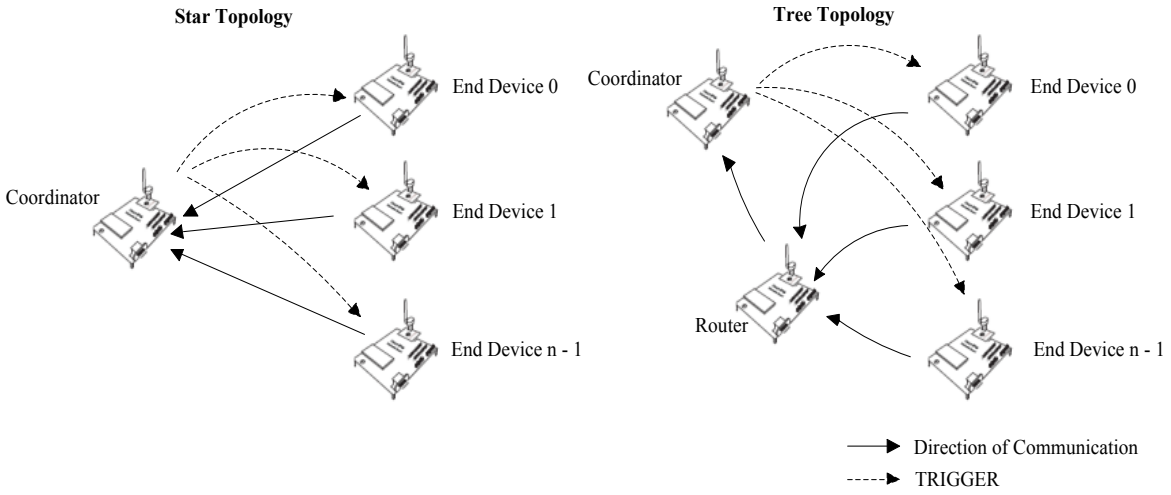


Figure 3.8 – Experimental configuration to measure the network delivery ratio and the delay in star and 2-hop tree topologies.

3.3 Clock Drift Analysis

Clock drift introduces a new problem to WBANs, particularly in non-beacon enabled networks that support applications that generate intensive and periodic traffic. Clock drift causes the de-synchronization of the sensor nodes and consequently the network performance may be degraded due to collision when packet transmission times of two or more devices start to approach. In beacon enabled networks, the clock drift may be minimized or even removed because the PAN coordinator constantly transmits a network beacon that can be used for synchronization.

Figure 3.9 illustrates the clock drift effect in beacon and non-beacon enabled networks. In the beacon enabled network, sensor devices may synchronize and transmit their packets. Therefore, the nominal transmission interval for each device n, T_{EDn}, is constant for applications that transmit periodic traffic during the nodes transmission lifetime. The network beacon may also suffer from the clock drift effect. Subsequently, it must be considered that $T_{EDn} = T_{Beacon} + T_{dBeacon}$, where T_{Beacon} is the nominal beacon transmission period and $T_{dBeacon}$ is the beacon time drift. Nevertheless, considering that all the associated nodes are perfectly synchronized with the beacon and separated in time, they do not contend for the network channel and therefore will not damage the network performance. In non-beacon enabled networks, the nodes transmission period is given by $T_{EDn} + T_{dEDn}$, where T_{EDn} is the nominal

transmission interval for node *n*, which is equal for all devices, and T_{dEDn} is the time drift occurred during the transmission interval, which depends on the device. Due to clock drift, the time interval between the transmission of the ED_1 and ED_2 nodes in the example is reduced, and it is increased relatively to the ED_3. In the course of time, ED_1, ED_2 and ED_3 will eventually contend for the wireless channel, interfering with each other transmissions, which may result in the network performance degradation.

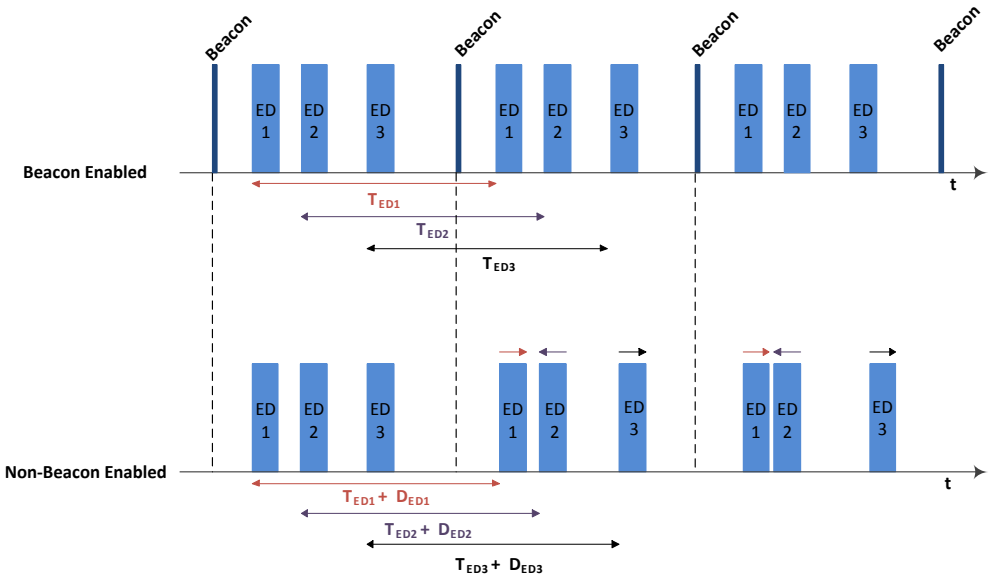


Figure 3.9 - Clock drift effects for periodic packet transmissions in beacon enabled and non-beacon enabled networks.

Next section introduces a model that evaluates the effect of clock drifts in a non-beacon enabled ZigBee BSN constituted by a group of end devices that generates periodic packets to the coordinator in a star network topology. Later, this model will be used to predict the clock drift effect on network performance using the measured sensor device's clock drift.

3.3.1 Clock Drift Evaluation

The clock drift effect in non-beacon enabled networks, especially on those supporting applications with highly intensive and periodic traffic, is of particular importance to evaluate. In this case, when the devices start to contend for the network channel, it is expected that the contentions may last for high amounts of time due to typical small clock drifts, which, consequently, may degrade the network performance. For instance, the CC2530 microcontroller datasheet [TICC2530-10] specifies a clock drift between ±40 ppm at 25ºC and with a voltage supply of 3V. The modeling of the network behavior caused by clock drift

will enable the creation of mechanisms to minimize this problem. In this work, measurements of clock drifts for a set of ZigBee devices were carried out and the results were introduced into the model to evaluate the clock drift effect in network performance.

In [López11], the author describes results obtained when observing the clock drift effect in the traffic generated by ECG wireless sensors in a non-beacon enabled ZigBee network. JN5139 modules were programmed to generate packets at 500 ms intervals. The author demonstrate the time drift through the measurement of packet timestamps, which allowed the conclusion that the order of the received packets from the sensor devices eventually change due to this clock drift. That is, if packets arriving from a device A are being followed from those transmitted form a device B, eventually, due to the clock drift, packets from node B will start to be received before packets from A. However, since the time drift was observed through the timestamp records of the received packets in a PC application, conclusions about the clock drift effect may not be very precise because the author did not measured the real clock drift of the network sensor devices The observed packets timestamps recorded in the PC application may have been influenced by other events, such as additional delays in the packet's timestamp introduced by other procedures on the PC operating system.

3.3.1.1 Clock Drift Measurement Setup

To measure the clock drift on end devices in the network, each end device EDn was connected to the coordinator in order to measure the differential clock drift between them. The differential clock drift between end device n (D_{EDn}) and the BS (D_{BS}) is given by:

$$D_{BS,EDn} = D_{BS} - D_{EDn}. \tag{3.5}$$

The differential clock drift between end devices ED1 and ED2 can be obtained from the respective differential clock drifts with relation to the BS:

$$D_{ED1,ED2} = D_{BS,ED1} - D_{BS,ED2} = D_{ED2} - D_{ED1}. \tag{3.6}$$

To obtain precise measurements of the $D_{BS,\,EDn}$ a hardware timer was programmed in the BS that toggles an output pin of the CC2530 generating a periodic signal of frequency $f = 0.2$ Hz on that pin. This pin was connected to an input pin on the device that we want to measure the clock drift. The signal generated by the BS was used to enable the reading of the value of a hardware timer in the end device. Both BS and end device timers were programed in the

same way to obtain the same timer frequency. Comparing the number of clock oscillations of the timer in the BS with the number of oscillations of the timer in the end device during the period $T = 1/f$, defined by the BS output pin signal, we obtain the number of oscillations ($ticks_{drifted}$) that are missing or that were added in the device due to the clock drift. Then, the $D_{BS,\,EDn}$ was calculated as:

$$D_{BS,EDn} = \frac{ticks_{drifted}}{f_{osc} \times T}, \tag{3.7}$$

where f_{osc} is the 32 MHz nominal clock frequency of the CC2530 microcontroller [TICC2530-10].

3.3.2 Clock Drift Model

This work proposes a model that uses the clock drift between ZigBee end devices to make an approximation of the interference periods during which the network devices will contend for the network channel and the intervals of repetitions of these periods. In this model it is assumed that nodes have the same nominal transmission period and the transmitted packets have equal lengths.

Several unsynchronized devices transmitting periodic traffic with the same nominal interval will eventually contend for the wireless channel due to the clock drift effect, even if they start transmitting at different instants of time. If the differential clock drift between the end device 1 and end device 2 is $D_{ED1,\,ED2}$ and the nominal transmission period is given by T_{ED}, then both nodes will contend for the wireless channel every T_{IntRep} seconds. This period, called the interference repetition interval, may be obtained through the next equation:

$$T_{IntRep} = \frac{T_{ED}}{D_{ED1,ED2}}. \tag{3.8}$$

The interference period T_{Int}, during which two devices will compete for the channel, can be obtained through the following equation:

$$T_{Int} = \frac{T_{Vul}}{D_{ED1,ED2}}, \tag{3.9}$$

where T_{Vul} represents the vulnerability time window. Figure 3.10 shows this vulnerability time window under which the transmissions of two nodes may interfere with each other. This interval is referred to as vulnerability window because it represents the time range where

transmissions from one device are vulnerable to collisions with the transmissions from the other device.

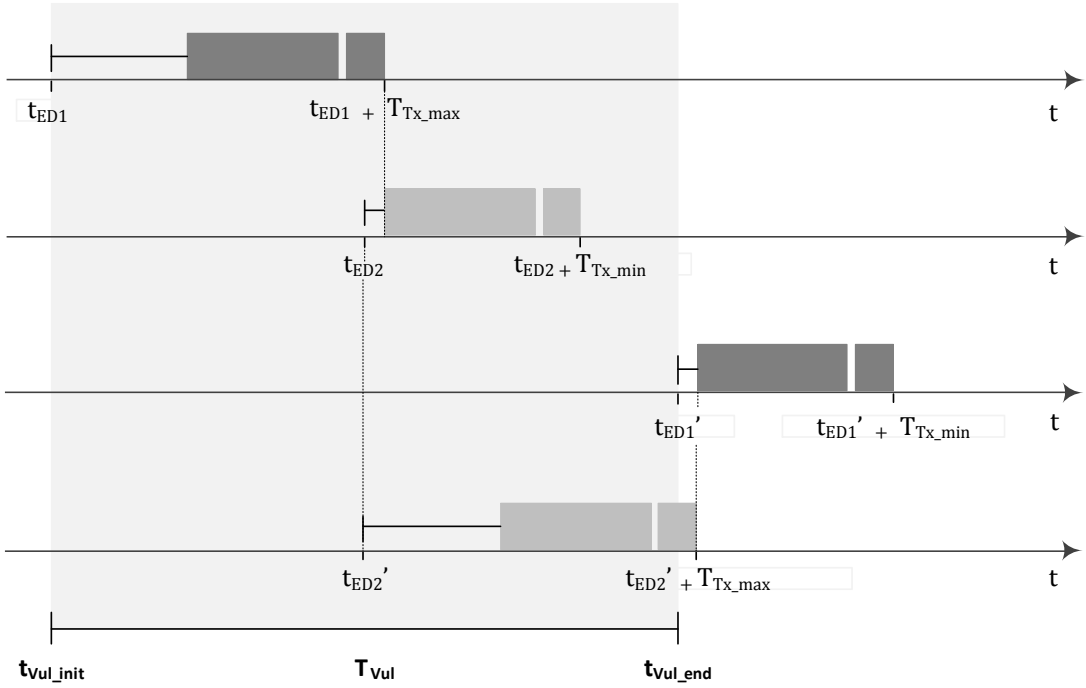


Figure 3.10 - Vulnerability window.

The interference period between devices *ED1* and *ED2* starts when:

$$t_{packet\_ED2} = t_{ED1} + T_{Tx\_max} \tag{3.10}$$

and ends when:

$$t_{packet_{ED1}}{}' = t_{ED2}{}' + T_{Tx\_max} \tag{3.11}$$

T_{Tx_max} represents the maximum period needed to transmit a packet and receive the respective acknowledgment; it can be calculated through equation 3.13. t_{EDn} is the instant of time where device *n* starts the IEEE 802.15.4 CSMA-CA algorithm, i.e., the start of backoff period, and t_{packet_EDn} is the instant of time when *EDn* starts to transmit the packet, which can be calculated through:

$$t_{packet\_EDn} = t_{EDn} + T_{Backoff} + T_{TA} \tag{3.12}$$

Figure 3.11 illustrates the time needed for the nodes to access the channel and transmit a packet for ZigBee/IEEE802.15.4 non-beacon enabled networks. This period (T_{Tx}) is

constituted by a random backoff interval ($T_{Backoff}$), the transceiver turnaround time (T_{TA}), the packet transmission time (T_{Packet}) and the acknowledgement transmission time (T_{Ack}).

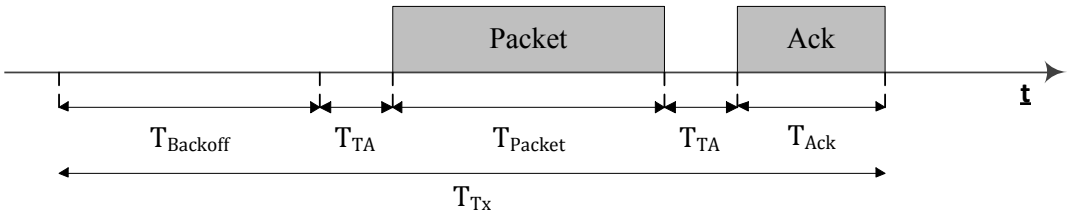


Figure 3.11 - ZigBee/IEEE 802.15.4 packet transmission associated times.

T_{Tx} is variable because it depends on a random period ($T_{Backoff}$) used by nodes to access the channel. T_{Tx_max} and T_{Tx_min} are given by equations 3.13 and 3.14, respectively, and represent the maximum and minimum period spent by a node for transmitting a packet and receiving the respective acknowledgment, respectively. $T_{Backoff\_max}$ and $T_{Backoff\_min}$ depend on a random backoff period, which is delimited by $2^{BE} - 1$ unit backoff periods (UBPs), where BE is the backoff exponent used by the CSMA-CA in non-beacon enabled ZigBee/IEEE802.15.4 networks.

$$T_{Tx\_max} = T_{Backoff\_max} + 2\ T_{TA} + T_{Packet} + T_{ACK} \tag{3.13}$$

$$T_{Tx\_min} = T_{Backoff\_min} + 2\ T_{TA} + T_{Packet} + T_{ACK} \tag{3.14}$$

In the model, the devices will start to interfere when

$$t_{ED1} + T_{Tx\_max} = t_{ED2} + T_{Backoff\_min} + T_{TA} \tag{3.15}$$

and the interference will end when

$$t'_{ED1} + T_{Backoff\_min} + T_{TA} = t'_{ED2} + T_{Tx\_max}, \tag{3.16}$$

where $t_{ED2}' = t_{ED2}$ because the clock drift of device ED2 is considered as absolute and it is used to derive the ED1 clock drift.

We obtain T_{Vul} from $t_{ED1}' - t_{ED1}$:

$$T_{Vul} = 2 \times (T_{Tx\_max} - T_{TA}). \tag{3.17}$$

This analysis does not consider the effect over T_{Vul} caused by the retransmission mechanism of the MAC layer, triggered in acknowledged transmissions when a packet collision occurs or errors in the packets are introduced by channel interference, which may result in a divergence between the results obtained in the model and the experiments. Retransmissions due to collisions will occur mainly due to the hidden node problem. Devices may backoff to avoid collisions, so the analysis does not consider further delays introduced by the carrier sense mechanism in the devices, which consequently may result in a new $T_{Backoff}$ period before the transmission, because this model was defined mainly for the case of an hidden node situation, in which the devices are more vulnerable to collisions.

If $D_{ED1,ED2} = 0$, and t_{ED1} and t_{ED2} are separated in time by $T_{ED1,ED2} > T_{Tx\_max}$, the devices will never contend for the wireless channel.

3.3.2.1 Clock Drift Model Validation Setup

The proposed model allows us to characterize the network so that we know for how long nodes will contend and what is the period between the contentions. For the default value $BE = 3$, $T_{Backoff\_min}$ is 0 UBPs and $T_{Backoff\_max}$ is 7 UBPs, which correspond to 0 ms and 2.240 ms, respectively. T_{TA} is defined by the IEEE 802.15.4 standard [IEEE4-06] and corresponds to 0.192 ms.

To validate our model, we evaluated a ZigBee network formed by two end devices that transmit packets of 62 bytes every 100 milliseconds to the coordinator in a star topology. Relevant IEEE 802.15.4 parameters are correspondent to those presented in Table 3.1. The packet transmission time is 1.984 milliseconds. No acknowledgment mechanism was used in this evaluation, so:

$$T_{Tx\_max} = T_{Backoff\_max} + T_{TA} + T_{packet}. \quad (3.18)$$

The validation process is done by estimating both T_{Int} and T_{IntRep} periods and comparing the estimations with the experimental measurements.

Figure 3.12 illustrates the experimental testbed used to measure the network delivery ratio, which allows us to observe the clock drift effect. To obtain the network delivery ratio, a window of 60 packets was used. In order to simulate a hidden-node situation, the experiment was performed in an anechoic chamber, metal plates separated the two end devices and the transmission power was reduced to -10 dBm. The acknowledgment mechanism was disabled,

so that there were no retransmissions. The coordinator collects the received packets and transmits them to a PC through the UART. The PC is placed outside the anechoic chamber and, once the test has finished, it calculates and presents the DR. The purpose of this experiment was to try to obtain more accurate results for the contention and non-contention periods. In a different scenario, if the two nodes could hear each other and if the retransmission mechanism were to be used, it would not be possible to observe and measure these periods.

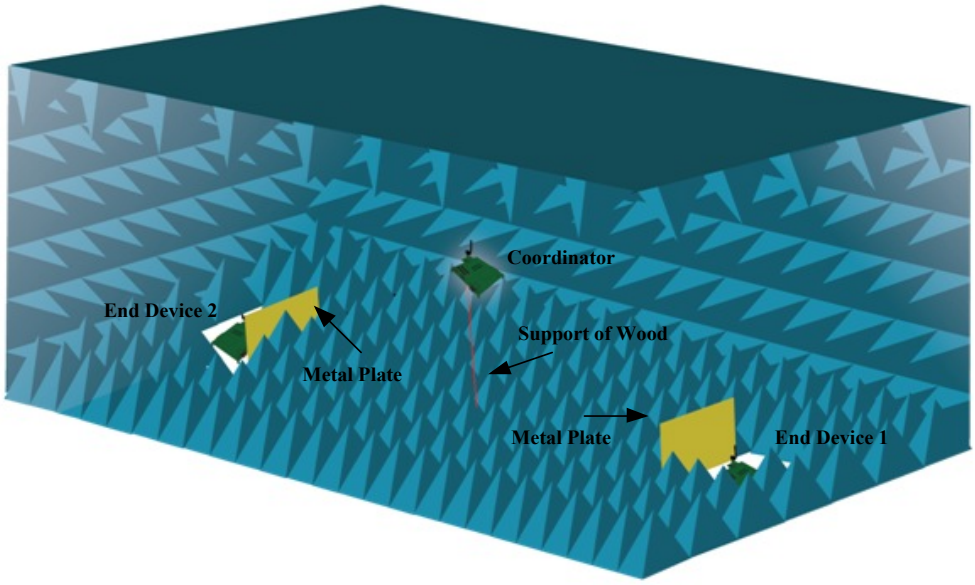


Figure 3.12 - Clock drift experiment test-bed in an anechoic chamber.

3.4 Hidden Node Analysis

In a network scenario composed by a coordinator and two associated nodes: A and B, when node A accesses the medium and it cannot sense that node B is transmitting a packet at the same time, or vice versa, this means that the two nodes are hidden from each other. The IEEE 802.15.4 standard provides a sensing procedure, the CCA mechanism, which is one of the main mechanisms to prevent collisions in carrier sense based networks. If this mechanism fails, the performance will degrade due to an increase in collisions. This degradation may be worse in the case of periodic and data-intensive WBAN applications, because nodes constantly transmit packets within the same interval, which will increase the probability of recurring collisions.

In this section, we discuss the experimental tests performed with ZigBee in the presence of hidden-nodes in order to evaluate the impact of the HNP in the network performance.

Knowing that WBAN applications demand specific QoS requirements, a solution to mitigate the HNP is necessary. To surpass this problem, we propose a solution to avoid the HNP in ZigBee/IEEE 802.15.4 networks: the HNPAvoidance protocol. This section also describes the setup of the experimental tests concerning the HNP.

3.4.1 Hidden Node Evaluation

This evaluation consists in two ZigBee end devices transmitting packets in mode B in a non-beacon enabled star network topology. In order to analyze the worst-case scenario, both nodes generate their packets simultaneously.

Figure 3.13 (a) illustrates the time period needed by nodes to access the channel and transmit a packet for non-beacon enabled ZigBee/IEEE 802.15.4 networks. In this particular case, the acknowledgment packet is not considered because it would turn this evaluation more complex due to the retransmissions mechanism. This period (T_{Tx}) is constituted by $T_{Backoff}$, T_{TA} and T_{Packet}. T_{Tx} is variable because it depends on $T_{Backoff}$, which is a random value delimited by $2^{BE} - 1$ unit backoff periods (UBP), where BE is the backoff exponent used by the CSMA-CA algorithm. Each UBP corresponds to 320 µs and T_{TA} is 192 µs. Figure 3.13 (b) shows the time boundaries of T_{Tx}. These boundaries, the minimum transmission time (T_{Tx_min}) and the maximum transmission time (T_{Tx_max}), depend, respectively, on $T_{Backoff\_min}$, which is the minimum backoff period of 0 UBPs, and on $T_{Backoff\_max}$, which is the maximum backoff period of 7 UBPs. T_{Packet} is 1.984 ms.

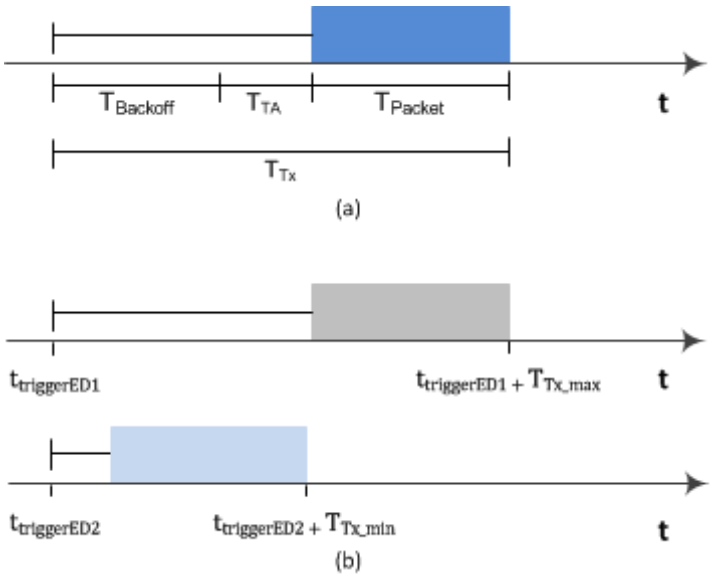


Figure 3.13 - Non-acknowledge IEEE 802.15.4 associated times (a) and its minimum and maximum time boundaries (b).

When the coordinator trigger sets a transmission event in both nodes ($t_{triggerEDn}$), the transmitted packets will not collide only if T_{Tx} of node ED1 is equals to the T_{Tx_max} and T_{Tx} of node ED2 is equals to the T_{Tx_min}, or vice versa. The probability for this specific case to occur (p_{TX}) can be obtained through the following equations:

$$p_{TX} = 2 \times p_{Backoff\_min} \times p_{Backoff\_max}. \tag{3.19}$$

$$p_{Backoff\_min} = P_{Backoff\_max} = {}^{1}/_{8} \tag{3.20}$$

Therefore, the probability of a successful transmission is only approximately 3.125%.

Through this theoretical analysis, we may conclude that, in a network composed by two end devices hidden from each other and in star network topology, the HNP may have serious implications in the network performance, especially when the acknowledgment mechanism is not used. In this case, the network performance is dependent on the probability of a successful transmission for a single attempt, which means that the theoretical network delivery ratio is 3.125%.

3.4.1.1 Experimental Evaluation Setup

Figure 3.14 illustrates the hidden-nodes experiment testbed. Two ZigBee end devices associated to a coordinator constitute the network configuration for this experiment. To simulate the HNP, the network was setup in an anechoic chamber to avoid the multipath effect; the nodes were separated by two metal plates and the transmission power was reduced, making it impossible for the end devices to sense each other. The coordinator was placed in such a way that it could communicate directly with both nodes.

The nodes transmit packets to the coordinator in mode B in the Star_With_Ack and Star_Without_Ack network modes (see Table 3.3 and Table 3.4 descriptions in section 3.2.2) and each test finishes when the coordinator receives 5000 packets. The coordinator collects the packets and then transmits them to a PC placed outside the anechoic chamber, through the UART. The PC calculates the network DR. The IEEE 802.15.4 parameters used in this experiment corresponds to those presented in Table 3.1. In order to observe the worst-case scenario, both nodes were synchronized using a trigger activated by the coordinator, which is used to set a new packet transmission event in the nodes.

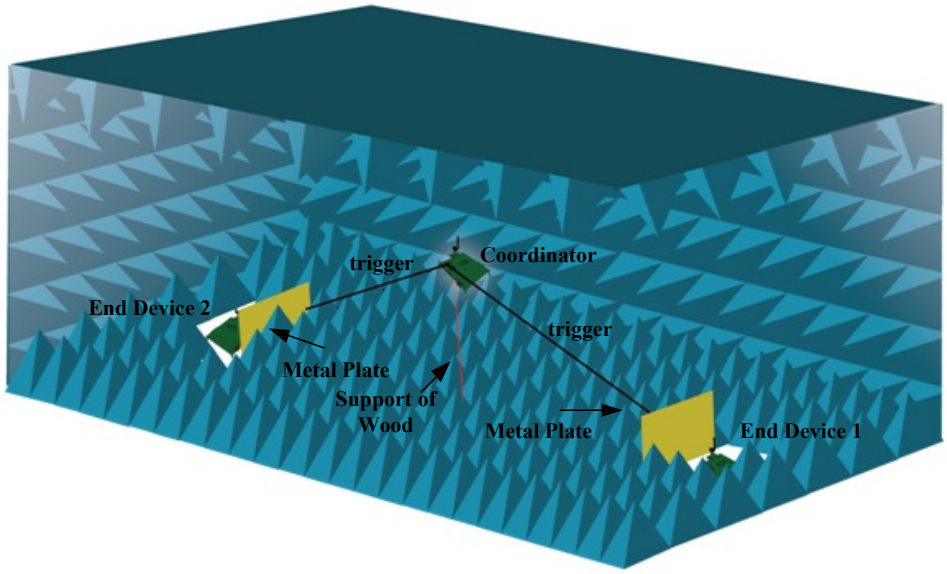


Figure 3.14 - Hidden-node experiment test-bed in an anechoic chamber

3.4.2 The HNPAvoidance Protocol

In this section, we present a solution to overcome the HNP in ZigBee/IEEE 802.15.4 networks: the HNPAvoidance protocol.

In general, WBANs are characterized by a network in which its nodes generate periodic and, in some cases, intensive traffic. Based on these characteristics, a solution to solve the HNP may be based on the synchronization of the network nodes, so they may transmit separated in time to avoid the packet collisions. If the network nodes are synchronized, both HNP and clock drift effect may be solved.

Several authors, as in [Koubâa09] [Kwon09] and [Hwang05], propose solutions to the HNP in ZigBee/IEEE 802.15.4 networks. These solutions are based on grouping strategies where the network nodes are grouped according with their hidden-node relationships. The network bandwidth is then divided into slots, each one attributed to a group. These strategies are usually very complex because grouping mechanisms use resource-intensive algorithms that may not be supported by some WBAN devices. The grouping and regrouping mechanisms used in these solutions when new hidden-nodes are introduced in the network may consume network bandwidth and cause packet transmission delays, which WBANs are usually intolerant to, because there are a series of procedures before nodes are correctly grouped. These procedures include the discovery of hidden node situations, group assignation and the notification of the grouping results to all the network nodes. Alterations to the IEEE 802.15.4 protocol, some of them at hardware level, are also required, in order to improve the

discovery of the hidden nodes, which may not be suitable for a ZigBee/IEEE 802.15.4 compliant solution. Furthermore, the mobility nature of many WBANs (for instance, a group of moving patients monitored wirelessly at a hospital) may increase the frequency of the grouping and regrouping events and consequently may cause a decrease in the network performance.

The HNPAvoidance protocol is a simple ZigBee/IEEE 802.15.4 application level algorithm, which has the main objective of mitigating the HNP. The great advantage of developing a ZigBee/IEEE 802.15.4 compliant mechanism without the constraints of modifying the protocol, more specifically, the IEEE 802.15.4 MAC layer, is to avoid further incompatibilities between different ZigBee systems. The proposed protocol uses the superframe structure of the IEEE 802.15.4 [IEEE04-06] standard, where a periodic beacon, transmitted by the MAC layer by the coordinator is used to synchronize the network nodes, avoiding some issues related to devices clock drift and mobility. The HNPAvoidance protocol explores the typical traffic configuration of WBAN applications, where devices usually transmit periodic data. So, the beacon is transmitted according with the device's transmission period.

Figure 3.15 illustrates the virtual superframe structure defined at the application level, where existing virtual time slots (VTSs) are assigned to the network's devices. In this illustration, the BO and SO parameter values [IEEE04-06] are equal (BO = SO), but a different configuration (SO ≤ BO) would also be valid. The reason for keeping BO and SO equal is to use the whole available bandwidth and increase the number of VTSs supported. The CAP occupies the whole superframe period because the HNPAvoidance should guarantee exclusivity to a device transmitting into its VTS, removing the need for the GTS mechanism of the IEEE 802.15.4 protocol. Otherwise, if the GTS mechanism is required, it must be defined a fixed period in the CAP for the VTSs, because the CFP is variable and depends on the number of GTSs.

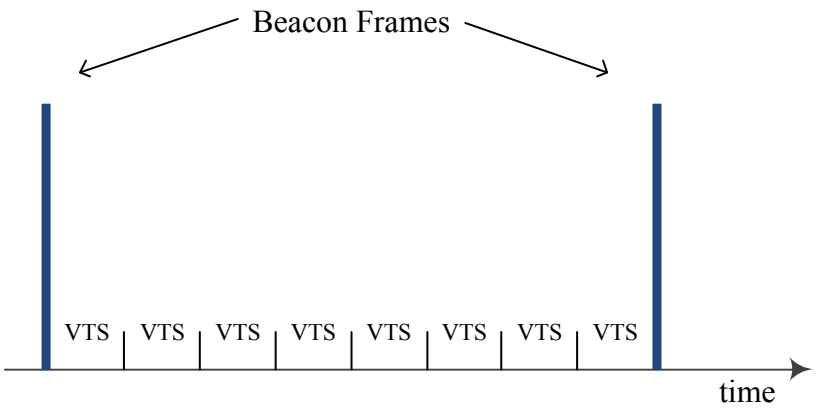


Figure 3.15 – HNPAvoidance application level virtual superframe structure.

The VTSs are an abstract concept to all the other components of the ZigBee/IEEE 802.15.4 stack. Our protocol creates a set of time slots at the application level that will be used by a node application to transmit its packets at the instant where the VTS that was assigned to it begins. The number of VTSs is defined in the application at the start of the network, and both network coordinator and nodes must know this value. In this particular example, the active period is divided into 8 VTSs, which corresponds to two IEEE 802.15.4 time slots[1] for each VTS. The number of VTSs may be configured to support all the network nodes.

HNPAvoidance is an algorithm coded in the application layers of the network coordinator and the nodes. The main functionalities of the algorithm in the coordinator application are:

- Assign VTSs to network nodes;
- Update the VTSs' state to unassigned, for those that are not in use anymore.

The algorithm in a node shall ensure that the node:

- Is synchronized with the network beacons;
- Transmit the packets in the VTS that was assigned to that node;

Figure 3.16 shows the HNPAvoidance algorithm in the coordinator application. When a data packet reception event occurs at the coordinator, and after the received packet is processed, the algorithm checks if the node that transmitted the packet has a VTS assigned. If

[1] The IEEE 802.15.4 superframe is divided into 16 equal time slots. The beacon is transmitted in the first slot of the superframe.

not, the algorithm searches in its application information base for an available VTS and assigns it to the node. Then, the coordinator transmits a Sync Packet to that node with the assigned VTS number, which identifies the VTS that the node will use to transmit its subsequent packets. Periodically, the VTS Update Event of the algorithm verifies if the VTSs are still in use, that is, if nodes are transmitting packets in the VTS that were assigned for them. If a node stops transmitting, the algorithm will release the VTS that was assigned to that node, so that the VTS can be available again. To keep the number of available VTSs updated, this event is executed every second, where a VTS status indicator defines if the VTS was used during the last second. The VTS status indicator is set when a packet is received and cleared at the VTS Update Event.

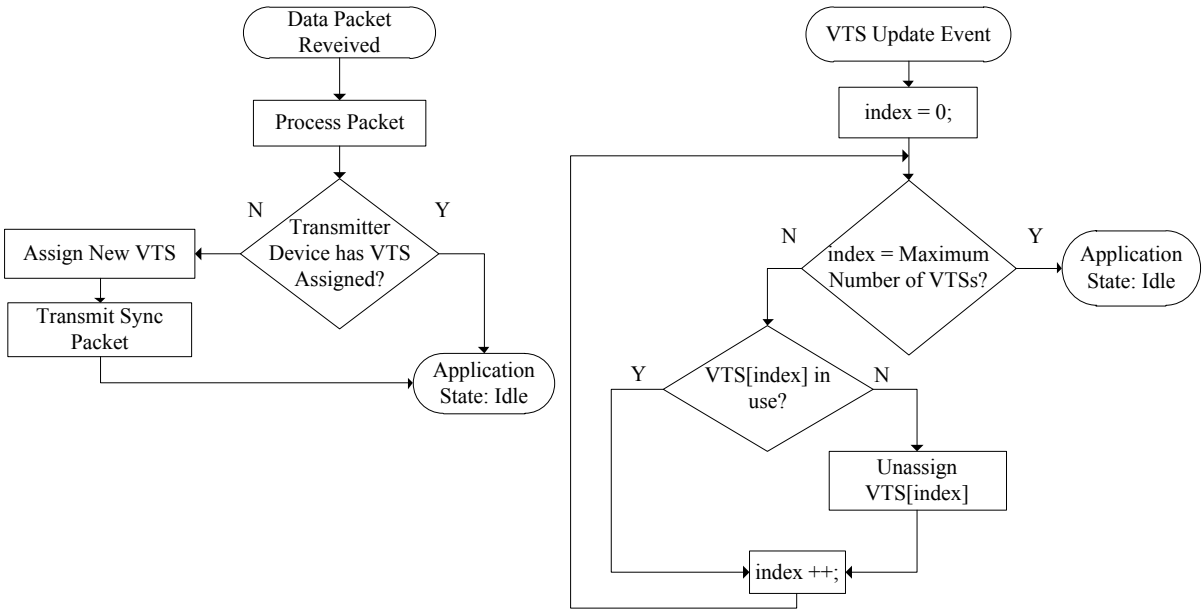


Figure 3.16 – HNPAvoidance application algorithm in the network coordinator.

The HNPAvoidance algorithm in the node application is shown in Figure 3.17. When a beacon is received and the node does not have a VTS assigned, it will set an event to transmit the packet in a random VTS. If it already has a VTS, the node will set an event to transmit the packet in that VTS. When the node application receives an event to transmit a packet, it simply transmits the packet. When the node receives a Sync Packet containing the information about the VTS number that was assigned to him by the coordinator, saves this information in the application information base. This information will be used later, when the beacon is received, to set the transmit packet event in the correct VTS.

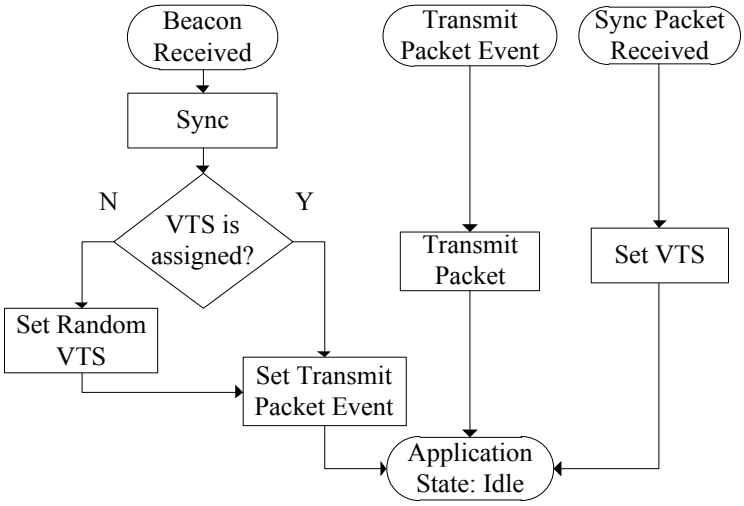


Figure 3.17 - HNPAvoidance application algorithm in a network end device.

Figure 3.18 illustrates the assignment and packet transmission sequence in the HNPAvoidance protocol in a superframe composed by 8 VTSs. End Device 1 starts to transmit a data packet to the coordinator. Since it was the first packet, which means that the node does not have a VTS assigned, it will transmit the packet in a random VTS. The coordinator verifies that node situation and transmits a Sync Packet that indicates to the node that, from now on, it must transmit in the first VTS. When the next beacon is received, the node transmits in the first VTS.

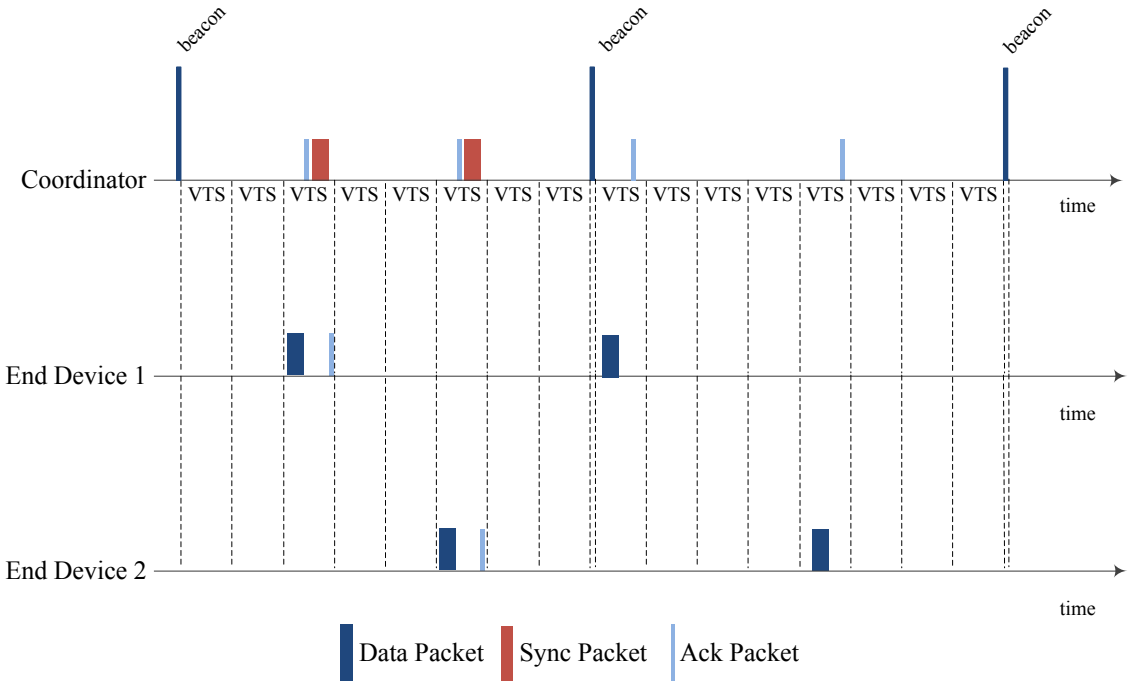


Figure 3.18 - VTS assignment sequence in the HNPAvoidance protocol.

In the previous figure, the first and the fifth VTS are assigned to End Device 1 and End Device 2, respectively. That is because the algorithm separates, as maximum as possible, the devices transmissions. This may be useful when there are few devices in the network because, if a packet transmission fails, the devices may retransmit the packet without interfere with transmissions from other devices, including the network beacons.

During the development of the HNPAvoidance protocol, a different mechanism for the allocation of VTSs was also considered. This mechanism uses the beacon payload to identify which VTS are allocated to the network devices. When a new device wants to use a VTS, it just listen for the beacon to verify which VTSs are available and then it may start to use it. In this solution, the coordinator is the entity responsible for detecting that a VTS is now being used by a new device and should update this information in the beacon payload. This solution would decrease the network load because the Sync Packet would not be transmitted. This would also decrease the time length of a VTS because, in the current implementation of the protocol, the VTS size should be adapted so that it may support the full VTS allocation procedure. Once the VTS time slot is smaller, the number of VTSs supported by the network should be greater. This solution was excluded because it shows a limitation in detecting when a new device starts transmitting in a previously non-allocated VTS, at the coordinator's application layer. Due to the scheduler of the Z-Stack operative system, the transmission of a packet from ZigBee's lower layers to the application layer may be interrupted if a new packet is received. Consequently, the delay experienced by the previous packet will increase, making it difficult for the application to determine, with precision, the instant of time it was received. Additionally, as the VTS length decreases, the probability of a packet being received while the previous packet is still being processed at the upper layers (e.g., the NWK layer) also increases.

Several issues may rise from the synchronization mechanism and from how the VTS assignment is managed by the network coordinator. Therefore the limitations and possible improvements for the proposed protocol are discussed next.

The algorithm assumes that there are VTSs available for all active nodes in the network and it does not guarantee that a time slot will be only used by one of the network nodes because the proposed protocol is not a pure TDMA protocol. At the application level, the nodes transmissions times are separated to avoid the HNP, but if there are insufficient VTSs to be assigned to all the network nodes, the protocol does not solve this problem and, consequently, several nodes may transmit in the same VTS. If a packet transmission fails in a

VTS, retransmission mechanisms may force that packet to be retransmitted into the next VTS, which may already be in use by other nodes. Consequently, the nodes would compete in the same VTS.

The VTS assignment mechanism of the HNPAvoidance is initiated by the transmitter device, which transmits the data packet in a random VTS and expects a Sync Packet containing information related to the VTS that was assigned to it. The random VTS that was selected to transmit the data packet may be assigned to another device. Consequently, the competition for the network channel may result in the problems discussed in the previous paragraph. To avoid this issue, a static VTS could be specifically used for devices transmit packets when they have no VTS assigned. However, this would decrease the number of available VTSs and competition between nodes with no VTS assigned may also occur although less frequently because nodes will only compete in this static VTS when they pretend to acquire a VTS. In the subsequent transmissions, their packets are all practically transmitted in the assigned VTS.

3.4.2.1 HNPAvoidance Validation Experimental Setup

In order to demonstrate that the HNP may be solved through the HNPAvoidance protocol, various tests were done, repeating the previous HNP experiment illustrated in Figure 3.14. Since the packet transmission events are set by the protocol, the triggers were not used. The beacon interval was set to approximately 122 ms, since it is not possible to configure the same 100 ms interval used in the previous HNP experiment due to limitations imposed by the IEEE 802.15.4 standard. Therefore, the nodes transmit data packets every 122 ms in this case. For the purpose of this test, the change in the traffic parameter is not relevant. The number of VTSs was set to 8, which means that each VTS has a time length of approximately 15.25 ms, which is more than sufficient for a node to transmit its packets inside the assigned VTS.

3.5 Analysis of Body Interference in RF Communications

In this section an analysis of human body interference on radio communications in a ZigBee-based WBAN is discussed. This topic is of great interest, so that we can understand the effects of interference due to the human body and assess if this protocol is a reliable framework for WBANs. The results of provided by this experimental evaluation may help

the future creation of realistic propagation models for Zigbee-based WBANs. This analysis takes into account the RSSI, which is measured by the CC2530, and the packet error ratio (PER), which corresponds to the number of erroneous packets that were received divided by the number of transmitted packets. The analysis is based on the WBAN posture monitoring system (PMS) discussed previously. In short, a set of sensor devices are placed over the human body, data packets containing the sensor data are transmitted to the network coordinator; and the RSSI and the PER are measured and calculated at the coordinator.

Several factors related to mobility, changes in posture, size, weight, and water content of the human body may affect the signal reliability in a WBAN. In fully wireless WBANs in which sensor nodes transmit directly to the BS, there may be some periods in which the human body may cause a lot of interference. This situation occurs frequently in the PMS that was studied, more specifically when a sensor is placed on the chest and the BS is on the opposite side.

In the experimental component of this analysis, a real implementation of the PMS was used in order to obtain accurate measurements for a typical WBAN where sensors are placed very close to the human body. Figure 3.19 shows the two modules that constitute a PMS sensor device: the sensor module and the communications module. In the former magnetic and inertial sensors are used. The latter corresponds to the communications hardware, which has a CC2530 unit and a PCB inverted F antenna. The modules are coupled in order to reduce the size of the sensor device.

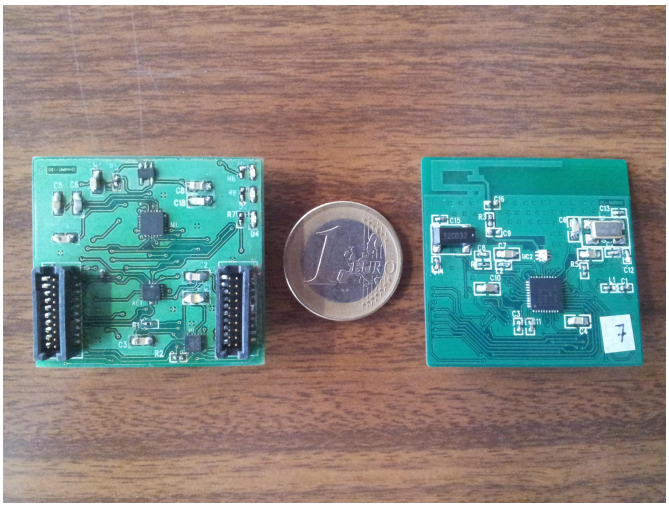


Figure 3.19- The sensor module and the communications module of a PMS device.

Two different schemes were tested, one in an RF anechoic chamber and the other in an

indoor classroom-type environment. The results achieved in these experiments are affected by fading and shadowing effects in the RF communications due to the environment and the body interference. Situations of non-line-of-sight (NLOS) communication between sensor devices and the BS are also of utter importance due to the mobility characteristics of the PMS.

3.5.1 Body Interference Experimental Setup

The network topology used in these experiments consists in a sensor device associated to a coordinator. The coordinator is a CC2530 evaluation module from the development kit. The sensor device transmits data packets of 62 bytes to the coordinator every 100 ms. Each test finishes when the coordinator application receives 2000 packets and each test was repeated three times. Once a test is finished, the results are transmitted via UART to a PC, which shows the results. Further experimental parameters are described in Table 3.1.

As mentioned before, the CC2530 in the coordinator automatically calculates the RSSI. On the other hand, the values of the PER are based in calculations at the application level. The application detects if a failure in a packet reception occurs by verifying the received packets sequence numbers. Therefore, to avoid the influence of the retransmission mechanism of the MAC layer in the PER calculation, this was disabled[1]. Otherwise, retransmissions could hide some packet errors from the application, despite these were observed at the PHY and MAC layers[2]. However, the PER can be affected by other errors not directly connected to the interference caused by the body. These are not detected at the PHY and MAC layers when, e.g., failures in the synchronization of the PHY frame preamble occur or when decoding the destination address, impeding the reception of the packets and the detection of further errors.

Figure 3.20 illustrates the experimental setup used to perform the evaluation of body interference on communication inside an anechoic chamber. These experiments were divided into two parts. In the first part, a sensor device was positioned on the user's chest and tests were made at several body positions, in function of θ, which represents the angle between the body direction and the direction of the BS, i.e., when $\theta = 0$ the sensor device is pointed

[1] For that effect, the IEEE 802.15.4 parameter *aMaxFrameRetries* was zeroed.

[2] Erroneous packets are detected in the radio hardware of the SoC CC2530 when the packet's CRC is calculated and verified. When an error is detected, an interruption is generated at the MAC sublayer of the Z-Stack that indicates it.

directly to the BS, and when θ = 180 the user's body is positioned between the sensor device and the BS. In the second part of the experiment, the same tests were performed, but this time with the sensor placed on the arm. In both cases, the user is 2 meters away from the network coordinator. Tests with various output power levels on the sensor device were performed in order to obtain as much information as possible. The results are presented and analyzed in the next chapter.

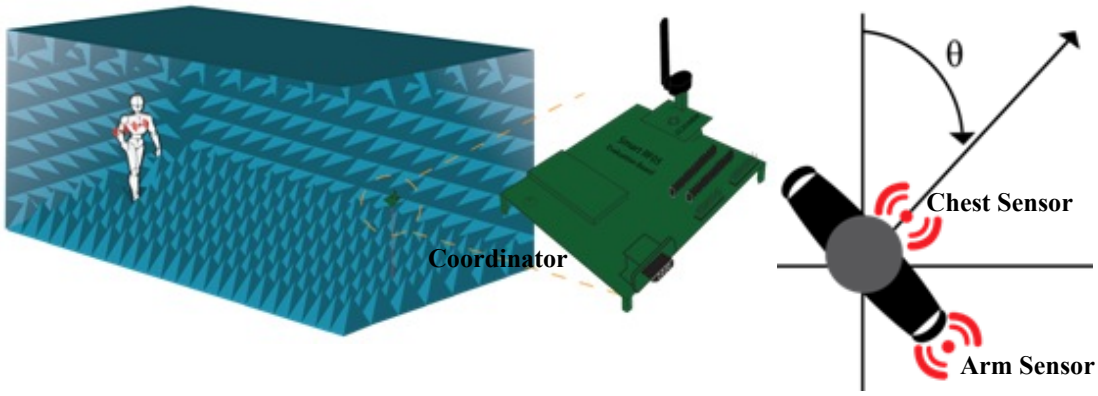


Figure 3.20 - Body interference experimental setup in an anechoic chamber.

The experimental setup for the indoor evaluation consists in a room of 6 m x 8 m, as shown in Figure 3.21. The tests performed in this experiment correspond to those executed in the anechoic chamber, where a sensor device was placed on the user's chest or on the arm and the body position in relation the BS was varied. In these experiments, instead of varying the output power of the transmitter, the measurements were made by changing the distance between the user and the BS.

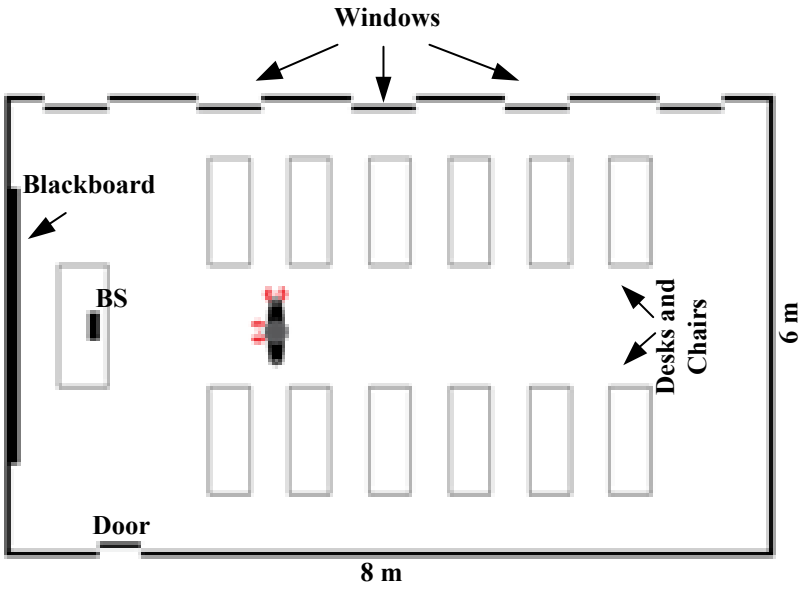


Figure 3.21 - Body interference experimental setup in a classroom.

3.6 ZigBee Software Delay Parametric Model

This section proposes a software delay model which establishes a set of parameters referring to the delay introduced by the ZigBee software components. The model makes general considerations in order to keep it simple and suitable for all ZigBee software implementations. However, particular considerations regarding the Z-Stack implementation and the CC2530 are used in modeling the software delay because the ultimate goal is to validate it by introducing the parameters into a simulator and comparing the simulations to the experimental results.

Network simulators are useful tools for studying several aspects of different wireless network protocols, including WBANs. The main advantage of using simulators is the time that can be saved in the evaluation of large-scale networks. However, simulation results may not achieve entirely accurate results because most simulators use simplified assumptions on some of theirs models [Kotz04]. For example, some simulators use radio models based on the node distances which assume that packets inside a given circular area around the transmitter are always perfectly received, ignoring the physical characteristics of the surrounding space that may cause fading and shadowing effects. In order to keep simulators simple, they may also ignore certain aspects of software and hardware specific implementations. The device's software may include delays due to its structure and due to the processing load in its

components. In the former case, the operating system used to manage the tasks running in the device may attribute different priorities to the tasks being processed at a certain moment, which may lead to an increase in delay, for example, when a task that is processing a packet is interrupted so a higher priority task may be processed. In the latter case, the amount of code to be executed by the stack may also introduce delay until the packet is fully processed. The hardware delays are related to the limited processing capacity of the devices, which increases the delay as the software computational requirements increase. The main reason to keep simulator implementations simple is to provide some portability between different application scenarios without the constraints of a simulator that may be adapted to a particular application.

In [Gama11], the author identified two main causes for the divergence between the results obtained in experimental tests and results from a simulation platform. The causes are due to the behavior of the software components and the devices' time drift. The author provided a software delay model for unslotted IEEE 802.15.4 networks and introduced it into Castalia[1], considering that a set of sensor nodes were transmitting intensive traffic (90 or 100 bytes of MAC payload every 250 ms or 50 ms, respectively). The model identifies a series of software characteristics and defines a set of parameters, which include the time needed for a sensor node to transmit a packet and the time spent by a base station to fully process the packet. A model, to distribute the instants of time that nodes transmit their packets, was also included due to limitations on the software platform used in the experimental tests. This is because the software being executed in the base station only receives a packet once the previous packet is fully processed. Otherwise, packets received while the BS is processing a packet are dropped. The model proposed by the author also includes a time drift model for the sensor nodes. Simulations showed that the results obtained with the proposed model match satisfactorily those obtained in real conditions.

In our work, the OMNeT++ simulation platform was used. The software delay model was introduced into the software simulation model of the unslotted CSMA-CA of the IEEE 802.15.4 protocol implemented by Pedro Macedo in his master´s degree thesis [Macedo10]. The evaluation scenario used to validate the model is based on the experimental tests provided in this work for the delivery ratio and the delay. Therefore, no time drift model and no traffic distribution model is defined in our model because a shared trigger sets nodes' transmissions

[1] Castalia is a discrete events simulator that is used for wireless sensor networks.

and no processing restraints in the BS software were found.

3.6.1 The IEEE 802.15.4 Unslotted CSMA-CA Simulator

In this work, we used a model of the unslotted CSMA-CA of the IEEE 802.15.4 protocol [Macedo10] developed in OMNeT++, in which we introduced our software delay model in order to increase the accuracy of the simulation results. The OMNeT++ principle of operation is based on a hierarchical modularization, where the simulator's implementation is divided into a set of modules that exchange messages which may contain complex data structures. Each module contains a set of functions and variables that are used to model its behavior. There are three types of modules:

- The "Simple Module", which is at the lowest level in the hierarchy and is implemented in C++ by the user;
- The "Compound Module", which may be composed by a set of Simple Modules and other nested Compound Modules, and whose code is automatically generated by OMNeT++ based on the network topology;
- The "Network Module" which is at the top of the hierarchy and may contain several Compound Modules.

Figure 3.22 presents the structure of the IEEE 802.15.4 Unslotted CSMA-CA simulator for a star network topology. The modular structure of the simulator is constituted by: The System Module, which corresponds to a Network Module and is composed by the Wireless Device and the Base Station Compound Modules and by the Wireless Channel "Simple Module". In the simulator, multiple Wireless Device modules may be defined to simulate a more complex network. The Traffic, Network, FIFO (First In First Out), MAC and PHY Simple Modules compose both the Wireless Device and the Base Station. Due to this, a single module called "Device" is defined, which is able to simulate the behavior of these two modules [Macedo10].

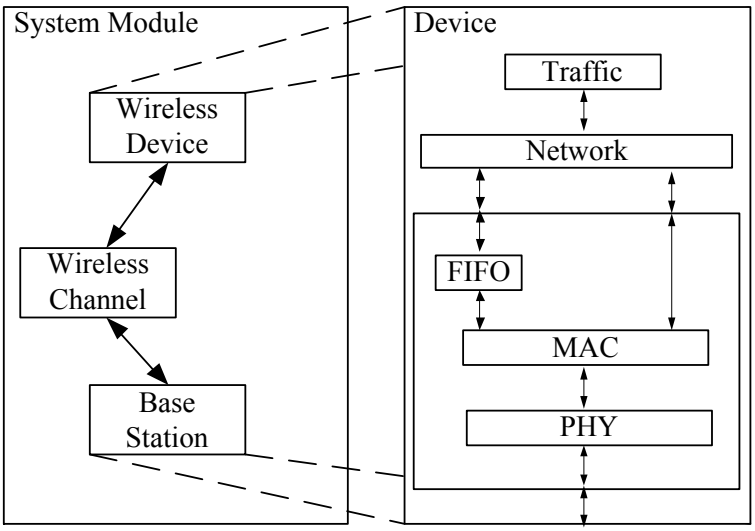


Figure 3.22 – System Module and the Device model structures implemented with OMNeT++.

In a Device module, the Traffic module is responsible for generating the traffic load in the Wireless Device by simulating an application. Three different applications are defined for this module: the coordinator application, which is responsible for outputting all the statistics of the simulation, i.e., the network delivery ratio, delay, etc.; the router application, which relays the packets to the coordinator in the case of a 2-hop tree network topology; and the sensor device application, which generates the traffic from the sensors, where the user can define the period for traffic generation.

The Network module was included in the model so that it may be possible to implement and simulate other IEEE 802.15.4-based networks; this module does not comply with the IEEE 802.15.4 standard, which does not include a network layer. By default, it relays all packets from a lower layer to the application and vice-versa; it also does not introduce delay overhead to the simulations.

The FIFO module represents a buffer to store messages from the Network module ready to be transmitted to the MAC module, while the previous packet is being transmitted [Macedo10].

The MAC module is where the medium access control mechanisms (CSMA-CA, in the case of the IEEE 802.15.4) are implemented and where the beacon and acknowledgment transmissions are enabled when required. It also manages the radio status (transmit, receive, sleep).

The PHY module is responsible for the communication with the Wireless Channel

module. It transmits/receives data packets and executes the CCA mechanism. In order to execute the CCA mechanism, the PHY transmits a message to the Wireless Channel module, which has a list of other active devices and responds whether there is some device transmitting or not.

The Wireless Channel simple module is responsible to simulate channel error models, where errors may be introduced in packets depending on static or dynamic bit error rate channel parameters, path loss models or other models. It is also responsible to detect collisions between simultaneous transmissions from different devices.

3.6.1.1 IEEE 802.15.4 Unslotted CSMA-CA Simulator Evaluation

In order to evaluate IEEE 802.15.4 unslotted CSMA-CA simulator, this section presents a set of results of simulations for the maximum goodput in a network device. The primary goal is to validate its compliance with the time model associated to a packet transmission in the IEEE 802.15.4 protocol, which was presented in section 3.2.1.

The simulation consists in an end device transmitting packets to a coordinator in star and 2-hop tree topologies. The application running on that device initiates a packet transmission after receiving the acknowledgement from the previous transmission. The IEEE 802.15.4 parameters and the respective values that were used in this simulation are specified in Table 3.1.

Figure 3.23 presents the theoretical and the simulated results for the maximum throughput in a network device in star and 2-hop tree topologies. The graphic shows practically identical results between the simulations and the theoretical results, which demonstrate that the model of the unslotted IEEE 802.15.4 of the simulator is equivalent to the model that we used to obtain the theoretical results, presented in section 3.2.1, and that the values of the parameters at the MAC and PHY level used by the simulator correspond to those in the theoretical model. This simulation also proves that no other delays were introduced by network and application layers of the simulator because otherwise this would have implications in the simulation results.

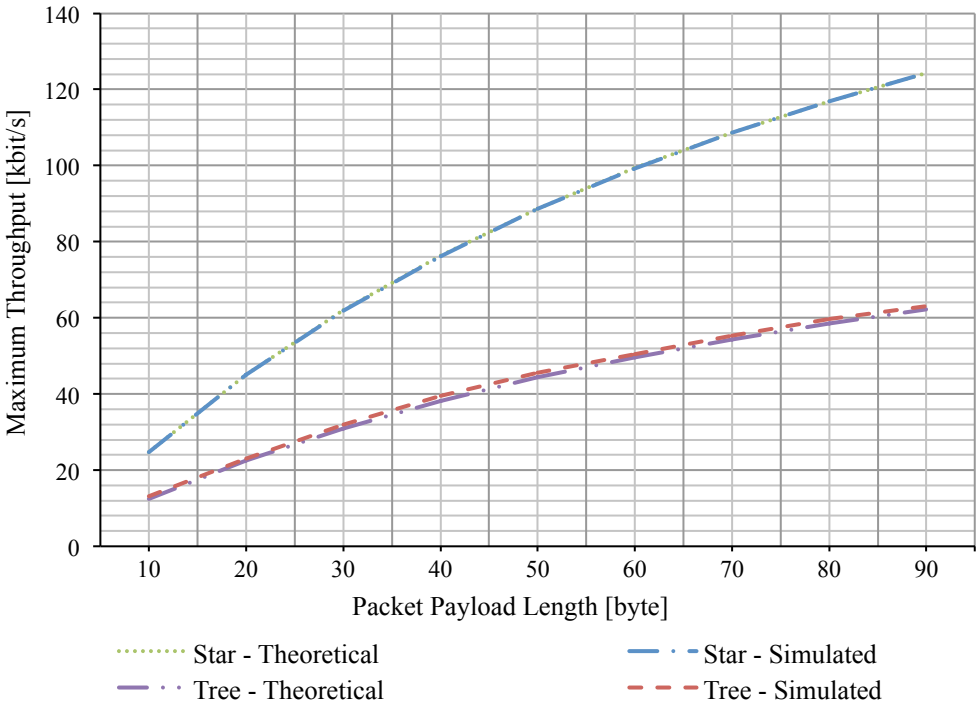


Figure 3.23 – Maximum theoretical and simulated goodput.

Concluding, the simulator of the IEEE 802.15.4 unslotted CSMA-CA used in this work complies with transmission model presented in section 3.2.1, which is used to define the model for the ZigBee software delay that is presented in the next section. This makes the simulator suitable to include our software delay model, allowing the provision of more accurate simulation results for ZigBee networks.

3.6.2 Software Delay Parametric Model

The proposed software delay model considers the extra time necessary for a packet to travel through the software stack in both directions, encompassing the time from the moment in which the end device application generates a data transmission event until it receives the confirmation that the packet was correctly transmitted (T_{TXtot}). The model also defines time elapsed in a base station for a packet that is received in the application layer, from its reception in the PHY layer (T_{RXtot}). Finally, it considers the time needed for a router to relay a packet (T_{Relay}) from the end device to the coordinator, i.e., the time elapsed since a packet is received in the PHY layer of the router until this layer relay this packet to the coordinator. These parameters are constituted by several delay components introduced by the stack layers, which are shown in Figure 3.24.

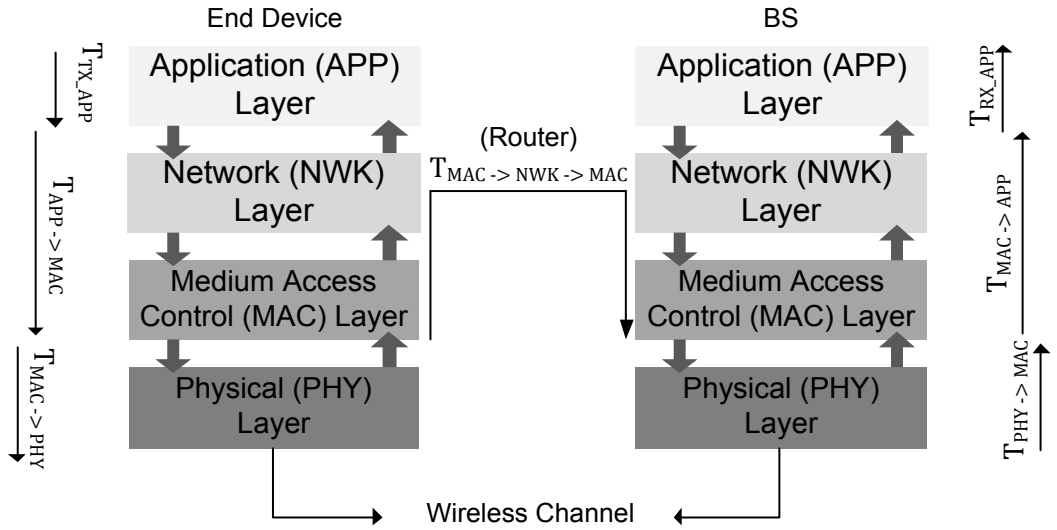


Figure 3.24 – Delay components involved in packet transmission, in a packet relaying and in a packet reception.

Equation 3.21 gives the T_{TXtot}, which depends on the packet length n.

$$T_{TXtot}(n) = T_{TX\_APP}(\mathrm{n}) + T_{APP \rightarrow MAC}(\mathrm{n}) + T_{MAC \rightarrow PHY}(\mathrm{n}) + T_{TX\_PHY} + T_{MAC\_Conf} \quad (3.21)$$

T_{TX_App} represents the time that the application needs to prepare the transmission. The parameter $T_{APP \rightarrow MAC}$ represents the time elapsed since the application calls the API function to transmit the packet until the instant it reaches the MAC sublayer. At this level, the final structure of the package is almost complete, lacking only the inclusion of the PHY overheads and the CRC[1]. $T_{MAC \rightarrow PHY}$ is the time needed to prepare the PPDU transmission, which includes the transmission of the packet to the radio module and the PHY overheads and CRC calculations. T_{TX_PHY}[2] corresponds to the backoff period ($T_{Backoff}$), the turnaround time (T_{TAT}) and the time required to transmit the packet (T_{Packet}). Considering the acknowledgment mechanism[3], T_{MAC_Conf} represents the time required for the MAC layer to receive the confirmation of the previous transmission and is constituted by a T_{TAT} and a T_{ACK}, which corresponds to the time period that is necessary to receive the acknowledgement frame.

$$T_{TX\_PHY}(\mathrm{n}) = T_{Backoff} + T_{TAT} + T_{Packet}(\mathrm{n}) \quad (3.22)$$

[1] The CRC is calculated and included automatically in the packet by the CC2530 radio hardware.

[2] The designation T_{TX_PHY} was chosen because its parameters are introduced by the CC2530 radio hardware.

[3] The receiver transmits the acknowledgment frame before the next layer processes the data frame.

$$T_{MAC\_Conf}(\mathrm{n}) = T_{TAT} + T_{ACK} \tag{3.23}$$

The $T_{MAC \rightarrow PHY}$ parameter is very difficult to measure because it is dependent on the firmware used in the radio module and it is impracticable to set a timer for measuring. However, this value can be obtained by measuring the round trip time (T_{MAC_RTT}[1]), which represents the period of time elapsed since the transmission of the packet by the MAC sublayer until the acknowledgment is received.

$$T_{MAC\_RTT}(\mathrm{n}) = T_{MAC \rightarrow PHY}(\mathrm{n}) + T_{TX\_PHY} + T_{MAC_{Conf}} \tag{3.24}$$

$T_{Backoff}$, T_{TAT} and T_{Conf} parameters have known values and T_{Packet} can be calculated using equation 3.25, in which R is the network data rate.

$$T_{Packet}(\mathrm{n}) = \frac{\text{Packet n Length (bits)}}{R} \tag{3.25}$$

This model also defines the period needed for the application to receive the confirmation that the packet was successfully transmitted by the MAC layer to the next hop in the network. With the $T_{MACtoAPP\_Conf}$ and T_{TXtot} parameters, it is possible to obtain the application level round trip time, which is shown in equation 3.26. The $T_{MACtoAPP\_Conf}$ parameter can be useful when it is required to simulate an application that generates the next packet only when the previous one has been correctly transmitted.

$$T_{APP\_RTT}(\mathrm{n}) = T_{TXtot}(\mathrm{n}) + T_{MACtoAPP\_Conf} \tag{3.26}$$

Equation 3.27 presents the T_{RXtot} period, which depends on the packet length n. T_{RX_PHY} is equals to T_{Packet}. The $T_{PHY \rightarrow MAC}$ parameter represents the entire processing period until the packet is delivered by the PHY layer to the MAC sublayer. As in the $T_{MAC \rightarrow PHY}$ parameter, the validation process does not distinguish the delay introduced by the communication between the microcontroller and the radio module in the $T_{PHY \rightarrow MAC}$ component. $T_{MAC \rightarrow APP}$ represents the period elapsed from when the MAC finishes the processing of the received packet until it is delivered to the application. T_{RX_APP} represents the processing time spent by the application.

[1] Excluding the $T_{MAC \rightarrow PHY}$ parameter, the T_{RTT} corresponds to the model presented in section 3.2.1.1, which shows the times associated to a packet transmission in the IEEE 802.15.4 standard.

Considering the acknowledgment mechanism, T_{MAC_conf} represents the time required to transmit the confirmation of the received packet[1].

$$T_{RXtot}(n) = T_{RX\_PHY}(\mathrm{n}) + T_{PHY \rightarrow MAC}(\mathrm{n}) + T_{MAC\_conf} + T_{MAC \rightarrow APP}(\mathrm{n}) + T_{RX\_APP}(\mathrm{n}) \quad (3.27)$$

$$T_{RX\_PHY}(\mathrm{n}) = \quad T_{Packet}(\mathrm{n}) \quad (3.28)$$

The T_{TX_APP} and T_{RX_APP} components were considered negligible in the T_{TXtot} and T_{RXtot} validation, respectively, because, in the experimental tests, the data generated by the source application is static and the receiver application does not process the data.

The T_{Relay} period can be calculated through the following equation, where n corresponds to the packet length.

$$\begin{aligned} T_{Relay}(n) = \quad & 2T_{MAC\_Conf} + T_{PHY \rightarrow MAC}(\mathrm{n}) + T_{RX\_PHY}(\mathrm{n}) + T_{MAC \rightarrow \mathrm{NWK} \rightarrow MAC}(\mathrm{n}) \\ & + T_{MAC \rightarrow PHY}(\mathrm{n}) + T_{TX\_PHY}(\mathrm{n}) \end{aligned} \quad (3.29)$$

The $T_{MAC \rightarrow NWK \rightarrow MAC}$ represents the period elapsed since the packet was received by the MAC sublayer of the router, from the PHY layer until it is received again by the MAC sublayer, after passing by the NWK layer, to be transmitted to the coordinator. Considering the acknowledgment mechanism, the doubled T_{MAC_conf} represents the time required to transmit and receive the confirmation of the relayed packet.

Table 3.7 summarizes the described parameters related with the transmitting, receiving and relaying processes.

[1] The acknowledgment frame is transmitted before the next layer processes the data frame.

Table 3.7 – Notation used in the parametric delay model.

Symbol	Meaning
T_{TXtot}	Time required for the end device to fully complete the process of transmission of an application data packet.
T_{RXtot}	Time elapsed in a base station for a packet to be received in the application since it was received in the PHY layer.
T_{Relay}	Time needed for a router to relay a packet from the end device to the coordinator.
T_{TX_App}	Time that the application needs to prepare the transmission
$T_{APP \rightarrow MAC}$	Time elapsed since the application calls the API function to transmit the packet until the instant it reaches the MAC sublayer.
$T_{MAC \rightarrow PHY}$	Time to prepare the PPDU transmission.
T_{TX_PHY}	Time to transmit the PPDU, in the PHY layer.
T_{TX_PHY}	Time to receive the PPDU, in the PHY layer.
$T_{Backoff}$	The backoff period.
T_{TAT}	The turnaround time.
T_{Packet}	The packet transmission time.
T_{MAC_Conf}	Time required to receive the confirmation of the previous transmission at the MAC layer.
T_{MAC_RTT}	Time elapsed since the transmission of the packet by the MAC sublayer until the acknowledgment is received.
$T_{PHY \rightarrow MAC}$	Time until a packet is delivered from the PHY layer to the MAC sublayer.
$T_{MAC \rightarrow APP}$	Time elapsed since the MAC finishes the processing of the received packet until it is delivered to the application.
T_{APP_RTT}	Time elapsed since the transmission of the packet by the application layer until the confirmation at the application that that the packet was successfully transmitted.
$T_{MACtoAPP\_Conf}$	Time elapsed since the MAC layer receives the confirmation that a packet was successfully transmitted until this information arrives at the application.
$T_{MAC \rightarrow NWK \rightarrow MAC}$	Time elapsed since the packet was received by the MAC sublayer and subsequently transmitted to the NWK layer until it is received again by the MAC sublayer.

3.6.2.1 Delay Measurements Setup

In order to obtain the delay values for the parameters of T_{TXtot}, T_{RXtot} and T_{Relay}, several delay measurements were performed in strategic points of the Z-Stack software with the use of an end device transmitting packets in modes A and B (see section 3.2.2) in star and 2-hop tree topologies.

To measure the $T_{APP \rightarrow MAC}$ component, a hardware timer was set to measure the interval of time while the packet crosses through the ZigBee stack from the application until it reaches

the MAC sublayer at the *macCspTxGoCSMA*[1] function in *mac_csp_tx.c* stack file.

To obtain the $T_{MAC \rightarrow APP}$ delay component, a measurement of the time interval was made while the packet crosses through the ZigBee stack from the MAC sublayer, at the *macRxAckTxDoneCallback*[2] function in *mac_rx.c* stack file, until it reaches the application, in its events processing function. This was achieved by using a hardware timer.

To obtain the $T_{MAC \rightarrow NWK \rightarrow MAC}$ delay component, the time interval was measured since the *macRxAckTxDoneCallback* function is executed, meaning that a packet was successfully received, until the *macCspTxGoCSMA* function is executed. During this period, the received packet is processed by the MAC layer and transmitted to the NWK layer, which then routes the packet to the next hop, so it may reach the destination. The NWK layer sends the packet back to the MAC to be transmitted by the *macCspTxGoCSMA* function.

The $T_{MACtoAPP\_Conf}$ time interval was measured since the *macRxAckTxDoneCallback* function is executed until and the AF_DATA_CONFIRMATION_CMD event is executed in the application. During this period, the confirmation message crosses the ZigBee stack from the MAC layer to the application.

As previously said, the $T_{MAC \rightarrow PHY}$ parameter does not distinguish the period for the transmission of a packet from the microcontroller to the radio transceiver because both are integrated onto the same chip.

The T_{TX_PHY} and the T_{RX_PHY} components are only dependent on the periods of the CSMA-CA of the IEEE 802.15.4 protocol, which are implemented into the unslotted CSMA-CA IEEE 802.15.4 simulator. So, in order to verify the compliance of the real values of these two components with the simulator, we measured the value of T_{MAC_RTT}, where the experimental results represent the average T_{MAC_RTT} of 1000 packet transmissions. This value was obtained by measuring the interval of time from when the *macCspTxGoCSMA* function is executed to the execution of the *macRxAckTxDoneCallback* function. $T_{MAC\_RTT}(n)$ is dependent on $T_{Backoff\_max}$ (7 UBPs) and $T_{Backoff\_min}$ (0 UBPs), along with the packet length n, which includes 33 bytes of overhead which is from the ZigBee protocol, Thus:

[1] The *macCspTxCGoCSMA* is a low-levelfunction that initiates the process of transmission of a packet, where the radio command strobe processor (CSP) is started to automatically proceed with the CSMA-CA mechanism and transmit the packet.

[2] The *macRxAckTxDoneCallback* is a low-level callback function executed when an outgoing acknowledgment frame has completed the radio transmission.

$$T_{Backoff\_min} + T_{TAT} + T_{Packet}(\mathrm{n}) + T_{MAC\_conf} \leq T_{MAC\_RTT}(\mathrm{n}) \quad (3.30)$$

$$T_{MAC\_RTT}(\mathrm{n}) \leq T_{Backoff\_max} + T_{TAT} + T_{Packet}(\mathrm{n}) + T_{MAC\_conf} \quad (3.31)$$

Table 3.8 presents the theoretical minimum ($T_{RTT\_min}(n)$) and maximum ($T_{RTT\_\mathrm{min}}(\mathrm{n})$) values that will be used to validate the experimental results of the $T_{RTT}(n)$ parameter, as the payload increases in the application. The results are expressed in milliseconds.

Table 3.8 – Minimum and maximum values for the T_{RTT} parameter.

Payload (byte)	$T_{RTT\_min}(\mathrm{n})$	$T_{RTT\_max}(\mathrm{n})$
10	2.112	4.352
20	2.432	4.672
30	2.752	4.992
40	3.072	5.312
50	3.392	5.632
60	3.712	5.952
70	4.032	6.272
80	4.352	6.592
90	4.672	6.912

3.6.2.2 Model Validation

In order to validate the proposed delay parametric model, its parameters were introduced into the IEEE 802.15.4 unslotted CSMA-CA simulator modules. The $T_{APP \rightarrow MAC}$ component was introduced into the network module of a simulated sensor device for when downlink packets that go from the application module to the MAC module. For the confirmation of the status of the previous transmission, the $T_{MACtoAPP\_Conf}$ was also introduced into this module whenever an application level acknowledgment message is transmitted from the MAC module to the Traffic module. This acknowledgment message is a new feature introduced into the unslotted CSMA-CA simulator, where, before, it only existed at the MAC level. The $T_{MAC \rightarrow APP}$ component was introduced into the network module for packets that move upwards in the simulated network BS. The $T_{MAC \rightarrow NWK \rightarrow MAC}$ delay component was included into the network module, but is only considered whenever the simulated device is a router.

The tests were carried out in the simulation platform using test conditions similar to the traffic configurations used in maximum goodput and the delivery ratio and delay experimental evaluation proposed in this work. The IEEE 802.15.4 parameters used in the simulations are consistent with those used in the experimental tests and are presented in Table 3.1.

For the maximum goodput, it was simulated the star and 2-hop tree topologies with a sensor device transmitting packets in mode 2, as the packet payload increases (see section 3.2.1.2).

For the delivery ratio and delay, the simulations were run in star and 2-hop tree topologies, while sensor devices transmitted packets in mode A and mode B (see Table 3.3 and Table 3.4 descriptions in section 3.2.2). Packets transmitted by the application are triggered by a simultaneous event in all network nodes. Therefore, as in the experimental tests, the worst-case scenario is evaluated too.

The validations process consists in the comparison of the simulations with the experimental results, for the same setup architecture. A simulation of maximum goodput and the network delivery ratio and delay metrics was performed with the configurations provided in the two previous paragraphs. If the simulation results are close to experimental results, we may prove that the proposed delay model, in combination with the IEEE 802.15.4 unslotted CSMA-CA model already developed, is suitable for the provision of accurate ZigBee network simulation results.

3.7 Summary

This chapter starts with a description of the hardware and software platforms that were used to perform the proposed evaluation for this work. The hardware consists on a CC2530 development kit. The software used in this work are implementations of the ZigBee 2007 specification and the IEEE 802.15.4 protocol: the Z-Stack and the TIMAC, respectively. Both hardware and software platforms were provided by Texas Instruments.

A set of experiments regarding the maximum goodput in a network device, the network delivery ratio and delay were evaluated. In these experiments, the network traffic is from an existing data-intensive WBAN: the posture monitoring system (PMS). These evaluations allow us to conclude that the maximum goodput in a device's application alongside with the network's normalized throughput is well below the network maximum data rate (250 kbit/s). The results obtained from the analysis of the delay enabled to delimitate the maximum and minimum delays for a packet to be transmitted over a star or 2-hop tree network topology. Knowing this, the detection of out-of-range delays in the experimental component of this work is now possible.

A model is also proposed to predict the clock drift effect in a non-beacon enabled ZigBee/IEEE802.15.4-based body area network because these standards do not specify any mechanism to solve this issue. In this model, an average clock drift based on precise measurements of each node's individual clock drift is defined to make an approximation of how much time two different nodes will contend for the wireless network channel (T_{Int}) and how long it takes for contention to repeat (T_{IntRep}). The approximation is based on a vulnerability window that defines when the two nodes will interfere with each other in the wireless channel (T_{Vul}).

In this chapter, we also describe an experimental evaluation setup for a WBAN in the presence of hidden-nodes in a network consisting of two ZigBee end devices associated and transmitting data packets to a coordinator in a star topology. The HNPAvoidance protocol is also presented to prove that the HNP may be solved by separating the instants of time in which the nodes transmit their packets. Since WBANs sensor devices usually generate periodic data, the HNPAvoidance uses the superframe structure of the IEEE 802.15.4 in order to synchronize transmissions for each node. At the application level, the HNPAvoidance protocol creates a set of virtual time slot (VTS) to be assigned to these nodes. Then, each node uses its assigned VTS to transmit at will.

The configuration adopted for the experimental tests regarding the interference of the human body in radio communications in a ZigBee-based WBAN is also presented. Several factors related to mobility, changes in posture, size, weight, and water content of the human body and other sources of interferences such as nearby WBANs, networks operating in these license-free frequency bands and other general sources of electromagnetic interference may affect the signal reliability in a WBAN. So, the experiments are based on the measurements of the received power and the packet error ratio using the PMS. The results obtained from these analyzes may be used later in the definition of propagation models for ZigBee-based networks.

Finally, this chapter provides a model for the delays introduced by the ZigBee's software layers. This model is to be introduced into a simulator of the unslotted CSMA-CA of the IEEE 802.15.4 protocol in order for it to produce more accurate simulation results of ZigBee networks. The description of the simulator in which our model was introduced is also present. The model considers three fundamental components of delay: T_{TXtot}, which corresponds to the time necessary for an end device to fully complete the process of the transmission of a packet,

T_{RXtot}, which is the time elapsed at the base station for a packet that is received in the application since it has been received in the PHY layer, and finally, the time needed for a router to relay a packet from the end device to the coordinator: the T_{Relay} component.

Chapter 4

Experimental Results and Models Validation

In this chapter, the results from the experiments which were described in the previous chapter are shown and discussed. These results were collected by a PC using a RS-232 connection and logged to files on the hard disk. After collecting the experimental results they were analyzed to obtain the maximum goodput, network delivery ratio and end-to-end delay; and also to determine the validity of the clock drift and software delay models. The effectiveness of the HNPAvoidance protocol is also validated through tests, and results from the experiments regarding the human body interference in RF communications are also evaluated.

4.1 QoS Metrics Results

4.1.1 Maximum Goodput Results

Figure 4.1 illustrates both the theoretical and measured maximum goodput for star and 2-hop tree topologies using the Z-Stack as a function of the payload length. The respective

experimental setup is described in section 3.2.1.2. This experiment shows that the measured values are significantly smaller than the corresponding theoretical values. This difference is caused by the delay between layers introduced by the Z-Stack operating system when packets are processed by its tasks.

Although IEEE 802.15.4 networks provide a data rate of 250 kbit/s in the 2.4 GHz band, the measured maximum goodput with 90-byte payload, in all experiments, was well below: 95 kbit/s in mode 1 (Star – Measured in mode 1) and 54 kbit/s in mode 2 (Star – Measured in mode 1) for the star topology, and 40 kbit/s in mode 2 (Tree – Measured in mode 2) for the 2-hop tree topology. The difference between the raw data rate and the theoretical maximum goodput is due to the overheads introduced by the protocol (backoff periods, packet headers, etc.). The difference between the experimental and theoretical results is due to the processing delay introduced by the stack implementation, as referred in the previous paragraph. The payload length could not be increased any further due to the maximum packet length limitation imposed by the IEEE 802.15.4 standard.

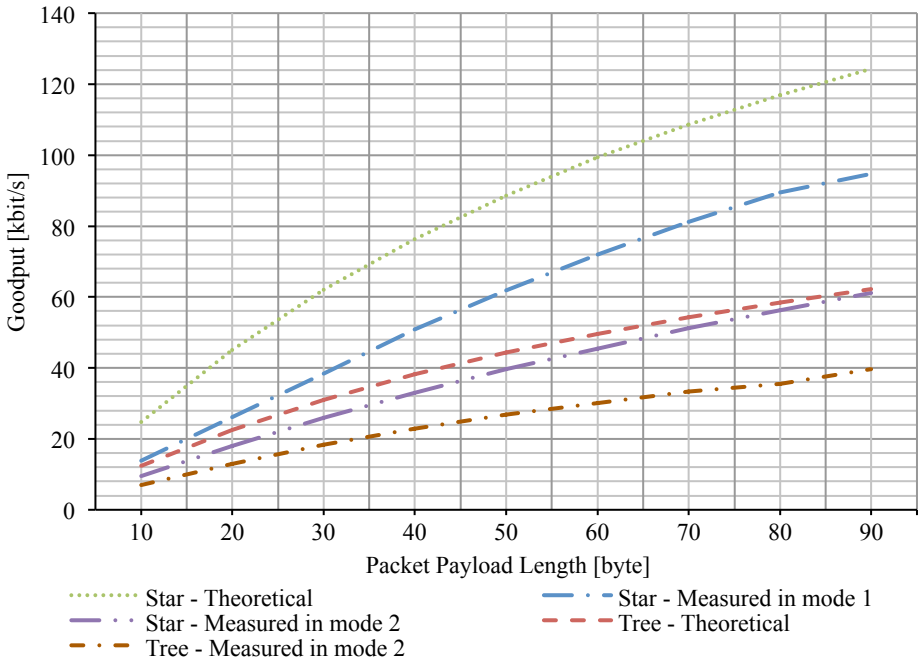


Figure 4.1 – Maximum goodput for star and 2-hop tree topologies.

It was not possible to obtain experimental results related to the maximum goodput in mode 1 with the 2-hop tree topology, because the router kept blocking during the respective tests. Some observations were made through the use of a packet sniffer. It was observed that the router relays packets for just a few seconds, then blocks for around 8 seconds, after that it becomes available again and the process repeats. Several other tests were performed in

different conditions, but this problem only occurred in tests where the router was subject to very high traffic load when receiving packets from one or more end devices. One possible explanation for this problem is that the router experiences an overload situation where it is not able to handle packet relaying at the NWK layer when new packets are constantly being received at the MAC layer, which is a higher priority task in the implementation of the Z-Stack. The router blocking problem does not occur when mode 2 is used. In this case, the time spent by the end device waiting for the reception of the ACK indication at application level gives the router enough time to relay the packet.

During the performance evaluation tests, it was detected that one of the ZigBee sublayers, more specifically the Application Support Sublayer, does not filter duplicated packets for the applications. This behavior is inconsistent with the APS characteristics defined in the ZigBee 2007 specification [ZigBee07].

In addition to all the experimental results described previously, three other transmission modes were tested. These tests were setup for the end device to transmit packets every 30 ms, 60 ms and 90 ms in star and 2-hop tree network topologies. The results showed no relevant differences in terms of goodput from the expected theoretical results because these intervals of transmission exceed the minimum period needed for the network to relay the packet without causing interference with the next transmission from the end device or without causing the router blocking problem. This minimum period can be calculated through the following equation:

$$minimum\ period\ = \frac{packet\ length\ [bit]}{measured\ goodput(Packet\ Payload\ Length)[bit/s]}. \qquad (4.1)$$

For instance, the minimum period for the star and 2-hop tree topologies with the sensor node transmitting packets of 90 bytes is of 7.58 ms and 18 ms, respectively. These are well below to the tested 30 ms, 60 ms and 90 ms transmission intervals.

4.1.2 Delivery Ratio and Delay Results

This section presents the results of the measurements regarding the network delivery ratio and end-to-end delay. Both metrics were measured in the same experiment; the reason for this is to keep them in a correlated context, enabling for a better analysis of the results. The respective experimental setup is described in section 3.2.2.3.

The router blocking problem described in the previous experiment was also observed in this scenario for the 2-hop tree topology with the acknowledgement mechanism enabled, although less frequently. Therefore, in order to allow the evaluation of the delivery ratio and delay during the period where the router is not blocked, the number of packets received by the coordinator before the experiment ends was reduced from 5000 to 1000 packets in this particular case.

4.1.2.1 Delivery Ratio

Figure 4.2 presents the measured delivery ratio with the Z-Stack in mode A as a function of the number of sensor nodes for the star and 2-hop tree topologies. For the star topology, the delivery ratio was close to 100% when the acknowledgment mechanism was used. However, the delivery ratio for the 2-hop tree topology with 3 to 5 end devices was lower (around 96%) in the same conditions. It was verified that the errors for packet delivery in these cases are associated to the route maintenance protocol, which manages the quality of the links and could not be disabled. Due to the high traffic load generated by the end devices, the route maintenance protocol initiates the router discovery procedure frequently (every 5 seconds on average, in the experiments with 3 to 5 end devices). During this procedure, which lasted for around 250 ms, the router interrupted the packet relaying, causing packet drops due to buffer overflow.

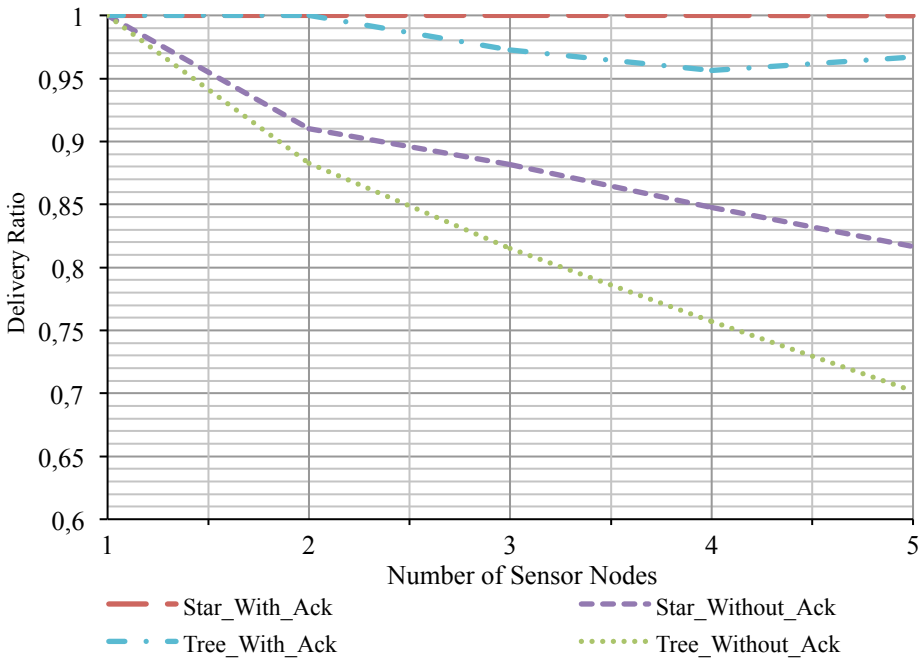


Figure 4.2 - Delivery ratio measured with Z-Stack for an increasing number of sensor nodes transmitting in mode A.

When the acknowledgments are disabled, the delivery ratio decreases significantly in both topologies as the number of sensor nodes increase. Due to the increase in network traffic the occurrence of channel access failures and collisions also increases. These results confirm the importance of the acknowledgment mechanism for the reliability of the system.

The results regarding a ZigBee network with end devices transmitting in mode B are shown in Figure 4.3. Decreasing the packet's length improved the results of these experiments, in relation to those obtained in mode A. For instance, in the acknowledged 2-hop tree topology experiment, better results were obtained because the route maintenance protocol is not executed so frequently. The route maintenance only occurred when the network was composed by four and five nodes due to the smaller packets' transmission time that increases probability of packets being successfully relayed and consequently reducing the probability of this procedure being executed. On the contrary, in experiments in mode A, this mechanism is activated when fewer nodes compose the network (three or more nodes). For the non-acknowledged 2-hop tree topology and star topologies, the differences that were found are also related to the packets smaller length.

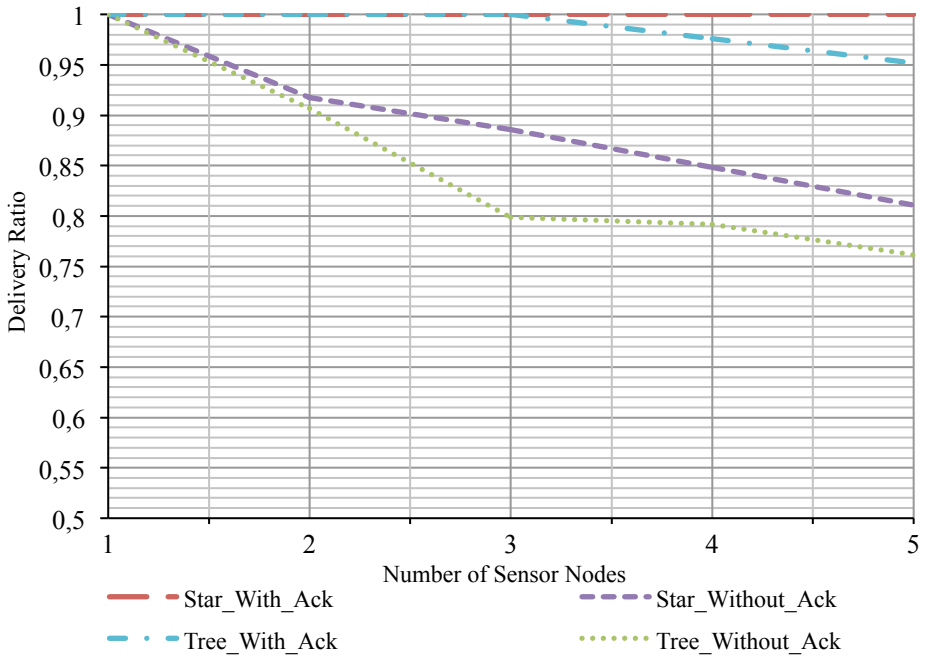


Figure 4.3 - Delivery ratio measured with Z-Stack for an increasing number of sensor nodes transmitting in mode B.

In [López11] the author detected a problematic situation, termed router deadlock, where the router receives a packet from an end device and then it rejects packets from other end devices until it relays the first packet to the next hop. This event was observed with the Jennic

implementation of the ZigBee stack on JN5139 devices, and it was shown to have a significant negative impact in the delivery ratio of the network.

During these performed tests which use the Texas Instruments implementation of the ZigBee standard, it was observed that router has the capability of interrupting the backoff process of CSMA-CA algorithm to receive and buffer new packets. Therefore, the router deadlock phenomenon does not occur with this ZigBee implementation and consequently the delivery ratio is not negatively affected. Figure 4.4 illustrates the behavior observed with the Z-Stack. In this example, the router receives two packets from different end devices first, and only after that it relays the packets.

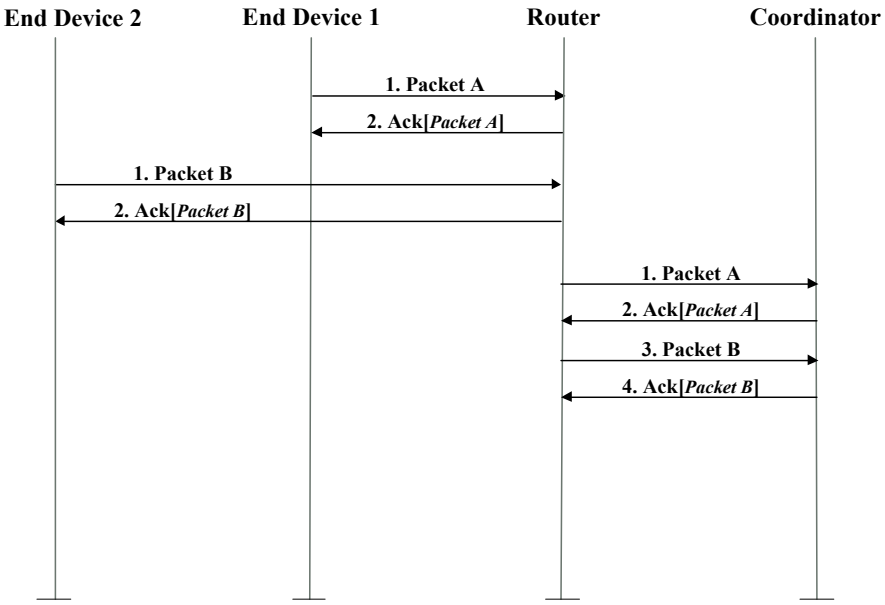


Figure 4.4 – Transmission model for tree topologies with Z-Stack.

The graphic in Figure 4.5 represents the delivery ratio with the TIMAC implementation when an increasing number of sensor nodes transmitting in mode A for the star and 2-hop tree topologies are used. In order to compare the TIMAC performance with the Z-Stack using the same traffic load, the lengths of the transmitted packets were made equal to those that were used in the Z-Stack measurements. Since the two stacks have different overheads, 16 bytes of dummy information were added to the payload of the TIMAC packets. The results with the acknowledgements enabled are worse than the ones obtained using the Z-Stack. This difference is explained by the fact that the Z-Stack network layer may retransmit a packet if the MAC layer has failed to transmit it. By default, the Z-Stack network layer is configured to perform one retransmission attempt.

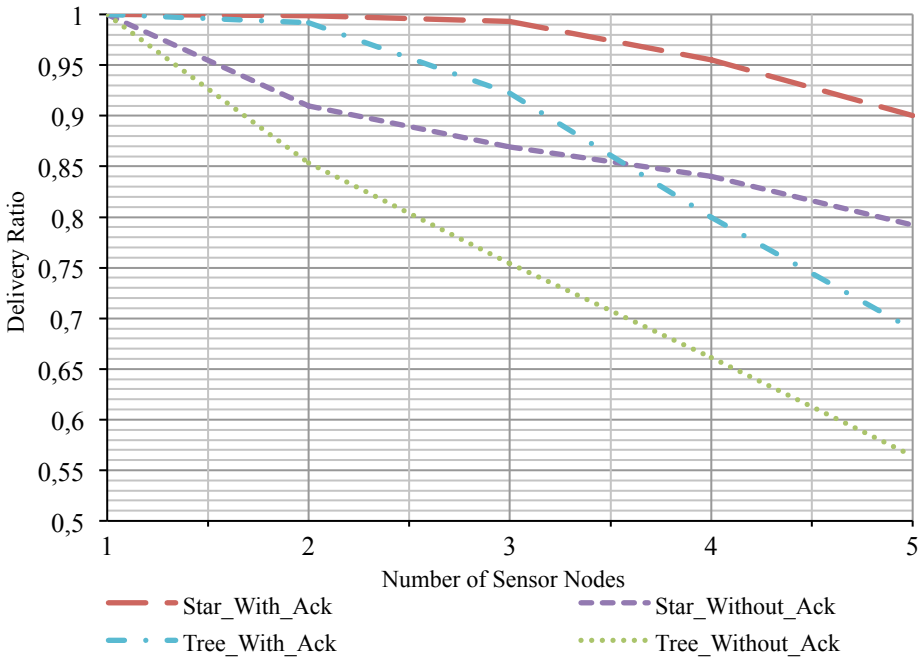


Figure 4.5 - Delivery ratio measured with TIMAC for an increasing number of sensor nodes transmitting in mode A.

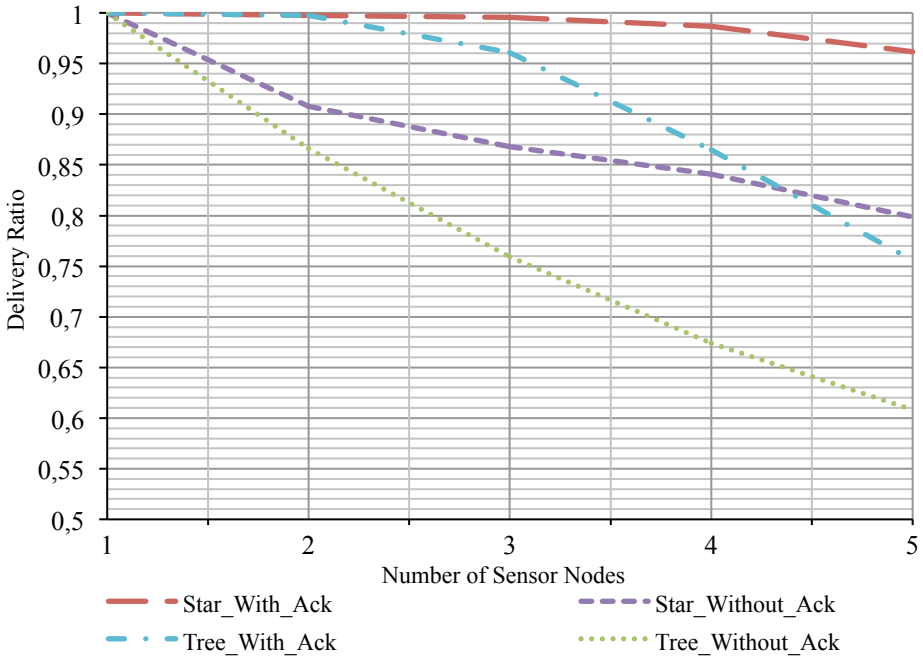


Figure 4.6 - Delivery ratio measured with TIMAC for an increasing number of sensor nodes transmitting in mode B.

Although the Z-Stack network layer retransmissions are disabled when the acknowledgments are not used, results for the tree topology without ACKs are also better than the ones observed with TIMAC. This is due to the router network layer, which has the capability of buffering the received packets and relaying them when the end devices are idle.

On the other hand, the application that simulates the router in the TIMAC relays the received packets immediately.

4.1.2.2 Network Delay

Figure 4.7 and Figure 4.8 show the measured average and maximum end-to-end delay, respectively, in function to the number of sensor nodes for both Z-Stack and TIMAC operating in mode A. Acknowledgments are used on both topologies. The TIMAC delays are lower than those measured with the Z-Stack due to the lower processing load introduced by the TIMAC. As expected, the average and maximum delays increase along with the number of sensor nodes, because the medium access contention, collisions and retransmissions also increase. For 3 to 5 end devices, the maximum delay for the tree topology with Z-Stack increased significantly. This higher delay is consequence of the packet buffering that occurs during the router discovery procedure, which is triggered by the route maintenance protocol.

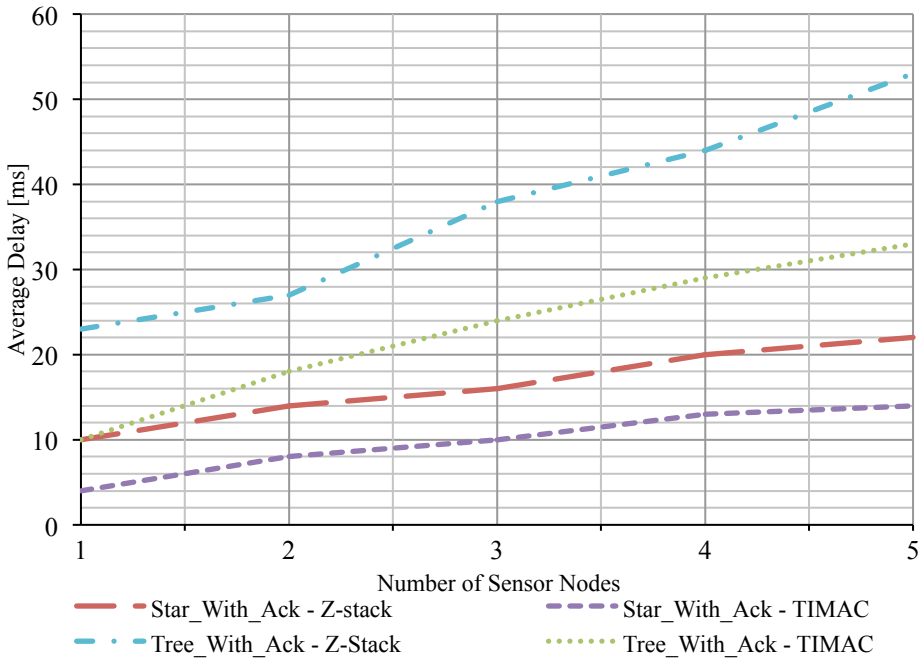


Figure 4.7 - Average delay as a function of the number of sensor nodes transmitting in mode A for both Z-Stack and TIMAC.

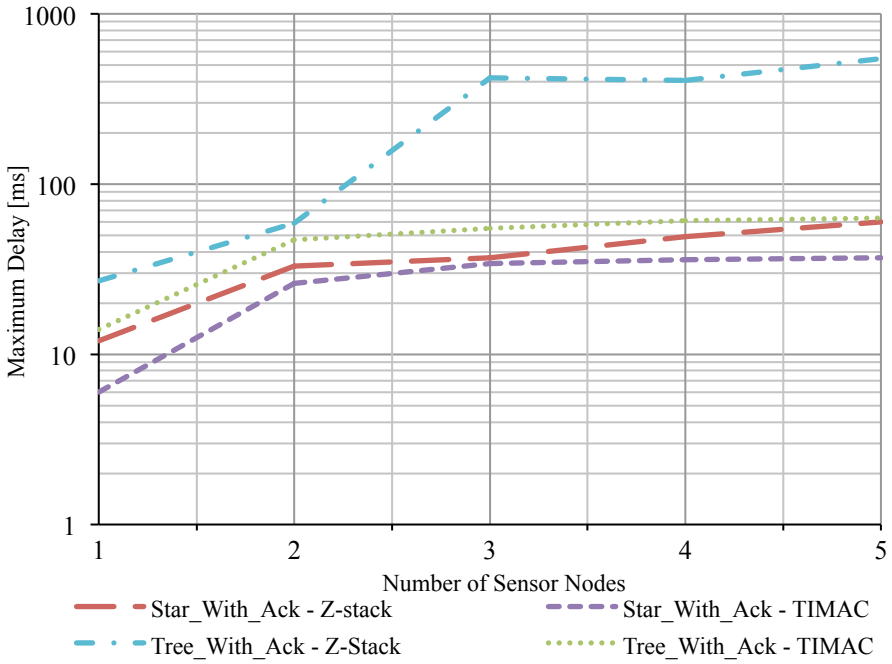


Figure 4.8 - Maximum delay as a function of the number of sensor nodes transmitting in mode A for both Z-Stack and TIMAC

In mode B, the observed delays were slightly smaller due to the smaller packet length, which in turn also decreases the packet transmission time. However, in Figure 4.9, which illustrates the maximum delay in function of the number of sensor nodes for both the Z-Stack and the TIMAC transmitting in mode B, it was noticed that the maximum delay for the tree topology with Z-Stack increases significantly, just as in the same experiment performed in mode A, but the higher delay was found just for 4 and 5 end devices. This is also the result of the packet buffering in the network layer caused by the router discovery procedure when the route maintenance protocol is executed. In this case, the higher delay was not verified for the 3 end devices due to the smaller packet length, which decreased the network load, and consequently, the failures in the packets relaying in the router also decreased; reducing the probability of the router discovery procedure being executed more frequently.

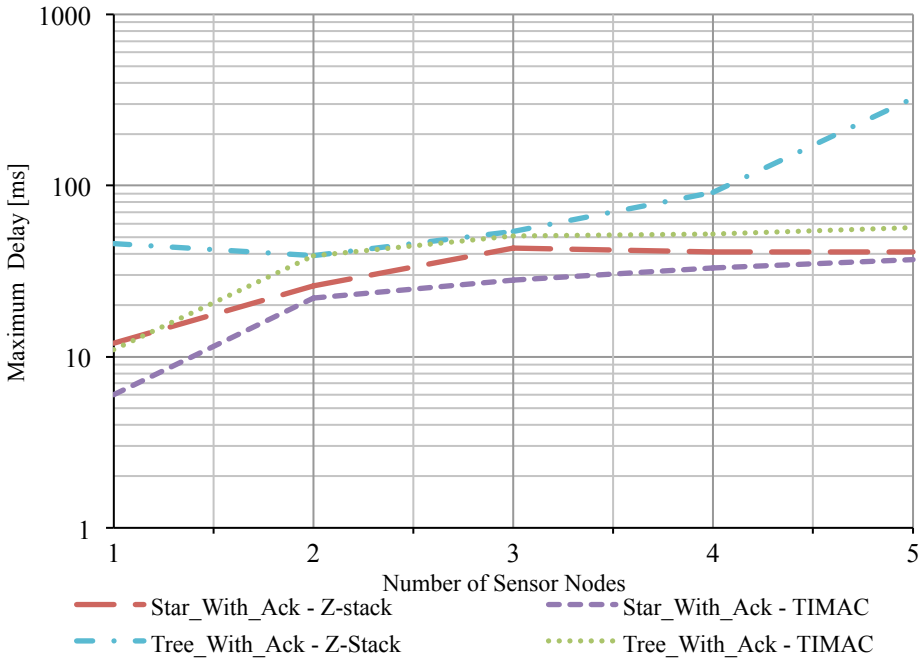


Figure 4.9 - Maximum delay as a function of the number of sensor nodes transmitting in mode B for both Z-Stack and TIMAC.

The delay results presented do not include the packetization delay, which approaches the value of the transmission period for the first sample of the packet and decreases for the subsequent samples.

4.2 Clock Drift Results

4.2.1 Clock Drift Measurements

Table 4.1 specifies the differential clock drifts between a device *n* and the BS ($D_{BS,EDn}$), measured using the process described in section 3.3.1.1, as well as the respective drift values between devices *n* and *m* ($D_{EDn,\ EDm}$), in ppm. The average clock drift among the five devices ($\overline{D_{ED}}$) that were tested is 1.48 ppm.

Table 4.1 - Measured and calculated differential clock drifts in ppm.

Device n	$D_{BS, EDn}$ (ppm)	$D_{EDn, ED0}$ (ppm)	$D_{EDn, ED1}$ (ppm)	$D_{EDn, ED2}$ (ppm)	$D_{EDn, ED3}$ (ppm)	$D_{EDn, ED4}$ (ppm)
0	3,6	0				
1	0,1	3,5	0			
2	-1	4,6	1,1	0		
3	-0,5	4,1	0,6	-0,5	0	
4	0,2	3,4	-0,1	-1,2	-0,7	0

4.2.2 Clock Drift Model Validation

Through the proposed network configuration described in section 3.3.2.1, we obtained a value for T_{Tx_max} (the maximum period needed by a device for transmitting a packet and receiving the respective acknowledgment) equal to 4.416 ms using equation 3.13, therefore T_{Vul} (the vulnerability time window), given by equation 3.17, is equal to 8.448 ms.

We have chosen devices 0 and 1 for the experimental measurements and model validation. For these nodes, the differential clock drift is $D_{ED1,ED0}$ = 3.5 ppm, as shown in Table 4.1, and T_{ED} is equals to 100 ms. Using these values, in equation 3.9, we obtain a T_{Int} value (the interference period during which two devices will compete for the channel) of approximately 40 minutes. The T_{IntRep} period (interference repetition interval), which can be obtained through equation 3.8, is approximately 7 hours and 56 minutes. If the average differential clock drift among all devices were used ($\overline{D_{ED}}$ = 1,48 ppm), T_{Int} and T_{IntRep} would be, in average, 1 hour and 35 minutes and 18 hours and 46 minutes, respectively, which means that the interference between devices, and possible network performance degradation, would last longer but would also take a longer period to repeat..

Figure 4.10 shows the results obtained in this experiment, which started at 18:15:10 p.m. and ended at 13:02:44 a.m. the next day. The DR was 100% most of the time of this experiment, which corresponds to non-interference periods. The DR decreases when the interference period starts, and reaches a minimum when both devices are generating packets at the same time. In the presented results, the first interference period started at 23:48:29 and ended at 00:25:47, while the second one started at 07:41:25 and ended at 08:18:25 respectively. Therefore, the interference periods lasted, on average, approximately 37 minutes. The interval between interferences is approximately 7 hours and 53 minutes. The

measured T_{Int} and T_{IntRep} periods have an error of approximately 7.5% and 0.6%, respectively, in relation to the same periods predicted by the theoretical model.

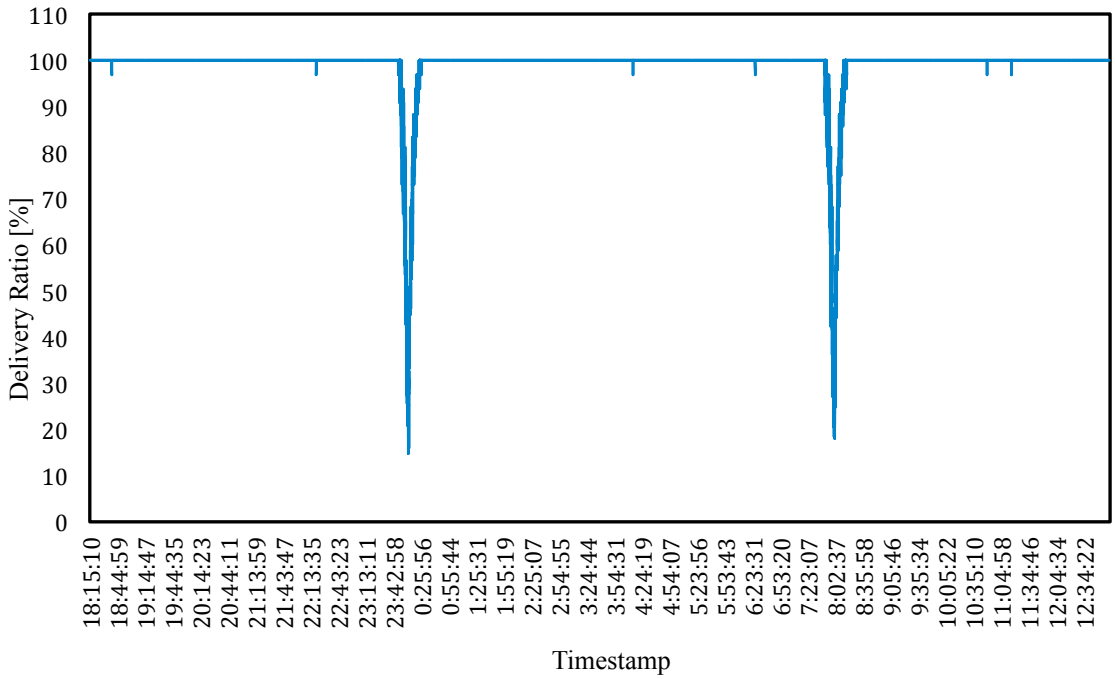


Figure 4.10 - Delivery ratio using a 60 message window in a two hidden-nodes start topology in an anechoic chamber.

Table 4.2 shows the results for the interference period and the intervals of the interference periods for the same experiment, but with the nodes transmitting packets every 50 ms instead of 100 ms, which allows us to decrease the value of the T_{IntRep} period.

Table 4.2 – Interference and interference repetition periods.

Interference Period			Interference Repetition Period		
Started	**Ended**	**Total**	**Started**	**Ended**	**Total**
18:37:20	19:13:40	00:36:20	18:37:20	22:32:44	03:55:24
22:32:44	23:10:11	00:37:27	22:32:44	2:28:36	03:55:52
2:28:36	3:06:34	00:37:58	2:28:36	6:25:53	03:57:17
6:25:53	7:04:04	00:38:11	6:25:53	10:25:33	03:59:40
10:25:33	11:02:35	00:37:02	10:25:33	---	---

The main aim of this new experiment was to record more occurrences of T_{Int} and T_{IntRep}, to prove the consistency of the model that we want to validate. For this new configuration, the interference period T_{Int} predicted by the model maintains the value of 40 minutes, but the interference repetition period T_{IntRep} is now approximately 3 hours and 58 minutes, since

T_{IntRep} depends on the packet transmission period. On average, the measured results show T_{Int} periods of approximately 37 minutes and T_{IntRep} periods of approximately 3 hours and 57 minutes, which result in errors of approximately 7,5% and 0,4% in relation to the results obtained through the model, respectively.

The experimental test proves the validity of the proposed model since the results are close to those that were predicted. The difference between the values that were predicted by the model and the experimental results may be related to errors in the measurement of the device's clock drift and in the measured time boundaries of T_{Int}

The errors in the T_{Int} measurements are related to the very low probability of collisions at the beginning and at the ending of these periods, which makes it difficult to determine their boundaries. Considering that t_{Vul_init} is the instant of time when a T_{Vul} period begins and t_{Vul_end} is the instant of time when a T_{Vul} period ends, there is a given probability of collision of approximately 1.6% between the time limits shown in equations 4.2 and 4.3.

$$t_{Vul\_init} \leq t_{ED1} < t_{Vul\_init} + T_{Vul\_UBP} \tag{4.2}$$

$$t_{Vul\_end} - T_{Vul\_UBP} \leq t_{ED1} < t_{Vul\_end} \tag{4.3}$$

That is because the nodes' transmissions will collide with each other if the following conditions are met:

- the $T_{Backoff}$ period of ED1 and ED2 is $T_{Backoff\_max}$ (7 UBPs) and $T_{Backoff\_min}$ (0 UBPs), respectively, during the interval presented in equation 4.2;
- the $T_{Backoff}$ period of ED1 and ED2 is $T_{Backoff\_min}$ and $T_{Backoff\_max}$, respectively, during the interval presented in equation 4.3.

T_{Vul_UBP} represents the period of a UBP in the vulnerability window, which is equal to 320 µs. The period of a UBP in T_{int}, T_{int_UBP}, can be obtained through:

$$T_{int\_UBP} = \frac{T_{Vul\_UBP}}{D_{ED1,ED2}}, \tag{4.4}$$

which corresponds to approximately 91 seconds. In the measurements, we assumed that the T_{Int} starts when the DR starts to drop continuously from 100% and ends when DR returns to a constant 100% level. However, in the first and last 91 seconds of T_{Int}, the probability of collision is very low, which makes it difficult to observe with precision these instants of time.

When these first and last 91 seconds are added to the measured 37 minutes in T_{Int}, it is obtained, approximately, the 40 minutes predicted by our model, which permits to confirm its validity.

4.3 Hidden Nodes Results

4.3.1 Hidden Node Scenario Results

In the proposed hidden node experimental setup, described in section 3.4.1.1, two synchronized ZigBee end devices hidden from each other transmit data to a coordinator in Star_With_Ack and Star_Without_Ack network modes, using the traffic mode B. The measured delivery ratio for the Star_With_Ack mode was approximately 90%. For the Star_Without_Ack mode, the result was of approximately 13%, which is very close to the minimum DR verified in the clock drift experiment, shown in Figure 4.10. This means that, in the worst-case scenario, the DR of a simple network constituted by two end devices decreases considerably. This fact may seriously compromise the reliability of the network and consequently make it unable to support WBANs because the network may not fulfill their applications requirements. Although this test case considers the worst-case scenario in terms of contention, due to the synchronization of packet generation times, the network is composed by only two end devices. If the network was to be constituted by several hidden nodes, the network performance could be seriously degraded, particularly in non-acknowledged transmission modes, due to the observed DR values in that case.

Previous measurements showed delivery ratios in the absence of hidden nodes of nearly 100% and 92% for two end devices transmitting in modes Star_With_Ack and Star_Without_Ack, respectively (see Figure 4.2). When compared with the results without hidden nodes, the experimental results with hidden nodes show accentuated decreases in the delivery ratio (10% for the Star_With_Ack mode and 79% for the Star_Without_Ack mode), especially in the non-acknowledged mode.

However, the hidden node experimental results differ from those obtained for the Star_Without_Ack network mode through the theoretical model presented in section 3.4.1, which was a DR of 3.125%. In order to discover the origin of this discrepancy we analyzed the log file in which all the information of the received packets during the experimental tests

were recorded. Figure 4.11 shows a record of some received packets during the hidden nodes experiment in the mode Star_Without_Ack_100. Each line of the board refers to a unique packet record, which includes the packet timestamp (Timestamp), the transmitting node (Node ID), the packet sequence number (Packet ID), the RSSI (Received Signal Strength Indicator) and the LQI (Link Quality Indicator). Through the Packet ID, we may observe that most of the packets were lost due to collisions caused by the HNP, but some of the packets were received when they weren't expected. For example, the packets with the PacketID 28, 30, 33, 39, 53 and 70 from the NodeID 1, in theory, should not have been received because the packets from NodeID 2 were not received.

```
1  Timestamp(ms)= 0     NodeID= 1  PacketID= 1   RSSI(dBm)= -78 LQI= 23
2  Timestamp(ms)= 4     NodeID= 2  PacketID= 1   RSSI(dBm)= -78 LQI= 23
3  Timestamp(ms)= 2699  NodeID= 1  PacketID= 28  RSSI(dBm)= -76 LQI= 28
4  Timestamp(ms)= 2899  NodeID= 1  PacketID= 30  RSSI(dBm)= -77 LQI= 28
5  Timestamp(ms)= 3200  NodeID= 1  PacketID= 33  RSSI(dBm)= -76 LQI= 28
6  Timestamp(ms)= 3799  NodeID= 1  PacketID= 39  RSSI(dBm)= -76 LQI= 28
7  Timestamp(ms)= 4300  NodeID= 1  PacketID= 44  RSSI(dBm)= -78 LQI= 23
8  Timestamp(ms)= 4304  NodeID= 2  PacketID= 44  RSSI(dBm)= -78 LQI= 23
9  Timestamp(ms)= 5199  NodeID= 1  PacketID= 53  RSSI(dBm)= -76 LQI= 28
10 Timestamp(ms)= 6902  NodeID= 1  PacketID= 70  RSSI(dBm)= -78 LQI= 23
```

Figure 4.11 - Record of received packets in the hidden-node experiment in mode star_without_ack.

The coordinator should only receive packets that were sent from the nodes in the absence of collision, which, according to the previous analysis, is only possible if node 1 selects the $T_{Backoff\_min}$ and node 2 selects the $T_{Backoff\_max}$ when the CSMA-CA is executed, or vice-versa. Therefore, it should not be possible receive packets from only one of the nodes, which was not the case. Using a packet sniffer, it was possible to observe that both nodes transmit their packets when triggered and if one of the nodes was disabled, the coordinator receives all the packets from the other node. It was also observed that if the transmit power of the nodes were controlled in a way for the coordinator to receive equal power from both nodes, the DR decreased, while it increased if the packets were received with different power. This suggests that the difference between theoretical and experimental results may be related with the capture effect, where, in the presence of other overlapping interfering packets, a packet may be correctly received if its power is sufficiently greater than the power of interfering packet.

The DER values obtained in the HNP experiment prove that hidden nodes have great influence in the network performance and in some cases it may be significantly degraded.

Since WBAN applications demand specific QoS requirements to be provided by the network, a solution to mitigate the HNP becomes necessary.

4.3.2 HNPAvoidance Protocol Evaluation Results

Figure 4.12 shows the results of the DR measured during the experimental evaluation described in section 3.4.2.1. The experiment started at 14:23:22 and finished at 16:57:30 on the next day, which allowed the observation of the HNPAvoidance protocol within a long period of time. Through this experiment, the validity of the proposed protocol to solve the HNP is confirmed, given that the measured DR was always 100%. This results contrast with the experimental results are present in section 4.2 (see Figure 4.10), where the network was affected by the clock drift effect. Once the transmissions are set by the beacon reception event in the network devices, node transmissions are perfectly synchronized with the clock of the coordinator and are scheduled to occur at distinct parts of the superframe period by the HNPAvoidance protocol.

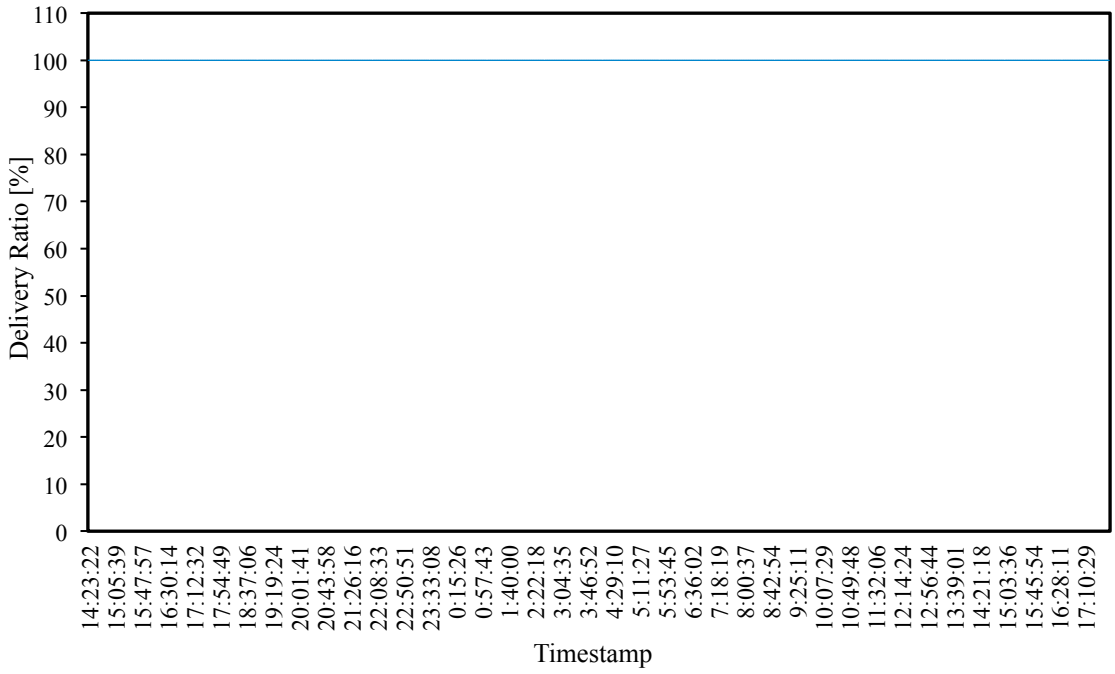


Figure 4.12 - Delivery ratio using a 60 message length window with two hidden nodes in a star topology.

4.4 Results of Body Interference in RF Communications

Table 4.3 and Table 4.4 present the results for the experiments regarding the interference

caused by the human body in radio communications, whose experimental setup is described in section 3.5.1. The former table shows results that were obtained within an anechoic chamber. The latter table shows the results obtained in an indoor environment. Each experiment was repeated three times where it was measured the average RSSI and the PER for each experiment. Thus, the results presented in Table 4.3 and Table 4.4 represents the average RSSI and PER of the three experiments as a function of the body segment where the sensor device was placed, the distance between the test subject and the BS in meters, the sensor device transmission power in dBm and the angle θ.

The results obtained in the anechoic chamber reveal that the RSSI in the coordinator decreases as the human body is positioned between the transmitting device and the coordinator (θ = 180º for a sensor placed in the chests; θ = 90º for a sensor placed in the arm). In this situation, the RSSI was found to be smaller when the sensor was placed on the arm instead of when it was placed on the chest.

In the worst-case scenario that was tested, where the sensor node's transmission power was set to -12 dBm and the sensor device was placed on the arm at an angle θ of 90º, the RSSI was on the lower limit of the sensitivity in the CC2530 transceiver. In this scenario, it was not possible to measure the PER due to losses of connectivity during the experiment caused by the lack of signal strength to overcome the obstruction caused by the body.

Further experiments showed that the received RSSI is smaller, for the sensor device placed on the chest, when θ is 90º than when it is 270º. Likewise, the received RSSI is smaller when θ is 0º for the sensor device placed on the arm than when it is 180º. These results suggest that the position of the sensor device's antenna also influences the RSSI and the PER in the coordinator.

Table 4.3 – PER and RSSI values obtained inside an anechoic chamber.

Body Segment	Distance from BS (m)	TX Power (dBm)	θ	PER (%)	RSSI (dBm)
chest	2	3	0	0	-47
chest	2	3	90	0.050	-61
chest	2	3	180	0.100	-86
chest	2	3	270	0.050	-63
chest	2	0	0	0.050	-52
chest	2	0	90	0.100	-65
chest	2	0	180	0.794	-85
chest	2	0	270	0.100	-72
chest	2	-3	0	0.050	-55
chest	2	-3	90	0.050	-70
chest	2	-3	180	34.167	-96
chest	2	-3	270	0.050	-76
chest	2	-12	0	0	-64
chest	2	-12	90	0.050	-77
chest	2	-12	180	37.343	-97
chest	2	-12	270	0	-80
arm	2	3	0	0.200	-55
arm	2	3	90	3.661	-92
arm	2	3	180	0.200	-64
arm	2	3	270	0.200	-48
arm	2	0	0	0	-58
arm	2	0	90	11.190	-93
arm	2	0	180	0.200	-67
arm	2	0	270	0.200	-51
arm	2	-3	0	0.200	-61
arm	2	-3	90	25.706	-97
arm	2	-3	180	0.200	-72
arm	2	-3	270	0.200	-55
arm	2	-12	0	0	-74
arm	2	-12	90	----------	----------
arm	2	-12	180	0.200	-79
arm	2	-12	270	0.200	-65

The results from the experiments performed in the indoor environment show some differences from those obtained in the anechoic chamber due to signal propagation effects such as multipath fading, which is consequence of signal reflections on surfaces of the classroom environment, and shadowing, which occurs when the human body is positioned between the transmitter and receiver devices. When compared with the results collected from the anechoic chamber experiment for the same distance and transmission power, these tests presented better RSSI and PER results for the sensor node placed on the chest with θ equals to 180º. Likewise, tests with the sensor node placed on the on the arm and with θ equals to 90º showed better RSSI results in this case but no significant different PER values were detected,

which can be a reflection of the interference caused by the multipath fading. These results show that multipath propagation effects have great influence on the received power when the human body obstructs the signal and there is no line of sight between the transmitter and the receiver.

Table 4.4 - PER and RSSI values collected in an indoor environment.

Body Segment	Distance from BS (m)	TX Power (dBm)	θ	PER (%)	RSSI (dBm)
chest	2	0	0º	0.200	-52
chest	2	0	90º	0. 200	-62
chest	2	0	180º	0. 200	-69
chest	2	0	270º	0. 200	-61
chest	5	0	0º	0. 200	-57
chest	5	0	90º	0. 200	-68
chest	5	0	180º	2.913	-85
chest	5	0	270º	1.186	-77
arm	2	0	0º	0.399	-72
arm	2	0	90º	11.817	-76
arm	2	0	180º	0. 200	-60
arm	2	0	270º	0. 200	-52
arm	5	0	0º	0.398	-69
arm	5	0	90º	0. 200	-68
arm	5	0	180º	0. 200	-71
arm	5	0	270º	0. 200	-53

4.5 Software Delay Results and Model Validation

4.5.1 Software Delay Results

Table 4.5 shows the values measured for diverse parameters of the software delay model defined in section 3.6.2, expressed in milliseconds. These results are specific to the CC2530 and the Z-Stack. $T_{APP \rightarrow MAC}(n)$ was measured on an end device, $T_{MAC \rightarrow APP}(n)$ was measured on a coordinator and the $T_{MAC \rightarrow NWK \rightarrow MAC}(n)$ parameter was measured on a router. Packets with payloads of 10 to 90 bytes were used. As the application level payload increased so did these delay values, due to the higher processing load and transmission times incurred when bigger packets are generated by the application. The $T_{MAC\_RTT}(n)$ values were found consistent with the theoretical values predicted by the model when $T_{MAC\_PHY}(n) = 0$, which proves that this parameter does not introduce significant delays in this testbed. The value measured for the $T_{MACtoAPP\_Conf}$ was 1.67 ms.

Table 4.5 – Values of the model parameters.

payload length *n* (byte)	$T_{APP \rightarrow MAC}(n)$	$T_{MAC \rightarrow NWK \rightarrow MAC}(n)$	$T_{MAC \rightarrow APP}(n)$	$T_{RTT}(n)$
10	3.28	4.32	1.78	3.26
20	3.37	4.41	1.87	3.8
30	3.48	4.47	1.90	3.91
40	3.57	4.53	1.94	4.25
50	3.68	4.61	2.01	4.55
60	3.77	4.67	2.07	4.87
70	3.90	4.72	2.15	5.29
80	3.95	4.80	2.16	5.48
90	4.04	4.89	2.23	5.84

4.5.2 Model Validation

In this section, results from simulation integrating the software delay model proposed in this work and using the measured values for its parameters are now assessed. The simulation results are compared to the experimental results for star and 2-hop tree topologies with sensor nodes transmitting in mode 2 and in mode A, using the same evaluation scenarios for the maximum goodput and for the delivery ratio and delay. Further results were obtained with the sensor nodes transmitting in mode B, but these are not described because similar conclusions to mode A were taken.

The simulations results were achieved after adding the retransmission functionality of the Z-Stack network layer to the network module of the unslotted CSMA-CA IEEE 802.15.4 simulator.

4.5.2.1 Maximum Goodput Simulation Results

Figure 4.13 shows the results for the maximum measured and simulated goodput with a sensor node transmitted in mode 2 for star and 2-hop tree topologies. In contrast with the simulation results without the model, these results show that the simulations are closer to the results obtained in the experimental analysis. Although the model approximates the simulations with the experimental results, the maximum relative error observed is significant, particularly in the 2-hop tree topology.

For the star network topology, the maximum relative error for the experimental results is of approximately 3.7% (10 byte payload). The deviations from the experimental results may be related to imprecisions in the measurements when the experimental tests were performed,

whose results may slightly vary from test to test. During the simulations a slight variation, in the order of hundreds of microseconds, for the values of the model's parameters allowed an even closer approximation of the simulations to the experimental results. This suggests that the measured delays may also have some imprecisions.

Influenced by the route maintenance protocol in the 2-hop tree topology, the experimental results show a higher deviation from the simulations. In this topology, the maximum relative error observed was approximately 12.6% for a payload length of 80 bytes. Since the route maintenance protocol is not implemented into the simulator and the delay parametric model does not consider the delays introduced by it, the results suggest that the proposed model may not be suitable when simulations of multi-hop topologies must be executed. As explained before, the route maintenance protocol may buffer packets in the router until it is ready to relay the packet when a new route to the coordinator is acquired. During this procedure, the router may drop packets.

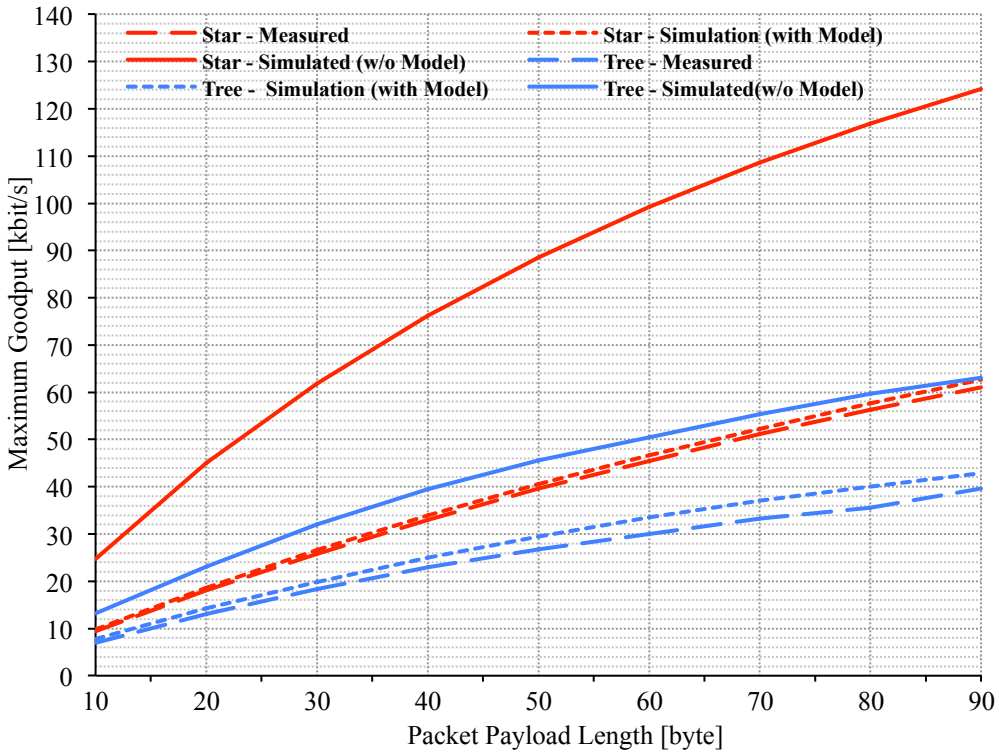


Figure 4.13 –Goodput measured and simulated for star and 2-hop tree topologies in mode 2.

4.5.2.2 Delivery Ratio Simulation Results

Figure 4.14 shows the simulation results for non-acknowledged star and 2-hop tree topologies in function of the number of sensor devices, for the evaluation scenario used in section 4.1.2. In order to validate the software delay model previously proposed, the results

from the Z-Stack experimental evaluation and the results of simulations without using the model for the delivery ratio are also presented.

For the star network topology, the simulations showed no significant differences from simulations without the model. In fact, simulations were worst and the divergences from experimental results are still accentuated. The simulations for the 2-hop tree topology presented better results when the model is used but also accentuated differences to the experimental results were found. These differences are caused mainly by the buffering mechanisms of the Z-Stack network layer of the router that has the capability to buffer packets during high-contention periods and relay them when the network channel is idle.

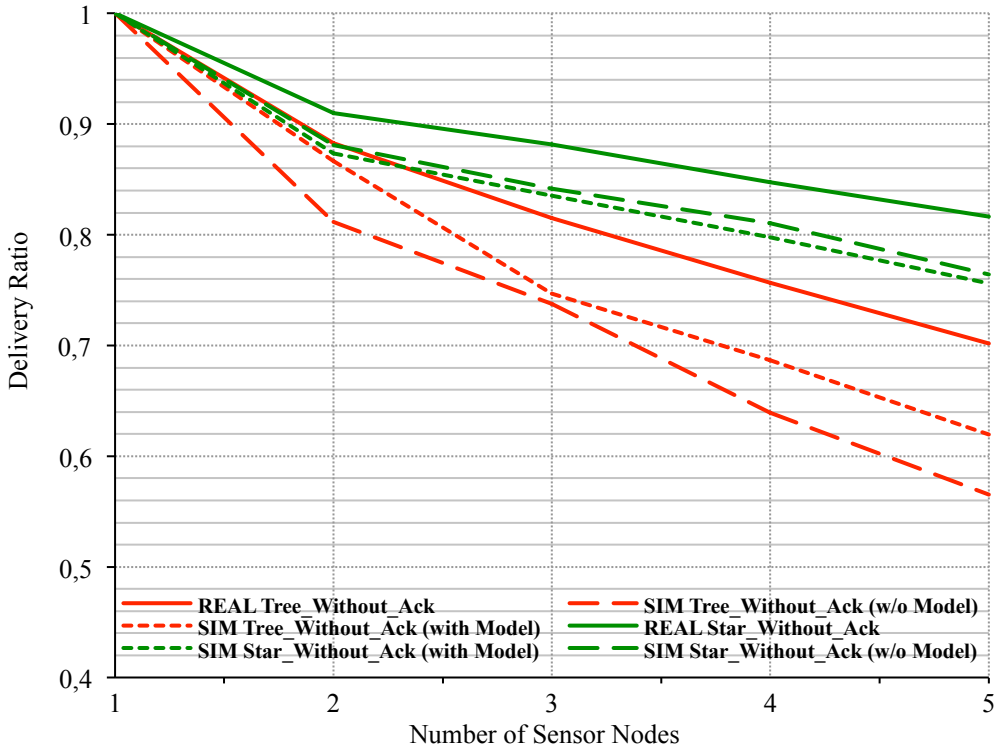


Figure 4.14 - Delivery ratio measured and simulated for an increasing number of sensor nodes transmitting in mode A.

In addition, simulations considering the acknowledgement mechanism were also carried. On those, for the star topology, it was found that both simulations with and without the delay model corresponds to the 100% of delivery ratio obtained in the experimental results. This was achieved mainly due to the network level retransmissions added into the network module of the simulator. Further simulations with 5 sensor nodes and without the network retransmissions showed a DR of approximately 99.4% and 99.5% with and without the model, respectively, which means that worst results are obtained when the model is used. For the 2-hop tree topology, the simulated results also showed 100% of delivery ratio, which was

well above of the experimental results for 3, 4 and 5 sensor nodes (see Figure 4.2). The divergence in the results can be explained by the Z-Stack route maintenance protocol, which is not implemented into the simulator.

Once again, these results suggest that the simulator may not be appropriate for simulating multi-hop topologies and introduction of the route maintenance protocol into the simulator may be a solution to improve simulations. However, this may not be entirely true because, in all simulations and experiments, the worst-case scenario was tested with nodes transmitting precisely at the same instant of time. In another situation, simulation results can be better, which also suggests that the definition of a model that distributes the nodes' traffic and its introduction into both the simulator and the physical platforms may be benefic for improving both simulation and experimental results, especially in multi-hop tree topologies with the Z-Stack.

4.5.2.3 Delay Simulation Results

In function of the number of sensor devices, the average and maximum delay results obtained through the simulations of acknowledged star and 2-hop tree topologies are shown in Figure 4.15 and in Figure 4.16, respectively. Non-acknowledged simulations were also performed but similar conclusions were found. Although most of the simulations using the model can approximate the simulations to the experimental results in relation to the same simulations without using the model, the average and maximum delays obtained show some discrepancy in relation to the experimental results, which are particularly significant in the 2-hop tree topology, as the number of sensor devices increases. This significant difference in the results for the acknowledged 2-hop tree topology is caused by the Z-Stack route maintenance protocol, which is triggered by the data-intensive traffic, introducing huge delays in the packets that are buffered in the network layer while it is being executed. Therefore, we conclude that, in order to increase the accuracy of simulation results, the route maintenance protocol should be implemented into the network layer of the simulator and a model to distribute the nodes' traffic should be implemented both into the physical platform and simulator in order to improve the experimental and simulation results, particularly for the evaluation of multi-hop network topologies with data-intensive and periodic traffic scenarios.

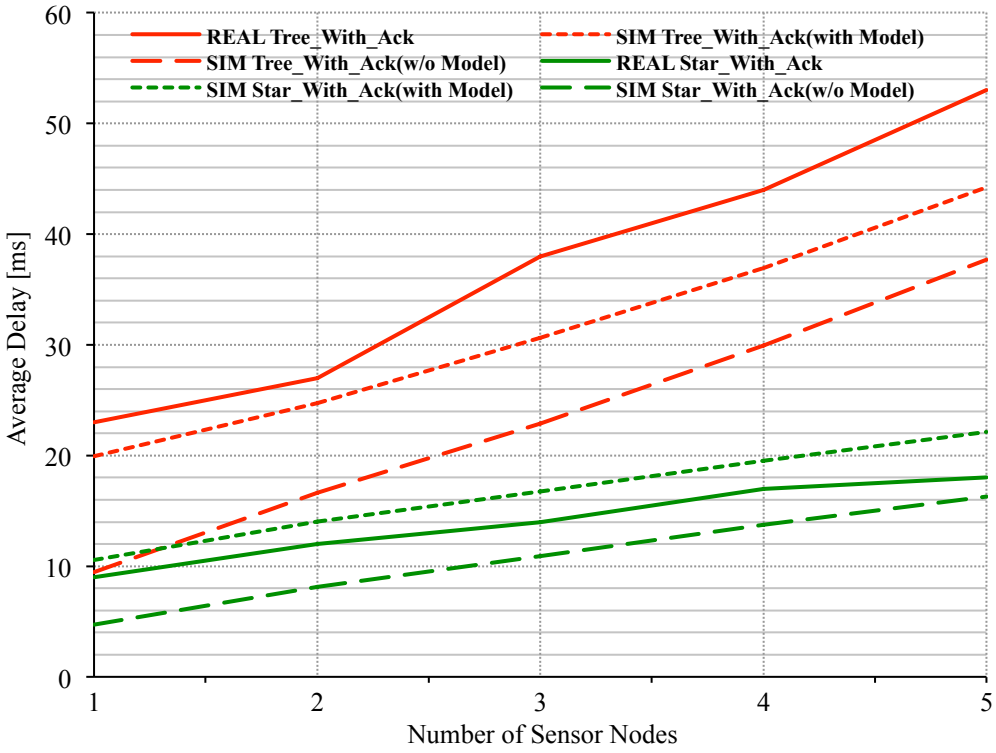


Figure 4.15 - Average delay measured and simulated for an increasing number of sensor nodes transmitting in mode A

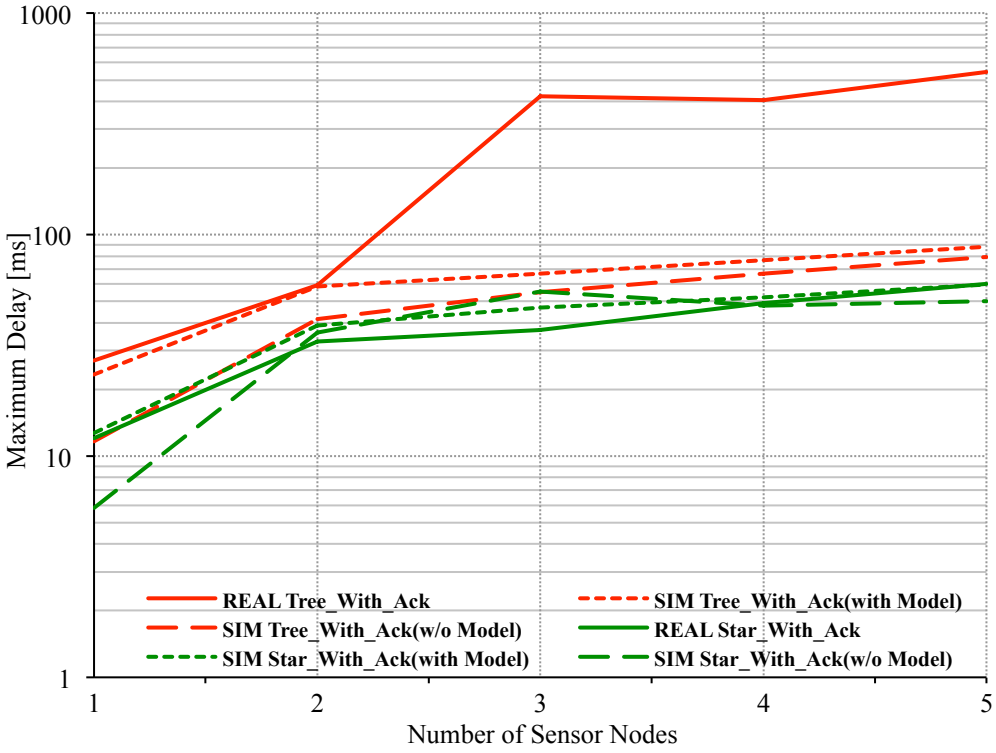


Figure 4.16 - Maximum delay measured and simulated for an increasing number of sensor nodes transmitting in mode A.

4.6 Summary

This chapter presented experimental performance evaluation results for BSNs using the ZigBee and IEEE 802.15.4 standards, with particular emphasis on high traffic load conditions and the usage of traffic parameters from a motion capture application.

Results confirm the importance of the acknowledgment and retransmission mechanism for increasing the reliability of the network. In this sense, a DR of 100% for the star topology and a DR exceeding 95% for the 2-hop tree topologies were achieved for networks with up to five sensor devices with the retransmission mechanism in the worst-case scenario of simultaneous traffic generation.

The router deadlock problem detected in other ZigBee implementations was not observed with the Z-Stack. On the other hand, in the 2-hop tree configuration, tests showed that successive periods of high traffic load caused the ZigBee router to start the route maintenance procedure, which has a negative impact on the delivery ratio and the network delay due to packets being dropped or buffered in the network layer while this procedure is running. A router blocking problem that lasted several seconds, caused by high traffic loads, was also observed. These results suggest that a mechanism to redistribute the traffic load generated by data-intensive devices along the time, in order to reduce contention, can be beneficial, since it would prevent the router from becoming overload with the traffic and, consequently, would contribute to maintain the expected level of network performance.

Differential clock drift measurements were provided for the devices used during the experiments. Based on these measurements, the validation of the proposed clock drift model was tested in two scenarios, each one comprising two hidden nodes with a differential clock drift of 3.5 ppm transmitting periodic traffic with packet length of 62 bytes to a coordinator in a star topology. The obtained results show that the interference period, where two nodes contend, lasts for approximately 37 minutes, while the interference repetition period is approximately 7 hours and 53 minutes for the scenario with packet transmission intervals of 100 ms. These long periods are due to the small clock drifts between the nodes. These experiments demonstrated the validity of the proposed clock drift model, where the predicted model results only showed a slight difference from those obtained in the measurements. The interference period may severely degrade the network performance, especially in the presence of hidden nodes, because nodes cannot backoff their transmissions, since they cannot hear each other.

Through an experimental test, we analyzed the performance of a network composed by two ZigBee end devices associated to a coordinator in a star topology. The end devices were hidden form each other and generated traffic simultaneously, in order to simulate a worst-case scenario. The end devices transmitted packets of 62 bytes every 100 ms to the coordinator. The results from this experiment showed that the network achieves a delivery ratio of 90% acknowledged mode, and only 13% for the unacknowledged mode. These results were well below those obtained in previous experiments in the absence of hidden nodes, which were approximately 100% and 92% for acknowledge and unacknowledged modes, respectively. These results, combined with the previous results regarding the clock drift effect, demonstrate that a mechanism to avoid the contention between nodes was needed, especially in the presence of hidden nodes, since WBAN application requirements would not be fulfilled. The proposed mechanism for this purpose, in the context of this work, is the HNPAvoidance protocol.

The HNPAvoidance protocol was implemented within a TIMAC application and was tested in a new HNP experiment. The results showed that the network delivery ratio was 100% in the unacknowledged mode. It can be concluded that the suggested protocol is able to solve the HNP. Besides this, the HNPAvoidance protocol also provides a solution for mitigating the clock drift effect in the network. This protocol can be beneficial to avoid contention between nodes even in the absence of hidden nodes.

Measurements to the signal's received power and to the packet error ratio were collected to analyze the influence of the human body on the radio communications. The results showed that, when the human body is positioned between the transmitter and the receiver device, the link degradation may be so severe that network connectivity is lost. It was also concluded that multipath effects in indoor environments may contribute to overcome the loss of connectivity due to non-line-of-sight communications in some situations, because the signal may be received through an alternative propagation path. However, there is no guarantee that these propagation effects will be beneficial in all circumstances.

Through the provided simulation results in this chapter, it was shown that the proposed software delay model, in conjunction with a previously implemented model for simulation of the IEEE 802.15.4 unslotted CSMA-CA, which were both described in the previous chapter, can increase the accuracy of the simulator, approaching it results to those observed in real ZigBee 2007 implementations. Nevertheless, simulations regarding multi-hop network

topologies based on the Z-Stack software with data-intensive traffic still showed significant deviations in relation to real measurements, due to the effect of route maintenance protocol implemented by the ZigBee network layer, which is not yet modeled in the simulator

Chapter 5

Conclusion

Wireless body area networks play an important and promising role of potential expansion in many industries such as: health, sports and entertainment. Ubiquitous environments are also expanding and in this context it is crucial to keep track of users' physical state for better understanding of how health problems arise, and in what conditions. Wireless monitoring, in indoor or outdoor environments, can bring benefits to patient's general well-being and can reduce caregivers' workload by allowing continued monitoring.

Standard-based low power wireless communication protocols were studied and evaluated, more specifically, the ZigBee and IEEE 802.15.4 standards, using hardware and software platforms from Texas Instruments. The first evaluations were conducted to achieve conclusions about the network maximum goodput. The IEEE 802.15.4 standard defines a maximum data rate of 250 kbit/s in the 2.4 GHz band, but the measurements showed that in all experiments the maximum goodput was well below. It was observed particularly that the overhead introduced by the stack implementation has a significant impact on the performance results. The design of future systems must take this into consideration because the obtained maximum goodput will be the upper limit in terms of achievable throughput for providing QoS to the end-user.

Overall, the performance of the ZigBee star topology was very good, even in the worst conditions, provided the acknowledgement mechanism was enabled. A router deadlock problem detected in other ZigBee implementations was not observed with the Z-Stack. However, we identified two different situations, triggered by periods of high traffic load, on which the ZigBee router stops relaying packets, causing a significant degradation on the

network performance.

A model to predict the clock drift effect in a non-beacon enabled ZigBee/IEEE802.15.4-based body area network is also proposed due to no support from specifications to overcome this issue. This model uses the average differential clock drift based on an accurate measurement procedure of each node's individual clock drift to estimate how much time two different nodes will contend for the wireless network channel (T_{Int}) and how long it takes for contention to repeat (T_{IntRep}). The estimation is based on a vulnerability window that defines when the transmissions of two nodes will interfere with each other in the wireless channel (T_{Vul}). The differential clock drift between two devices was measured and used in the experiments performed to test and validate the clock drift model. The obtained results showed that the interference and the interference repetition periods may last for a long time due to the short clock drifts between nodes. These experiments have also demonstrated the validity of the proposed clock drift model, where the predicted model results only showed a slight difference from those obtained in the experimental measurements. The interference period may significantly degrade the network performance or even cause stability issues, especially in the presence of hidden nodes, where nodes cannot backoff their transmissions because they cannot hear each other.

Through an experimental test, the performance of a ZigBee network with two end devices hidden from each other associated with a coordinator in a star topology was analyzed. The results from this experiment showed that, in the worst-case scenario where the nodes generate packets at the same time, the network achieves a delivery ratio of 90% in the acknowledged mode, and only 13% in the unacknowledged mode. These results were well below those obtained in previous experiments in the absence of hidden nodes, which was approximately 100% for the acknowledged mode and 92% for the unacknowledged mode. These results, combined with the previous results regarding the clock drift effect, demonstrate that a mechanism to avoid the contention between nodes was needed, especially in the presence of hidden nodes, since WBAN application requirements cannot be fulfilled. The mechanism proposed for this purpose, in the context of this work, is the HNPAvoidance protocol. This protocol aims to solve the HNP by separating the instants of time in which the nodes transmit their packets. Since WBANs sensor devices usually generate periodic data, the HNPAvoidance uses the superframe structure of the IEEE 802.15.4 in order to synchronize transmissions for each node. At the application level, the HNPAvoidance protocol creates a set of virtual time slots (VTS) to be assigned to these nodes. Then, each node uses its assigned

VTS to transmit at will using the unslotted CSMA-CA protocol**.** The results showed that the network delivery ratio was 100% in the unacknowledged mode. It can therefore be concluded that the suggested protocol is able to solve the HNP. Apart from this, the HNPAvoidance protocol also eliminates the clock drift effect in the network. This protocol can be beneficial to avoid contention between nodes even in the absence of hidden nodes.

Experiments regarding the interference of the human body in radio communications in a ZigBee-based WBAN were also accounted for. The signal reliability in WBANs may suffer from several aspects related to the body's posture, size, weight, and water content. Other sources of interference such as nearby WBANs, networks operating in the ISM license-free frequency bands or even other general sources of electromagnetic interference may also affect the signals reliability. These experiments were based on the measurements of the received power and the packet error ratio using the posture monitoring system (PMS). The results showed that, when the human body is positioned between the transmitter and the receiver device, the link quality may become degraded to a point where network connectivity may be completely lost. Another important conclusion is that multipath effects in indoor environments may contribute to overcome the loss of connectivity due to non-line-of-sight communications in some situations, because the signal may be received through an alternative propagation path. However, there is no guarantee that these propagation effects will be beneficial in all circumstances.

A model to analyze the delay introduced by ZigBee's software layers was developed. This model was then introduced into a simulator of the IEEE 802.15.4 protocol in order for it to give more accurate simulation results for ZigBee networks in general. The description of the simulator in which the model was introduced is also given. Essentially the model considers three fundamental components of delay: T_{TXtot}, which corresponds to the time necessary for an end device to fully complete the process of the transmission of a packet, T_{RXtot}, which is the time elapsed at the base station for a packet that is received in the application since it has been received in the PHY layer, and finally, the time needed for a router to relay a packet from the end device to the coordinator: the T_{Relay} component. Through the provided simulation results, it was concluded that the proposed software delay model, in conjunction with a previously implemented model for simulation of the IEEE 802.15.4 unslotted CSMA-CA, can increase the accuracy of the simulator in terms of delivery ratio in star topologies with up to five sensor nodes and approach results to those observed in real ZigBee 2007 implementations. This was achieved mainly due to the network level

retransmissions added into the simulator's network module. Nevertheless, simulations regarding multi-hop network topologies based on the Z-Stack software with data-intensive traffic and acknowledgements still showed significant deviations in relation to real measurements, due to the route maintenance protocol implemented by the ZigBee network layer, which wasn't modeled into the simulator.

As future work the following points can be considered:

- The implementation of a mechanism to redistribute the traffic load generated by data-intensive devices over time for 2-hop tree network topologies would reduce contention and prevent the router from becoming overloaded. Reducing the frequency that the route maintenance protocol is to be executed would contribute to maintain a better level of network performance;
- The inclusion of the route maintenance protocol of the ZigBee network layer in the simulator, which includes the modeled parametric delay presented in this work and the model of unslotted CSMA-CA of the IEEE 802.15.4 standard, in order to produce more accurate simulation results, particularly when multi-hop networks are simulated.

References

[Akyildiz02] I.F. Akyildiz, S. Weilian, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks", in Communications Magazine, IEEE, 40 (8). pp. 102-114, 2002.

[Barakah12] D. M. Barakah and M. Ammad-uddin, "A Suvey of Challenges and Applications of Wireless Body Area Netwok (WBAN) and Role of a Virtual Doctor Server in Existing Architecture", in Third International Conference on Intelligent Systems, Modelling and Simulation (ISMS), pp. 214-219, February, 2012.

[Batra11] A. Batra, A. Xhafa, "An Overview of IEEE 802.15.6", in Berkeley Wireless Research Center (BWRC) Sensor Workshop, June, 2011.

[Bradai11] N. Bradai, S. Belhaj, L. Chaari and L. Kamoun, "Study of Medium Access Control Mechanisms under IEEE 802.16.5 Standard", in 4th Joint IFIP Wireless and Mobile Networking Conference, pp. 1-6, October, 2011.

[Brzozowski09] M. Brzozowski, H. Salomon and P. Langendoerfer. "On Efficient Clock Drift Means and Their Applicability to IEEE 802.15.4", in 8th International Conference on Embedded and Ubiquitous Computing (UEC), IEEE/IFIP, pp. 216-223, 2010.

[Chen11] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, V. Leung, "Body Area Networks: A Survey", in ACM/Springer Mobile Networks and Applications, pp. 171–193, April 2011

[Chin12] C. A. Chin, G. V. Crosby, T. Ghosh and R. Murimo, "Advances and challenges of wireless body area networks for healthcare applications", in International Conference on Computing, Networking and Communications (ICNC), pp. 99-103, February, 2012.

[Curtis08] D. Curtis, E. Shih, J. Waterman, J. Guttag, J. Bailey, T. Stair, R. Greenes, L. Ohno-Machado, "Physiological Signal Monitoring in the Waiting Areas of an Emergency Room", in 3rd International Conference on Body Area Networks, Article No. 5, March, 2008.

[Falck07] T. Falck, J. Espina, J. Ebert, D. Dietterle, "BASUMA—The Sixth Sense for Chronically Ill Patients". in International Workshop on Wearable and Implantable Body Sensor Networks, pp. 4-60, April, 2006

[Farella08] E. Farella, A. Pieracci, L. Benini, L. Rocchi, A. Acquaviva, "Interfacing human and computer with wireless body area sensor networks: the WiMoCA solution", in Journal Multimedia Tools and Applications, Vol. 38 Issue 3, pp. 337–363, July, 2008

[Fiorini08] P. Fiorini, I. Doms, C. Van Hoof and R. Vullers, "Micropower Energy Scavenging", in 34th European Solid-State Circuits Conference, ESSDERC 2008, pp. 4-9, September, 2008.

[Gama09] O. Gama, P. Carvalho, J. A. Afonso, P. M. Mendes, "Quality of Service in Wireless e-Emergency: Main Issues and a Case-study", in Advances in Soft Computing, Vol. 51 , pp. 95-102, 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008, J. M. Corchado, D. I. Tapia and J. Bravo (Eds.), Springer-Verlag, 2009.

[Gama11] O. Gama, "A MAC Protocol for Quality of Service Provisioning in Adaptive Biomedical Wireless Sensor Networks", PhD thesis, University of Minho, 2011.

[Gislason08] D. Gislason, "Zigbee Wireless Networking", Newnes, 2008.

[Gopalan10] S. A. Gopalan, J. Park, “Energy-Efficient MAC Protocols for Wireless Body Area Networks: Survey”, in International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 739-744, 2010.

[Hwang05] L. Hwang, S. Sheu, Y. Shih, Y. Cheng. “Grouping Strategy for Solving Hidden Node Problem in IEEE 802.15.4 LR-WPAN”. in First International Conference on Wireless Internet, pp. 26-32, 2005.

[IEEE4-03] IEEE Std 802.15.4-2003: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks IEEE ed., 2003.

[IEEE4-06] IEEE Std 802.15.4-2006 – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), September 2006.

[IEEE4-07] IEEE. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) - Amendment 1: Add Alternate PHYs, 2007.

[IEEE4-09I] IEEE. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) - Amendment 2: Alternative Physical Layer Extension to support one or more of the Chinese 314-316 MHz, 430-434 MHz, and 779-787 MHz bands, 2009.

[IEEE4-09II] IEEE. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) - Amendment 3: Alternative Physical Layer Extension to Support the Japonese 950 MHz Band, 2009.

[IEEE6-08] 15-08-0644-09-0006-tg6-technical-requirements-document, Available at https://mentor.ieee.org/802.15/dcn/08/15-08-0644-09-0006-tg6-technical-requirements-document.doc, IEEE, 2008.

[Jennic10] Jennic, the Jn5139 ieee802.15.4/jennet evaluation kit. Jennic, Available at http://www.jennic.com/products/development_kits/jn5139_ieee802154_jennet_evaluation_kit , 2010.

[Kotz04] D. Kotz, C. Newport, R. Gray, J. Liu, Y. Yuan, C. Elliott, “Experimental Evaluation of Wireless Simulation Assumptions”, in 7th ACM/IEEE Inter. Symposium on Modelling, Analysis and Simulation of Wireless and Mobile Systems, Venice, Italy, October, 2004.

[Koubâa09] A. Koubâa, R. Severino, M. Alves and E. Tovar. “Improving Quality-of-Service in Wireless Sensor Networks by Mitigating “Hidden-Node Collisions””. In IEEE Transactions on Industrial Informatics, Special Issue on Real-Time and Embedded Networked Systems, Volume 5, No. 3, August 2009.

[Kwon09] C. H. Kwon, R. J. Tek, K. H. Kim and S. H. Yoo. “Dynamic Allocation Scheme for Avoiding Hidden Node Problem in IEEE 802.15.4”. in IEEE Transactions on Industrial Informatics, Vol. 5, Issue 3, August, 2009.

[Lamprinos06] Lamprinos et al., "Communication protocol requirements of patient personal area networks for telemonitoring", Technology & Health Care; vol. 14, issue 3, pp. 171-187, 2006.

[Li09] C. Li, H. B. Li, and R. Kohno, "Performance Evaluation of IEEE 802.15.4 for Wireless Body Area Network (WBAN)", in ICC 2009, June, 2009.

[Liang07] X. Liang and I. Balasingham, "Performance Analysis of the IEEE 802.15.4 based ECG Monitoring Network", in 7th IASTED International Conferences on Wireless and Optical Communications, Montreal, Canada, 2007.

[Lo2005] B. Lo and G. Z. Yang, "Key technical challenges and current implementations of body sensor networks", in 2nd International Workshop on Body Sensor Networks (BSN 2005), London, UK, April 2005.

[López11] H. F. López, "Remote Vital Signs Monitoring Based on Wireless Sensor Networks", PhD thesis, University of Minho, 2011.

[Macedo10] P. J. Macedo, "Desenvolvimento de Modelos de Simulação de Redes de Sensores sem Fios", Master thesis, University of Minho, 2010.

[Mattos96] M. K. Mattos, P. H. Biagioni and W. Bassi, "Electric field measurement on time domain generated by corona on insulators on distribution systems", in Conference Record of the 1996 IEEE International Symposium on Electrical Insulation, vol.321, pp. 328-330, 1996.

[Patel10] M. Patel and J.Wang, "Applications, challenges, and prospective in emerging body area networking technologies", in Wireless Communications, vol. 17, no.1, pp. 80-88, 2010.

[Pentland04] A. Pentland, "Healthwear: Medical Technology Becomes Wearable", in IEEE Computer Society Press, vol. 37, No.. 5, May, 2004.

[Ramli11] S. N. Ramli, R. Ahmad, "Surveying the Wireless Body Area Network in the real of wireless communication", in 7th International Conference on Information Assurance and Security, pp. 58-61, December, 2011.

[Shnayder05] V. Shnayder, B. Chen, K. Lorincz, T. R. Fulford-Jone and M. Welsh (2005) Sensor networks for medical care. In Harvard University Technical Report TR-08-05.

[Silva11] H. D. Silva, P. Macedo, J. A. Afonso and L. A. Rocha, "Design and implementation of a Wireless Sensor Network applied to Motion Capture", 1st Portuguese Conference on Wireless Sensor Networks (CNRS 2011), Coimbra, Portugal, pp. 45-52, March, 2011.

[Soomro06] A. Soomro and D. Cavalcanti, "Opportunities and Challenges in Using WPAN and WLAN Technologies in Medical Environments", in IEEE Communications Magazine, pp. 114-122, February, 2007.

[Suriyachai12] P. Suriyachai, U. Roedig and A. Scott, "A Survey on MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks", in IEEE Communications Survey & Tutorials, vol. 14, pp. 240-264, 2012.

[TICC2530-10] Texas Instruments. Second generation system-on-chip solution for 2.4 ghz ieee802.15.4/rf4ce/zigbee. Texas Instruments. [Online]. Available at: http://focus.ti.com/docs/prod/folders/print/cc2530.html, October, 2010.

[TI-HALAPI09] Z-Stack HAL Driver API. Available: http://www.ti.com.

[TI-MACAPI09] Z-Stack MAC API. Available: http://www.ti.com.

[TI-OSALAPI09] Z-Stack SAOL API. Available: http://www.ti.com

[TI-Z-StackAPI09] Z-Stack API. Available: http://www.ti.com

[Trigui09] I. Trigui, A. Laourine, S. Affes and A. Stéphenne, "Performance Analysis of Mobile Radio Systems over Composite Fading/Shadowing Channels with Co-located Interference", in IEEE Transactions on Wireless Communications, pp. 3448-3453, July, 2009.

[Uddin11] M. S. Uddin, N. B. Ali, N. H. Hamid, "Wave Propagation and Energy Model for Dynamic Wireless Body Area Networks", in International Conference on Electrical, Control and Computer Engineering (INECCE), June, 2011.

[Vieria07] M. A. Vieira, C. N. Junior, D. Junior and J. M. da Mata, "Survey on Wireless Sensor Network Devices", in IEEE Conference in Emerging Technologies and Factory Automation (ETFA'03), September, 2003.

[ZigBee04] ZigBee Alliance. ZigBee Specification - ZigBee Document 053474r06, Version 1.0, 2004.

[ZigBee07] The ZigBee Alliance, ZigBee Alliance Document 053474r17, ZigBee Specification, v. 1.0 r17. Alliance, Z. ed., 2007.

[ZigBee06] ZigBee Alliance. ZigBee Specification – ZigBee Document 053474r13, December, 2006.

[ZigBee11] ZigBee Alliance. ZigBee Alliance: Wireless Control that Simply Works, 2011. http://www.zigbee.org.