UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

ANALYSIS OF THE IEEE 802.15.4A ULTRA WIDEBAND PHYSICAL LAYER THROUGH WIRELESS SENSOR NETWORK SIMULATIONS IN OMNET++

by

Marthinus Alberts

Submitted in partial fulfillment of the requirements for the degree

Master of Engineering (Computer Engineering)

in the

Faculty of Engineering, the Built Environment and Information Technology

UNIVERSITY OF PRETORIA

February 2011

Document ver. 2.0

ACKNOWLEDGEMENTS

Throughout my life all personal accomplishments, even those meager in the eyes of the world, are attributed to two constants ... God and my family.

First of all it delights me to be able to thank my creator, Jesus Christ. Without God's ever abundant grace I would not have been able to achieve anything. To the Master of the universe be all glory, power and praise. May His Word and His Will find perfection in my life.

> *"Be strong and of a good courage; be not afraid, neither be thou dismayed:*
> *for the LORD thy God is with thee whithersoever thou goest." - Jos. 1:9*

Secondly God blessed me with a loving, caring family. To my beautiful wife, Simone, thank you for sacrificing so much time and being devoted to me through the fast falling troughs and slow rising crests that every voyage has to weather. To the rest of my family, close by and far, I am indebted to each of you for all the prayers, true words of motivation and ongoing support. I am a lucky man indeed.

A special word of thanks to my supervisor, Prof. Gerhard Hancke of the Department of Electrical, Electronic and Computer Engineering at the University of Pretoria for his guidance and utmost patience with me throughout this research project.

Finally my gratitude also goes out to Jérôme Rousselot, Ph.D. student at the Ecole Polytechnique Federale de Lausanne, for the Mixim-UWB framework and for granting me access to his research.

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Analysis of the IEEE 802.15.4a ultra wideband physical layer through wireless sensor network simulations in OMNET++

by

Marthinus Alberts

Supervisor:	Prof. G.P. Hancke
Department:	Electrical, Electronic and Computer Engineering
Degree:	Master of Engineering (Computer Engineering)

KEY TERMS

Ultra Wideband; Wireless Sensor Networks; IEEE 802.15.4; IEEE 802.15.4a; OMNET++; Zigbee; Wireless Personal Area Networks; WPAN; LR-WPAN; Network Simulator; PAN; Broadband; Sensor Mobility;

ABSTRACT

Wireless Sensor Networks are the main representative of pervasive computing in large-scale physical environments. These networks consist of a large number of small, wireless devices embedded in the physical world to be used for surveillance, environmental monitoring or other data capture, processing and transfer applications.

Ultra wideband has emerged as one of the newest and most promising concepts for wireless technology. Considering all its advantages it seems a likely communication technology candidate for future wireless sensor networks.

This paper considers the viability of ultra wideband technology in wireless sensor networks by employing an IEEE 802.15.4a low-rate ultra wideband physical layer model in the OMNET++ simulation environment.

An elaborate investigation into the inner workings of the IEEE 802.15.4a UWB physical layer is performed. Simulation experiments are used to provide a detailed analysis of the performance of the IEEE 802.15.4a UWB physical layer over several communication distances. A proposal for a cognitive, adaptive communication approach to optimize for speed and distance is also presented.

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Analise van die IEEE 802.15.4a ultra wyeband fisiese laag deur middel van draadlose sensor netwerk simulasies in OMNET++

deur

Marthinus Alberts

Studieleier: Prof. G.P. Hancke

Department: Elektriese, Elektroniese en Rekenaar-Ingenieurswese

Graad: Meester in Ingenieurswese (Rekenaar Ingenieurswese)

SLEUTELTERME

Ultra wyeband; Draadlose Sensor Netwerke; IEEE 802.15.4; IEEE 802.15.4a; OMNET++; Zigbee; Draadlose Persoonlike Area Netwerke; Netwerk Simulator; Breëband; Sensor Mobiliteit;

OPSOMMING

Draadlose Sensor Netwerke is die hoof verteenwoordiger vir deurdringende rekenarisering in groot skaal fisiese omgewings. Hierdie tipe netwerke bestaan uit ’n groot aantal klein, draadlose apparate wat in die fisiese wêreld ingesluit word vir die doel van bewaking, omgewings monitering en vele ander data opvang, verwerk en oordrag applikasies.

Ultra wyeband het opgestaan as een van die nuutste en mees belowend konsepte vir draadlose kommunikasie tegnologie. As al die voordele van dié kommunikasie tegnologie in ag geneem word, blyk dit om ’n baie goeie kandidaat te wees vir gebruik in toekomstige draadlose sensor netwerke.

Hierdie verhandeling oorweeg die vatbaarheid van die gebruik van die ultra wyeband tegnologie in draadlose sensor netwerke deur ’n IEEE 802.15.4a lae-tempo ultra wyeband fisiese laag model in die OMNET++ simulasie omgewing toe te pas.

’n Breedvoerige ondersoek word geloots om die fyn binneste werking van die IEEE 802.15.4a UWB fisiese laag te verstaan. Simulasie eksperimente word gebruik om ’n meer gedetaileerde analiese omtrent die werkverrigting van die IEEE 802.15.4a UWB fisiese laag te verkry oor verskillende kommunikasie afstande. ’n Voorstel vir ’n omgewings bewuste, aanpasbare kommunikasie tegniek word bespreek met die doel om die spoed en afstand van kommunikasie te optimiseer.

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
AOA	Angle Of Arrival
BAN	Body Area Network
BER	Bit Error Rate
BI	Beacon Interval
BO	Beacon Order
bps	bits per second
BPSK	Binary Phase Shift Keying
CAP	Contention Access Period
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CFP	Contention-Free Period
CFR	Code of Federal Regulations
CW	Continuous Wave
DCC	Dynamic Channel Coding
DoD	Department of Defense
DS	Direct Sequence
DS-UWB	Direct Sequence Ultra Wideband
ECMA	European Computer Manufacturers Association
FCC	Federal Communications Commission
FCS	Frame Check Sequence
FEC	Forward Error Correction
FEL	Future Event List
FES	Future Event Set
FFD	Full Function Device
FLL	Frequency Locked Loop
Gbps	Giga bits per second
GHz	Giga Hertz
GPL	General Public License
GUI	Graphic User Interface
Hz	Hertz
IC	Integrated Circuit

IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronic Engineers
IR	Impulse Radio
IRA	Impulse Radiating Antenna
ISI	Information Sciences Institute
Kbps	Kilo bits per second
kHz	Kilo Hertz
LAN	Local Area Network
LLC	Logical Link Control
LOS	Line-Of-Sight
LR-WPAN	Low-rate Wireless Personal Area Network
MAC	Medium Access Control
MB-OFDM	Multi-Band Orthogonal Frequency Division Multiplexing
Mbps	Mega bits per second
MCPS	Medium Access Control Common Part Sublayer
MCPS-SAP	Medium Access Control Common Part Sublayer (MCPS) Data Service
MFR	Medium Access Control Footer
MHR	Medium Access Control Header
MHz	Mega Hertz
MLME	Medium Access Control Sublayer Management Entity
MLME-SAP	Medium Access Control Sublayer Management Entity Service Access
MPDU	Medium Access Control Protocol Data Unit
MSDU	Medium Access Control Service Data Unit
NED	Network Description
NLOS	Non-Line-Of-Sight
OFDMA	Orthogonal Frequency Division Multiple Access
OSI	Open Systems Interconnection
PAN	Personal Area Network
PAR	Project Authorization Request
PD	Physical Layer Data
PD-SAP	Physical Layer Data Service Access Point
PDU	Protocol Data Unit
PHR	Physical Layer Header
PHY	Physical Layer
PLL	Phase Locked Loop

PLME	Physical Layer Management Entity
PLME-SAP	Physical Layer Management Entity Service Access Point
PPDU	Physical Protocol Data Unit
PR	Pseudo Random
PSD	Power Spectral Density
PSDU	Physical Service Data Unit
R&D	Research and Development
RAM	Random Access Memory
RFD	Reduced Function Device
RFID	Radio Frequency Identifier
RSSI	Received Signal Strength Indicator
SAP	Service Access Point
SDU	Service Data Unit
SFD	Start-of-Frame Delimiter
SHR	Synchronization Header
SNR	Signal-to-Noise Ratio
SO	Superframe Order
SSCS	Service-Specific Convergence Sublayer
TOA	Time Of Arrival
TDOA	Time Difference Of Arrival
TG	Task Group
TH	Time Hopping
U.S.	United States
USB	Universal Serial Bus
USB-IF	Universal Serial Bus – Implementers Forum
USC	University of Southern California
UWB	Ultra Wideband
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

TABLE OF CONTENTS

Chapter 1

Introduction

Ultra wideband (UWB) refers to a wireless technology that employs very narrow pulses to transmit energy across a wide spectrum of frequencies, in comparison with existing narrowband technologies that makes use of a frequency carrier to transmit data. This wideband nature of ultra wideband systems incorporates a lot of promising concepts for wireless communications.

Wireless sensor networks (WSNs) consist of a network of small autonomous wireless devices with sensors to cooperatively monitor the environment, capture data for processing, perform home or industrial automation tasks, or to be used for surveillance.

Wireless sensor networks have certain limitations which the features of ultra wideband technology seem to potentially address.

Tools to evaluate and understand the performance of new wireless communication technologies are necessary and invaluable. The OMNET++ network simulation environment provides the necessary framework and tools to enable the simulation of a low-rate ultra wideband implementation in wireless sensor networks. This low-rate ultra wideband implementation is defined by the IEEE 802.15.4a standard.

1.1 SCOPE

The scope of this research pertains to the analysis of the IEEE 802.15.4a standard for low-rate wireless personal area networks (WPANs) through simulations in software. The IEEE 802.15.4a standard defines two new alternate physical layers for the IEEE 802.15.4 standard better known as Zigbee. This research focuses on the ultra wideband physical layer introduced by the IEEE 4a task group. Furthermore, this dissertation also explores the fundamentals of ultra wideband technology and its applicability to wireless sensor networks by evaluating the performance of a software implemented IEEE 802.15.4a low-rate ultra wideband model in a simulated wireless sensor network.

1.2 PROBLEM STATEMENT AND MOTIVATION

Due to advances in high-speed switching technology ultra wideband gained attraction in low-cost consumer electronics and computer equipment. While these kinds of applications do not have a big problem with power, the same can not be said for wireless sensor networks. Sensor nodes are usually deployed once-off, sometimes without dedicated power supplies, and even for those that do employ batteries the recharging and replacing of those batteries is not an option. Size and cost constraints on sensor nodes result in further constraints on node resources such as memory, speed and bandwidth. Ultra wideband technology has proven to provide very robust communications with high data rates over short distances while being very conservative with energy consumption. The carrierless property of ultra wideband also implies that such radios can be manufactured inexpensively.

The choice of frequency band is an important factor for a device wanting to communicate wirelessly because it determines the capacity and possible interference from other systems that might be communicating in the same frequency band. The public ISM bands for example have no usage restrictions and systems operating in these bands need to coexist with each other. Specific ultra wideband impulse radio spreading techniques can be utilized to ensure coexistence with the numerous other radio systems. In addition, due to the large bandwidth available, a multiple access system may accommodate many users.

As of this writing, ultra wideband hardware is difficult to get hold of and any hardware that can be acquired carries an expensive price tag. In such a case, network simulator tools are of great value in demonstrating the capabilities of new network technologies if the required models are available. The two most popular network simulators used by academia are NS-2 [1] and OMNET++ [2]. For each of these network simulators an ultra wideband model contributed by former research exists [3], [4]. Both of these models provide excellent frameworks but do not claim to be complete and will therefore greatly benefit from any additional academic contributions to address shortcomings.

Much research has been done in recent years on the IEEE 802.15.4 standard and Zigbee but this is not the case for the low-rate ultra wideband physical layer amendment to the standard defined in IEEE 802.15.4a. This dissertation contributes to the knowledge area of low-rate ultra wideband and wireless sensor networks by providing an investigation into the viability of using low-rate ultra wideband as the communication medium for wireless sensor networks.

1.3 OBJECTIVES

The objective of this research is to do an in depth study into the intricacies of the IEEE 802.15.4a standard as it is defined for the new ultra wideband physical layer in order to clarify its complex inner workings and examine its viability for wireless sensor network applications.

To achieve this goal the most suitable simulation environment and an existing IEEE 802.15.4a ultra wideband physical layer simulation model will be chosen, investigated and adapted to ensure it accommodates all required features. Moreover, this model will be employed in a wireless sensor network for various performance analyses.

1.4 RESEARCH METHODOLOGY

Ultra wideband is an exciting new technology with lots of potential and this excitement sparked an extensive literature study to better understand the technology with all of its principles and characteristics that distinguishes it from other wireless technologies.

Thereafter a thorough study was made into the IEEE 802.15.4 and IEEE 802.15.4a standards to acquire the necessary know-how to be able to investigate and complement existing ultra wideband simulation models following this standard.

The study continued into the area of wireless sensor networks with emphasis on the limitations such networks currently experience.

When implementing a network simulation model, a key component is the network simulation environment and corresponding framework. A comprehensive study was made into different wireless network simulators after which the most suitable simulator and corresponding framework was chosen and closely examined to make proper use of its features.

The practical simulation work was mainly supported by and built upon the Mixim-UWB framework [4] implemented by Jérôme Rousselot for the OMNET++ network simulator.

1.5 DOCUMENT OUTLINE

This dissertation consists of several chapters in which the various aspects of WPAN standards, ultra wideband technology, Zigbee technology, wireless network simulators, amendments to the IEEE 802.15.4 standard and simulation thereof are explored and discussed.

Chapter 2 will present a history and background into wireless communications. The organization of the different IEEE 802.15 Working Group activities is outlined to give a bird's eye view of where the theme of this paper fits into the big picture of communication standards. The chapter finishes with an overview of wireless sensor networks.

Chapter 3 will engage into a thorough study and discussion of ultra wideband technology, looking at its advantages, disadvantages, unique characteristics, signal generation, modulation schemes, antenna considerations, and transmitter and receiver design.

Chapter 4 will engage a thorough investigation of the IEEE 802.15.4 and IEEE 802.15.4a standards to introduce all the detailed features, methods and techniques specified and applicable to a low rate ultra wideband PHY and corresponding MAC implementation.

Chapter 5 will regard the OMNET++ discrete event simulation environment in comparison to other network simulators. A brief outline of the OMNET++ features is also provided.

Chapter 6 discusses the chosen IEEE 802.15.4a UWB model and the utilization of the model in a simulated wireless sensor network.

Chapter 7 provides the results of the simulation sets and discusses the conclusions reached from these results.

Chapter 8 finishes with a summary of the research performed and what has been achieved in this dissertation. It also includes suggestions for further work and future research based on this topic.

Chapter 2

Background

2.1 WIRELESS DATA COMMUNICATION

During the past few years the wireless communication world saw the rise of an old wireless radio technology with booming potential. This section takes an in depth look into the heart of Ultra Wideband.

2.1.1 History of wireless

The date is December 12, 1901. Using a crude spark-gap transmitter and a balloon-supported antenna for reception, Guglielmo Marconi repeatedly transmitted the Morse code letter *S* from Poldhu, Cornwell, in England, across the Atlantic Ocean to St John's, Newfoundland, in Canada, totaling a distance of about 3500 kilometers. With this technological breakthrough Marconi sparked a wireless revolution [5], [6].

Despite the fact that Marconi proved the concept of wireless telegraphy with the spark-gap transmitter, it had some severe shortcomings. One of which is that each spark caused a burst of electromagnetic radiation which created pulses of energy with extremely short duration inducing excessive interference with sidebands. After different frequency and power limitations over the following years, spark-gap systems were totally banned in the United States by 1927 [7].

2.1.2 Narrowband transmission

To demonstrate the effect of such a short energy pulse, we have to look at the correlation between the time domain and frequency domain of the pulse as expressed by the Fourier transform.

Modeling the short energy pulse with a unit impulse function defined as

$$\delta(t) = \begin{cases} 1, & if\ t = 0 \\ 0, & if\ t \neq 0 \end{cases} \tag{2-1}$$

we can see from Figure 2-1 that the Fourier transform $X(f)$ of $\delta(t)$ demonstrates how such a pulse signal radiates as every frequency in the spectrum due to the infinitesimal time duration of the impulse signal.

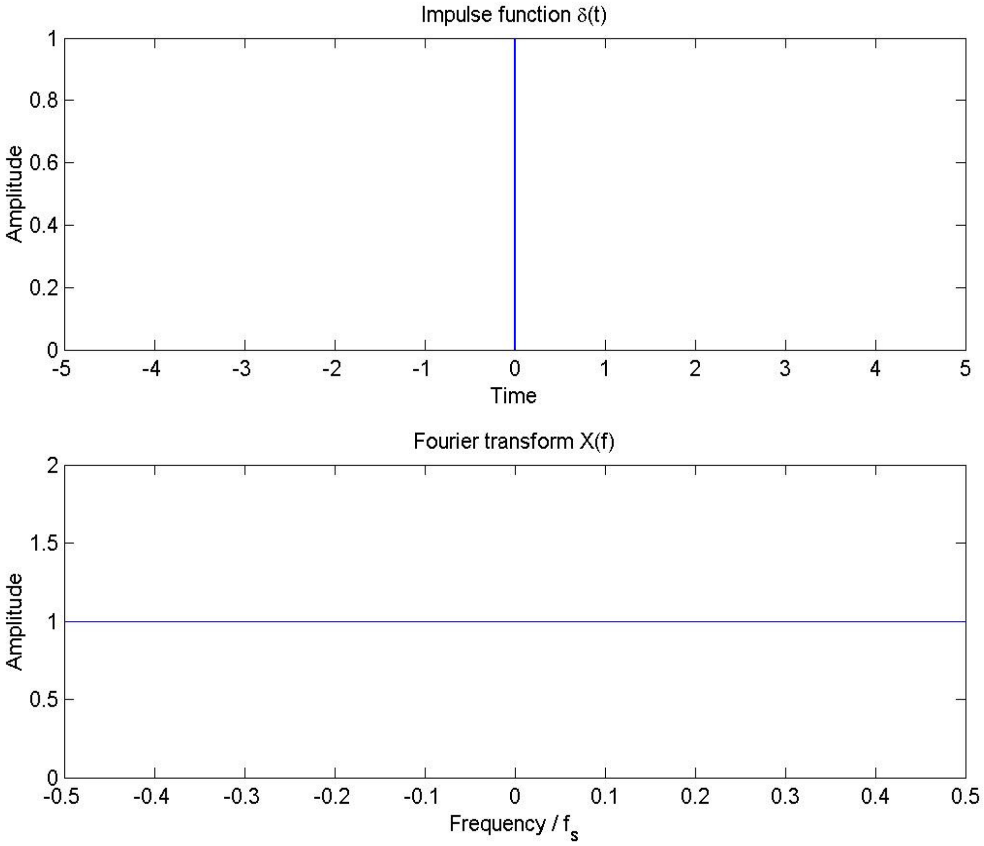


Figure 2-1. Fourier transform of an impulse function

The unit impulse signal illustrates the ideal case. In real life the time duration of such a signal will be finite and somewhere in the nanosecond time range. Still, it is clear that the transmit power of such signal should be severely limited in order to prevent interference with other transmitters and receivers wanting to share the same airwaves. Furthermore, since transmission occurs in many frequency bands, it becomes difficult for the receiver to distinguish between information and noise.

Marconi’s spark-gap transmitters (largely based on the ingenious research done by Nikola Tesla [8]) generated fairly broad signals having wavelengths between 250 meters (1.2 MHz) and 550 meters (545 kHz). Frequency band crowding and interference worsened and as a result legislation were put in place to prohibit the use of spark-gap transmitters.

It is mostly because of these reasons that scientists came up with the concept of separation by frequency where a specific frequency, which can easily be isolated from other frequencies via filtering, is used to transmit information.

Shorter wavelengths in the form of continuous wave (CW), which modulate a carrier signal in some way to convey information, are characterized by these frequency separation properties. Figure 2-2 shows the Fourier transforms of two sinusoidal waves. If a sinusoidal wave is employed as a carrier signal it appears as a very narrow pulse in the frequency domain, making it easy to filter out other transmissions. This method is best known as *narrowband* transmission.

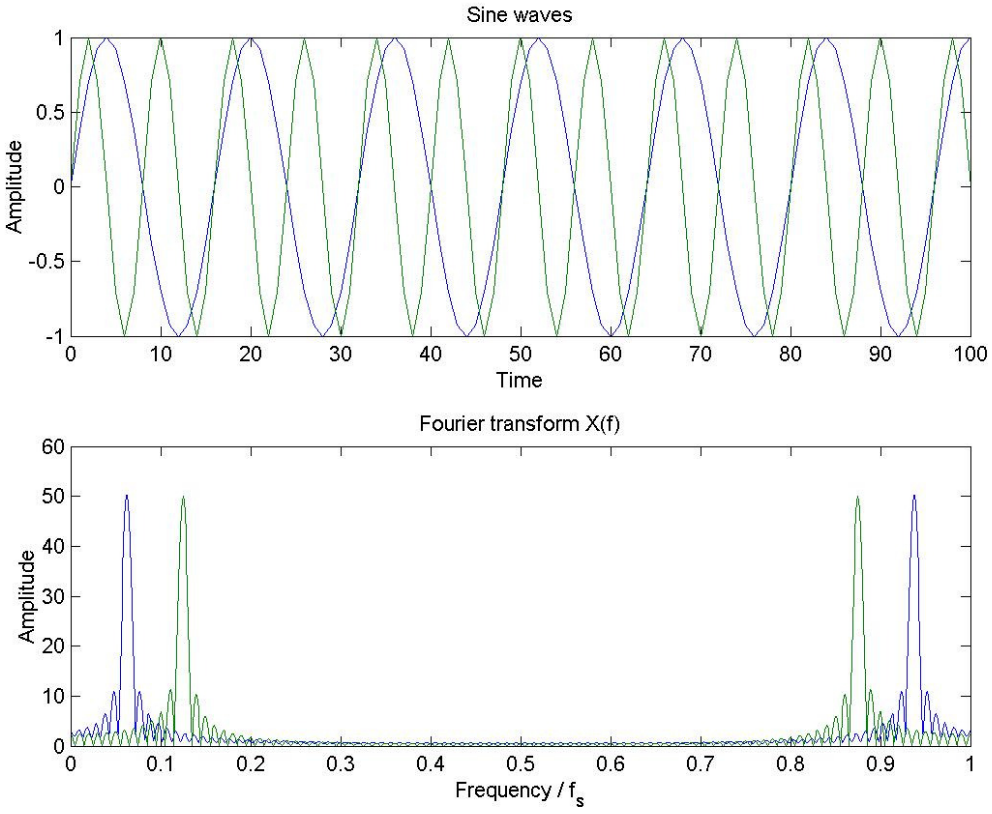


Figure 2-2: Fourier transforms of sine waves

To keep people from interfering with valuable wireless services such as emergency and military broadcasts, the frequency spectrum is divided up and regulated by various governments of which the Federal Communications Commission (FCC) in the United States is the largest.

The limited bandwidth nature of a narrowband signal places limitations on the amount of information the signal can transmit.

2.1.3 Shannon's information capacity theorem

Shannon's information capacity theorem is expressed as

$$C = B\ \log_2\left(1 + \frac{\mathrm{S}}{\mathrm{N}}\right)\ \text{bps} \tag{2-2}$$

where

C is the information capacity of the channel (bps),

B is the channel bandwidth (Hz),

S is the total signal power over the bandwidth (Watt),

N is the total noise power over the bandwidth (Watt),

$\frac{\mathrm{S}}{\mathrm{N}}$ is the received signal-to-noise (SNR) ratio.

[9] defines information capacity as the maximum rate at which information can be transmitted across the channel without error.

Equation (2-2) distinguishes three things we can do to improve the capacity of the channel.

- Increase the bandwidth B
- Increase the signal power S
- Decrease the noise N

Furthermore we also see that the capacity of a channel grows linearly with increasing bandwidth B, but only logarithmically with signal power S.

With the frequency spectrum becoming more and more crowded and the ever increasing demand for higher data rates, the bandwidth limitation of narrowband systems coerced scientists to develop new transmission techniques without using up and interfering with other reserved parts of the spectrum. One of these new techniques goes back and mimics the first technology used to transmit wireless signals.

2.1.4 History of Ultra Wideband

The origin of ultra wideband (UWB) technology stems from work in time-domain electromagnetics begun in 1962. However, it was not until the advent of the sampling oscilloscope in the early 1906s and the development of techniques for generating sub-nanosecond baseband pulses, that proper observation and measurements could be done. Impulse measurement techniques were used to characterize the transient behavior of certain microwave networks.

From measurement techniques the main focus moved to radar and communication devices of which radar was given a lot of attention because the low-frequency components were useful in penetrating objects.

The first UWB communications patent was awarded in 1973.

In 1989 the U.S. Department of Defense (DoD) started to use the term “ultra wideband” for this baseband, carrier-free, impulse technology which by then had experienced nearly 30 years of extensive development. Most applications and development occurred in classified military programs whose main driving force was accurate radar and communications technology that cannot be easily intercepted. Other UWB applications included automobile collision avoidance, positioning systems, liquid level sensing and altimetry [10].

In recent years UWB for consumer communication got a lot of attention and companies such as Alereon [11], Time Domain [12], Wisair [13] and XtremeSpectrum (acquired by Motorola [14]) were started to investigate and provide solutions for the use of UWB in personal computing, consumer electronics, and mobile devices.

2.2 STANDARDS ACTIVITY OF WPANs

The standards activity of Wireless Personal Area Networks (WPANs) is taken care of by the IEEE 802.15 standards working group [15]. IEEE 802.15 is responsible for creating and maintaining WPAN standards and is divided into six major task groups as shown in Figure 2-3.

.

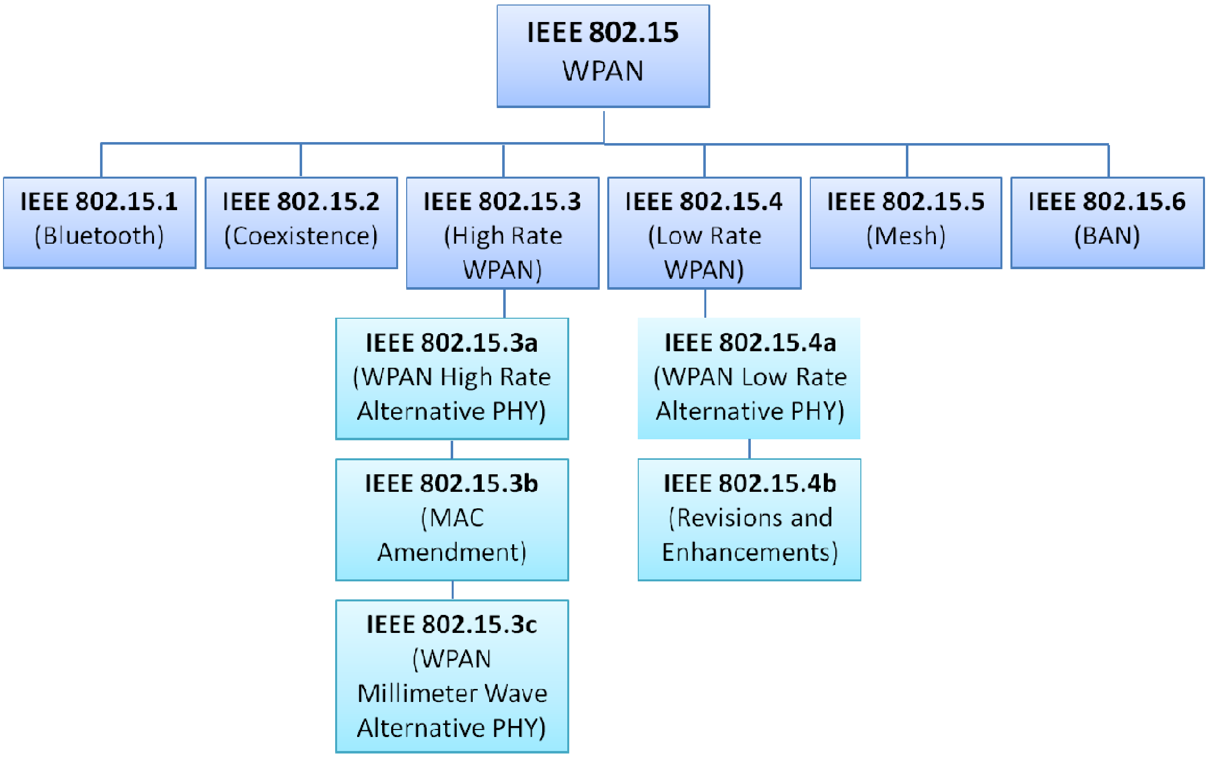


Figure 2-3: Organization of IEEE 802.15 Working Group activities

2.2.1 Task group 1 (Bluetooth)

The IEEE Project 802.15.1-2002 used the Bluetooth v1.1 Foundation Specifications to derive a WPAN standard. It incorporates a medium access control (MAC) and physical layer specification. The standard was updated to include the additions of Bluetooth v1.2 and published as IEEE 802.15.1-2005 [16].

2.2.2 Task group 2 (Coexistence)

The IEEE Project 802.15.2 developed a Recommended Practices to facilitate coexistence of WPANs (802.15) and WLANs (802.11). The Task Group also developed a set of Coexistence Mechanisms to facilitate coexistence of WLAN and WPAN devices [17].

2.2.3 Task group 3 (High Rate WPAN)

The IEEE Project 802.15.3 was chartered to define a new standard for high-rate WPANs. The new standard will also address the need for portable consumer digital imaging and multimedia applications by providing low power, low cost solutions [18].

IEEE 802.15.3a (WPAN High Rate Alternative PHY)

The IEEE 802.15 Task Group 3a (TG3a) was established with the purpose to draft and publish a higher speed alternative physical layer concept for the existing 802.15.3 standard. With a minimum data rate of 110 Mbps at 10m, the intend were to address the high data rate demand of applications incorporating video, imaging and multimedia links [19].

The IEEE 802.15 TG3a managed to consolidate a total of 23 UWB PHY specifications into 2 proposals backed by two different industry alliances:

- Multi-Band Orthogonal Frequency Division Multiplexing (MB-OFDM) UWB, supported by the WiMedia Alliance [20]
- Direct Sequence UWB (DS-UWB), supported by the UWB Forum [21]

On January 19, 2006 the TG3a members unanimously voted to withdraw the 802.15.3a project authorization request (PAR) because the process was in total standstill. Consensus could not be reached between the two industry alliances as to which technology would become the 802.15.3a standard for high-speed wireless [22].

IEEE 802.15.3b (MAC Amendment)

The IEEE 802.15 Task Group 3b (TG3b) is working on an amendment to the 802.15.3 standard in order to improve implementation and interoperability of the MAC [23].

IEEE 802.15.3c (WPAN Millimeter Wave Alternative PHY)

The IEEE 802.15 Task Group 3c (TG3c) is developing an alternative physical layer for the 802.15.3 standard based on millimeter-wave technology. The millimeter-wave WPAN will operate in the new and clear band defined by FCC 47 CFR 15.255 and will allow for very high data rates over 2Gbps [24].

2.2.4 Task group 4 (Low Rate WPAN)

IEEE 802.15.4 (Zigbee)

The IEEE 802.15 Task Group 4 (TG4) was tasked to investigate a low data rate solution with very low complexity and high energy efficiency that will allow for batteries to live from multi-months to multi-years. The potential applications are sensors, interactive toys, smart badges, remote controls, and home automation. The current version of the standard is the 2006 revision [25].

The Zigbee protocol stack employs the 802.15.4 standard as its base and provides a complete networking solution by developing the upper layers which are not covered by the standard. This paper takes an in depth look into the 802.15.4 standard together with its amendment, 802.15.4a, which is discussed next.

IEEE 802.15.4a (WPAN Low Rate Alternative PHY)

The IEEE 802.15 Task Group 4a (TG4a) provides alternative physical layers for low rate WPANs and is an amendment to the 802.15.4 standard.

It allows for additional capabilities over the existing 802.15.4 standard by

- providing communications and high precision ranging capability,
- high aggregate throughput,
- ultra-low power,
- as well as adding scalability to data rates,
- longer range, and
- lower cost.

The current status of 802.15.4a is complete and requires no further TG4a effort [26].

The main focus of this dissertation is the analysis of the 802.15.4a ultra wideband physical through a simulation model that can be used to simulate a WSN. An in-depth discussion of this amended standard is presented in Chapter 4.

IEEE 802.15.4b (Revisions and Enhancements)

The IEEE 802.15 Task Group 4b (TG4b) was chartered to provide for specific enhancements and clarifications to the 802.15.4-2003 standard by resolving ambiguities, reducing unnecessary complexity and allocating frequency that recently became available. IEEE 802.15.4b was approved in June 2006 and was published in September 2006 as IEEE 802.15.4-2006 [27].

2.2.5 Task group 5 (Mesh networking)

The IEEE Project 802.15.5 is responsible for defining a PHY and MAC layers standard to enable mesh networking in WPANs [28].

Mesh networking is a way to route data between nodes. Packets are “hopping” from node to node until the destination is reached. Connections between nodes can be reconfigured to get around broken or blocked paths.

Two connection arrangements are defined:

1. *Full mesh topology*: All nodes are connected to each other.
2. *Partial mesh topology*: Some nodes are connected to all the others while some only connect to those they communicate with most of the time. All nodes can still communicate with each other through multiple hops.

Mesh networks extend network coverage without increasing the transmit power or receive sensitivity. Furthermore, they also enhance reliability through route redundancy, drain less battery power due to fewer transmissions and make it easier to configure the network.

2.2.6 Task group 6 (BANs)

The IEEE Project 802.15.6 is focusing on Body Area Network (BAN) technologies. The goal is to develop a communication standard optimized for low power devices operating on, in or around the human body. Typical applications areas include medical, consumer electronics, personal entertainment, and other [29].

2.3 WIRELESS SENSOR NETWORKS

Wireless Sensor Networks (WSNs) are the main representative of pervasive computing in large-scale physical environments. These networks consist of a large number of small, wireless devices embedded in the physical world to interact with their environment by sensing or controlling physical parameters.

2.3.1 Applications

WSNs show a clear difference between sources of data and sinks of data. Sources are the actual sensor nodes that sense the data. Sinks are nodes where the data should be delivered to and can be part of the sensor network or an outside system.

A sensor node reports to the sink(s) once it has detected the occurrence of some event it was tasked to monitor. For more complicated tasks a sensor node can collaborate with other sensor nodes to decide whether an event has occurred, as would be required when the source of the event is mobile. Sensor nodes can also be tasked with periodic reporting of measured values.

Sensor nodes can be deployed in a number of ways ranging from fixed to random deployment depending on the application and environment. This causes concerns for the lifetime of the network because of the limited options available for maintenance and power source replacement.

The following extensive list of real-life application examples for WSNs is provided by Karl *et al* [30]:

- Disaster relief
- Environment control and biodiversity mapping
- Intelligent buildings
- Facility management
- Machine surveillance and preventative maintenance
- Precision agriculture for irrigation and fertilizing
- Medicine and health care
- Logistics and passive RFID tags
- Telematics for traffic information

2.3.2 Sensor nodes

Usually a single sensor node is incapable of fulfilling the tasks of the WSN on its own and has to collaborate with other sensor nodes using the wireless radio that forms part of the sensor node. The components a basic sensor node comprises of are shown in Figure 2-4.

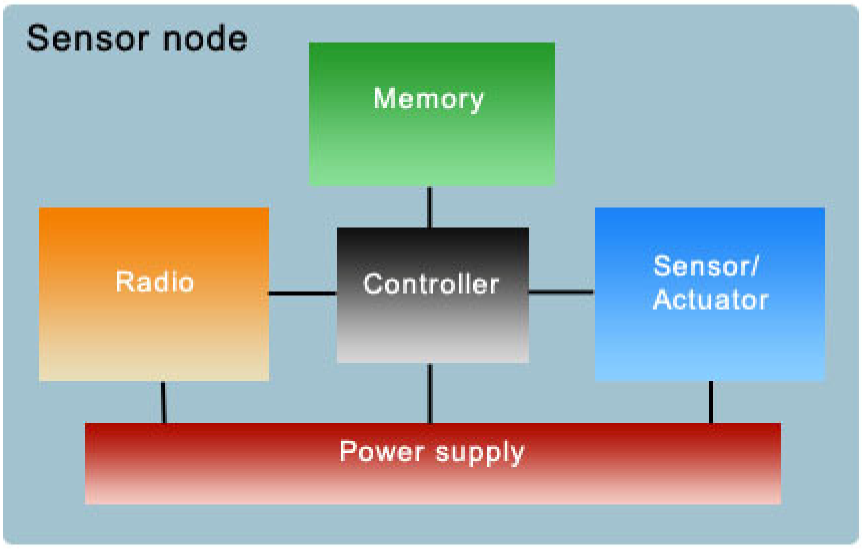


Figure 2-4: Hardware components of basic sensor node

Each of these components have to be operated in such a way as to consume as little power as possible while still fulfilling their required tasks. This is because energy is a very scarce resource in wireless sensor networks. Mains power is not an option as a truly wireless system is one where all nodes run without any attached cables, therefore a local power supply is required. Unfortunately hundreds of wireless sensors can be deployed in an environment where it is near impossible to reach the nodes and replacement of batteries is not an option.

2.3.3 Power sources

Mahlknecht [31] lists the following main energy sources available for wireless sensor networks which is categorized as either stored energy, or salvaged energy.

Stored energy

- Batteries: Stored energy is conventionally provided through batteries. Batteries have a finite lifetime and need to be replaced or recharged which can be very difficult and expensive. The following list indicate the variety of different batteries available:

- Alkaline Manganese
- Lithium Cells
- Zink Air
- Silver Oxide

And for rechargeable batteries:

- Nickel Metal Hydride
- Lithium Ion
- Lithium Manganese
- Lithium Vanadium Penthoxide

- Capacitors: Apart from traditional batteries, there are also other forms of energy reservoirs. High energy capacitors represent an interesting alternative. “Gold caps” or “Ultra capacitors” are the names given to these high capacity energy stores which can easily and quickly be recharged and do not wear out over time.

Salvaged energy

The concept is also known as “scavenged” energy and it aims to provide methods that will allow a node to tap into energy from the environment the node operates in. The drained energy can then be saved in an energy supply on the node itself. The following approaches exist:

- Photovoltaic cells: These are the well-known solar cells. If sufficient light is available on a regular basis this method can ensure continuous operation of the sensor node.
- Mechanical vibrations: With this method mechanical vibrations from sources such as walls or windows that vibrate, due to the operation of heavy machinery or the passing of vehicles nearby, are used to generate electrical energy. Several means

exist to convert vibrations into electrical energy based on electromagnetic, electrostatic and piezoelectric principles [32].

- Thermo elements: In this method temperature differences are directly converted to electrical energy. The Peltier effect and Seebeck effect are both thermoelectric phenomenon that can be used to generate electric energy between metals.

2.3.4 Challenges

The typical application areas of WSNs demand for certain required mechanisms to be in place before such networks can be properly realized. Therefore, WSNs pose certain challenges [33] that are discussed next.

Scalability

Since WSNs can have a large number of nodes, the employed architectures and protocols need to be able to handle these large numbers and any additional node increases. To transmit information to a destination node not in the source node's vicinity, might require higher transmission power which can cause inter-node interference and drain the power of the transmit node.

Energy-efficient operation

To extend the lifetime of the entire network, power conservation at individual node level is of utmost importance. The major culprit at exhausting the power supply of a sensor node is the wireless radio. Hence, avoiding unnecessary communications can save energy considerably. For long distance communications a multi-hop wireless communication scheme where intermediate nodes are used as relays can also reduce the energy consumption of a single node.

Self-organization

A WSN needs to autonomously configure most of its operational parameters so that the network is able to tolerate the failure of nodes and the integration of new nodes. Self-organization should be performed such that the overall WSN performance is improved while reducing power consumption.

Synchronization

Due to the large sensor population, global synchronization of the whole network is not viable and therefore focus has to be given to node-by-node synchronization. If nodes are not synchronized in time with each other they have to continuously poll the network, listening for transmissions, and needlessly drain power.

Channel estimation

Many methods exist to estimate the quality of the wireless link, but basically it involves receiving packets and gauging their quality or calculating packet loss ratios. There are two types of estimators:

- Active estimator: Special packets are sent out by the node which requests certain measurement responses from neighbor devices.
- Passive estimator: Nearby transmissions are observed by the node to estimate loss rates and quality.

The channel estimation knowledge is crucial since it can be used to overcome the detrimental effects of noise, multipath, intentional jamming and inter-node interference by ensuring reliable data transfer between the nodes.

Extensive research has taken place to determine the best possible hardware and software solutions to all of the above challenges, all based on conventional narrowband technology. Ultra wideband technology addresses most of these issues just with its core characteristics. Some of these WSN challenges are addressed by the IEEE 802.15.4 and Zigbee standards, hence this study also aim to look at the role the IEEE 802.15.4a UWB physical layer amendment can play in complementing these solutions and dealing with the remaining challenges.

Chapter 3

Overview of Ultra Wideband

In this chapter we take an in-depth look at the exciting wireless technology known as ultra wideband (UWB). We start with the definition of ultra wideband and list the properties distinguishing it from other wireless technologies. UWB waveforms generation, modulation schemes and multiple access strategies are investigated. Channel models for UWB are presented with a brief discussion on UWB transceiver and antenna design.

3.1 DEFINING ULTRA WIDEBAND

UWB is a spread spectrum technology. Spread spectrum is an RF communications technology in which the bandwidth of the baseband signal is intentionally spread over a larger bandwidth by injecting a higher-frequency signal. Consequently, the transmit energy is spread over a wider bandwidth and the signal appears as noise. Refer to [34], [35] and [36] for a broad overview of spread spectrum principles.

Even so, UWB differs from conventional spread spectrum technologies because in a UWB system information is transmitted through a series of short pulses or a “chirped” signal while with traditional spread spectrum systems information is transmitted by modulating a continuous carrier signal. Also, a UWB system occupies a lot more bandwidth than a spread spectrum system.

The Federal Communications Commission (FCC) formally defines a UWB device as any intentional radiator with a fractional bandwidth greater than 0.20, or with a UWB bandwidth equal to or greater than 500 MHz [37].

UWB bandwidth is defined as the frequency band bounded by the points that are 10 dB below the highest radiated emission.

Fractional bandwidth (FB) is defined as

$$FB = 2\frac{(f_H - f_L)}{(f_H + f_L)} \tag{3-1}$$

where f_H is the upper frequency of the -10 dB emission point and f_L is the lower frequency of the -10 dB emission point.

Furthermore, the transmission center frequency f_c is defined as the average of the upper and lower -10 dB points, i.e.

$$f_c = \frac{(f_L + f_H)}{2} \tag{3-2}$$

3.1.1 UWB Power Spectral Density

Power spectral density (PSD) describes the distribution of signal power over frequency. It can be calculated as:

$$PSD = \frac{P_t}{BW} \tag{3-3}$$

where P_t is the transmit power in Watts and BW is the bandwidth of the signal in Hz. Therefore PSD is measured in Watts/Hz.

Narrowband technologies have a high power spectral density compared to UWB because in UWB the transmit energy is spread over a very large bandwidth (see Figure 3-1).

It is noteworthy to mention that a very low PSD allows for covert communication.

3.1.2 UWB Regulations

Due to the wide bandwidth occupied by UWB emissions, it could potentially interfere with other licensed bands in the frequency spectrum if left unregulated.

Many organizations and government entities around the world set rules and recommendations for UWB usage. On the international level there is the International Telecommunication Union (ITU). In the Asia-Pacific region the Asia-Pacific Telecommunity (APT) is responsible for telecommunication recommendations and guidelines. Japan has the Ministry of Internal Affairs and Communication (MIC) as regulatory body. The European Conference of Postal & Telecommunications Administrations (CEPT) created a task group under the Electronic Communications Committee (ECC) to draft a proposal regarding the use of UWB for Europe with Ofcom being the independent regulator and competition authority for the communication industries in the United Kingdom. In the USA, the Federal Communications Commission (FCC) is charged with regulating interstate and international communications. The USA was the first country to legalize UWB for commercial use.

The FCC first set in motion a Notice of Inquiry (NOI) in September of 1998 following the argument that low power wireless services could operate below authorized out-of-band emissions limits described in the FCC Part 15 rules for intentional and unintentional radiators in unlicensed bands.

The emission limits are defined in micro volts per meter (uV/m) and in order to express this in terms of radiated power, the following formula can be used:

$$P = \frac{E_0^2 4\pi R^2}{\eta} \tag{3-4}$$

where E, represents the electric field strength (V/m), R is the radius of the sphere at which the field strength is measured, and η = 377 ohms is the characteristic impedance of a vacuum [38].

In May of 2000, the FCC issued a Notice of Proposed Rule Making (NPRM) that could allow UWB emitters under the Part 15 rules.

On February 14 of 2002, the FCC issued a first Report and Order [37], revising the Part 15 rules regarding UWB transmission systems by permitting UWB intentional emissions subject to certain frequency and power limitations. A total of 7500 MHz of spectrum in the 3.1 GHz to 10.6 GHz frequency band was allocated for the unlicensed use of UWB devices.

Figure 3-1 shows the spectrum masks for indoor and outdoor operation permitted under Part 15 of the Commission's rules. UWB signals must be transmitted at low radiated power, with the rules specifying a mean EIRP of -41.3 dBm/MHz. Effective isotropic radiated power (EIRP) is the amount of power supplied to an isotropic antenna multiplied by the antenna gain in given direction. An isotropic radiator radiates power equally in all (theoretically) directions. The gain of an antenna represents how well it increases effective signal power in a particular direction. So, the EIRP refers to the highest signal strength measured in any direction at any frequency from the UWB device.

Table 3-1 and Figure 3-1 gives a comparison between the spectrum allocations for unlicensed bands in the USA.

Table 3-1: FCC spectrum allocation for unlicensed use in the USA

Unlicensed bands	Operating frequency (GHz)	Bandwidth (MHz)
ISM	2.4 to 2.4835	83.5
U-NII	5.15 to 5.35	300
UWB	3.1 to 10.6	7500

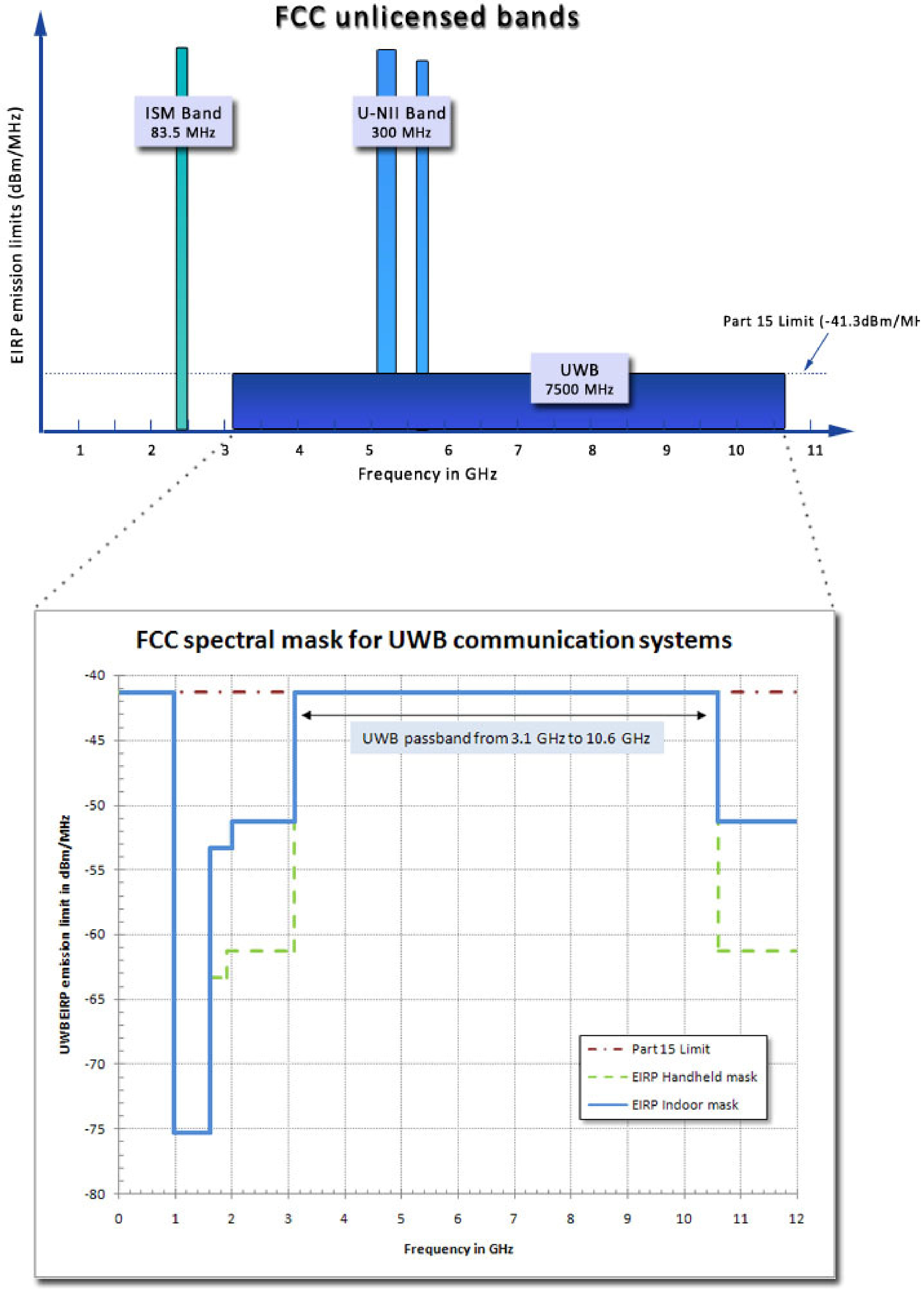


Figure 3-1: FCC unlisensed bands and UWB spectral mask

3.2 ADVANTAGES OF UWB

UWB offers many advantages over narrowband technology making it very attractive for consumer communications applications.

3.2.1 Improved channel capacity

In Section 2.1.3, the channel capacity equation of Shannon (Equation 2-2) showed that increasing the channel capacity requires a linear increase in bandwidth and exponential increase in power.

It is this property that makes UWB technology ideal for applications requiring very high data rates whilst using very lower power. However, useful range is limited to about 10 meters due to the low power levels mandated by the FCC. Data rates of over 100 Mbps have been demonstrated with the potential of even higher data rates over shorter distances.

3.2.2 Inherent robustness to multipath fading

The large bandwidth contributes to another advantage by making UWB very robust to multipath fading.

Multipath is a signal propagation phenomenon where transmitted signals reach the receiving antenna by multiple paths. The effects are constructive and destructive interference, and phase shifting of the signal. The sum of the out of phase signals causes destructive interference with multipath fading as a result.

Due to the wideband nature of UWB the transmitted signal is resistant to severe multipath propagation. The narrow pulses prevent multiple reflections from combining destructively at the receiver. Furthermore, the multipath components can be resolved and used to improve signal reception using a rake receiver or channel equalization techniques [39], [40].

3.2.3 Noise-like signal

As mentioned in Section 3.1.1, UWB pulses are transmitted at very low power levels and in addition with some pseudo-randomness the signal can be made to appear noise-like ensuring a very low probability of interception or detection.

3.2.4 Low complexity, low equipment cost and small form factor

UWB transmitters directly modulate a baseband signal onto an antenna. There is no need for a RF mixer to do any frequency up-conversion or inject a carrier frequency. UWB receivers may require more complex architectures but again, no frequency down-conversion or complex delay and phase lock loops circuits are required. Consequently, all tuned circuitry can be reduced and UWB radios can be implemented with low cost, low power CMOS processes.

The reduction in passive components allow for devices with small size to be manufactured, fulfilling a crucial requirement of certain electronic devices such as wireless sensors.

Designing for small but effective antennas still remains a challenge.

3.2.5 Low power consumption

The simple architecture also allows for the design of power economic circuits where information is transmitted with very short, low energy pulses. Conserving power, even on such small scale, holds huge benefits for wireless sensor networks.

3.2.6 Penetration ability

Certain applications benefit or rely on the ability of the communication system to penetrate through physical objects typically found in home and office environments. UWB systems operating on the lower center frequencies still provide this capability. The higher the center frequency the lower the ability of the UWB pulses to pass through objects due to the shorter wavelength. The relationship between frequency and wavelength is shown in equation 3-5.

$$\lambda = \frac{c}{f} \tag{3-5}$$

where λ is the wavelength in meters, f is the frequency in Hz and c is the speed of light defined as 299,792,458 m/s.

3.2.7 Accurate ranging and location detection

The very narrow transmit pulses give UWB radios a very good time domain resolution allowing for location and position determination. UWB can be used as a short range

RADAR (Radio Direction And Ranging). A single receiver can determine the range of its transmitter, and with the location information of three receivers triangulation can be used to determine position with much better accuracy than a GPS (Global Positioning System).

3.3 UWB WAVEFORM GENERATION

FCC regulation 47 CFR Section 5.5 (d) [41] state that intentional radiators are prohibited to produce class B emissions (damped waves). The damping oscillations of such waveforms cause sharp peaks in the spectrum with only small bandwidth and these peaks can cause serious interference with existing communication systems.

Desired UWB waveforms therefore, should provide a flat frequency domain spectrum in order to meet FCC regulations. Gaussian, Rayleigh, Laplacian, cubic waveforms and modified Hermitian monocycles are all examples of nondamped waveforms with nearly flat spectrum. In the following pages we will take a closer look at Gaussian waveforms.

3.3.1 Gaussian waveforms

Gaussian waveforms get their name from their mathematically resemblance to the Gauss function defined as

$$G(x) = \frac{1}{\sqrt{2\Pi\sigma^2}} e^{-x^2/2\sigma^2} \qquad (3\text{-}6)$$

where σ is the standard deviation.

A Gauss pulse can be represented by the following equation where τ is the time decay constant:

$$g(t) = e^{-\left(\frac{t}{\tau}\right)^2} \qquad (3\text{-}7)$$

To create a waveform suitable for UWB transmission, the Gaussian pulse is filtered which gives the same effect as taking the derivative of equation 3-7. The waveform produced by the first derivative of the Gaussian pulse is called a Gaussian monocycle (see Figure 3-2) and has a single zero crossing. A Gaussian monocycle is given by

$$g'(t) = \frac{d}{dt}\left[e^{-\left(\frac{t}{\tau}\right)^2}\right] \qquad (3\text{-}8)$$

$$= \frac{-2t}{\tau^2} e^{-\left(\frac{t}{\tau}\right)^2}$$

The waveform produced by the second derivative of the Gaussian pulse is called a Gaussian doublet (see Figure 3-2) and has two zero crossings. A Gaussian doublet is given by

$$\begin{aligned} g''(t) &= \frac{d}{dt}\left[\frac{-2t}{\tau^2} e^{-\left(\frac{t}{\tau}\right)^2}\right] \\ &= \frac{-2}{\tau^2} e^{-\left(\frac{t}{\tau}\right)^2} - \frac{2t}{\tau^2}\left(\frac{-2t}{\tau^2} e^{-\left(\frac{t}{\tau}\right)^2}\right) \\ &= \frac{-2}{\tau^2} e^{-\left(\frac{t}{\tau}\right)^2} + \frac{4t^2}{\tau^4} e^{-\left(\frac{t}{\tau}\right)^2} \\ &= \frac{-2}{\tau^2}\left(1 - \frac{-2t^2}{\tau^2}\right) e^{-\left(\frac{t}{\tau}\right)^2} \end{aligned} \tag{3-9}$$

Each additional derivative of the Gaussian pulse will result in an additional zero crossing decreasing the relative bandwidth and increasing the center frequency for a fixed time decay value τ [42]. Figure 3-2 and Figure 3-3 shows the waveforms given by the first four derivatives of the base Gauss pulse defined in Equation 3-7 with τ = 50ps.

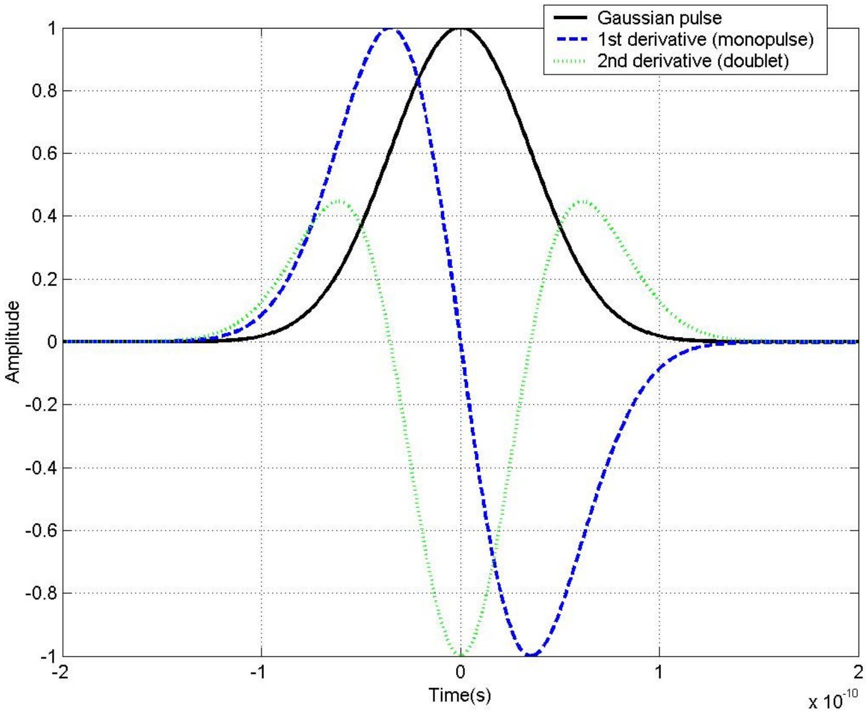


Figure 3-2: Time domain of Gaussian pulse with 1st and 2nd order derivatives

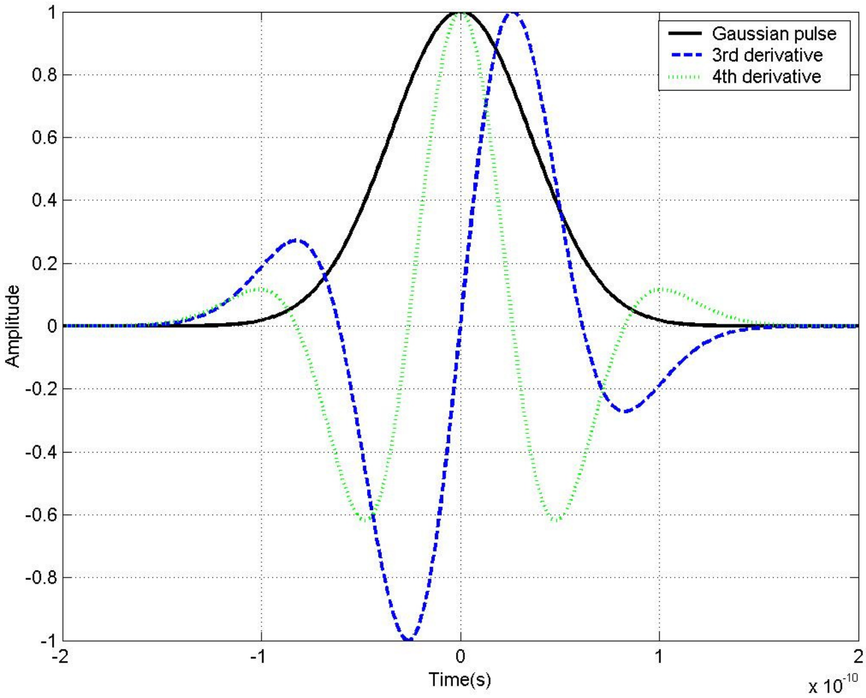


Figure 3-3: Time domain of Gaussian pulse with 3rd and 4th order derivatives

In order to get the frequency domain representation of the Gaussian waveforms discussed, we have to take the Fourier transform of these waveforms. The Fourier transform of the base Gauss pulse is given by

$$\begin{aligned} G(f) &= \mathcal{F}[g(t)] \\ &= \int_{-\infty}^{\infty} e^{-\left(\frac{t}{\tau}\right)^2} e^{-j2\pi f t} dt \\ &= \tau\sqrt{\pi} e^{-(\pi f t)^2} \end{aligned} \tag{3-10}$$

The same can now be done for the derivative functions of the Gauss pulse. To simplify the math we make use of the Fourier transform derivative theorem which states that the derivative of a function $f(x)$ is equal to the Fourier transform of the function multiplied by $(j2\pi f)^n$ where n equals the derivative order. Equations 3-11 to 3-14 gives the Fourier transforms of the first four derivative functions of the base Gauss pulse with Figure 3-4 and Figure 3-5 illustrating them graphically.

$$G(f) = \tau\sqrt{\pi}(j2\pi f) e^{-(\pi f t)^2} \tag{3-11}$$

$$G(f) = \tau\sqrt{\pi}(j2\pi f)^2 e^{-(\pi f t)^2} \tag{3-12}$$

$$G(f) = \tau\sqrt{\pi}(j2\pi f)^3 e^{-(\pi f t)^2} \tag{3-13}$$

$$G(f) = \tau\sqrt{\pi}(j2\pi f)^4 e^{-(\pi f t)^2} \tag{3-14}$$

From these figures we can clearly see the effect of each derivative order. As mentioned each order will cause a decrease in the relative bandwidth and an increase of the center frequency. It can also be seen that these waveforms are almost uniformly distributed across the frequency spectrum.

3.3.2 Choice of waveform

Pulse characteristics of the waveform can be varied to define the energy in the frequency spectrum to meet the design criteria and regulations. The design criteria for a UWB system typically define the operating bandwidth, the spectral mask and the center frequency within the spectrum of interest. The following characteristics of the generated waveform influence these factors:

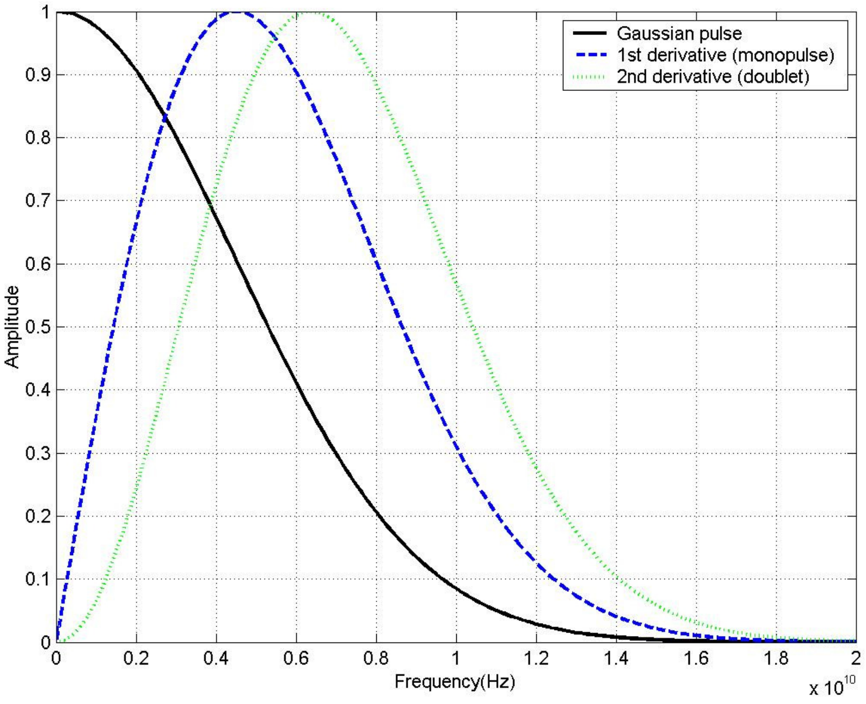


Figure 3-4: Frequency domain of Gaussian pulse with 1st and 2nd order derivatives

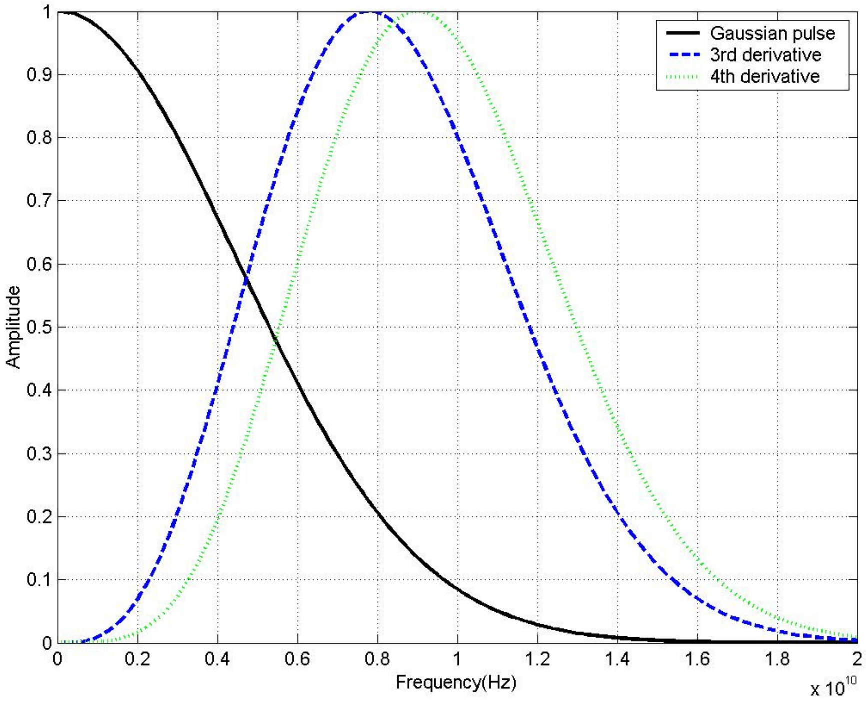


Figure 3-5: Frequency domain of Gaussian pulse with 3rd and 4th order derivatives

- Pulse duration: The time duration of the pulse and determines the bandwidth in the frequency domain. As a guideline we can state that the reciprocal of the pulse duration is approximately equal to bandwidth of the pulse.
- Pulse repetition frequency (PRF): The number of UWB pulses that is transmitted per second. If the pulse repetition is almost periodic, the PRF can be used to determine the center frequency of the spectrum.
- Pulse shape: The physical shape of the transmitted waveform. As was shown with the different derivative waveforms of the Gauss pulse (Figure 3-2 to Figure 3-5).

3.4 UWB IMPULSE RADIO

In an impulse radio single short baseband pulses are generated for communicating data. We now know that each of these pulses has a very wide spectrum which must adhere to certain spectral requirements. Each pulse will have very low energy to ensure that the low power levels permitted for UWB transmission is met which reduces the risk of interference with other narrowband communication technologies. Because of the very low energy for any given pulse, many pulses are combined to carry one bit of information.

3.4.1 Pulse trains

Such long sequences of very short duration pulses are known as pulse trains. If we take a Gaussian monocycle $g_m(t)$ defined in equation 3-8 we can write an unmodulated periodic monocycle pulse train $p_{train}(t)$ as

$$p_{train}(t) = \sum_{n=-\infty}^{n=\infty} g_m(t - nT) \tag{3-15}$$

where T is the period between pulses. The reciprocal of T gives us the pulse repetition frequency (PRF). Figure 3-6 shows a monopulse train.

Pulse trains with a constant pulse interval T, introduce energy spikes as strong spectral lines into the spectrum of the transmitted signal which might interfere with other RF communication systems at short range [43]. Figure 3-7 shows that the resulting spectrum for the pulse train of Figure 3-6 will have an overall envelope with the same shape as the spectrum of the single monocycle, but it will consist of harmonics of the PRF rather than a continuous spectrum.

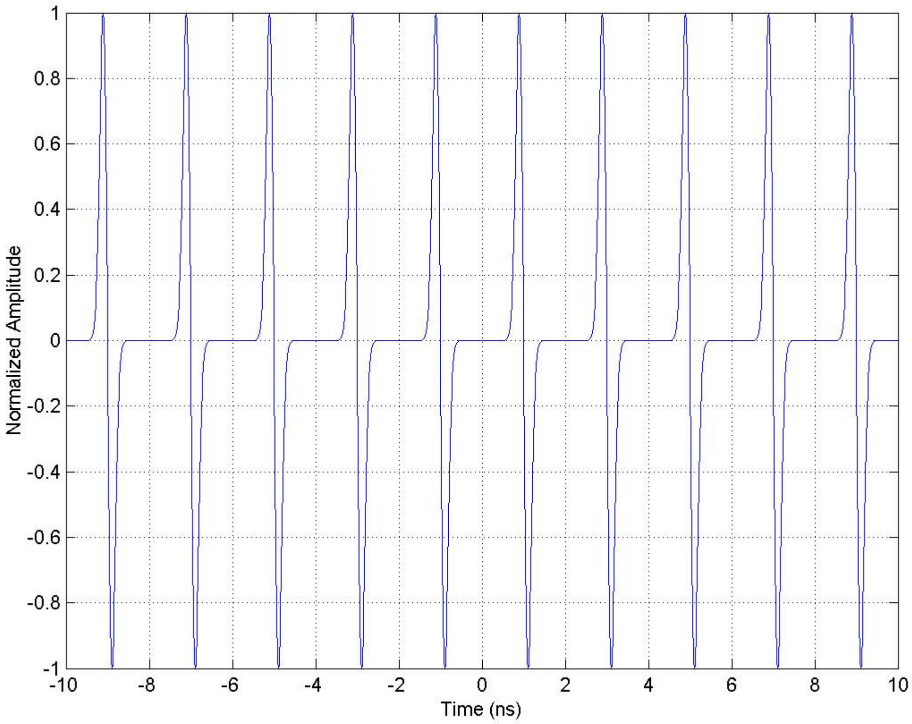


Figure 3-6: Monocycle pulse train

Examination of the period *T* and PRF by Ghavami *et al.* [44] have shown certain properties:

- Increasing the PRF in time domain also increases the magnitude in the frequency domain.
- Decreasing the duration of the pulse in the time domain increases the spectrum width in the frequency domain.
- If a random pulse interval is used, the frequency components are unevenly spread across the spectrum which produces a much lower peak magnitude spectrum.

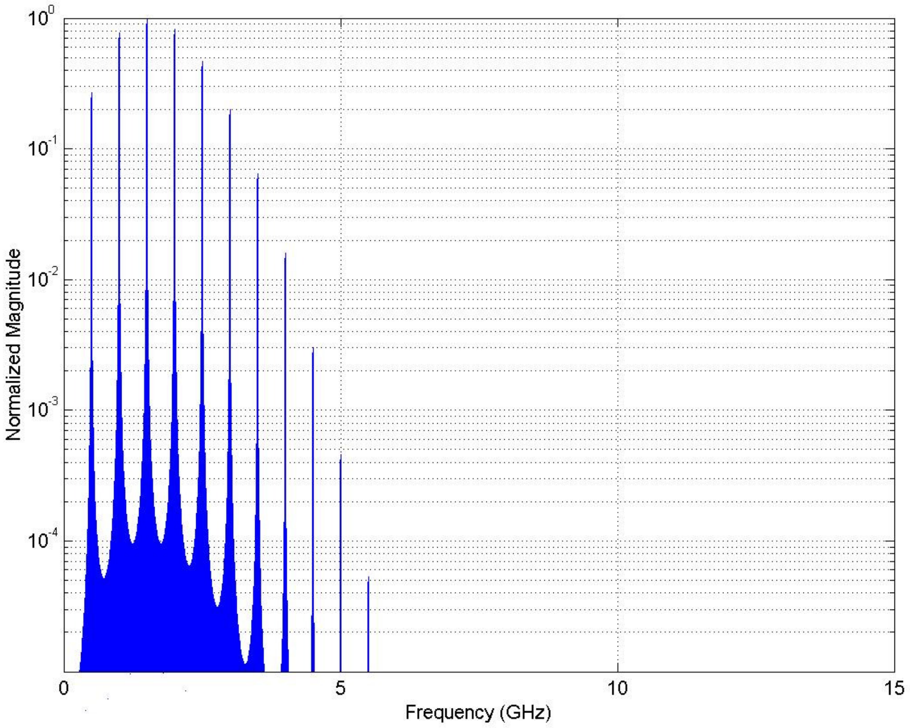


Figure 3-7: Spectrum of monocycle pulse train

Therefore, to overcome these spectral peaks, randomization techniques are used which adds a random offset to each pulse to remove common spectral components. Figure 3-8 shows a monocycle with two other exact monocycles offset in time.

By combining the offset pulses of Figure 3-8 randomly, the pulse train in Figure 3-9 is obtained (notice that the pulses are not evenly spaced). If we look at the resulting spectrum of this pulse train with the random offset monocycles as depicted in Figure 3-10, we can clearly see the smoothing effect these offsets have on the magnitude of the spectrum.

Because the random offset is not known by the receiver, a special cyclic sequence is used instead. This sequence is called a pseudo-random noise (PN) code. At the receiver end tracking is made easier since the PN codes are known and can easily be reproduced. Time hopping (TH) and direct sequence (DS) randomization techniques are examined in section 3.7 when we look at multiple access techniques as well.

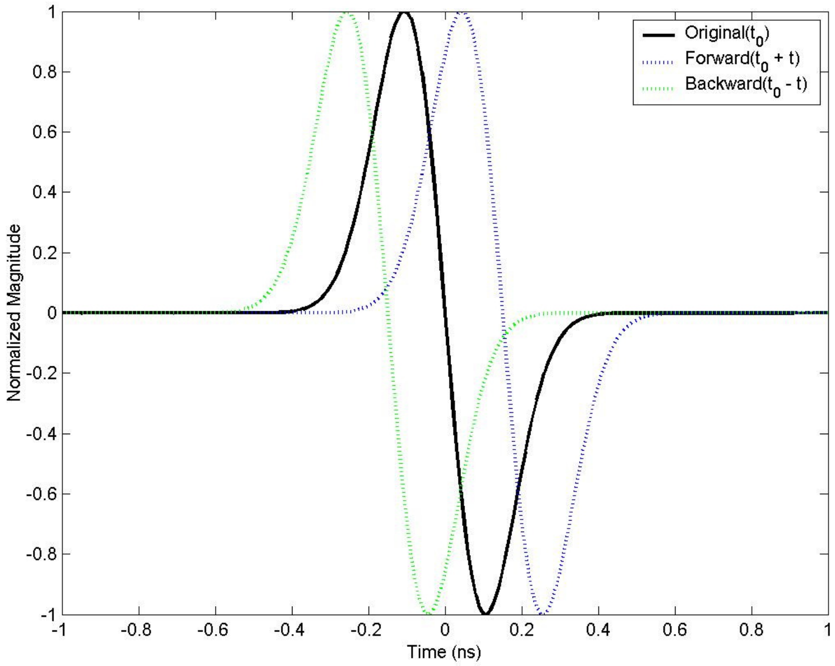


Figure 3-8: Monocycle pulse with offset pulses

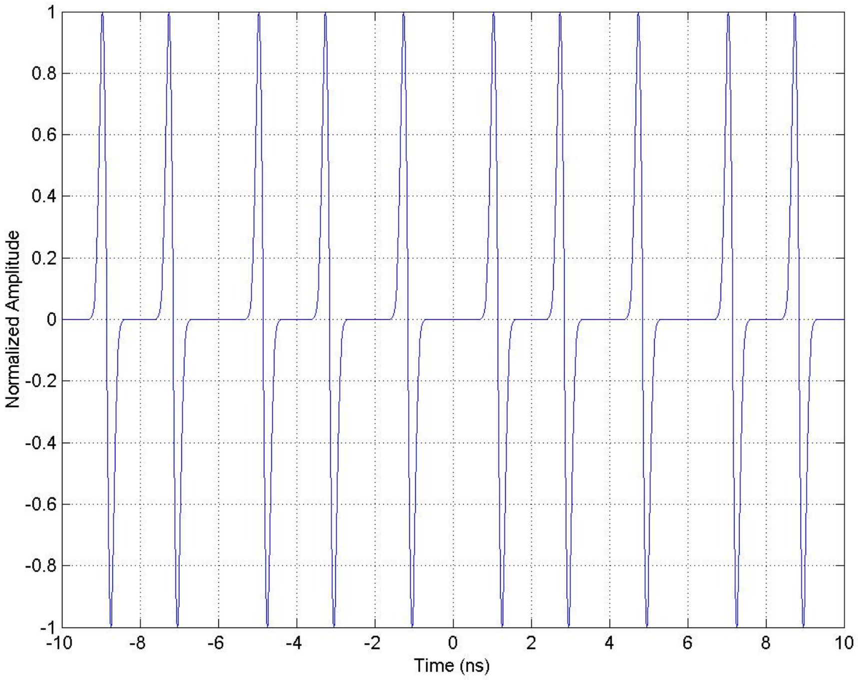


Figure 3-9: Monocycle pulse train with offset pulses

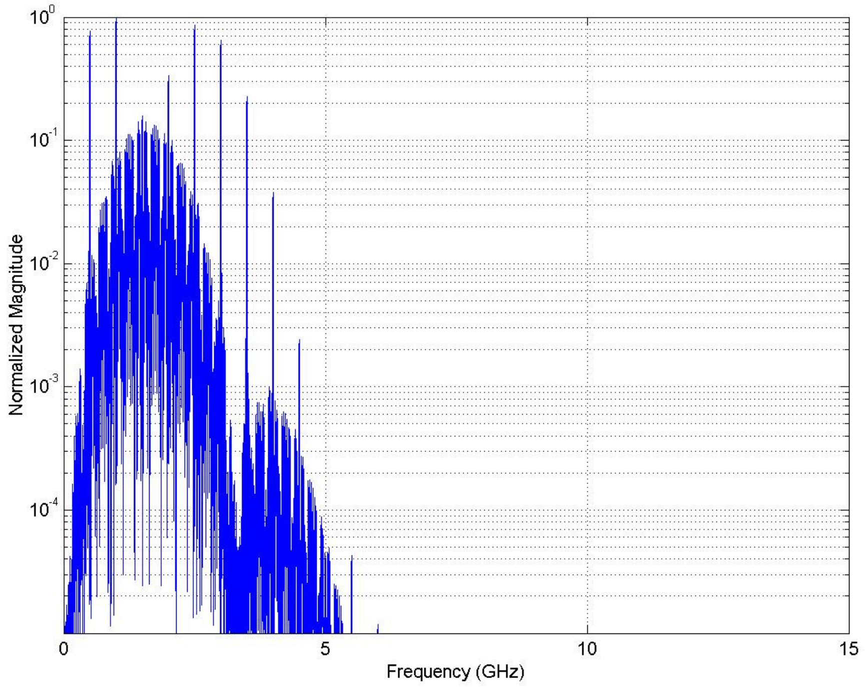


Figure 3-10: Spectrum of monocycle pulse train with offset pulses

3.5 MULTIBAND UWB

With this approach the available spectrum is divided into smaller frequency bands, each with a bandwidth of at least 500 MHz as shown in Figure 3-11. To accomplish this multiple UWB signals are transmitted at the same time but because they operate at different frequencies they do not interfere with each other.

The main advantages of multiband are its efficient use of the available spectrum and its flexibility to coexist with other wireless technologies because the separate bands may be treated independently.

Section 2.2.3 mentioned that the IEEE 802.15.3a high-speed UWB standard was never completed because consensus could not be reached for the choice of technology. The proposals were for Multi-Band Orthogonal Frequency Division Multiplexing (MB-OFDM) UWB and Direct Sequence (DS-UWB) UWB.

It is interesting to know that after IEEE disbanded the 802.15.3a Task Group, the DS-UWB approach was abandoned while the WiMedia Alliance continued to support the MB-

OFDM approach through the European Computer Manufacturers Association (ECMA). Today two international ISO-based specifications from ECMA, [45] and [46], exist with an adoption of the WiMedia Alliance MB-OFDM UWB radio platform by USB-IF for the highly anticipated Wireless USB [47].

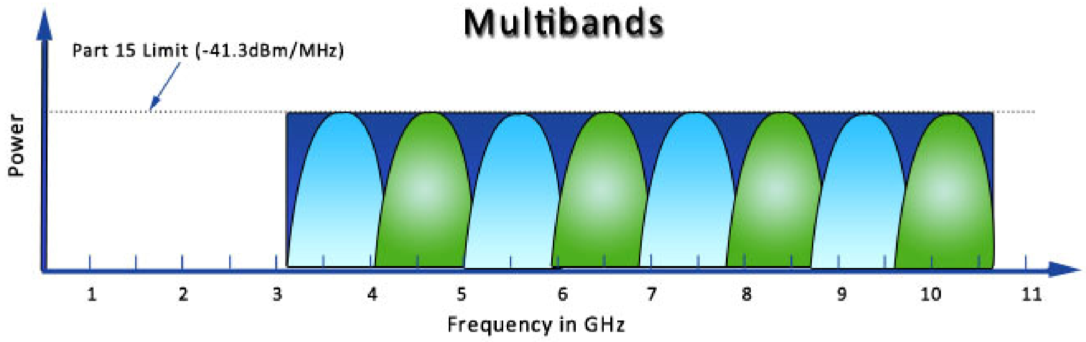


Figure 3-11: Multibands over the UWB spectrum

3.6 UWB MODULATION TECHNIQUES

To limit interference caused by a periodic pulse train and to provide access for multiple users wanting to communicate on the channel at the same time, the basic modulation must include techniques to allow these requirements. TH and DS are such techniques which we discuss in section 3.7.

First, we look at the different modulation methods available. The choice of modulation scheme depends on the operating conditions and desired system complexity.

3.6.1 Pulse Position Modulation

Modulation method where bit information is encoded by transmitting a time shifted version of the pulse used for communication. The concept is illustrated in Figure 3-12.

The time shift parameter is very important because the smaller the time shift, the more synchronized the receiver will have to be (implying complex receiver design). If the receiver is not properly synchronized it will make a lot of errors while trying to distinguish between the different pulses. The larger the time shift, the more bandwidth is wasted (compare with BPSK discussed in section 3.6.4).

If the time shift between the pulses equals one pulse width, PPM is categorized as an orthogonal modulation method.

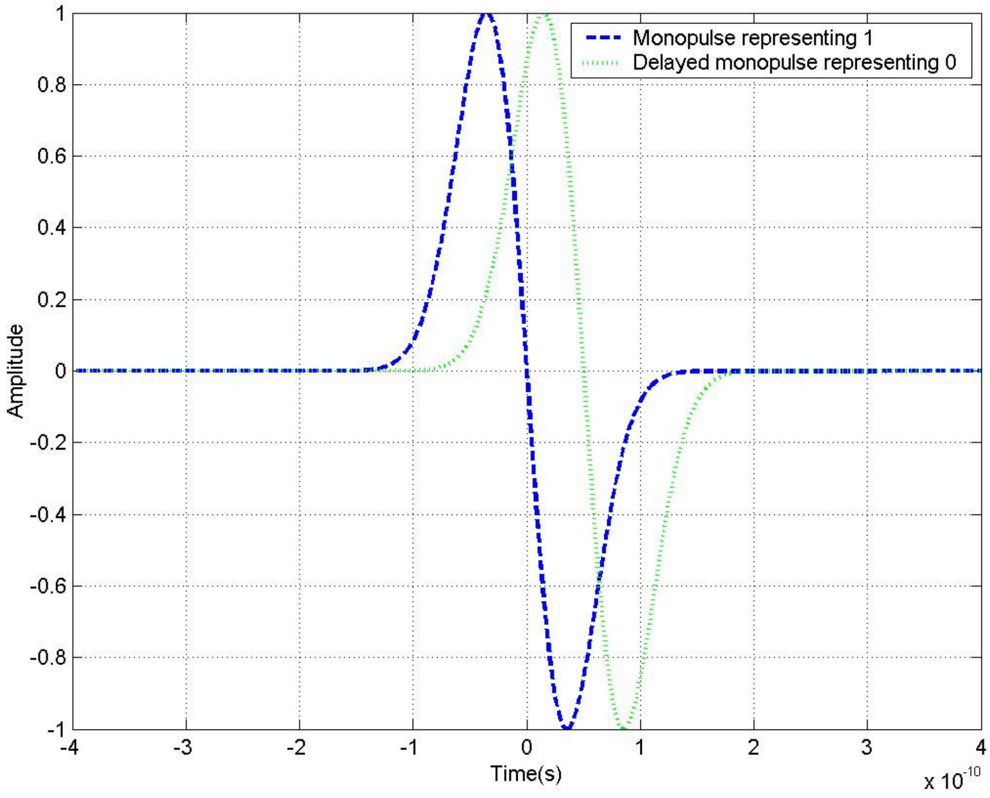


Figure 3-12: Example illustrating PPM pulses

3.6.2 Pulse Amplitude Modulation

PAM is a modulation method where the bit information is encoded into the amplitude of the transmitted pulse. The concept is illustrated in Figure 3-13.

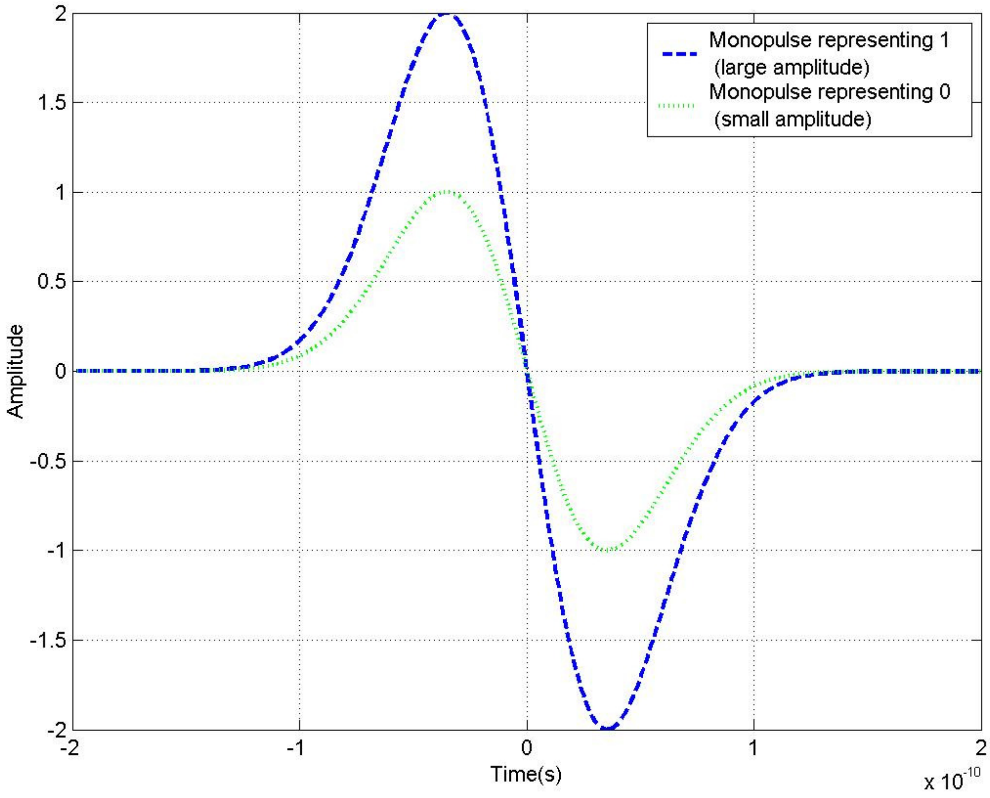


Figure 3-13: Example illustrating PAM pulses

3.6.3 On-Off Keying

In this modulation method, the absence of a pulse indicates a ‘0’ bit and the presence of a pulse a ‘1’ bit.

Albeit a simple modulation scheme it is very susceptible to noise. When no pulse is transmitted to indicate a ‘0’ bit, interfering signals might constructively add up and cause an incorrect bit decision by the receiver.

3.6.4 Binary Phase Shift Keying

Binary Phase Shift Keying (BPSK), also known as Bi-Phase Modulation, is classified as an antipodal modulation method. Basically information is encoded by inverting the transmitted pulse shape (reverses the pulse phase by 180°). The concept is illustrated in Figure 3-14.

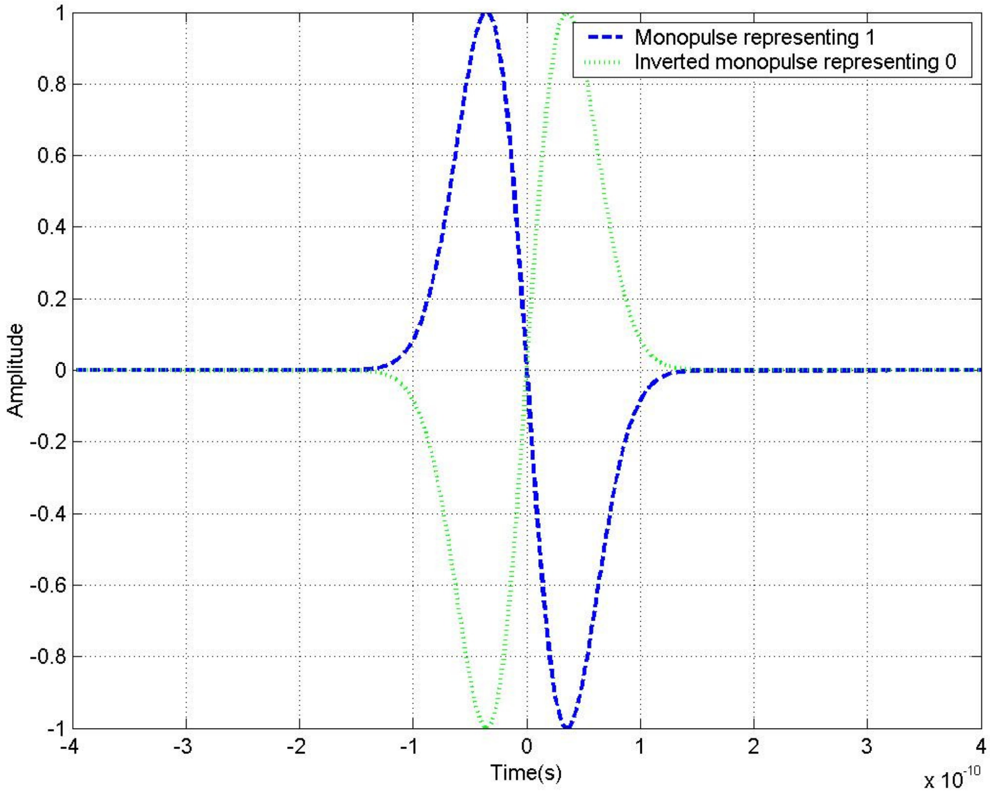


Figure 3-14: Example illustrating BPSK pulses

Compared to orthogonal PPM, BPSK provides for better bandwidth utilizations and therefore increased data throughput. An orthogonal PPM system will have to delay the pulse by at least one pulse width. During this delay period nothing is transmitted while in a BPSK system pulses can immediately be transmitted after each other allowing for twice the throughput.

3.6.5 Pulse Shape Modulation

Information can also be encoded by using two different pulse shapes. This technique is employed by PSM schemes and is illustrated in Figure 3-15.

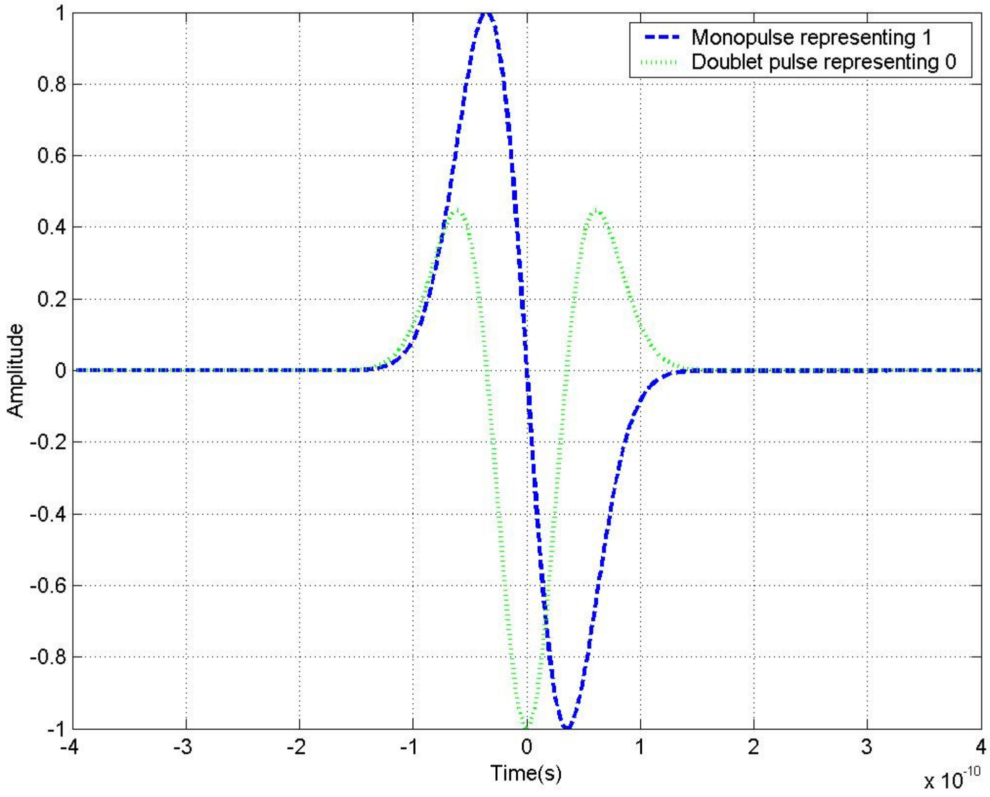


Figure 3-15: Example illustrating PSM pulses

3.7 MULTIPLE ACCESS STRATEGIES

There are three main approaches to enable multiple access in UWB systems [44]. Two of these techniques make use of unique pseudo random (PR) codes to identify users. The randomization also serves to limit interference caused by a UWB pulse train by spreading the RF energy across the frequency band. However the addition of a PR sequence requires that the receiver knows the PR sequence of every user it communicates with. The third approach defines unique frequency codes for each user.

PR codes and frequency codes are chosen to be orthogonal to ensure multiple signals from multiple users do not interfere with each other.

3.7.1 Direct Sequence

Direct Sequence is the first of the two techniques that makes use of PR codes. Each user employs its unique PR code to spread the data into multiple chips. A chip is represented by a UWB pulse.

In order to transmit a bit of information, a sequence of chips gets transmitted instead of transmitting bit per bit (see equation 3-16).

$$T_d = N_c . T_c \tag{3-16}$$

where the chip period T_c equals the UWB pulse period T_p. Therefore the chip rate is higher than the data rate and is defined by the Pulse Repetition Frequency (PRF).

If the PRF is kept constant, increasing the number of chips N_c will increase the processing gain as shown in equation 3-17 but it will also reduce the data rate.

$$PG = 10 \log_{10} N_c \tag{3-17}$$

DS allows users to transmit in the same bandwidth at the same time.

3.7.2 Time Hopping

Time Hopping is the second technique making use of PR codes. TH are usually used in conjunction with PPM modulation i.e. TH-PPM. Each user employs its unique PR code to hop transmissions in time. Therefore each user has its own time slice to transmit in with a bit being represented by the position of a pulse in its time slice [48].

With the TH approach the PRF determines the nominal transmit time of each pulse [49]. TH-PPM encodes bit information as a pulse occurring before the nominal transmit time and a pulse occurring after the nominal transmit time.

The number of users a TH system can support without interference is limited by the PR code length. A PR code of length L_{PR} provides L_{PR} unique moments in time that can be utilized for transmitting information. By increasing the number of possible hops (defined by L_{PR}) Multiple Access Interference (MAI) can be reduced but with a penalty in the data rate.

If a pulse train is used instead of a single pulse, the processing gain defined by equation 3-17 applies. In addition, the low duty cycle also adds to the total processing gain. The low duty cycle is due to the fact that the time hopping frame width $T_f >> T_p$ requiring the receiver to only listen for a small timeslot between pulses. The additional processing gain is defined in equation 3-18 [12].

$$PG_{dutyCycle} = 10\log_{10}\left(\frac{T_f}{T_p}\right) \tag{3-18}$$

3.7.3 Orthogonal Frequency Division Multiple Access

In OFDMA the frequency band used for communication is divided into sub-bands and each sub-band is assigned a unique frequency code. Each user gets pre-assigned one of these frequency codes that it uses to frequency hop a UWB pulse train to be transmitted. Consequently, users can transmit data simultaneously across multiple frequency ranges.

The divided frequency sub-bands are spread apart at precise frequencies to guarantee the frequency codes are orthogonal.

3.8 UWB CHANNEL MODELS

The channel represents the physical medium through which communication signals will propagate. In the case of a wireless technology, the signals travel through space and encounter various distortions from multiple paths due to reflections, electromagnetic interference from other wireless signals occupying the same space and inter symbol interference from neighbor devices.

Several path loss models exist to allow for experimentation and simulation of the attenuation these signal undergo before arriving at the receiver.

3.8.1 Free space propagation model

The free space path loss is the loss in a transmitted signal’s strength due to the spreading of the electromagnetic wave. The longer the transmit distance, the larger the spread. This model assumes a line-of-sight (LOS) path through normal free space to the receiver without any obstacles and as such does not include the effects of reflection, refraction or diffraction. The resulting path loss is given by

$$L(d) = 20\log\left(\frac{4\pi}{\lambda}\right) + 20\log{(d)} \tag{3-19}$$

where λ is the signal wavelength and d is the distance the signal travels [38].

3.8.2 Saleh-Valenzuela path loss model

The Saleh-Valenzuela model is for indoor environments only. It assumes that the multipath components arrive in clusters made up from the multiple reflections bouncing off objects close by. The arrival of multipath components follow a Poisson distribution with interarrival times corresponding to an exponential distribution [50].

3.8.3 Ghassemzadeh path loss model

The Ghassemzadeh model is a simpler and faster stochastic path loss model for indoor environments. It presents different path loss model parameters of UWB signals with a nominal frequency of 5 GHz. The two main parameters used for characterization are the path loss exponent and shadow fading standard deviation. These differ from location to location [51].

3.8.4 IEEE 802.15.4a path loss models

A. F. Molisch *et al* [52] presents path loss models for IEEE 802.15.4a specified by its working group. Different CM models are specified based on the Saleh-Valenzuela path loss model. CM1 defines a Residential LOS model, whereas CM2 can be used to model Residential non-line-of-sight (NLOS) scenarios. C3 is for Indoor Office LOS, CM5 for Outdoor LOS, CM6 for Outdoor NLOS and CM7 for Open Outdoor NLOS.

3.9 UWB TRANSCEIVER

A brief look at typical transceiver design is provided next. For a comprehensive discussion refer to the excellent work done by Aaron Orndorff [53].

3.9.1 UWB Transmitter

The transmitter has the task of converting a binary data stream into symbols and to map these symbols to analog waveforms. Using an antenna, the transmitter then has to transmit these signals through the channel while adhering to all required regulations.

A simple UWB transmitter block diagram is shown in Figure 3-16.

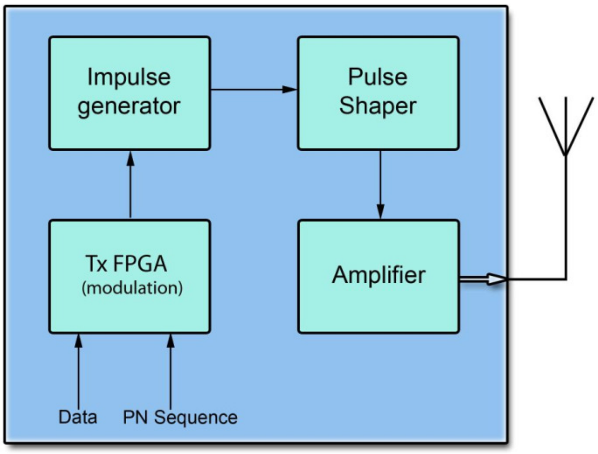


Figure 3-16: Simple UWB transmitter block diagram

3.9.2 UWB Receiver

The receiver has the task of amplifying and converting the received electromagnetic energy from its antenna to reconstruct a transmitted pulse shape. The receiver then maps these analog pulse shapes to the appropriate symbols and then converts those symbols into a binary bit stream representing the data.

An example of a simple energy detection receiver is shown in Figure 3-17.

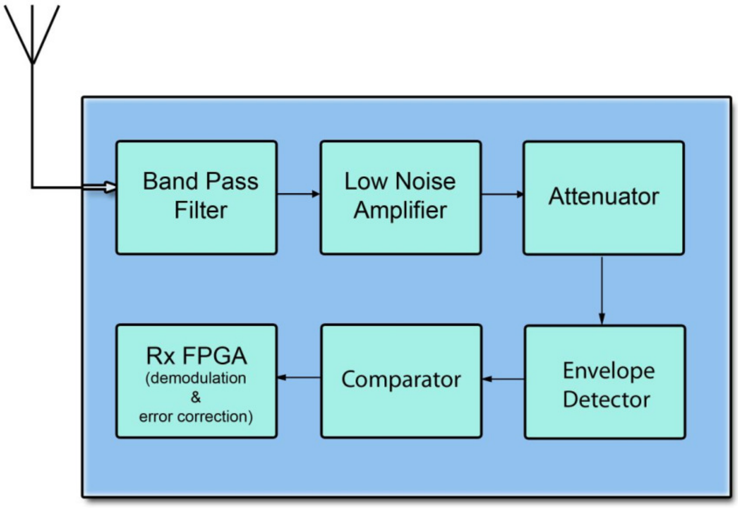


Figure 3-17: Simple UWB receiver block diagram

Forward Error Correction (FEC)

Error detection and correction plays an important role in receiver design. Redundant data gets added to the transmitted data at the sender and this extra data allows the receiver to detect and correct bit errors.

Error correction codes fall into two categories:

- Block codes: Performed on fixed-size blocks of bits
- Convolutional codes: Performed on bit streams of arbitrary length

This research study will go into great detail to compare the effects of forward error correction in the IEEE 802.15.4a UWB physical layer.

3.10 UWB ANTENNAS

Conventional antennas are not suited for UWB systems because they were designed to radiate over the narrow range of frequencies used in narrowband communication systems.

The following is a list of antennas considered for UWB systems:

- Monopole antenna
- Dipole antenna
- Conical antenna
- D-dot antenna
- Folded horn antenna
- TEM (Transient Electromagnetic) horn antenna
- IRA (Impulse Radiating Antenna)

It is desirable for a UWB antenna to be small, be embedded as part of the transceiver circuitry and be portable. An omnidirectional antenna, radiating power uniformly in all directions, is therefore implied. Furthermore, because of the wide range of frequencies a nonresonant antenna seems appropriate but the size of nonresonant antennas makes it unsuitable for mobile applications.

3.11 POSITIONING AND RANGING

There are many estimation techniques that makes use of radio signals to determine position and range. A non-exhaustive list of these techniques is:

- Received Signal Strength Indicator (RSSI)
- Time Of Arrival (TOA)
- Angle Of Arrival (AOA)
- Multilateration by computing the Time Difference Of Arrival (TDOA)

Although a major feature of UWB, this paper will not go into the details of ultra wideband positioning and ranging. The reader is referred to [44] and [49] for more information.

3.12 TYPICAL UWB APPLICATION AREAS

At the time of this writing, UWB technology is starting to move from prototype to commercial applications. The list of application areas for UWB will certainly grow in the future. A few areas are identified:

- Asset location
- Home networking
- Body area networks
- Sensor networks
- Video and audio distribution
- A great deal of suggestions exists for the medical field focusing mainly on imaging and monitoring applications of which there are a myriad of useful propositions.

3.13 UWB DISADVANTAGES

UWB might sound like a too-good-to-be-true technology, but it is not without certain disadvantages.

Disadvantages of UWB:

- Requires complex and very accurate timing components making receiver design difficult
- Only short range, due to strict limits on transmit energy by regulatory bodies
- Complex signal demodulation required
- Antenna design still a huge challenge
- Material penetration causes high losses

Chapter 4

Overview of IEEE 802.15.4a

A comprehensive study into the IEEE 802.15.4a standard is offered in this chapter.

The IEEE 802.15.4a standard [54] is an amendment to the 2006 revision of the IEEE 802.15.4 standard [55] (original IEEE 802.15.4 standard published in October 2003). Therefore, the body of work represented by this chapter also includes a study of the IEEE 802.15.4-2006 standard.

4.1 INTRODUCING THE PARTS

4.1.1 IEEE 802.15.4

IEEE 802.15.4 specifies the physical (PHY) and medium access control (MAC) layers of a low-rate Wireless Personal Area Network (WPAN). It targets applications for wireless devices requiring low data rates and low power consumption.

4.1.2 Zigbee

Zigbee is an emerging standard from the Zigbee Alliance [56] that is easily confused with IEEE 802.15.4. Zigbee is based on the IEEE 802.15.4 standard but provides a complete protocol stack that can be used for low cost, low power wireless personal area network applications. It adds a network layer with additional capabilities allowing for self-organizing mesh networks to be employed with features such as high security, multi-casting, and many-to-many routing.

4.1.3 IEEE 802.15.4a

IEEE 802.15.4a is the result of an amendment project to IEEE 802.15.4 for an alternative PHY, successfully completed by the IEEE 802.15 Low Rate Alternative PHY Task Group (TG4a) for Wireless Personal Area Networks (WPANs).

The focus of the committee was to draft an alternate PHY specification that provides communications with high precision ranging capability, high aggregate throughput, ultra-low transmit power, scalable data rates. In addition, the alternate PHY should provide for longer range, consume less power, and should be inexpensive.

On 22 March 2007 a baseline consisting of two optional PHYs was approved by the IEEE-SA Standards Board as a new amendment to IEEE 802.15.4-2006. The two optional PHYs consisted of a UWB Impulse Radio (IR) operating in the unlicensed UWB spectrum, and a Chirp Spread Spectrum (CSS) operating in the unlicensed 2.4GHz spectrum. This paper focuses exclusively on the UWB IR physical layer.

Note: IEEE 802.15.4 standard and IEEE 802.15.4a standard will be referred to as 802.15.4 and 802.15.4a, respectively, hereinafter.

4.2 GENERAL WPAN DESCRIPTION

4.2.1 Node types

Two types of devices are defined by the standard:

- Full Function Device (FFD): It is a device that supports all network functionalities. A FFD can have one of three roles:
    - PAN Coordinator: A device with the responsibility to set up, manage and maintain the PAN.
    - Coordinator: A device linking RFDs to the PAN coordinator.
    - Device: A normal device with dedicated tasks.

  A FFD can talk to both FFDs and RFDs.

- Reduced Function Device (RFD): It is a device that supports only a reduced set of network functionalities and is intended for applications that are extremely simple like a typical sensor node. A RFD can only talk to a FFD.

4.2.2 Topology

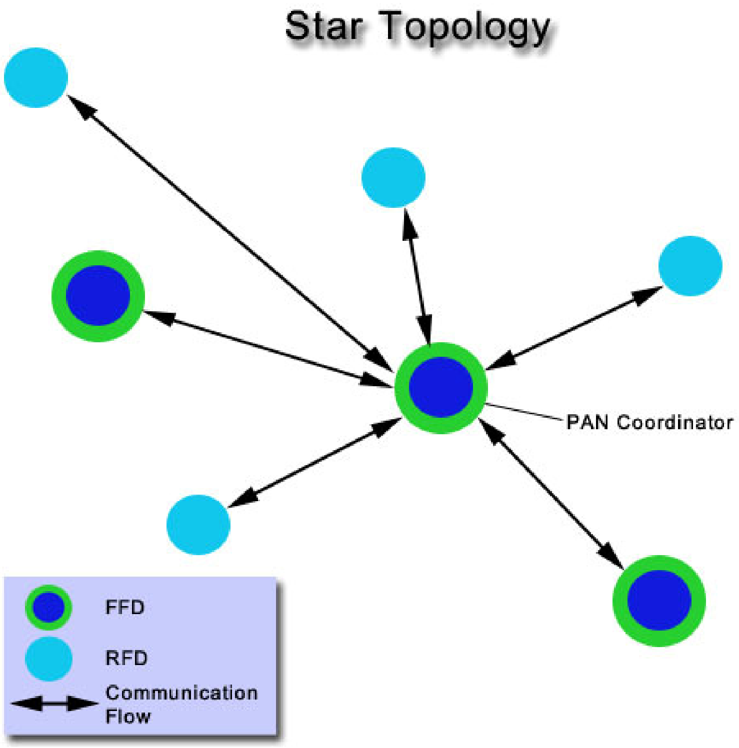
A WPAN requires at least one FFD to ensure there is a PAN coordinator. The FFD and RFD devices in a PAN organize themselves by talking to the PAN coordinator responsible with setting up and maintaining the PAN. To join the PAN a device needs to send an associate request to a coordinator.

All 802.15.4/4a devices have unique 64-bit addresses or alternatively a short 16-bit address can be allocated to a device by the PAN coordinator upon association. To enable communications across different PANs, each PAN has a unique identifier.

The standard defines two topologies a WPAN may operate in:

- Star topology: All devices can only communicate with the PAN coordinator.
- Peer-to-peer topology: Each FFD can communicate with any other FFD or RFD in its radio sphere of influence. A RFD can only communicate with a coordinator due to its limitations.

Figure 4-1 illustrates these two topologies and also a derivative of a peer-to-peer topology known as a cluster tree or mesh topology where small clusters of networks communicate peer-to-peer.



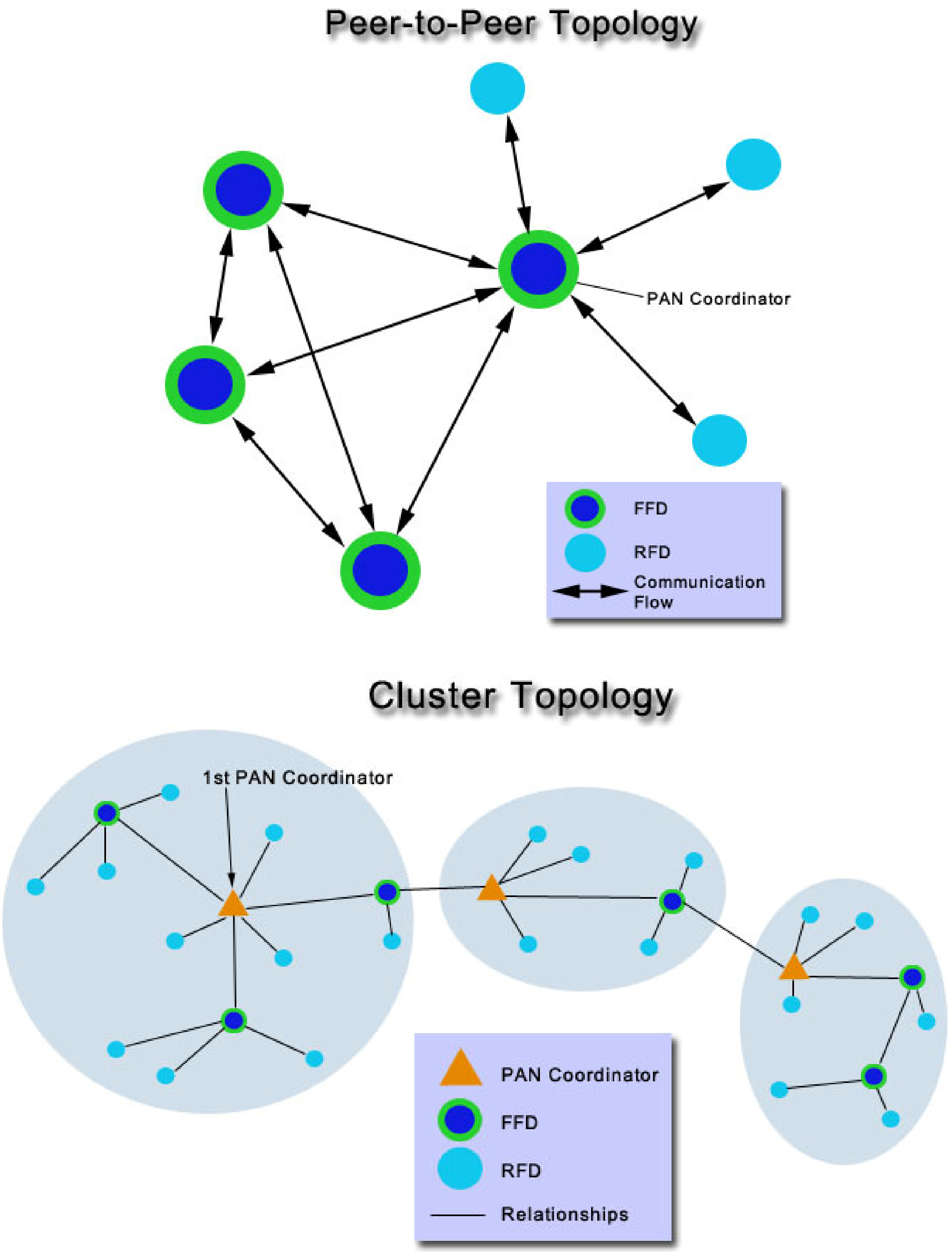


Figure 4-1: Examples of star, peer-to-peer and cluster topologies

4.2.3 Architecture

The 802.15.4/4a architecture is based on the layered approach followed for the open systems interconnection (ISO) seven-layer model. Each layer has its own set of responsibilities and offers its services to the next higher layer.

The standard defines a low-rate wireless personal area network (LR-WPAN) device's architecture as shown in Figure 4-2.

Physical (PHY) layer

The PHY specifies the radio frequency (RF) transceiver and low-level control mechanisms. The PHY provides two services, accessed through service access points (SAPs):

- PHY data service, accessed through the PHY data SAP (PD-SAP)
- PHY management service, accessed through the physical layer management entity SAP (PLME-SAP)

The PHY layer specification is discussed in section 0.

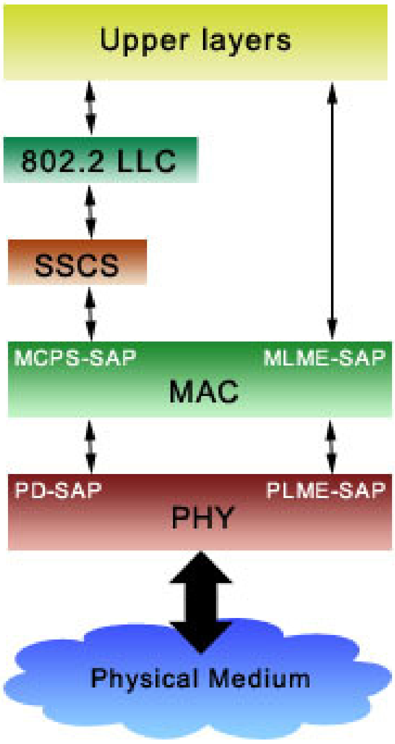


Figure 4-2: LR-WPAN device architecture

Medium Access Control (MAC) layer

The MAC provides access to the physical channel. The MAC provides two services, accessed through service access points (SAPs):

- MAC data service, accessed through the MAC common part sublayer (MCPS) data SAP (MCPS-SAP)
- MAC management service, accessed through the MAC sublayer management entity SAP (MLME-SAP)

SSCS

The service-specific convergence sublayer (SSCS) provides an interface through which the LLC can access the MAC sublayer.

802.2 LLC

The IEEE 802.2 Type I logical link control (LLC) layer presents a uniform interface to the upper layers, usually the network layer.

Upper layers

Not defined by the 802.15.4/4a standard. Basically a network layer responsible for network configuration and message routing tasks, and an application layer providing the functions for a device to fulfill its purpose.

4.3 FUNCTIONAL OVERVIEW

4.3.1 Medium access strategies

In a LR-WPAN access to the medium is based on a combination of random access and scheduled access [57]. Medium access is controlled by the PAN coordinator that may choose between two different modes of operation:

- Beacon-enabled: The PAN coordinator broadcasts a periodic beacon frame containing PAN information and providing synchronization. This mode is only used in star topology.

- Nonbeacon-enabled: The PAN coordinator makes exclusive use of random access for medium sharing because it does not transmit any beacon frames and as such no synchronization is provided. This mode is usually used for peer-to-peer topologies but can be adopted in a star topology network as well.

4.3.2 Superframe structure

The coordinator of a star topology network can choose to operate in the beacon-enabled mode. If so, the coordinator periodically sends beacons with the period between two consecutive beacons defining a specific fixed length superframe structure. The superframe structure is displayed in Figure 4-3 and serves to organize channel access and data transmission.

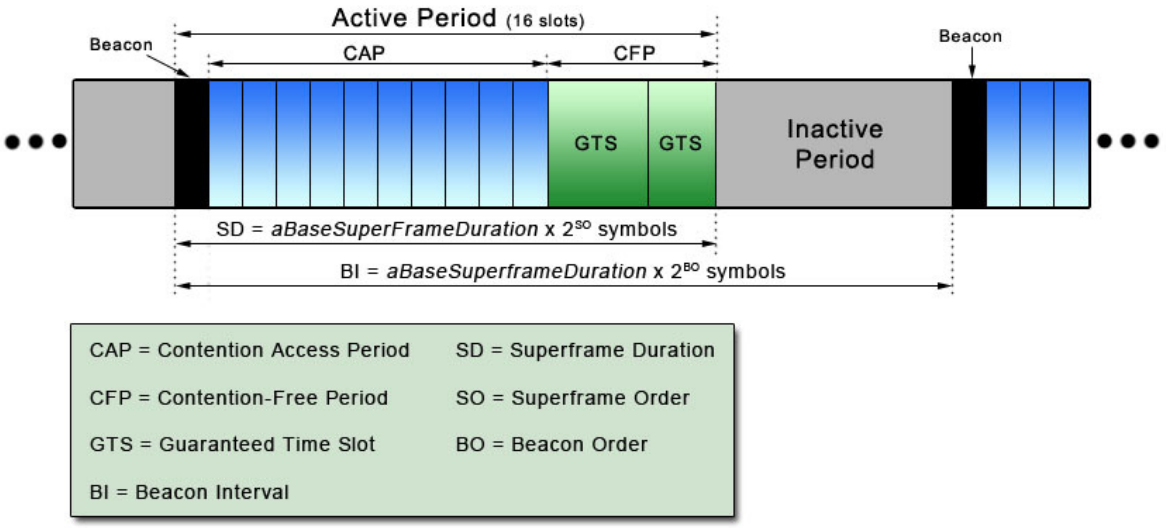


Figure 4-3: Superframe structure between beacons

The coordinator defines the format of the superframe. A superframe can have an active period and an inactive period bounded by beacons. Beacons get transmitted in the first slot of each superframe.

Active Period

The active period is divided into 16 equally spaced slots with a duration known as the superframe duration (SD) and consists of three parts: a beacon, a contention access period (CAP) and a contention-free period (CFP).

The CAP starts immediately following the beacon and is the time duration in symbols during which devices contend with each other to access the channel and transmit data. The contention-free period

The CFP is partitioned into a number of (maximum of seven) guaranteed time slots (GTSs). A GTS slot may span over more than one superframe slot. A GTS is a period of time during which certain low-latency application devices are given exclusive rights over the channel and can directly start transmitting data without having to content with other devices. The CFP starts immediately after the CAP.

Inactive Period

The inactive period is an interval during which the coordinator may enter a low-power sleep mode. No PAN interaction takes place so all other devices can also switch of their transceivers and sleep for the duration of the inactive period. The inactive period can be zero length.

4.3.3 Interval and duration calculations

Devices in the network compete with each other to get a chance to transmit data. In order to compete, devices need to know when the CAP starts and for how long it is available or when a GTS for a specific device starts. This information is provided to each device by the superframe structure. The length of the active and inactive period as well as the length of a single time slot can be configured. Basically, the structure of a superframe is determined by two parameters:

- Superframe Order (SO): Variable that determines the superframe duration (SD).
- Beacon Order (BO): Variable that determines the beacon interval (BI). The time duration between two successive beacons is known as the BI and is given in symbols.

The BI can be calculated as follows:

$$BI = aBaseSuperframeDuration.2^{BO} \tag{4-1}$$
$$for\ 0 \leq BO \leq 14$$

where $aBaseSuperframeDuration$is defined as the number of symbols in a superframe when SO = 0. $aBaseSuperframeDuration$ is calculated as follows:

$$\begin{aligned} &aBaseSuperframeDuration \\ &= aBaseSlotDuration.aNumSuperframeSlots \end{aligned} \tag{4-2}$$

where $aBaseSlotDuration$is the number of symbols in a superframe slot when $SO = 0$ and $aNumSuperframeSlots$ is the number of equally spaced slots in the active period of a superframe. The superframe slot duration is given by

$$slotDuration = 2^{SO}.aBaseSlotDuration \tag{4-3}$$

The superframe duration (SD), given in symbols, is calculated as follows:

$$\begin{aligned} SD &= aBaseSuperframeDuration.2^{SO} \\ &for\ 0 \le SO \le BO \le 14 \end{aligned} \tag{4-4}$$

The standard defines $aBaseSlotDuration = 60$ and $aNumSuperframeSlots = 16$ which gives us $aBaseSuperframeDuration = (60)(16) = 960$ symbols, according to equation 4-2.

The relationship between the SD and the BI can be given by

$$1 : 2^{(BO-SO)} \tag{4-5}$$

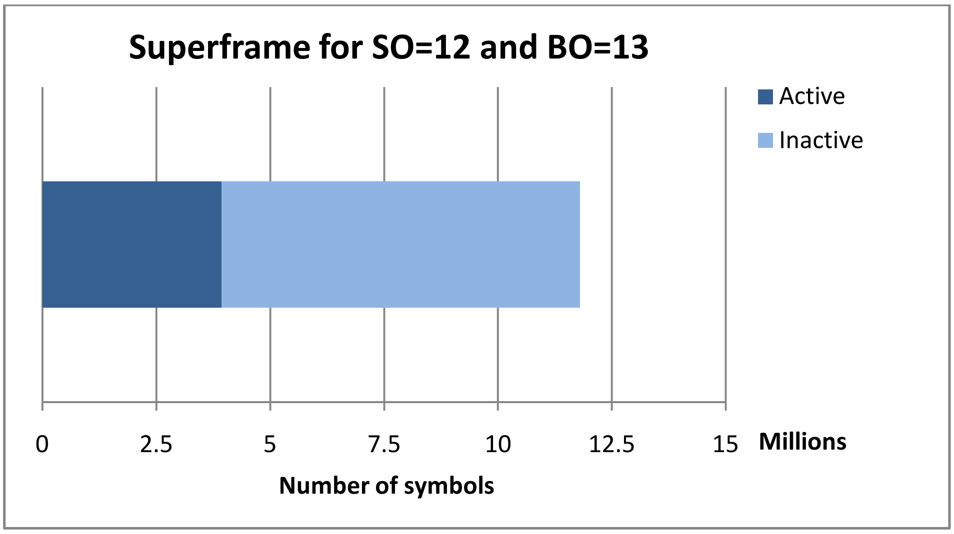
Using $aBaseSuperframeDuration = 960$ symbols this exponential relationship is illustrated in Figure 4-4.

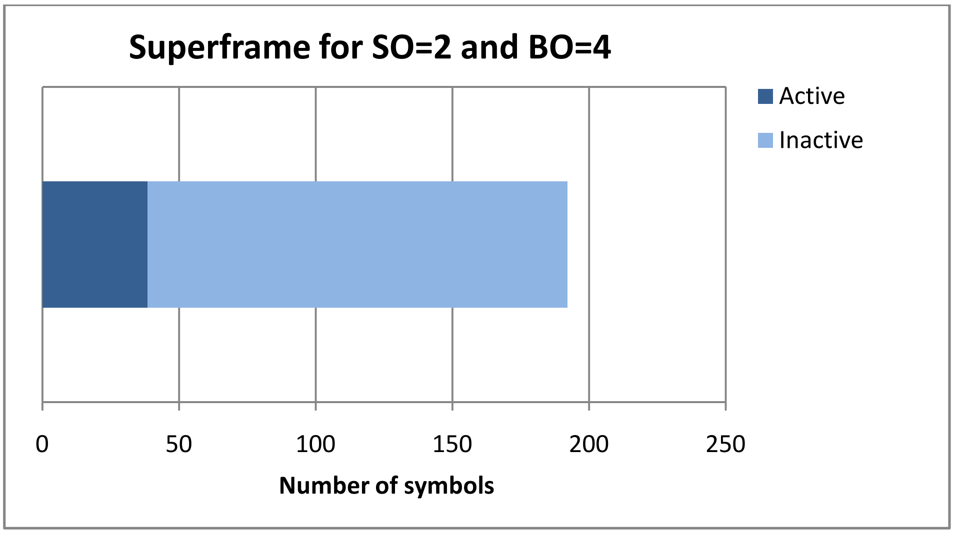
A value of 15 for BO indicates that the coordinator does not transmit any beacon frames except when requested to do so through a beacon request command. In such a case the value of SO is ignored because the superframe does not exist. In addition, GTSs is not allowed.

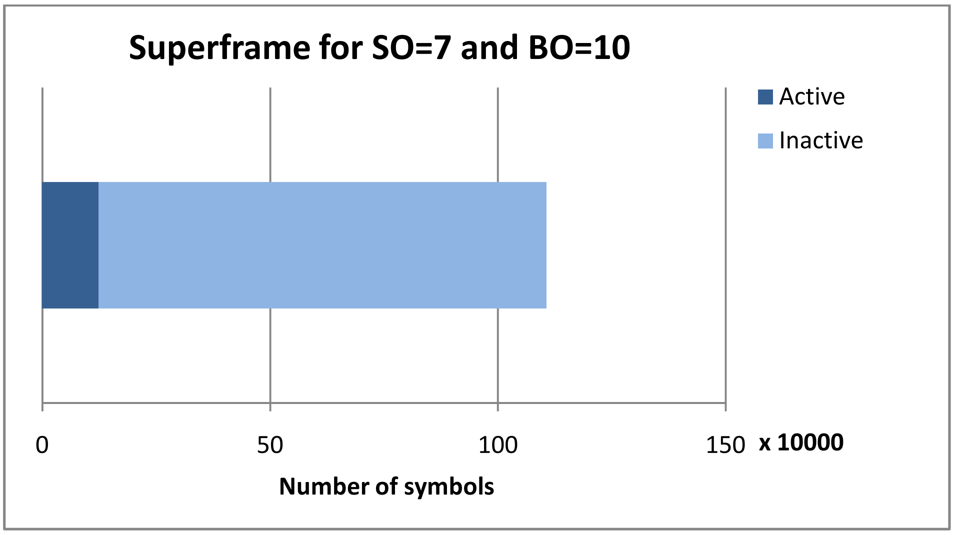
Therefore, we can specify the possible values for BO and SO as shown in Table 4-1.

Table 4-1: Permitted values for BO and SO

	BO	SO
Beacon-enabled	$0 \le BO \le 14$	$0 \le SO \le BO$
Nonbeacon-enabled	$BO = 15$	$SO = 15$


Superframe for SO=12 and BO=13
Active
Inactive
0
2.5
5
7.5
10
12.5
15
Millions
Number of symbols


Superframe for SO=2 and BO=4
Active
Inactive
0
50
100
150
200
250
Number of symbols


Superframe for SO=7 and BO=10
Active
Inactive
0
50
100
150
x 10000
Number of symbols

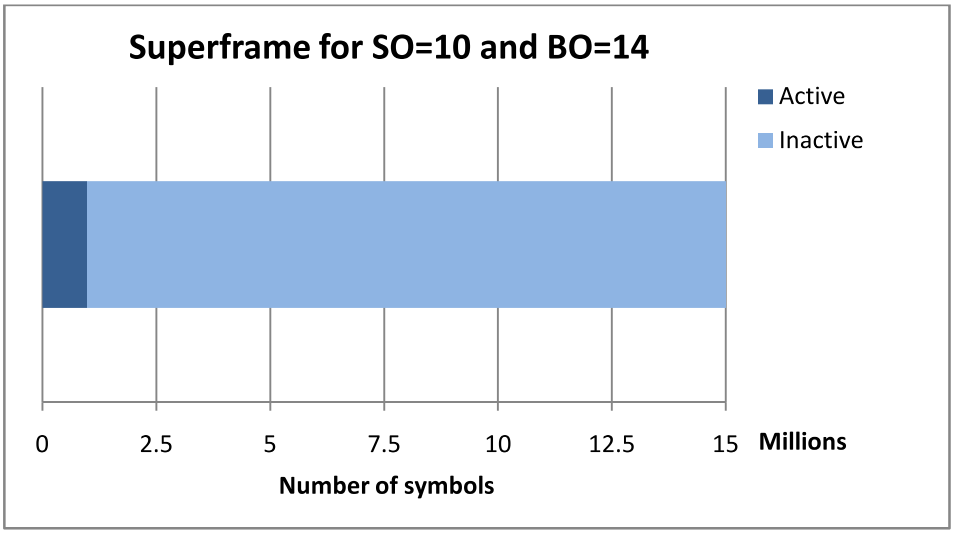


Figure 4-4: Illustrations showing the relationship between SO and BO

4.3.4 Data transfer

If a PAN requires synchronization or support for low-latency devices the PAN is setup to use beacons. If there is no need for synchronization or low-latency device support, the PAN can be setup not to make use of beacons for normal transfers. Beacons will still be required for the network discovery phase.

Three types of data transfer exist for LR-WPANs with different mechanisms for beacon-enabled and nonbeacon-enabled networks. Data can be transferred to the coordinator, from the coordinator or between two peer devices. Let’s take a closer look at each of these data transfer types.

To coordinator

Beacon-enabled PAN

The device listens for a network beacon and synchronizes to the superframe structure when a beacon is received. To transmit data, the device uses a slotted Carrier Sensing Multiple Access with Collision Avoidance (CSMA-CA) scheme to contend for channel access. The coordinator may reply with an optional acknowledgement frame if the data was received successfully.

Nonbeacon-enabled PAN

The device simply starts to transmit its data using unslotted CSMA-CA to the coordinator. The coordinator may reply with an optional acknowledgement frame if the data was received successfully.

From coordinator

Beacon-enabled PAN

If coordinator has data pending for a device, it announces so in the network beacon. Because the device periodically listens to the beacon it will be able to determine if a pending message is intended for itself. If so, the device transmits a MAC command request frame, using slotted CSMA-CA, to notify the coordinator that it is ready to receive the message. The coordinator will acknowledge the data request command and transmit the data to the device using slotted CSMA-CA. The device may reply with an optional acknowledgement frame if the data was received successfully. The coordinator will remove the successfully transmitted message from its list of pending messages to ensure it is not advertised in the next beacon.

Nonbeacon-enabled PAN

If coordinator has data pending for a device, it stores the data and waits for the device to make contact and request the data. A device may request the data through a MAC command request frame using unslotted CSMA-CA. The coordinator will acknowledge the data request command and if it has data pending for the device, the coordinator will transmit the data to the device using unslotted CSMA-CA. It the coordinator does not have any data pending for the device it will indicate so. The device may reply with an optional acknowledgement frame if the data was received successfully.

Between two peer devices

Communication between peer devices is not applicable to star topology networks because with a star network communication is only allowed between device and coordinator.

In peer-to-peer communications, all FFDs can communicate directly with each other if they are in range. To do this, the devices will have to receive and listens constantly to the channel and transmit its data using unslotted CSMA-CA. Alternatively, the devices will have to synchronize with each other through certain measures. The 802.15.4/4a standard does not specify any methods or techniques for peer device synchronization because it is beyond the scope of the standard. This is where the Zigbee Alliance with their Zigbee protocol stack with mesh networking support comes in. Refer to [58] for the Zigbee network layer standard which provides detailed information on how to accomplish peer-to-peer communications between devices.

4.3.5 Frame structure

The standard defines four frame structures to keep things simple.

- Beacon frame
- Data frame
- Acknowledgement frame
- MAC command frame

Beacon frame

Used by a coordinator for beacon transmissions in a beacon-enabled PAN. It originates from the MAC sublayer. The structure of a beacon frame is shown in Figure 4-5.

MAC beacon frame

This is the MAC Protocol Data Unit (MPDU) of the MAC sublayer. It consists of a MAC header (MHR), MAC payload and MAC footer (MFR). The MAC payload contains all superframe specific information related to the GTSs, pending data and beacon payload. The MFR is a 16-bit frame check sequence (FCS).

PHY packet

The MAC beacon frame (i.e. MPDU) is passed to the PHY layer as the PHY service data unit (PSDU). The PSDU becomes the payload of the PHY packet (refer to Addendum 8.2.7B for a description of service and protocol data units). The physical protocol data unit (PPDU) consists of the PSDU, PHY header (PHR) and synchronization header (SHR). The

SHR is made up of the Preamble Sequence and Start-of-Frame Delimiter (SFD) fields used to provide symbol synchronization at the receiver.

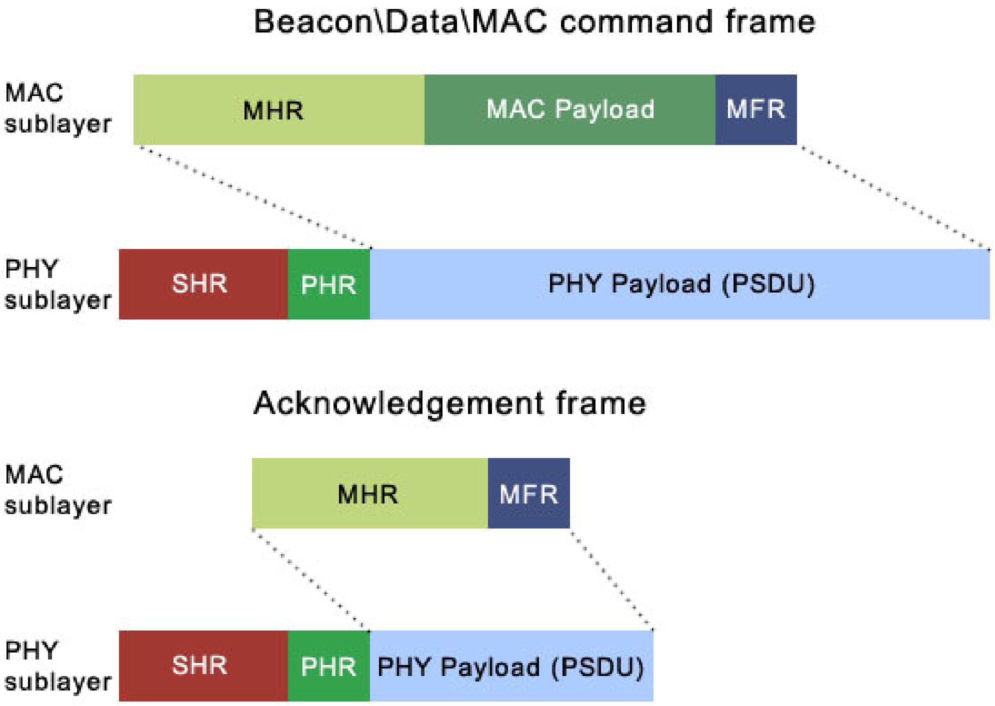


Figure 4-5: Structure of a MAC beacon frame and PHY packet

Data frame

Used for all data transfers and originates from the upper layers. The data payload is passed from the upper layers to the MAC as the MAC service data unit (MSDU). The structure of a data frame is shown in Figure 4-5. It is similar to the beacon frame except for some differences in the fields that make up the MHR and the MAC payload is a data payload.

Acknowledgement frame

Used to confirm the successful reception of frames and originates from within the MAC sublayer. The acknowledgement frame does not have a MAC payload. The structure of an acknowledgement frame is shown in Figure 4-5. It is similar to the other frame structures except for the missing MAC payload and some differences in the fields that make up the MHR.

MAC command frame

Used to handle all MAC peer entity control transfers and originates from within the MAC sublayer. The structure of a MAC command frame is shown in Figure 4-5. It is similar to the beacon and data frame structures except some differences in the fields that make up the MHR and MAC payload. The MAC payload of a MAC command frame contains the Command Type field and the command payload.

4.3.6 Delivery mechanisms

The standard employs various mechanisms to ensure the highest probability of successfully delivering data to its destination.

Data verification

As mentioned, the MFR consists of a FCS used to detect bit errors. The FCS is calculated over the MHR and MAC payload in the frame. The FCS is a 16-bit International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) cyclic redundancy check (CRC). The FCS is calculated using a standard generator polynomial of degree 16 given by

$$G_{16}(x) = x^{16} + x^{12} + x^{5} + 1 \tag{4-6}$$

Frame acknowledgement

An optional acknowledgment can be sent to confirm the successful reception and validation of a data or MAC command frame. If an acknowledgement is requested, the process is simple with the originator of the message waiting for some period of time for the acknowledgement after it has sent the message. If an acknowledgement is not received after the wait time, the message is sent again. This process repeats for a configured number of times after which the originator will terminate the transaction. If an acknowledgement is not requested, the originator assumes the message was transmitted successfully.

CSMA-CA mechanism

802.15.4 specifies two types of CSMA-CA for channel access depending on the network configuration:

- Slotted CSMA-CA, used in beacon-enabled PANs
- Unslotted CSMA-CA, used in nonbeacon-enabled PANs

Carrier Sense Multiple Access (CSMA) is a protocol in which a node firsts checks if the communication channel is idle before transmitting. The node listens for a carrier wave to determine this idle state of the channel.

CSMA-CD (Collision Detect) improves the CSMA scheme by breaking transmission if a node detects another signal while it is busy transmitting.

CSMA-CA (Collision Avoidance) improves the CSMA scheme by first checking if no communication is taking place on the channel for a predetermined period of time before declaring the channel as idle. Furthermore the node wishing to transmit sends a backoff message to all other nodes telling them not to transmit before it starts using the channel.

The details of CSMA-CA and its slotted and unslotted approaches are not discussed in this document.

ALOHA mechanism

802.15.4a specifies the ALOHA method as an alternative channel access strategy. The ALOHA method, developed at the University of Hawaii, defines a very simple communication scheme. Any device in the network that has data to transmit will send the data immediately. If the data was delivered successfully to the destination device, the next data packet is sent. If the data was not delivered successfully to the destination device, the same data packet gets sent again.

Obviously the setback with the ALOHA method is collision problems. If two devices transmit at the same time the transmitted frames collide with each other and the data of both have to be resent. The number of collisions increases when the volume of communications increases.

Two types of ALOHA are defined:

- Pure ALOHA
- Slotted ALOHA

Pure ALOHA is the approach discussed above. Slotted ALOHA try to optimize network efficiency by minimizing the number of collisions. The system employs beacon signals sent at precise intervals with the effect of splitting time into slots. A device can only transmit on the beginning of a timeslot, thus reducing the chance of collisions.

If the arrival of packet follows a Poisson distribution pure ALOHA has a maximum throughput of about 18.4%. This is a loss of about 81.6% of the total available bandwidth. Optimizations resulting from the slotted ALOHA approach, increased the maximum throughput to 36.8%.

Di Benedetto *et al* [59] demonstrated that for light and medium network traffic loads the ALOHA approach offers satisfactory throughput in UWB networks due to the robustness UWB technology provides against multi-user interference. Packet loss due to collisions is also reduced if a time hopping scheme is adopted which will introduce a different delay one each burst in a packet.

4.3.7 Ranging

This paper will not go into the details of 802.15.4a ranging. The reader is referred to pp. 8 - 20 of [54] for a detailed ranging overview.

4.4 ULTRA WIDEBAND PHYSICAL SPECIFICATION

We will now take a detailed look into the workings of the ultra wideband alternate physical layer as specified in the IEEE 802.15.4a standard.

The Direct Sequence UWB approach was chosen for the standard due to its spectral efficiency, robustness at low transmit powers and support for high-precision ranging. UWB PHY waveforms employ an impulse radio scheme.

4.4.1 Channels and operating frequency bands

The standards specify three independent frequency bands and a total of 16 channels (or 32 complex channels). The bands, channels and their center frequencies are listed in

Table 4-2 with a pictorial representation provided in Figure 4-6.

Table 4-2: 802.15.4a UWB operating frequency bands and channel information

Band	Channel Number	Center Frequency	Bandwidth
0 (Sub-GHz band)	***0***	***499.2 MHz***	***499.2 MHz***
1 (Low band)	1	3494.4 MHz	499.2 MHz
	2	3993.6 MHz	499.2 MHz
	3	***4492.8 MHz***	***499.2 MHz***
	4	3993.6 MHz	1331.2 MHz
2 (High band)	5	6489.6 MHz	499.2 MHz
	6	6988.8 MHz	499.2 MHz
	7	6489.6 MHz	1081.6 MHz
	8	7488.0 MHz	499.2 MHz
	9	***7987.2 MHz***	***499.2 MHz***
	10	8486.4 MHz	499.2 MHz
	11	7987.2 MHz	1331.2 MHz
	12	8985.6 MHz	499.2 MHz
	13	9484.8 MHz	499.2 MHz
	14	9984.0 MHz	499.2 MHz
	15	9484.8 MHz	1354.97 MHz

A single mandatory channel is specified for each band, one of which a compliant device must implement to adhere to the standard. The rows marked bold and italic in Table 4-2 indicate these mandatory channels.

Inside each channel there is support for two complex channels. A complex channel has a unique 31 bit preamble code used to construct the synchronization header part of a UWB PHY frame. The channel number together with the preamble code makes up a complex channel. A compliant device must support both complex channels of each channel it implements.

Channels 4, 7, 11 and 15 are optional channels and have a bandwidth > 500 MHz. To meet regulatory PSD constraints the larger bandwidth allows for transmission at higher power resulting in a longer communication distances and more accurate ranging.

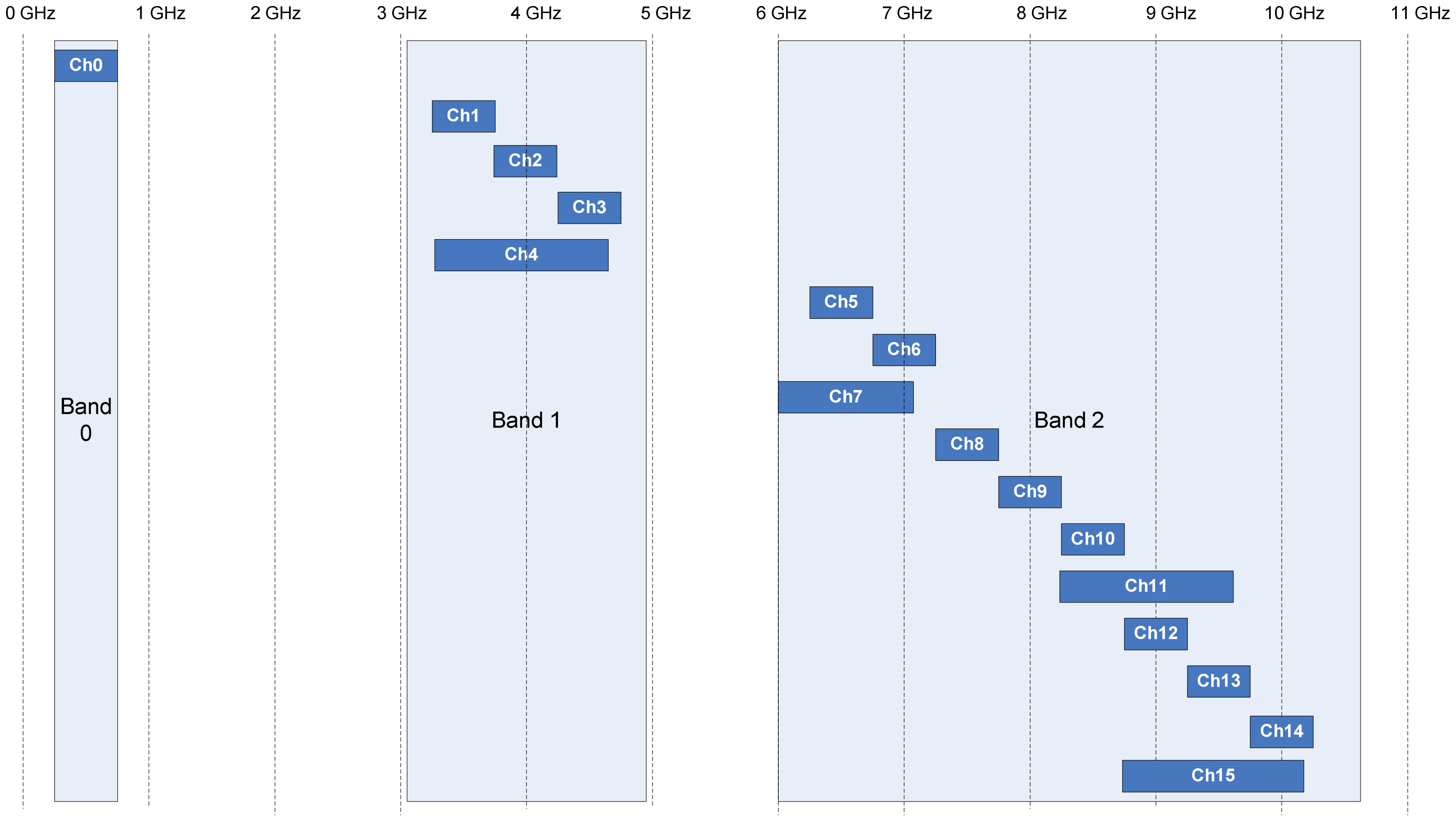


Figure 4-6: Graphical view of 802.15.4a UWB operating frequency bands and channels

4.4.2 Signal flow

The typical steps involved to modulate and create an UWB PHY frame at the transmitter and then detect and demodulate the frame at the receiver are illustrated in Figure 4-7.

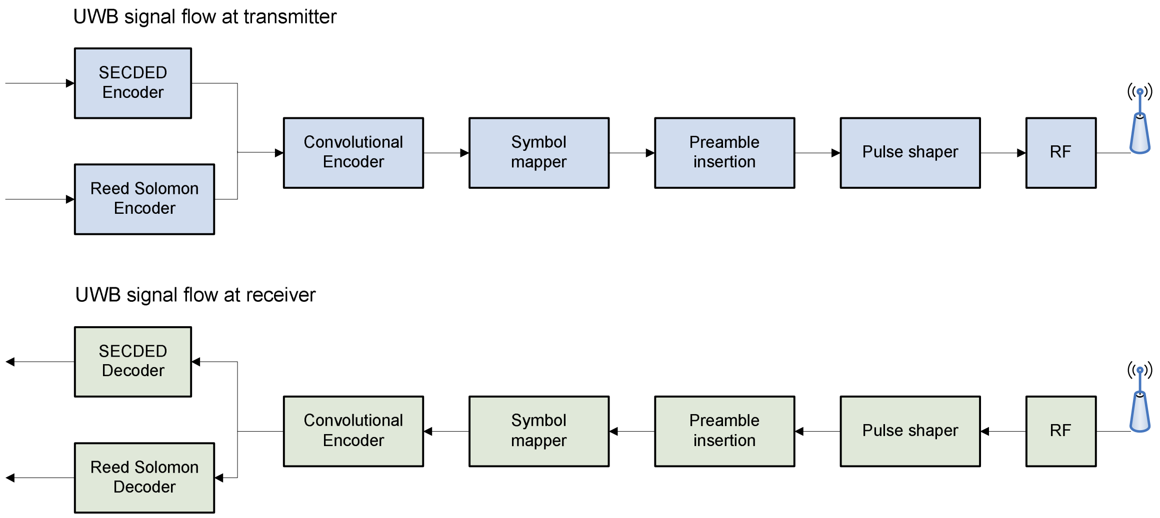


Figure 4-7: Typical 802.15.4a UWB PHY frame processing at transmitter and receiver

The reasons behind and the functions performed by the different steps in the illustration will become clear during the discussions in the next sections.

4.4.3 UWB frame format

Figure 4-8 shows the format of an 802.15.4a UWB frame. It consists of three parts:

1. Synchronization Header (SHR)
    a. Synchronizations (SYNC) portion
    b. Start Of Frame Delimiter (SFD) portion
2. Physical Header (PHR)
3. Payload or Data part

Figure 4-8: 802.15.4a UWB PHY frame format

Various UWB pulse trains lengths are used to obtain various transmit data rates for the different frame parts. The data rates for each part can be found in Table 4-3.

Table 4-3: Different data rates for 802.15.4a UWB PHY frame parts

Part	Data rate(s)
SHR	1. Msymbols/s or 0.25 Msymbols/s depending on the preamble PRF
PHR	If data rate is above 850kb/s then 850kbps, else 110kb/s
Data	At the information data rate defined by the various UWB symbol timing parameters

4.4.4 UWB symbol structure

The modulation scheme adopted is a combination of Burst Position Modulation (BPM) and Binary Phase-Shift Keying (BPSK). BPSK was discussed in section 3.6.4. BPM makes use of the UWB PHY symbol structure to encode information. The 802.15.4a UWB PHY symbol structure is shown in Figure 4-9. Notice how the symbol contains two separate intervals in which a burst of UWB pulses can occur. Only one of these burst intervals in a symbol are allowed to contain a burst at a time and therefore the position of the interval containing the burst can be used to carry information.

As such, a single UWB PHY symbol can carry two bits of information at a time. The burst position determines one bit and the other bit gets modulated in the phase of that same burst.

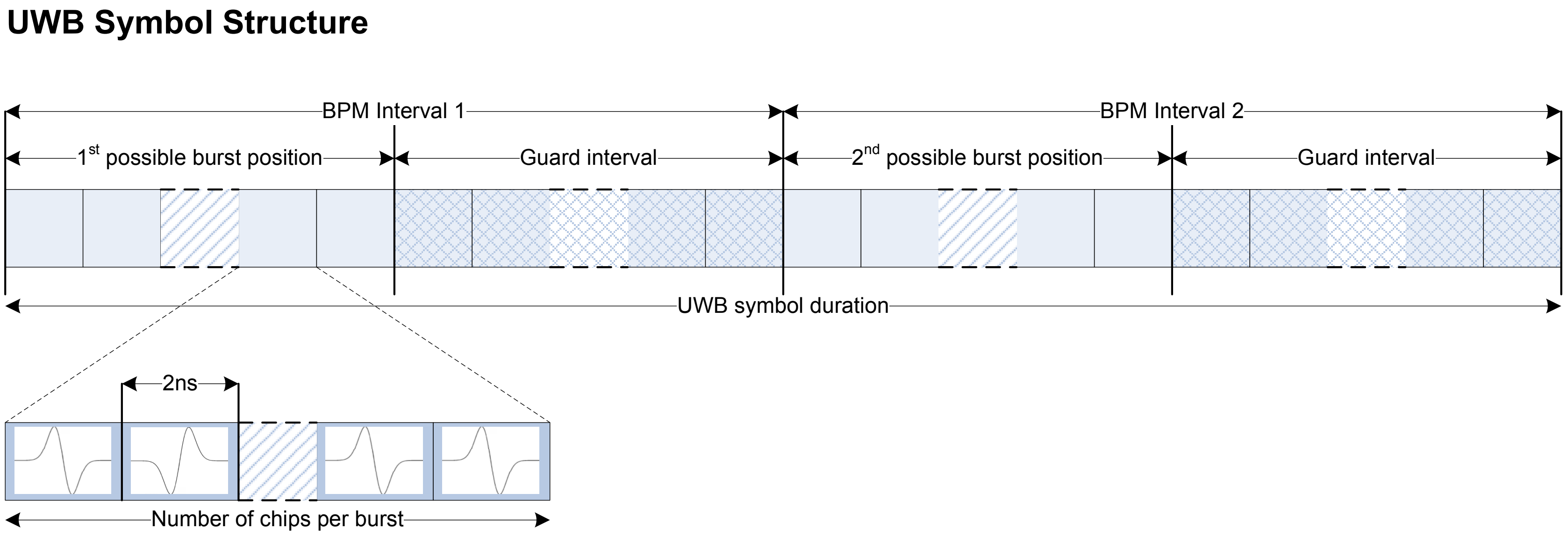


Figure 4-9: Structure of a 802.15.4a UWB PHY symbol

UWB symbol timing parameters

The UWB symbol duration (T_{dsym}) is defined as the total number of chip positions in the symbol (N_c) multiplied by the duration of one chip interval ($T_c \approx$ 2ns) – see equation 3-16. For BPM modulation T_{dsym} is divided by 2 to get the duration of a single BPM interval (T_{BPM}). Only the first half each T_{BPM} may contain a burst. The second half is used as a guard interval to limit inter-symbol interference (ISI).

A burst is a grouping of consecutive UWB pulses (chips) and has a duration T_{burst} defined as the number of chips in a burst (N_{cpb}) multiplied by T_c. The phase of the burst is modulated with the second bit of information. The total number of intervals in a symbol that may host a burst is given by N_{burst} and is defined as the duration of the UWB PHY symbol (T_{dsym}) divided by the duration of a burst (T_{burst}).

Note that $T_{burst} \ll T_{BPM}$ which allows for the time hopping scheme to provide some multi-user access. For every symbol the position of a burst is varied according to a predefined time hopping code.

The calculations for the timing parameters introduced in this section are given by equations 4-7 to 4-10.

$$\begin{aligned} T_{dsym} &= N_c . T_c \\ &= N_{burst} . T_{burst} \end{aligned} \tag{4-7}$$

$$T_{BPM} = \frac{T_{dsym}}{2} \tag{4-8}$$

$$T_{burst} = N_{cpb} . T_c \tag{4-9}$$

$$N_{burst} = \frac{T_{dsym}}{T_{burst}} \tag{4-10}$$

4.4.5 UWB PHY rate and timing parameters

Several data rates are defined for each channel. By keeping N_{burst} constant and varying N_{cpb} the duration of the symbol (T_{dsym}) will change resulting in different data rates. In addition to the UWB symbol timing parameters, the data rate is also affected by other parameters as presented in Table 4-4.

Table 4-4: 802.15.4a UWB PHY rate and timing parameters

Peak PRF
For each UWB channel the peak PRF must be 499.2 MHz. The peak PRF defines the chip duration: $T_c = \frac{1}{PRF_{peak}} \approx 2ns$
Mean PRF
The average PRF during the data part of the UWB PHY frame. $PRF_{mean} = \frac{N_{cpb}}{T_{dsym}}$
Preamble code length
Together with the channel number the preamble defines a complex channel. Preamble code is used during the SHR part of the UWB PHY frame. Preamble code length can be 31 bits or 127 bits.
Viterbi Rate
Indicates the effect of the convolutional encoding on the data rate.
RS Rate
Indicates the effect of the Reed-Solomon encoding on the data rate.
FEC Rate
Indicates the total effect of the Forward Error Correction (FEC) applied. $FEC_{rate} = Viterbi_{rate}. RS_{rate}$
Symbol Rate
The inverse of the duration of the UWB symbol on the air. $Sym\ Rate = \frac{1}{T_{dsym}}$
Bit Rate
The information throughput rate including the effects of FEC. $Bit\ Rate = \frac{2. FEC_{rate}}{T_{dsym}}$

A complete list of rate-dependent parameters and timing-related parameters for the PSDU are given in Table 39a of the 802.15.4a specification [54].

Peak PRF vs Mean PRF

In the data part of the UWB PHY frame, the peak and mean PRFs are different because of the pulse bursts in the UWB symbols. In the SHR preamble part they are the same because pulses are distributed uniformly throughout the preamble symbol.

4.4.6 SHR preamble

The SHR preamble aids the receiver with frame detection and synchronization. The SHR preamble is made up of a SYNC portion and SFD portion.

Table 4-5 introduces the preamble parameters. For the full list with the possible values of each the reader is referred to Table 39c of the 802.15.4a specification [54].

Table 4-5: 802.15.4a UWB SHR preamble parameters

Mean PRF
The SHR preamble is sent at a slightly higher mean PRF than the data. The PRF_{mean} has the following three values: - 4.03 MHz - 16.10 MHz - 62.89 MHz
Delta length (δ_L)
The number of chip durations to insert between preamble code symbols.
$N_{c/psym}$
The number of chips per preamble symbol. Defined as the preamble code length multiplied by the delta length. $N_{c/psym} = \delta_L . x$ where x = 31 or 127.
T_{psym}
The duration of a single preamble symbol.
N_{sync}
The number of preamble symbols in the SYNC portion of the SHR preamble. The standard defines four N_{sync} sizes. - Short = 16

- Default = 64 - Medium = 1024 - Long = 4096
$\boldsymbol{N_{sfd}}$
The number of preamble symbols in the SFD portion of the SHR preamble.
$\boldsymbol{T_{sync}}$
The duration of the SYNC portion.
$\boldsymbol{T_{sfd}}$
The duration of the SFD portion.
$\boldsymbol{T_{pre}}$
The duration of the whole SHR preamble.

SHR SYNC

The SYNC portion provides for packet synchronization and channel estimation. It is also employed for the ranging functionality incorporated in the standard.

Each channel has two unique 31 length preamble codes and extra optional 127 length preamble codes are also specified. The combination of channel number and preamble code is known as a complex channel. Preamble codes consist of the code symbols from the ternary alphabet {-1, 0, 1}. They were chosen such that the codes used in the same channel have the lowest cross-correlation.

The SYNC portion is constructed by repeating the preamble symbol a certain number of times. The repeat number is defined by the value of the parameter N_{sync}.

The preamble symbol is built up by taking the preamble code and inserting a specific number of chip durations between the ternary symbols of the preamble code. The number is defined by the value of the delta length parameter δ_L. The construction of the preamble symbol is defined by the Kronecker operation $\otimes$ as shown in equation 4-11. The concept is illustrated with the example provided in Figure 4-10. Note that the values used in the example are not supported by the standard but are for explanation purposes only. Spreading is accomplished by adding (L-1) zeros after every ternary preamble code symbol.

$$S_i = C_i \otimes \delta_L(n)$$

$$with\ \delta_L = \begin{cases} 1 & ,n = 0 \\ 0 & ,n = 1,2\ldots,L-1 \end{cases} \tag{4-11}$$

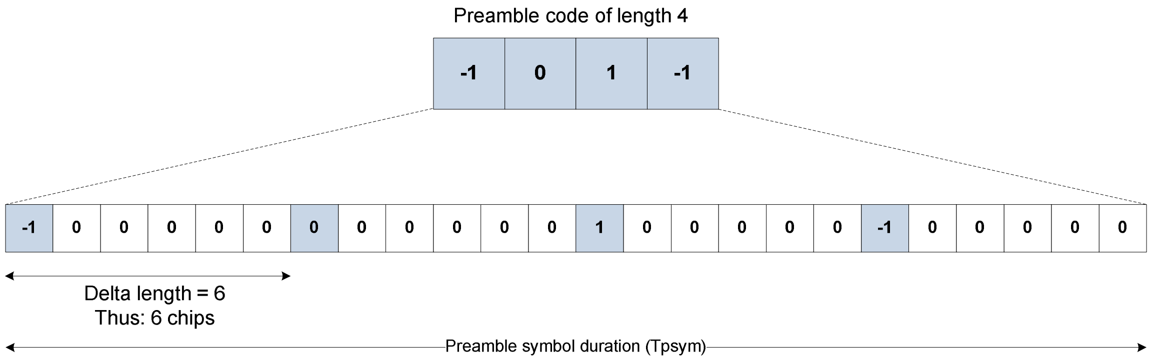


Figure 4-10: Example illustrating SHR preamble symbol construction

The construction of the SYNC portion by making use of the preamble symbol is shown in Figure 4-11.

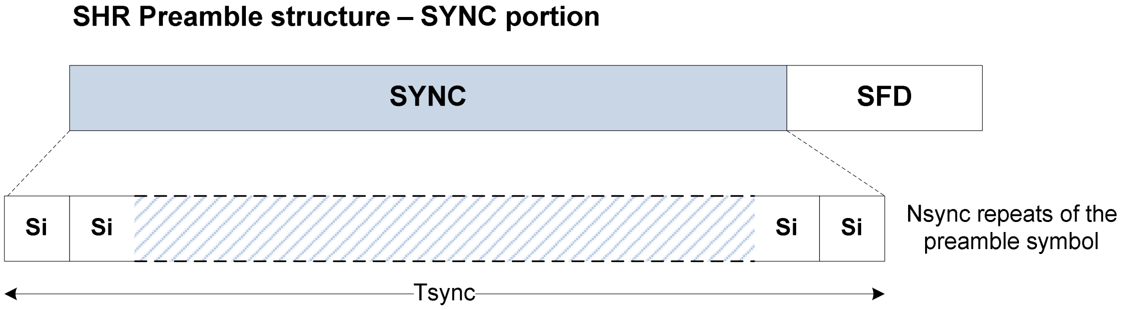


Figure 4-11: SHR preamble structure – SYNC portion

SHR SFD

The SFD acts as the boundary between the synchronization part and rest of the UWB PHY frame and is used to establish frame timing.

Two ternary SFD codes are defined - one of length 64 for the nominal data rate of 110 kbps and one of length 8 for the other data rates. The SFD code is spread by the preamble symbol to make up the SFD part of the UWB PHY frame (see Figure 4-12).

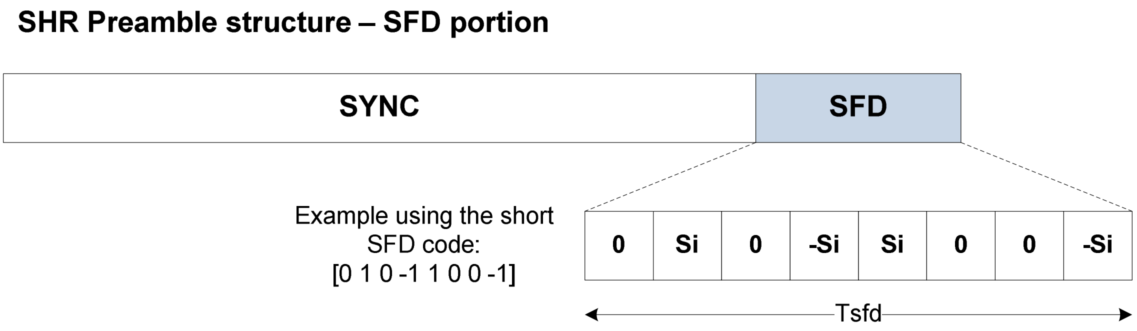


Figure 4-12: SHR preamble structure – SFD portion

A detailed description of the construction of the SYNC and SFD portions is provided in section 6.2.8.

4.4.7 PHR

The physical header (PHR) follows the SHR preamble. It is used to help the receiver successfully decode the packet. The PHR is 19 bits in size and it contains the following information:

- Data rate
- Frame length (number of bytes in PSDU from MAC)
- Ranging indicator
- Header extension
- SHR preamble SYNC portion duration
- SECDED parity check bits

SECDED bits

To protect the PHR information bits from errors a simple Hamming error detection and correction block code, consisting of 6 bits, is used. SECDED is the abbreviation for single-error correct, double-error detect.

4.4.8 Data

The data field is encoded as follows:

1. A systematic Reed-Solomon block code encodes the data.

2. The encoded Reed-Solomon block code is then encoded with a systematic convolutional encoder.
3. The encoded convolutional bits are then spread and modulated using BPM-BPSK modulation.

Reed-Solomon encoding

The Reed-Solomon (R-S) encoder appends 48 parity bits for every 330 bit blocks of data. For blocks of data with less than 330 bits, dummy zero bits are added. The code is defined as an R-S (63, 55) code.

Note that the R-S code is systematic and therefore decoding is optional. The original data can be obtained by just removing the R-S parity bits.

Convolutional encoding

Unlike the R-S coder working with data blocks, convolutional codes work with data bit streams.

The specified convolutional encoder makes use of the two generator polynomials given in equation 4-12 which were chosen to be noncatastrophic and systematic.

$$\begin{aligned} g_0 &= (010)_2 \\ g_1 &= (101)_2 \end{aligned} \tag{4-12}$$

The convolutional encoder has a rate of R $= 1/2$, signifying that for each input bit it will produce two output bits. Sequence g_0 is known as the position bits and they encode the position of a burst while sequence g_1 is known as the sign bits and encode the polarity of the pulses in a burst.

Note that a non-coherent receiver is not able to decode the polarity of a burst and as such it will not be able to make use of the sign bits for decoding. In such a case the receiver can comfortably ignore the sign bits because the convolutional code is systematic. The original data will still be intact in the position bit sequence.

Decoding

The 802.15.4a standard does not specify any details about decoding at the receiver.

If a receiver does not support forward error correction at all, it can simply ignore the code bits because both the RS-code and convolutional code is systematic.

The mathematical properties of the generator polynomials defined by the Reed-Solomon encoding algorithms allows for the R-S decoder to detect and correct a limited amount of errors.

To decode the convolutional code at the receiver the most popular universal decoder is based on maximum likelihood estimation and is known as the Viterbi decoder.

Spreading and Modulation

As mentioned, two information bits are encoded in one UWB PHY symbol. One into the burst position and one into the polarity of the burst. To suppress the negative effects of interference, time hopping and polarity scrambling schemes are employed.

Section 3.7.2 discussed how time hopping can be used to minimize multiple access interference (MAI) caused by multiple users communicating at the same time. Only one burst interval in a UWB PHY symbol will host an active burst. The remaining burst intervals will be empty. The time hopping scheme determines the active burst interval, also known as the hopping position, in the symbol.

The polarity scrambling aids with interference rejection but in addition provides the randomness necessary to smooth the spectrum of the transmitted waveform – a side effect of transmitting impulse trains (bursts). For a review, see the discussion on pulse trains in section 3.4.1.

A single Pseudo Random Binary Sequence (PRBS) generator will be used to generate both the spreader sequence and the burst hopping sequence.

The generator polynomial used by the PRBS generator is chosen such that it is a primitive polynomial and maximum length sequence (MLS) and is given by equation 4-13.

$$g(D) = 1 + D^{14} + D^{15} \tag{4-13}$$

where D is a delay of one chip interval ($T_c \approx 2ns$).

An MLS is generated using a linear feedback shift register (LFSR) of which the Fibonacci LFSR and Galois LFSR implementations are the most common. The 802.15.4a standard specifies a Fibonacci implementation clocked at the peak PRF of 499.2 MHz. The output of this LFSR gives the scrambler output as

$$s_n = s_{n-14} \oplus s_{n-14} \qquad \text{(4-14)}$$
$$with\, n = 0,1,2 \ldots$$

For symbol k, the LFSR is clocked a number of times equal to the number of chips per burst (N_{cpb}). The resulting output becomes the scrambling code for symbol k.

To calculate the burst hopping position for symbol k the standard presents the following three formulas for the three different mean PRFs:

PRF_{mean} = 3.90 MHz	
$h^k = s_{kN_{cpb}} + 2s_{1+kN_{cpb}} + 4s_{2+kN_{cpb}} + 8s_{3+kN_{cpb}} + 16s_{4+kN_{cpb}}$	(4-15)
PRF_{mean} = 15.60 MHz	
$h^k = s_{kN_{cpb}} + 2s_{1+kN_{cpb}} + 4s_{2+kN_{cpb}}$	(4-16)
PRF_{mean} = 62.4 MHz	
$h^k = s_{kN_{cpb}}$	(4-17)

4.5 CURRENTLY AVAILABLE 802.15.4A HARDWARE

It was mentioned in the introduction that it is getting a little easier to get hold of UWB hardware but the available UWB integrated circuits (ICs) out there is still very expensive. We will briefly look at two ICs implementing the IEEE 802.15.4a standard for UWB communications.

4.5.1 IMECs Digital UWB Transmitter IC

IMEC [60] is a world-leading independent research center in the fields of nanoelectronics, bioelectronics and nanotechnology. Its research focuses on future generation chips and state-of-the-art technologies for ambient intelligence.

The IMEC transmitter IC was designed according to the signal structure of the 802.15.4a standard. To reduce startup time and save on energy, the IC is equipped with a phase-aligned FLL (Frequency Locked Loop) instead of the traditional Phase Locked Loop (PLL).

4.5.2 TES IEEE 802.15.4a transceiver with ranging capability

TES Electronic Solutions specializes in custom electronic design and manufacturing services [61].

The TES IC implements the complete LR-UWB functionality specified by the IEEE 802.15.4a standard.

Typical features of the IC solution include:

- Highly integrated, low-power transceiver IP core
- IEEE 802.15.4a conformant PHY
- IEEE 802.15.4 MAC
- Localization and tracking algorithms
- Advanced Encryption Standard (AES) security engine
- Embedded Flash/RAM and signal conditioning
- Omni-directional monopole, small footprint dipole and directional UWB antennas

Chapter 5

The OMNET++ Simulation Environment

In this chapter we will take a look at different network simulators for wireless networks. After a comparison we will look at the reasons for choosing the OMNET++ simulation environment and provide an introduction into the general concepts of OMNET++.

5.1 NETWORK SIMULATORS

5.1.1 Available network simulators

A wide variety of network simulators, all with different levels of complexity and flexibility are available to set up network test scenarios and simulate the network behavior. Table 5-1 lists the most common network simulators as identified by Andrea Rizzoli [62].

Table 5-1: List of available network simlators

Simulator Name	Description
OMNET++	OMNeT++ is a component-based, modular and open-architecture simulation environment with strong GUI support and an embeddable simulation kernel. The simulator can be used for modeling: communication protocols, computer networks and traffic modeling, multi-processors and distributed systems, etc. OMNeT++ also supports animation and interactive execution. OMNeT++ is freely distributed under an academic public license [2].
NS-2	Network Simulator v2 (NS-2) is the second version of a discrete event simulator targeted at networking research. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. NS is developed by the Information Sciences Institute at the University of Southern

	California (USC) school of engineering. NS-2 is free software covered by the GNU GPLv2 license and is publicly available for research, development, and use [1].
NS-3	NS-3 is the next major revision of the NS-2 simulator. NS-3 is presently (April 2008) a work in progress, before even an initial alpha release. The current source code is available for experimentation [63].
GloMoSim	A wireless and wired network simulator providing a scalable simulation environment. GloMoSim is designed using parallel discrete-event simulation capability provided by the Parsec C-based simulation language. GloMoSim is only freely available to academic institutions for research purposes. The commercial version of GloMoSim is QualNet [64].
SWANS	Scalable Wireless Ad hoc Network Simulator (SWANS) is a scalable wireless network simulator built atop the Java in Simulation Time (JiST) platform. It was created primarily to provide for additional research requirements existing network simulation tools do not cater for. The SWANS is organized as independent software components that can be composed to form complete wireless network or sensor network configurations. Its capabilities are similar to NS-2 and GloMoSim, but it is able to simulate much larger networks. SWANS leverages the JiST design to achieve high simulation throughput, save memory, and run standard Java network applications over simulated networks. In addition, SWANS implements a data structure, called hierarchical binning, for efficient computation of signal propagation [65].
QualNet	A commercial network simulation suite, providing modeling software that predicts performance of networks through simulation and emulation. QualNet enable the deployment of a plethora of applications in wireless, wired and mixed network platforms. QualNet claim to have the fastest real-time traffic modeler and focuses on

	speed, accuracy, portability and extensibility to successfully deploy virtual networks [66].
OPNET	A leading commercial network simulator, providing solutions for managing network operations, network R&D, capacity management and application performance management. OPNET provides customers the capabilities to design, deploy and manage network infrastructure, the network equipment and network applications [67].
TOSSIM	A Discrete Event Simulation platform for TinyOS sensor networks. No physical hardware motes running the TinyOS are necessary. It is primarily targeted for the simulation of TinyOS applications and therefore focuses on simulating the TinyOS environment rather than the real world. Even so, TOSSIM is very useful in cause-effect analysis of sensor networks. It allows a user to separate out the environment noise to gain better understanding of the implemented algorithms [68]. Although an excellent simulator, TOSSIM was not chosen because at the time of this writing the OMNET++ community support and tool availability far exceeds that of TOSSIM. No radio propagation models are provided and one has to rely on external programs to accomplish this. Furthermore, the learning curve is steeper because one has to learn the TinyOS way of implementing simulation tasks. The biggest benefit of the TOSSIM simulator is that the applications written for it can be run on real modes without any modification. For this reason it is considered as an excellent future work contribution for the work presented in this paper.

5.1.2 OMNET++ vs NS-2

The two most popular network simulators, especially for research purposes, are OMNET++ and NS-2 (Note: TOSSIM is also fast gaining acceptance but was not considered for the reasons mentioned earlier). In choosing a network simulator to use for this project, the following features were considered important:

- Ease of installation and multiple platform support
- Programming model
- Flexibility
- Model management and structure
- Existing UWB frameworks
- Debugging ability
- Documentation and support
- Logging of results
- Visualization

Table 5-2: OMNET++ vs NS-2 - Comparing important features

Ease of installation and multiple platform support	
OMNET++	Supports both Windows and Linux platforms. Windows installation: Very easy. Linux installation: Moderately easy.
NS-2	Supports Windows (using Cygwin) and Linux platforms. Windows\Cygwin installation: Not tested, but should be similar to Linux installation. Linux installation: Moderate to difficult. NS depends on several externally available components that need to be installed.
Programming Model	
OMNET++	Object-oriented, event-driven and written in C++. Topology descriptions are either written in the NED language which is very basic and easy to learn, or can be dynamically created in run-time. The newly released version 4.0 of the simulation software boasts with a very intuitive IDE providing a graphical interface for various

	tasks such as creating and editing topologies, configurations and network descriptions.
NS-2	Mixed-mode programming. Use OTcl (Object-oriented Tcl) scripting language with underlying C++ classes. OTcl is also used for creating and configuring networks.
Flexibility	
OMNET++	OMNeT++ is a flexible and generic simulation framework. The basic active components of the model are called simple modules and every simple module lives on its own and exchanges messages with others to communicate. One can simulate anything that can be mapped to active components that communicate by passing messages.
NS-2	NS-2 has been designed as a (TCP/IP) network simulator, and it is difficult to impossible to simulate things other than packet-switching networks and protocols with it. NS-2 concepts are highly detailed and deeply hardcoded making it very difficult to do things differently.
Model management and structure	
OMNET++	Models in OMNeT++ are independent of the simulation kernel because the OMNeT++ simulation kernel is a class library. Simple modules are reusable and can be combined like blocks to create compound modules and simulations. A complex module can be assembled from these self-contained building blocks which can be reused in different simulations. By organizing modules in this way, complexity is dealt with in a hierarchic fashion.
NS-2	In NS-2 the boundary between simulation core and models is blurred, without a clear API. Installation involves a lot of patching. Furthermore, in NS-2 all models are "flat" and therefore does not allow for subnetworks, or complex protocols to be implemented as a composition of several independent units.

Existing UWB frameworks	
OMNET++	OMNeT++ has a good variety of models for simulating computer systems, queuing systems etc., but lags behind the ns-2 simulator on availability of communication protocol models. OMNET++ has the Mobility framework (recently superseded by the Mixim framework) and the INET framework, both which include support for mobile and wireless simulations. The Mixim framework combines various wireless and mobile framework features and provides for simulation up to signal level. The Mixim-UWB variation of the framework follows the 802.15.4a standard and is an ideal candidate.
NS-2	Since NS-2 has been designed as a network protocol simulator, it has a much richer set of communication protocol models than OMNET++. A model exist that implements an impulse radio UWB physical layer with time-hopping, pulse position modulation and convolutional channel codes. The model also supports a DCC-MAC layer.
Debugging ability	
OMNET++	Simple macro is provided to write debug messages to standard output. Debugging can be enabled or disabled through easy parameter changes. Watches can also be set on variables. The Eclipse IDE included with version 4.0 of the software allows the ability to perform step execution, to set breakpoints and inspect variables and values while actively running a simulation.
NS-2	Have to consider debugging at both OTcl and C++ levels. Care has to be taken to prevent memory leaks.
Documentation and support	
OMNET++	A lot of documentation and online support available with tutorials and demos to ease the learning curve.

NS-2	A lot of documentation and online support available but documentation seems to be fragmented.
Logging of results	
OMNET++	Allows for the capturing of time series data through output vectors and single value statistics through scalar output files. Visualization and analysis tools are also provided to interpret and graph the result data.
NS-2	Available through OTcl and trace files with the network animator (NAM) of NS. NAM is a Tcl/Tk based animation tool for viewing network simulation traces and real world packet tracedata.
Visualization	
OMNET++	Provide an interactive graphical user interface through TkEnv and a command line interface through CmdEnv. TkEnv is a Tcl/Tk based animation tool. OMNET++ allows individual modules to be inspected and viewed while the simulation is running.
NS-2	Provide an interactive graphical user interface through NAM. Like OMNET++ it supports topology layout, packet level animation, and various data inspection tools.

Simulations are very useful in the pursuit of research, especially for cases where physical hardware is not readily available or too expensive and time consuming to implement. Additionally it provides for a flexible environment in which relatively easy experimentation of various scenarios can be performed.

Even so, J.G. Page *et al* [69] states that "*if a simulation is not accurately mimicking reality then the results are meaningless*". Furthermore, modeling of the physical layer is essential in WSN simulations because it affects the performance of the upper layers.

There are several reasons why the research community sometimes questions the credibility of simulation results:

- Covering too little detail. If the simulation model provides too little detail the results will be inaccurate and impractical.

- Covering too much detail. If the simulation model provides too much detail the performance of the simulation might suffer and the model will be difficult to reuse.

- Difficult or near impossible to repeat results. Study has shown that most of the existing research provides very little simulation model information and almost no support documentation, making it hard for anyone to reproduce the results published.

After considering the above comparison and taking into account the issues that surrounds network simulations, OMNET++ version 4.0 was chosen for implementing the simulation of the 802.15.4a standard. More information on the alternative NS-2 UWB MAC and PHY simulator can be found at [3].

OMNET++ is much easier to work with than NS-2, provides a far better model architecture with its hierarchical structure, and has a lot more potential in scope and features. The most useful feature is the availability of an ultra wideband framework based on the IEEE 802.15.4a standard through the Mixim framework.

The Mixim framework together with OMNET++ version 4.0 addresses a lot of the concerns associated with networks simulations. OMNET++ and the Mixim framework is very easy to install, works on both the Windows and Linux platforms, includes proper documentation backed up by an online community. Results can be repeated exactly by using the same set of random generator seeds used in the original experiment runs. The Mixim framework models the physical layer up to the signal level for the most accurate results.

Next we will now look into the basic modeling concepts of discrete-event simulation and introduce the rudiments of OMNET++.

5.2 SIMULATION MODELING CONCEPTS

5.2.1 Discrete Event Simulation

A discrete-event simulation is one in which the state of a model changes only at discrete, but possibly random, instances in time.

An event is an occurrence that changes the state of the model. In a DES events take zero time to happen and it is assumed that no state changes in the model takes place between events. The time when an event occurs is called an event timestamp and the time within the model is referred to as simulation time. Real time (or wall-clock time) refers to how long the simulation is actually running.

The opposite of discrete event simulation models are continuous simulation models where states change all the time. Numerous systems like manufacturing, transportation, health care, communication, defenses, information processing, and queuing systems can be modeled using discrete event simulation.

An entity can be seen as a traffic unit that moves from one point to another in the model. An entity can be in one of five states [70]:

1. Active state: The active state is the state in which the currently moving entity is.
2. Ready state: The ready state is a state for entities waiting to enter the active state (i.e. there is more than one entity ready to move but they have to do it one at a time).
3. Time-delayed state: The time-delayed state is a state in which an entity is waiting for a known future simulated time to be reached before entering the Ready state.
4. Condition-delayed state: The condition-delayed state is a state in which an entity is waiting for a certain condition to be satisfied before entering the Ready state.
5. Dormant state: The dormant state represents a sort of hold or sleep state. When an entity is in the Dormant state, changes in the model conditions cannot trigger a state transfer. Modeler logic is required to transfer an entity from the Dormant to the Ready state.

A DES provides a set of lists for each of the five states to organize and track entities. We will quickly look at the Future Events List (FEL) used for entities in the Time-delayed state.

5.2.2 The event loop

The Future Events List is also known as the Future Event Set (FES) and contains the set of future events, i.e. entities with a time-based delay are inserted into the FES. Figure 5-1 illustrates the basic steps a DES would typically take to implement the event loop. To ensure causality (that no event effect earlier events), all events are processed in a strict timestamp order.

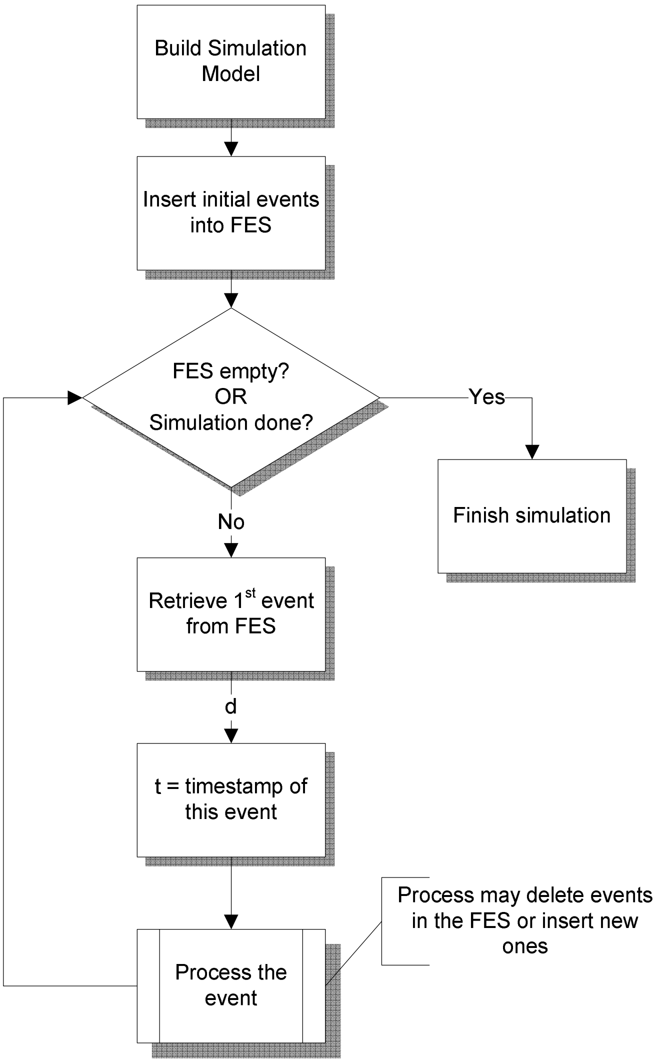


Figure 5-1: Flowchart illustrating typical event loop

5.3 OMNET++ INTRODUCTION

OMNeT++ is an object-oriented modular discrete event simulation environment. Its primary application area is the simulation of communication networks, but it can be used to simulate almost any discrete event system [71].

OMNeT++ provides a component based architecture for models. The components of an OMNET++ model are modules programmed in C++. Using a high-level language called NED, the modules are hierarchically assembled into larger compound modules and models that can be reused. OMNET++ also has strong GUI support and an embeddable simulation kernel.

Modules communicate by passing messages that may contain complex data structures to each other. Messages can be sent directly or along a predefined path to their destination.

Modules can have their own parameters to customize module behavior. At the lowest level of the module hierarchy we have simple modules implemented in C++ which encapsulate the behavior.

5.3.1 OMNET++ model structure

Figure 5-2 shows the OMNET++ model structure with its hierarchically nested modules [72].

The top level module is the system module. The system module contains submodules, which can also contain further submodules with no limit on the depth of module nesting.

As mentioned, at the lowest level of the module hierarchy we have the simple modules containing the model algorithms implemented by the user. Modules that contain submodules are termed compound modules.

OMNeT++ models are also known as networks. In a given network, simple and compound modules are instances of user defined module types. When describing the network, these module types are used to define more complex module types. So the system module is also an instance of a previously defined module type with all other network modules as submodule or sub-submodule instances of the system module.

Therefore the OMNET++ module structure allows the user to reflect the logical structure of the actual system.

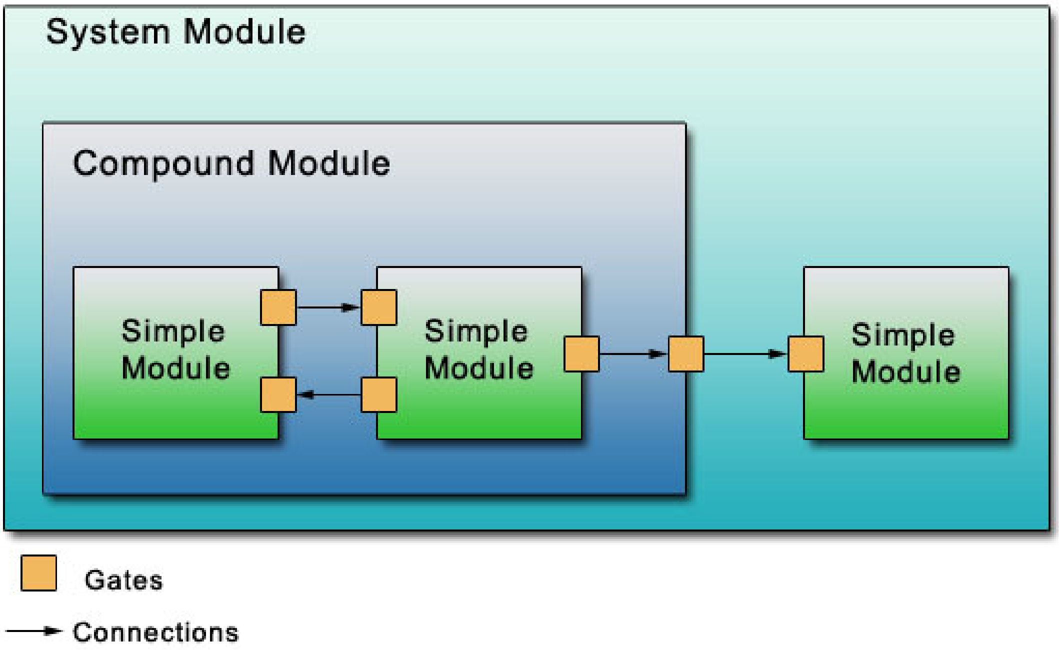


Figure 5-2: OMNET++ model structure

5.4 THE NED LANGUAGE

NED (NEtwork Description) is a simple yet powerful topology description language employed by OMNET++. NED facilitates the network description process by allowing different component descriptions for simple modules, compound modules and channels. These component descriptions can be reused in another network description. A NED description can contain the following components:

- Import directives
- Channel definitions
- Simple module definitions
- Compound module definitions
- Network definitions

Without going in too much detail, let us take a brief look at each of these components.

5.4.1 Import directives

The `import` directive is used to import component descriptions from another network description file. After importing a network description the simple modules, compound modules and channels defined in it can be used. A NED file has a `.ned` extension.

5.4.2 Channel definitions

Different connection types can be defined by subclassing from the predefined channel types. OMNET++ provides three predefined channel types and the `extends` keyword to indicate inheritance. Two of the three predefined channel types also provide parameters to specify common channel attributes as discussed below.

IdealChannel

Lets messages pass through without any side effects or delay.

Parameters: (None)

DelayChannel

Introduces a propagation delay.

Parameters:

- `delay`: Propagation delay in simulated seconds
- `disabled`: If `true`, the channel object will drop all messages

DataRateChannel

Adds to the parameters of a `DelayChannel` to allow for data transfer rates and basic error rates to be taken into account.

- `ber` and `per`: The bit error rate (BER) and packet error rate (PER) with range [0,1]
- `datarate`: Channel bandwidth in bits per second (bps) or its multiples

The values for these attributes are specified by declaring a connection type with a channel definition inheriting the required predefined channel type. The newly specified channel name can then be used to create connections with the attribute values assigned for that channel type.

The syntax for a channel definition is:

```
channel ChannelName

{

//Optional declaration of attributes can be done here

}
```

Or to make use of the predefined channel type parameters:

```
channel ChannelName extends ned.DataRateChannel

{

   datarate = 100Mbps;

   delay = 100us;

   ber = 1e-10;

}
```

5.4.3 Simple module definitions

Simple modules are the basic building blocks for compound modules. A simple module's definition consists of parameter and gate declarations.

The syntax for a simple module definition is:

```
simple SimpleModuleName

{

   parameters:

   //Define the simple module variables here

   gates:

   //Define in and out gates here
```

```
}
```

5.4.4 Compound module definitions

Compound modules consist of one or more submodules. These submodules can be simple modules or other compound modules. Compound modules can be seen as groupings that allow the simulation model to be organized and structured, i.e. compound modules have no active behavior. Like simple module definitions, compound module definitions also have gates and parameter sections but they also have two additional sections - submodules and connections. All compound module sections are optional.

The syntax for a compound module definition is:

```
module CompoundModule

{

types:

//Define any channel and module types used locally by
//the compound module

parameters:

//Define the compound module variables here

gates:

//Define in and out gates here

submodules:

//Define the submodules this module is composed of
//here

connections:

//Define the connections between all the module gates
//here

}
```

Submodules are identified by names. The syntax to declare submodules is:

```
module CompoundModule

{

//...

submodules:

   submodule1: ModuleType1 {

         parameters:

         //Assign  values  to  the  parameters  of  this
         //submodule ModuleType1

         gatesizes:

         //Specify  the  sizes  for  any  gate  vectors  of
         //submodule ModuleType1

   }

   submodule2: ModuleType2 {

         parameters:

         //Assign  values  to  the  parameters  of  this
         //submodule ModuleType2

         gatesizes:

         //Specify  the  sizes  for  any  gate  vectors  of
         //submodule ModuleType2

   }

//...

}
```

The connections section specifies how the gates of the compound module and the gates of its submodules are connected. The syntax to specify connections is:

```
module CompoundModule

{

   //...

   connections:
```

```
   // Connect output gate with -->

   node1.output --> node2.input;

   // Connect input gate with <--

   node1.input <-- node2.output;

   // Connect inout gate with <-->

   node1.inout <--> node2.inout;

}
```

5.4.5 Network definitions

A network definition is needed to actually get a simulation model that can be run. Module declarations only define module types. To declare a network simulation model a previously defined compound module type is instantiated.

The syntax of a network definition is:

```
network networkName extends CompoundModuleType1

{

   parameters:

   // Example parameters

   param1 = 10;

   param2 = true;

   param3 = truncnormal(100,60);

}
```

5.5 SIMPLE MODULES

Simple modules are the active elements in a model with events occurring inside them. Simple module types are programmed in C++ by subclassing the `cSimpleModule` class of the OMNET++ class library. The `cSimpleModule` class has some virtual member functions that need to be redefined by the user to implement the required model behavior.

5.5.1 Handling events

Whenever the simulation kernel receives a message, the `handleMessage()` method gets called. The `handleMessage()` method acts as an event-handler and is one of the virtual functions of the `cSimpleModule` class that needs to be redefined with message processing code. It will get called for every message that arrives at the module. No simulation time elapses within a call to `handleMessage()`.

5.5.2 Passing messages

Simple modules mostly keep themselves busy with sending and receiving messages. Messages can be sent via output gates or they can be sent directly to another module. Sometimes we need to model events that occur within the same module. Consider the case where we want to implement timers or schedule events that need to happen a certain time in the future. OMNET++ caters for this by allowing a module to send a message to itself. Such a message is known as a self message.

When a message is received by a module the simulation kernel registers an event and calls the `handleMessage()` function of that module to process the message.

The OMNET++ simulation library provides the functions listed in Table 5-3 to facilitate the transmission of messages between modules.

Table 5-3: Send functions provided by OMNET++

Function	**Description**
`send(..)`	To send a message through an output gate
`sendDelayed(..)`	To send a message through an output gate after some delay (to simulate processing time etc.)
`sendDirect(..)`	To ignore any gates or connections and send a message directly to a remote destination module
`scheduleAt(..)`	To send a self message

5.6 COMPOUND MODULES

Compound modules are made up of submodules which are other compound modules or simple modules. Compound modules do not have any active behavior. They transparently relay messages, acting like black boxes. Refer to section 5.4.4 for more detail.

5.7 GATES AND CONNECTIONS

Modules are connected through points called gates. Gates can be of type `input`, `output` or `inout`. OMNET++ only supports simplex (one-way) communications for its modules. Hence, for a module to send and receive it has to define an `inout` gate type for itself or both an `input` and `output` gate type.

Gates are declared by listing their names, which is also used to identify the gate, in the gate section of a module description. OMNET++ allows gate vectors to be declared. A gate vector contains a number of single gates.

A connection is made between a starting point (source module output gate) and an ending point (destination module input gate).

5.8 MESSAGES

In OMNET++ messages are used to represent events. The `cMessage` class incorporates the message functionality in OMNET++. `cMessage` objects and classes derived from `cMessage` may model a lot of different things like events, messages, packets, frames, bits etc. `cMessage` also provides a number of message object attributes for greater programming convenience.

5.8.1 Simulating packets

When simulating telecommunication networks in OMNET++, protocol layers are usually implemented as modules that exchange packets. The packets are message objects instantiated from a `cMessage` derived class. To send additional information OMNET++ allows control info objects subclassed from `cPolymorphic` to be attached.

Another essential when modeling layered protocols is encapsulation and decapsulation of packets. OMNET++ provides an `encapsulate()` function to encapsulate a message into another one. To get the encapsulated message back the `decapsulate()` function can be used.

Message definitions, discussed next, allows one to add parameters or objects to a message.

5.8.2 Message definitions

To add parameters and objects to a message class involves a lot of coding. Message definitions are a more convenient way to describe message contents. The OMNET++ message definition syntax is very compact and allows for C++ code to be automatically generated from the message definition. Message definitions files have a `.msg` extension.

An example of a message definition is given below:

```
message MyPacket
{
    int srcAddress;
    int destAddress;
    int hops = 32;
}
```

A `MyPacket` message class with setter and getter methods for each of the declared fields will automatically be generated by OMNET++ when the message definition file gets compiled.

5.9 SUMMARY

Figure 5-3 gives a quick overview of the steps involved to create an OMNET++ simulation model.

Refer to the OMNET++ user manual [71] for detailed information about OMNET++.

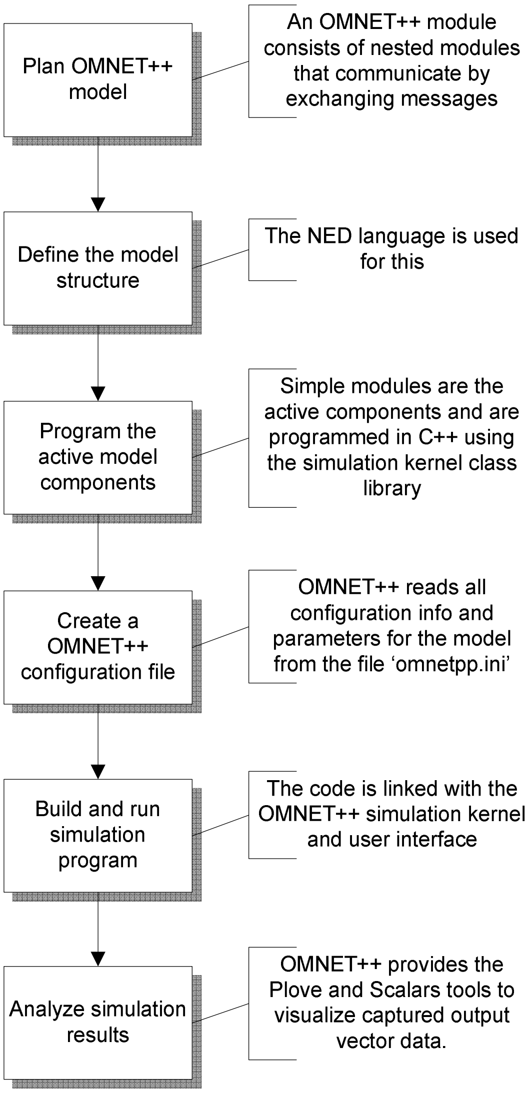


Figure 5-3: Basic steps to create an OMNET++ simulation model

Chapter 6

Simulating a 802.15.4a Ultra Wideband Physical WSN

This chapter will explain how the 802.15.4a UWB PHY is simulated in OMNET++. It will describe the simulation goals of this project, identify the focus areas in the 802.15.4a standard required to meet these goals and finally discuss the application of the UWB model in a WSN simulation.

6.1 SIMULATION GOALS

The project aims to achieve the following goals to appropriate the analysis of the IEEE 802.15.4a UWB physical layer:

1. Simulate the 802.15.4a UWB PHY as closely and accurately possible to ensure credible simulation results.
2. Apply the UWB PHY simulation model to a wireless sensor network.
3. Compare the distance, throughput and bit-error-rate (BER) for the various channels and timing parameters using several channel models.
4. Analyze the impact of the Reed-Solomon coder on the distance, throughput and BER.
5. Analyze the impact of the convolutional encoder and Viterbi decoder on the distance, throughput and BER.
6. Analyze the combined impact of both forward error correction schemes on the distance, throughput and BER.
7. With the various results consider a mechanism allowing for the adaptive change of channel, timing parameters and forward error correction to guarantee the best BER for a given minimum throughput or minimum distance.

6.2 OMNET++ 802.15.4A UWB PHY SIMULATION MODEL

For reasons mentioned in chapter 5, the Mixim-UWB framework for OMNET++ version 4.0 was chosen for simulation purposes.

The Mixim framework combines several existing frameworks for wireless and mobile simulation in OMNET++ to furnish a rich protocol library and infrastructure allowing accurate modeling of the environment, mobility, connectivity, reception, obstacles and collision [73].

6.2.1 Previous work

Jérôme Rousselot built upon the Mixim framework version 1.0 to develop the Mixim-UWB framework according to the 802.15.4a specification. The Mixim-UWB framework is still under development and the source code can be downloaded from the GitHub code sharing site at [74]. The Mixim-UWB baseline used in this project can be found on the master branch stamped with the date 9 July 2009.

The Mixim-UWB framework provides the following useful components:

- Symbol-level UWB Impulse Radio simulator
- IEEE 802.15.4a transceiver with energy detection based receiver
- Accurate modeling of path-loss and fading
- Relatively easy evaluation of the BER

With regards to the 802.15.4a UWB PHY standard, the Mixim-UWB framework includes:

- UWB signal simulation on a symbol level.
- SHR and Data parts of the UWB PHY frame format.
- Burst Position Modulation.
- Time Hopping.
- Calculation of spreading sequence.
- Energy detector (non-coherent receiver).
- Limited synchronization.

- Impact of Multiple Access Interference (MAI).
- Simple UWB pulse.
- Limited error correction. Caters for Reed-Solomon coder by making sure the number of bit errors at receiver does not exceed the number of bits the R-S code can actually fix.
- Multiple channel models.
- Easy adjustment of receiver sensitivity.
- Mandatory channel of frequency band 1 (channel 3).
- Mean PRF of 15.60MHz and bit rate of 850kbps.
- Preamble with default length sequence only with implementation limited to preamble code with index 5.

6.2.2 Contributions

To be able to achieve the simulation goals the following modifications and additions were made to the Mixim-UWB framework:

- Added capability to configure channel, bit rate, mean PRF and number of preamble symbols with automatic calculation of all UWB PHY rate and timing parameters based on these configuration values.
- Improvements to spreading sequence calculations including support for all length 31 and length 127 preamble codes.
- Moved generation of the UWB symbol from the MAC layer to the PHY layer where FEC encoding is also performed.
- Provided a SECDED implementation.
- Added Binary Phase Shift Keying modulation.
- Modified energy detector to be able to demodulate the polarity of a burst in addition to its position.
- Added Reed-Solomon encoding and decoding.

- Added convolutional encoding and Viterbi decoding.

6.2.3 Channels

The Mixim-UWB framework only implements the mandatory channel of frequency band 1, i.e. channel number 3. A new class called `IEEE802154A_Config` was added to the model allowing real time selection of a different channel number and\or bit rate.

Upon simulation startup the OMNET++ initialization file requires the parameters listed in Table 6-1 to be configured for each node in the wireless network.

Table 6-1: Configurable channel parameters and permissible values

Parameter Name	Description	Permissible values
channelNum	The channel number.	0 to 15
complexChannelNum	(Optional – Default = 1) The complex channel number indicating the index of the length 31 preamble code to use.	1 = 1st preamble code specified for channel 2 = 2nd preamble code specified for channel
meanPRF	The mean PRF.	1 = 3.90MHz 2 = 15.60MHz 3 = 62.40MHz
dataRate	The data rate.	0 = (non-UWB PHY) 1 = 110kbps 2 = 850kbps 3 = 1.70Mbps (for PRF of 3.90MHz) else = 6.81Mbps 4 = 6.81Mbps (for PRF of 3.90MHz) else = 29.24Mbps
nbPreambleSyncSymbols	The number of repetitions of the preamble symbol used in the SHR.	8 or 64

The physical layer of each node maintains a pointer to the `IEEE802154A_Config` instance configured for that node allowing its physical layer to issue a channel change command if requested from the upper layer. The arguments supplied to the channel change function are the same parameters listed in Table 6-1 and hence a new channel and\or new bit rate can be configured through such a call.

Only the PAN coordinator node should be permitted to initiate the broadcast of a channel change command.

6.2.4 UWB PHY rate, timing and preamble parameters

The `IEEE802154A_Config` also contains all the UWB PHY rate, timing and preamble parameter values as class members. Most of these values are automatically calculated when the `IEEE802154A_Config` object is initialized or reconfigured. The parameters that are not calculated make use of constants and lookup tables to obtain their values.

All of these parameters are easily accessible at each node's physical layer through the pointer it maintains to the configuration object.

6.2.5 UWB simulation model process flow

An outline of the processes the UWB simulation model performs are illustrated in Figure 6-1 for the transmitter and Figure 6-2 for the receiver. Although a lot of complexity sits behind these processes, the illustrations provides a logical overview of the steps involved at the lower layers to transmit a data message from one node to another.

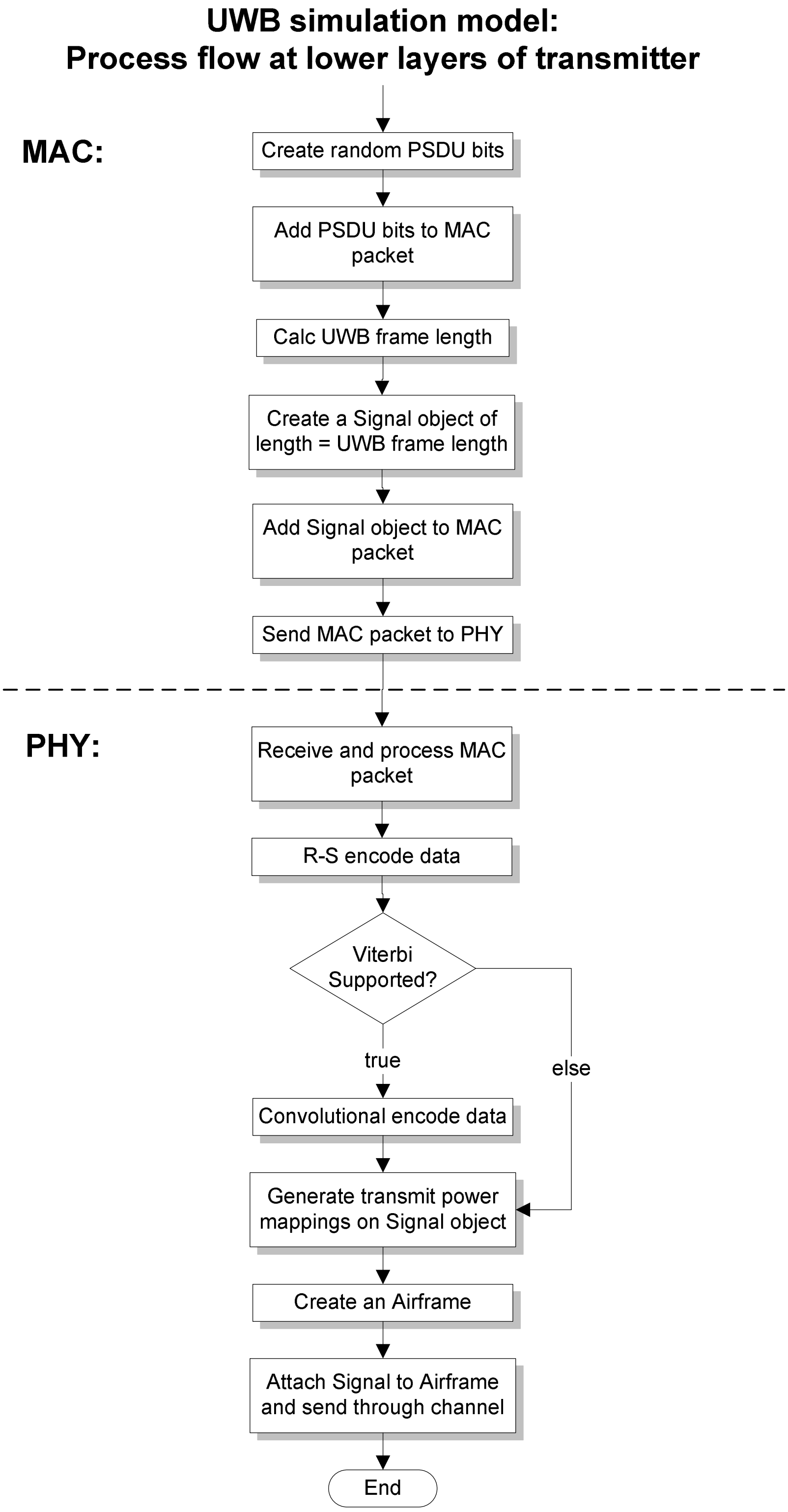


Figure 6-1: Transmitter: UWB simulation model process flow

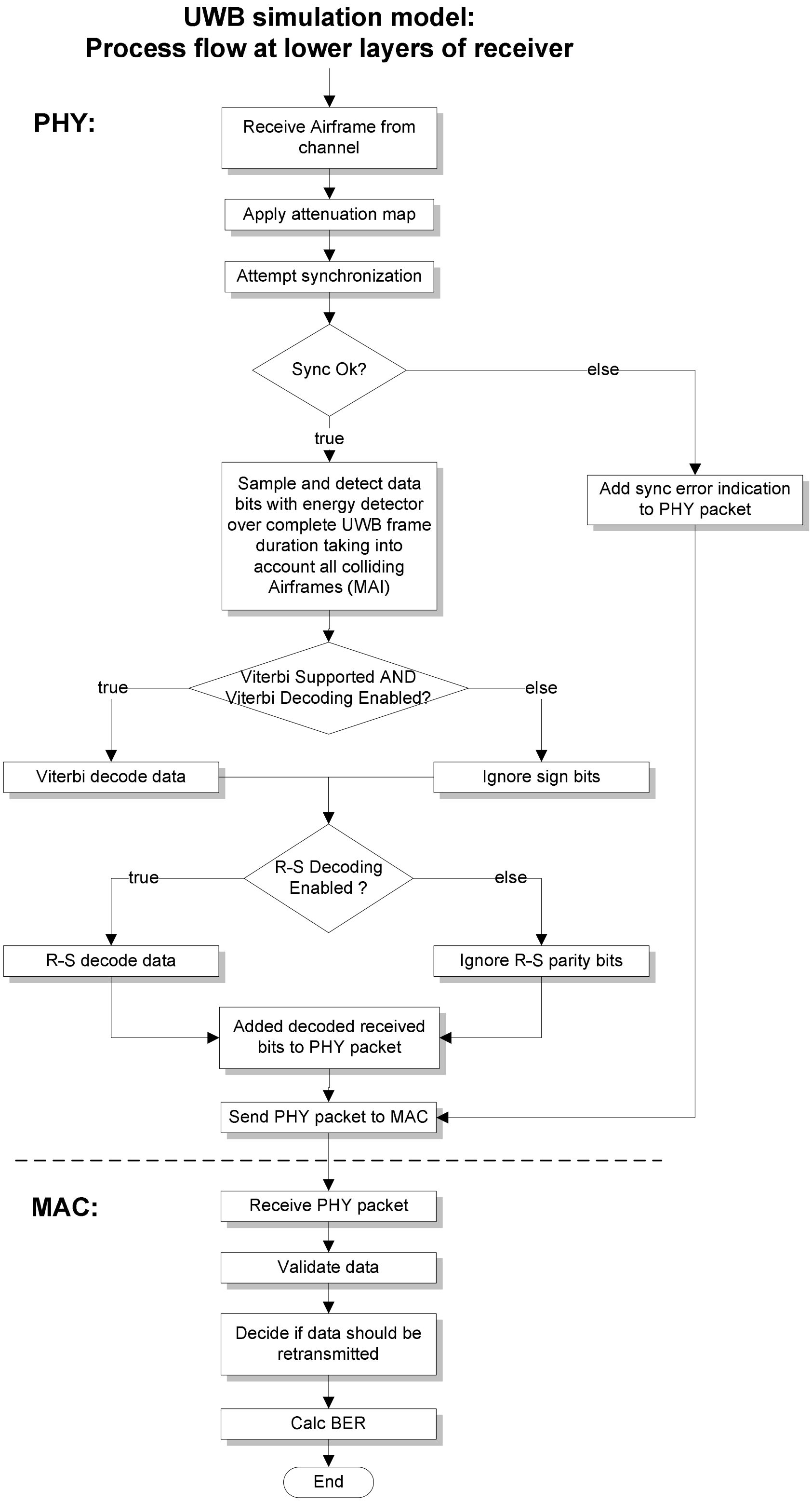
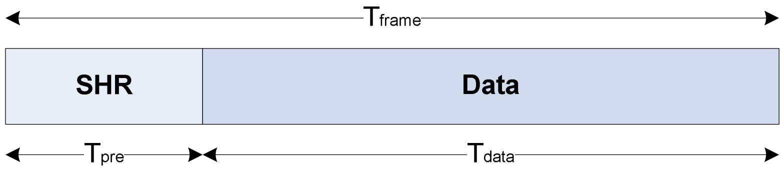


Figure 6-2: Receiver: UWB simulation model process flow

6.2.6 UWB frame format

The frame format was kept the same – that is, only the SHR and Data parts are generated as shown in Figure 6-3.



SHR consists of a repetition of preamble symbols of length T_{psym}

Data consists of a repetition of UWB PHY symbols of length T_{dsym}

Figure 6-3: UWB Simulation: Frame format

A Mixim `Signal` object is used with a time mapping container storing the various transmit power samples of the preamble and UWB PHY symbol repetitions over time. It is exactly this modeling of the low level detail of each symbol that gives the Mixim-UWB framework its novel “symbol-level simulator” characteristic.

Evidently the duration of the `Signal` object, representing the UWB frame, had to be increased to include the addition of error correction bits. It also made more sense to create the `Signal` object at the MAC layer but to generate and populate the transmit power mapping sample values of the symbols at the PHY layer where the error correction is performed. The original Mixim-UWB framework differs in that it creates the `Signal` object and generates the sample values at the MAC layer.

6.2.7 UWB pulse

The pulse employed is a triangular pulse chosen for its simplicity. To store a single triangular pulse only requires three samples values and their corresponding times to be saved in a time mapping container (see Figure 6-4).

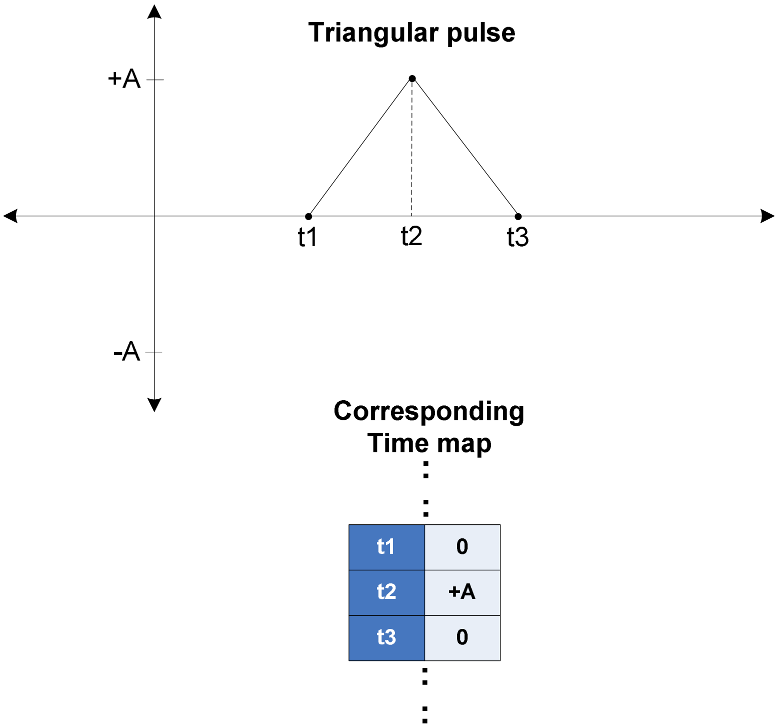
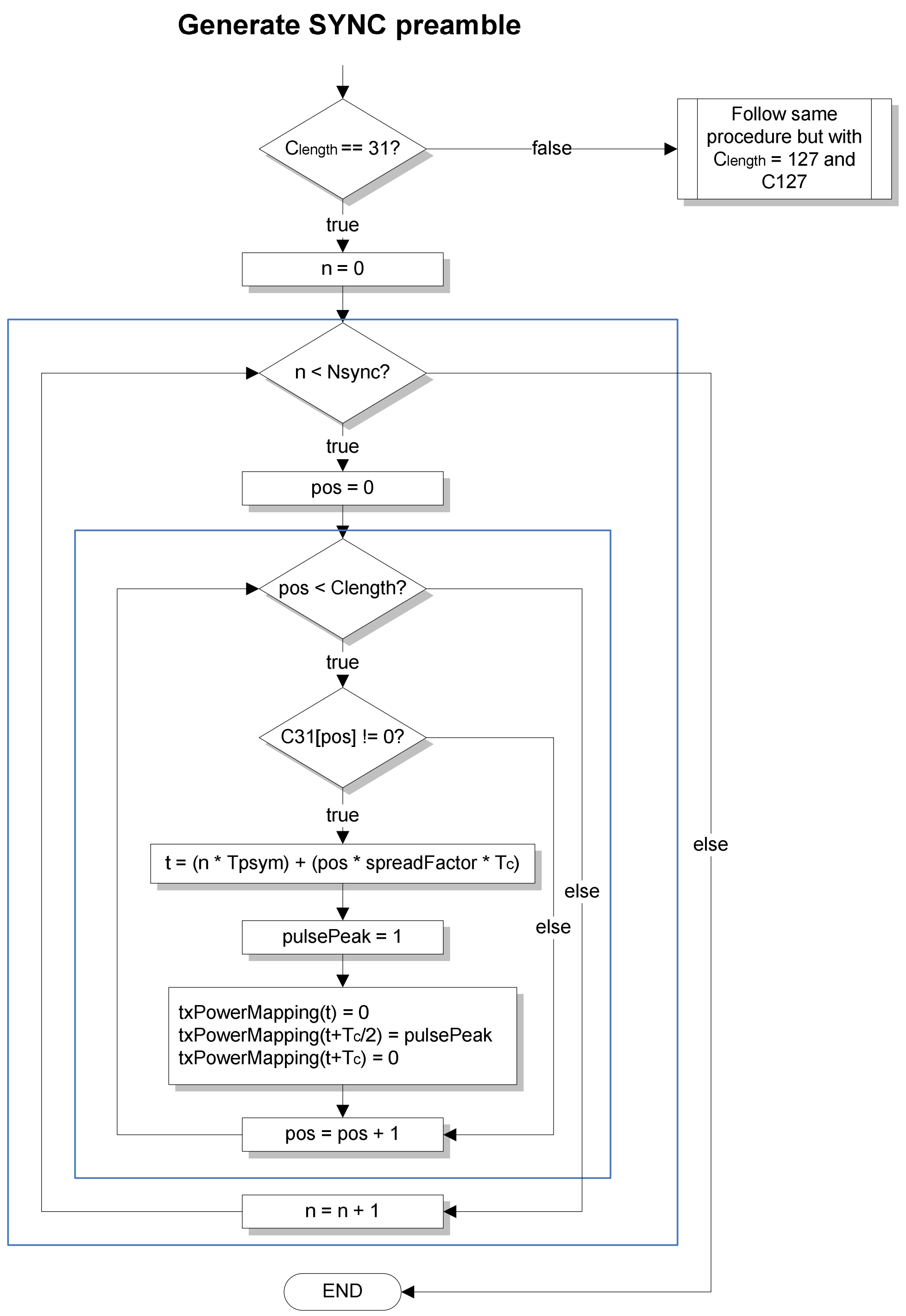


Figure 6-4: Triangular pulse and corresponding time map

6.2.8 SHR preamble generation

Both the SYNC and SFD portions of the SHR are simulated by the Mixim-UWB framework. First the SYNC portion is generated and added to the transmit power mapping of the `Signal` object after which the same is done for the SFD portion. The algorithms used to generate these are shown in Figure 6-5 and Figure 6-6.



N_{sync} = Number of symbols in SYNC portion of SHR

C_{length} = Preamble code length

C31 = Length 31 preamble code for configured channel

T_c = Length of a chip interval (UWB pulse duration)

Figure 6-5: UWB simulation: Generation of SHR SYNC

SYNC preamble algorithm

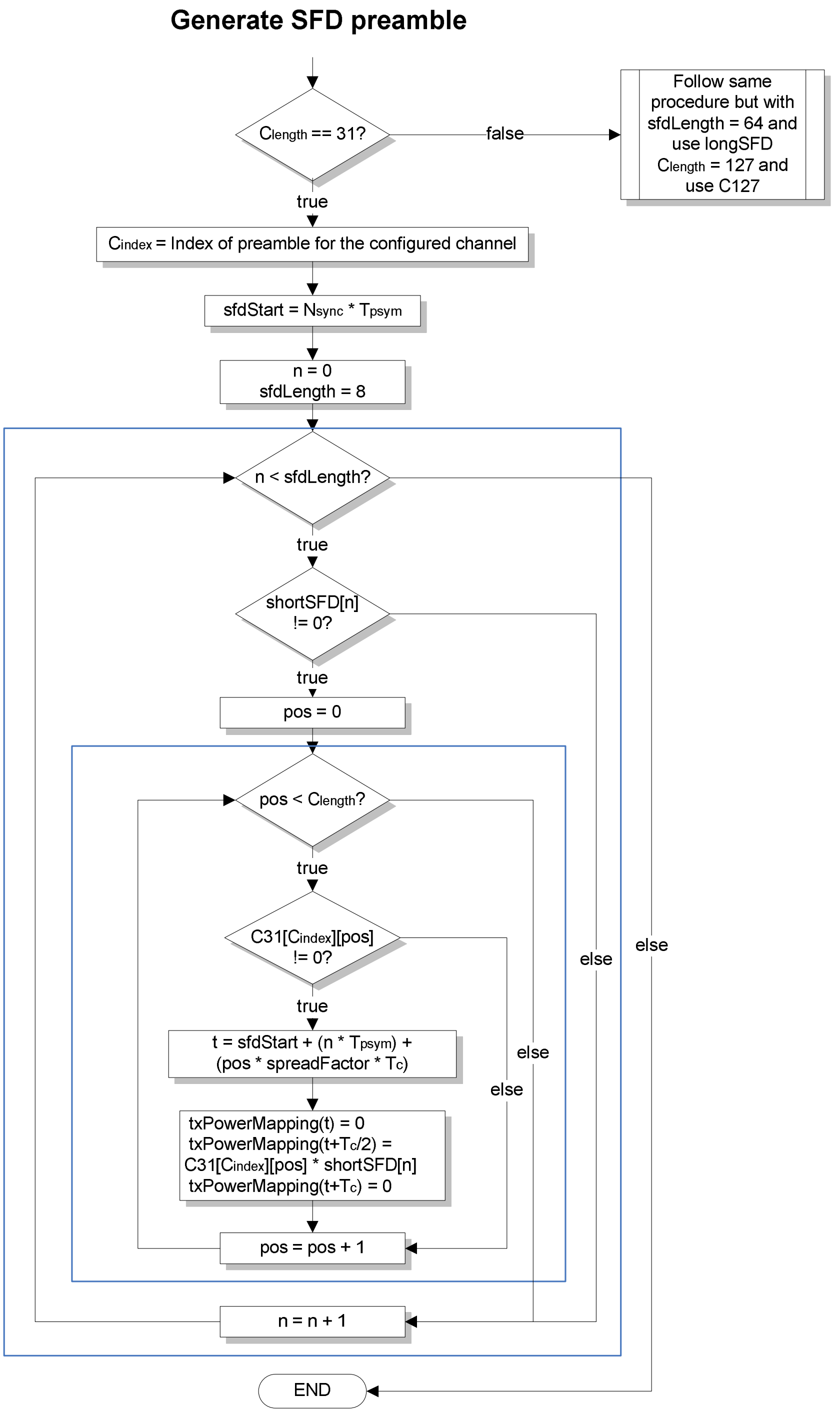
The first step in generating the SYNC preamble is to determine the configured preamble code length (C_{length}). According to the IEEE 802.15.4a standard the preamble code length value can only be 31 or 127. The preamble codes are also defined by the standard and were chosen for their perfect periodic autocorrelation properties.

The next step is to generate a total of N_{sync} preamble symbols, where N_{sync} is the number of symbols in the packet SYNC sequence also defined by the standard.

A single preamble symbol is created as follows:

1. Look up the corresponding preamble code sequence given the code length and channel number.
2. Spread the preamble code sequence using the delta function and Kronecker product as defined in equation 4-11. Zeros are added after every preamble code symbol (from the ternary symbol alphabet {-1, 0, 1}). The number of zeros that gets added determines the spread and is given by the L parameter of the delta function. L-1 zeros are always added.
3. To represent the preamble code sequence symbols, the time of each ternary symbol in relation to the whole UWB frame is calculated and at that point in time the transmit power mapping is set to have a pulse peak of 1 or -1 depending on the value of the ternary symbol (this result in the triangular pulses as illustrated in Figure 6-4).

Therefore, a preamble symbol is the preamble code sequence spread by the delta function of length L with every non-zero preamble code sequence symbol represented by a triangular pulse.



N_{sync} = Number of symbols in SYNC portion of SHR

C_{length} = Preamble code length

C31 = Length 31 preamble code for configured channel

T_c = Length of a chip interval (UWB pulse duration)

T_{psym} = Preamble symbol duration

Figure 6-6: UWB simulation: Generation of SHR SFD

SFD preamble algorithm

The first step in generating the SYNC preamble is to determine if the short or long length SFD should be used. According to the IEEE 802.15.4a standard the short SFD length is 8 and the preferred SFD to use. The longer SFD has length 64 and is optional for use with transmissions at data rate of 110kbps. The actual short and long SFD codes are also defined by the standard.

The next step is to spread the SFD code with the preamble symbol.

The preamble symbol is created in the same way as was done for the SYNC portion. The steps are as follows:

1. Look up the corresponding preamble code sequence given the code length (C_{length}) and channel number.
2. Spread the preamble code sequence using the delta function and Kronecker product.
3. To represent the preamble code sequence symbols, the time of each ternary symbol in relation to the whole UWB frame is calculated and at that point in time the transmit power mapping is set to have a pulse peak of 1 or -1 depending on the value of the ternary symbol (this result in the triangular pulses as illustrated in Figure 6-4).

To spread the SFD code with the preamble symbol, each SFD ternary symbol is multiplied by the preamble symbol.

As an example, the short SFD is defined as [0 +1 0 -1 +1 0 0 -1]. If the preamble symbol is given by pS, then the result after spreading will be $[0\ \ +pS\ \ 0\ \ -pS\ \ +pS\ \ 0\ \ 0\ \ -pS]$.

6.2.9 UWB symbol generation

The same transmit power mapping container is used for the sample values representing the absence or presence of UWB bursts over the time duration of the Data part. The method behind the generation of all these UWB symbols making up the Data part of the frame is shown in Figure 6-7.

UWB data symbol algorithm

At this point the data encoding is already performed and as such there exist both a position bit and sign bit for every data bit.

The first step is to set the time where the data starts in the UWB data symbol. Relative to this time, all the UWB pulse bursts are added.

For every data bit the following takes place:

1. The position of the burst is determined. The varying time hopping code is used to determine the position of the burst taking into account the position bit and allowing for a guard interval.
2. For every chip in the burst (number of chips per burst given by N_{cpb} as defined by the standard) a triangular pulse is generated as illustrated in Figure 6-4.
3. To represent the triangular pulse, the time of every chip in the burst is determined in relation to the start of the data in the UWB symbol. At that point in time the transmit power mapping is set to have a pulse peak of 1 or -1 depending on the sign bit value.

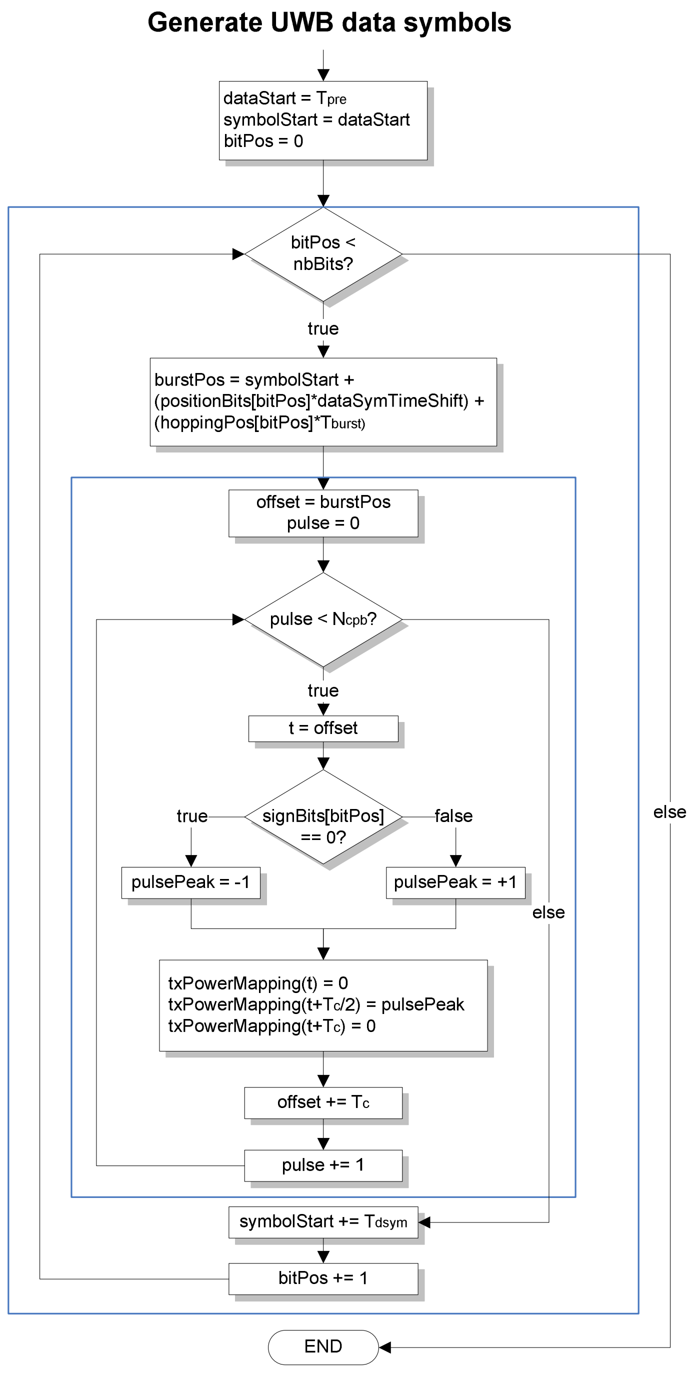
6.2.10 Data bits generation

The simulation model utilizes a uniform random number generator with seed values that can easily be varied for different experiment runs. The MAC layer receives the required number of bytes to transmit from the upper layers and automatically generates a uniform random bit stream representing the message data.

6.2.11 PHR

The PHR part is not yet included in the simulation model. It consists of 19 bits containing its own parity through the SECDED bits making it very robust against transmission errors.

Without too much difficulty the PHR part can be generated by future contributors. More work will have to be done at the receiver though, to process the PHR bits and utilize the PHR information effectively to aid with reception and synchronization of any new incoming frames. A structure for the PHR and a SECDED implementation was added to the model to assist any research requiring the PHR.



N_{cpb} = Number of chips per burst

nbBits = Total number of data bits to create symbols for

T_{pre} = Preamble symbol duration

T_c = Length of a chip interval (UWB pulse duration)

T_{burst} = Total duration of a burst

dataSymTimeShift = Duration specifying position in 1st or 2nd BPM interval

T_{dsym} = Duration of one UWB data symbol

Figure 6-7: UWB simulation: Generation of SHR SYNC

6.2.12 Reed-Solomon encoding and decoding

To be able to perform R-S encoding and decoding a new class called `IEEE802154A_ReedSolomon` was added to the model. This class employs the C++ implementation of the Schifra Reed-Solomon error correction library maintained by Arash Partow [75].

The Schifra library is free to use for academic purposes under the GNU General Public License. The Schifra library is completely configurable allowing for the primitive polynomial and finite field of the 802.15.4a standard to be set up. The library features errors and erasures, different symbol sizes, variable code block lengths and an optimized architecture for various hardware platforms.

A few new classes had to be added under the `bitio` namespace. These allow the data bitstream to be converted to 6-bit symbols and back. The 6-bit symbol size is defined by the R-S code Galois field, GF(2^6), specified by the 802.15.4a UWB standard.

Every 330 bits of data gets an additional 48 parity bits after R-S encoding. If the data is less than 330 bits, zero bits are added to the front until the block size is 330 bits. R-S encoding is then performed and the zero bits are removed before transmission. In such a case the receiver will follow the same procedure by adding zero bits before R-S decoding the block.

6.2.13 Convolutional encoding and Viterbi decoding

The algorithms used for the convolutional encoder and Viterbi decoder were written by Chip Fleming and the source together with an excellent tutorial on Viterbi decoding can be found online at [76]. Through personal correspondence, Chip Fleming granted permission to use his convolutional encoder and Viterbi decoder implementations with the prerequisite that it may only be used for academic and research purposes.

The original implementation is in C, which posed a problem if it were to be integrated in the modular OMNET++ environment. Hence the C implementation was ported to C++ classes to provide configurability and to facilitate the integration of the convolutional encoder and Viterbi decoder in the Mixim-UWB model. The class diagram of the C++ implementation is made available in Figure 6-8.

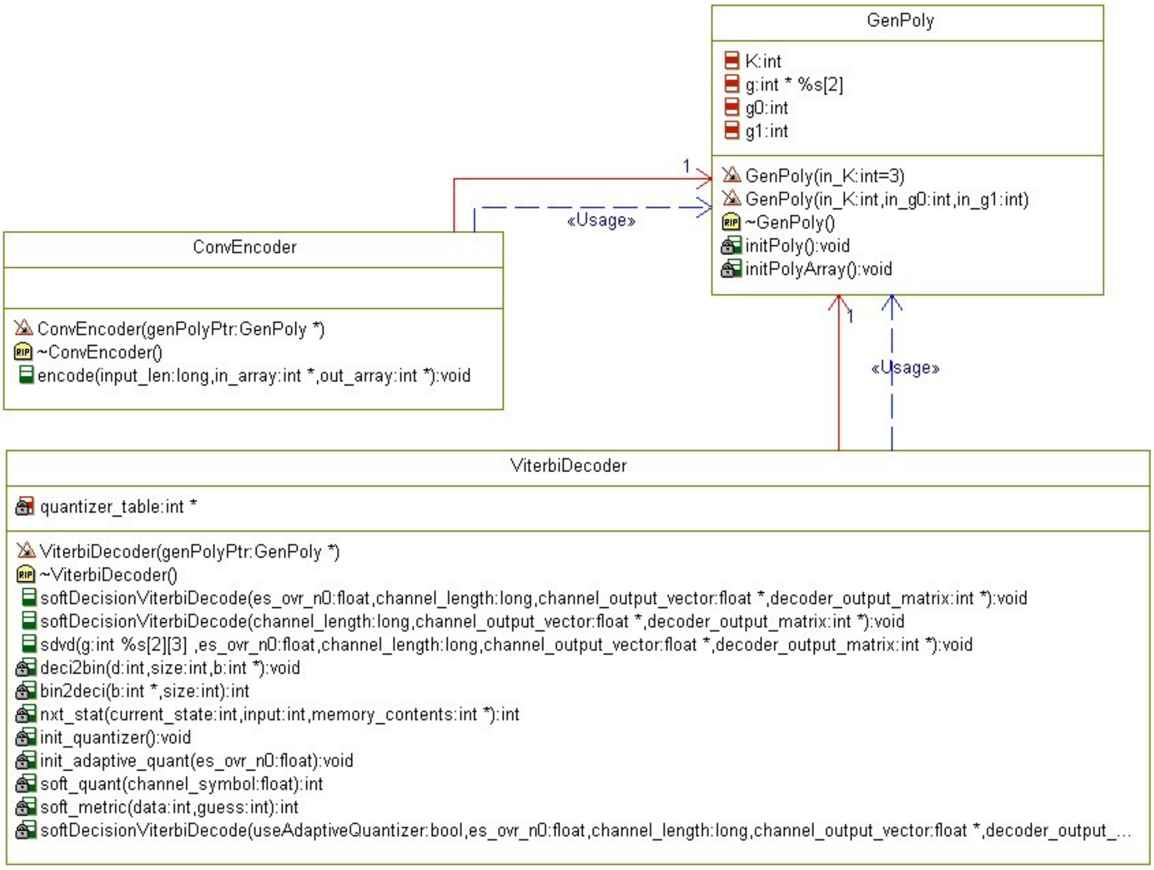


Figure 6-8: Convolutional Encoder & Viterbi Decoder class diagram

Implications on energy detection receiver

An energy detection receiver is a non-coherent receiver and as such it cannot make use of the polarity information modulated in the UWB PHY symbol. It is exactly this polarity information that is stored by the sign bit stream output from the convolutional encoder. In order to be able to evaluate the performance of the convolutional code and decode the sign bit stream at the receiver, the energy detection receiver was modified to take the polarity of each burst into account before that information is lost.

Note that this is definitely not the preferred way of making use of the polarity information. The correct approach is to use a correlation receiver.

A correlation receiver is a coherent receiver because it knows something about the transmitted waveform used by the transmitter. A correlation receiver incorporates a known reference signal. By multiplying the received signal with the reference signal and then integrating over the signal duration, the receiver can very accurately distinguish noise and decide if it is the recipient of a particular signal. Such receivers require very accurate and fine synchronization to line up the received signal with the reference signal.

The implementation of a correlation receiver for the simulation model is outside the scope of this project and therefore the energy detection receiver was modified instead.

6.3 SIMULATING 802.15.4A WSN

The OMNET++ environment allows for easy configuration of the number of nodes in a network and various other network parameters. Network description files specify the different components (OMNET++ modules) to use in your network and it is through these description files that the 802.15.4a UWB IR implementation is incorporated. A graphical view of the components specified in the network description files for the simulated 802.15.4a WSN is given in Figure 6-9.

6.3.1 Application layer

The Mixim-UWB model makes use of a simple `TestApplicationLayer`. It basically requests a configured number of packets of a certain size to be transmitted at certain time intervals throughout the simulation. If all the number of packets had been send, transmission stops. Incoming packets are deleted after some statistics were collected.

6.3.2 Network layer

As network layer `DummyRoute` is used. No special routing functions are performed – it simply translates the network packet information to MAC packet information.

6.3.3 MAC layer

An `AlohaMacLayer` is defined by the Mixim-UWB model. This MAC layer inherits from the base UWB MAC layer provided by the framework, called `UWBIRMac`. The `AlohaMacLayer` supports the Pure ALOHA approach – if a node has data to send, it sends it. If this message collides with another node’s message then retransmission will occur sometime later.

The `UWBIRMac` also creates the `Signal` object used to carry the preamble and data symbol information and generates random bits to serve as the physical data bits of a message.

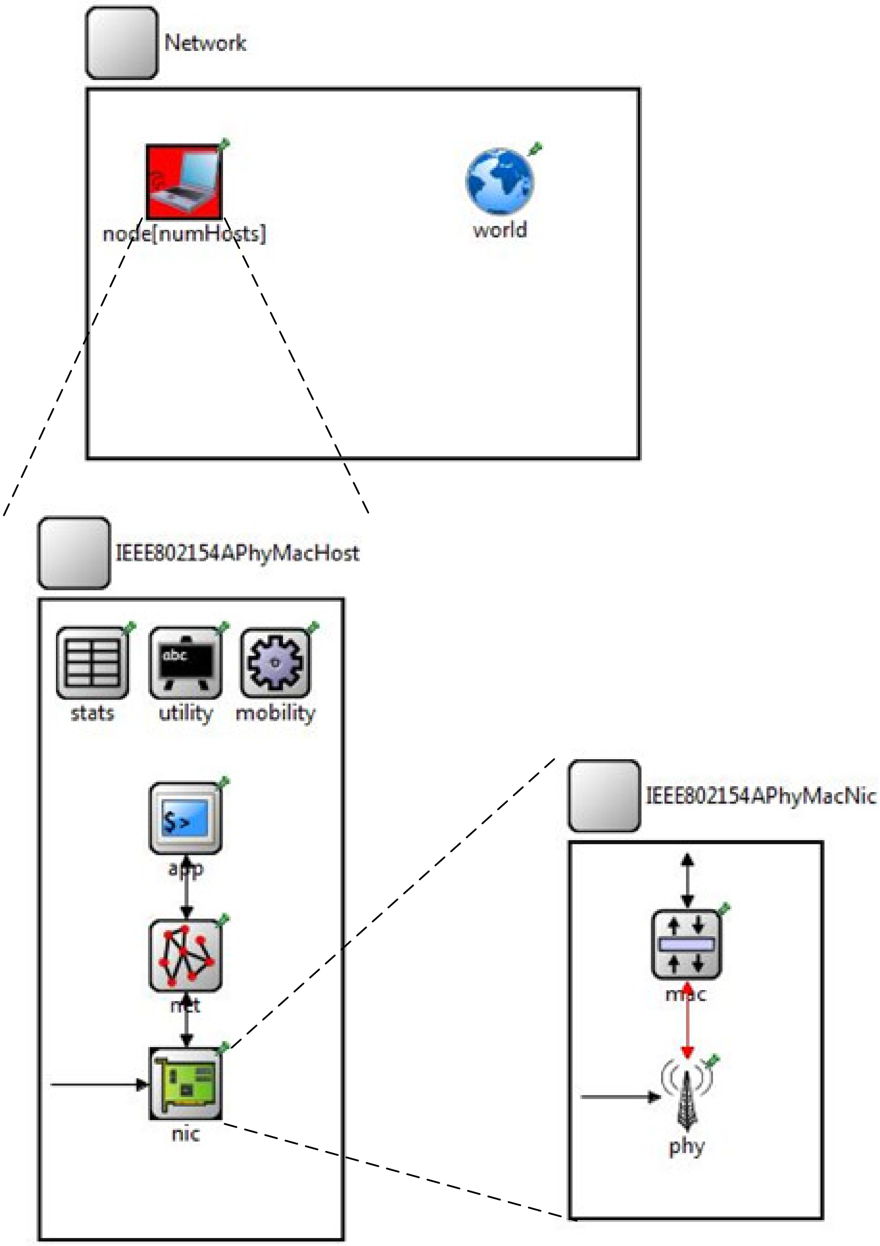


Figure 6-9: Simulated 802.15.4a WSN components

Chapter 7

Simulation Results

The various simulation results and how they were obtained will be presented and discussed in this chapter. All results are easy to reproduce and the reader can browse through the contents on the accompanied DVD to acquire more information and the required software tools (see Addendum A).

7.1 ASSUMPTIONS

For the simulations performed using the 802.15.4a UWB model the following assumptions were made:

1. Perfect synchronization: The Mixim-UWB model allows for basic synchronization on the first preamble. The simulations experiments performed assumed perfect synchronization.
2. Outside interference from other technologies: Other communication systems such as 802.15.11 also introduce interference. This is not taken into account.
3. MAI results not considered: Although the simulation model provides for MAI modeling and a multiple node WSN is presented, no MAI results were considered.
4. Uniform random number generator: Data bits are randomly generated with a uniform distribution.
5. Energy detection receiver used to demodulate bit burst polarity: As mentioned before, an energy detection receiver is non-coherent in nature and cannot take the burst polarity into account. The simulations that make use of the Viterbi decoding algorithm employ a modified version of the energy detection receiver that can read the polarity of the burst. The correct approach will be a correlation receiver.
6. Triangular pulse: A triangular pulse is used as UWB pulse in a chip interval. Although such a pulse might not meet the exact power constraints set by the FCC, it enables a simple UWB signal implementation that does not require a lot of memory to store.

7. Clock drift: Different systems have different clocks making clock drift unavoidable. Clock drift is not modeled or taken into account.

7.2 CONVOLUTIONAL ENCODER AND VITERBI DECODER PORTING

In order to verify that the ported C++ implementation of the convolutional and Viterbi algorithms is working, some tests were performed to compare the results against those of the original C implementation.

Gaussian random noise variables were generated and added to the signal by calculating the standard deviation of Additive White Gaussian Noise (AWGN) for varying E_b/N_o values.

NASA did a lot of experiments and proved that for the constraint length K = 3 the best polynomials is (7, 5). For the comparison tests between the C and C++ implementations these polynomials were used by the convolutional encoder.

Figure 7-1 shows the average results after a few test runs. It is clear that the C++ Viterbi implementation is working properly and gives almost the exact same results as the C Viterbi implementation.

Although the polynomials (7, 5) were proven to be the best for constraint length K =3, the 802.15.4a standard specifies the polynomials (2, 5) be used for the UWB PHY convolutional encoder. The same test were performed using the C++ implementation but with polynomials (2, 5) and the results are also shown on the graph in Figure 7-1. The results clearly show that the polynomials (2, 5) perform considerably worse.

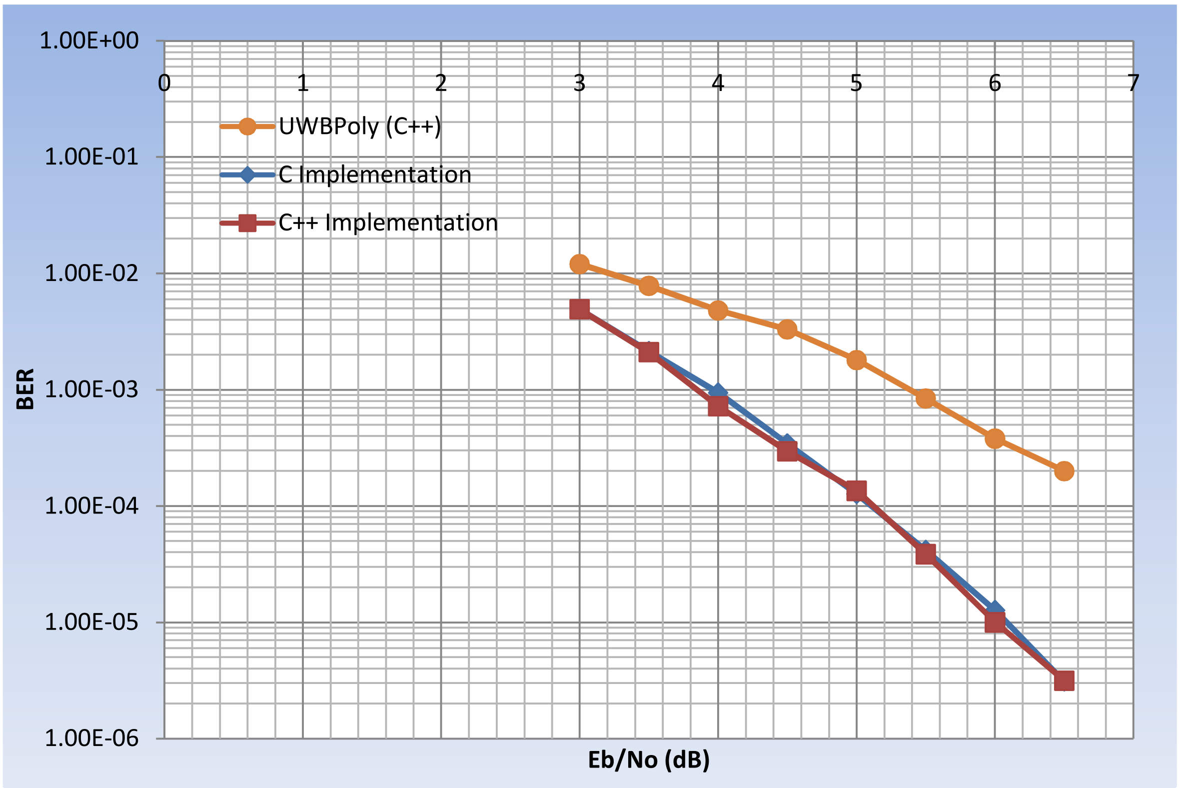


Figure 7-1: Comparing C & C++ implementations of Viterbi algorithm

7.3 SIMULATION MESSAGE TRACE BETWEEN 2 NODES

The scheduling of simulation timers (for self messages) and flow of messages between two communicating nodes are illustrated in Figure 7-2 where node1 transmits a single packet to node0. Events points in the figure are identified by a number prefixed with a # character.

Event	Description
#0	At time 0 the simulation starts. The app layer of node1 schedules a timer (self message) called “app-delay-timer”. This provides a time delay before data is transmitted by the app layer. During this time delay the node can initialize and prepare itself to transmit data. The phy layer of node 1 also schedules a timer message called ‘RadioSwitchingOver” to simulate the time it would take for the radio circuitry of a real node to change state. The default state for a node is RECEIVE but because node1 has to transmit data, the node state is set to TRANSMIT and a “RadioSwitchingOver” timer is scheduled.

#1	The “RadioSwitchingOver” timer expired (self message was received) and the node1 phy radio state is now considered to be TRANSMIT. Another “RadioSwitchingOver” timer is scheduled which simulates the time it would take for the phy layer to notify the mac layer about the radio state change.
#2	The mac layer of node1 receives the message and prepares the mac to receive data from the network layer.
#3	The “app-delay-timer” expires after 5 seconds. The node is now initialized and ready to transmit data. The node1 application layer creates a message called “DataMessage” and sends it down to the node1 network layer.
#4	The node1 network layer gets “DataMessage”, does the necessary processing and sends “DataMessage” off to the node1 mac layer.
#5	The node1 mac layer gets “DataMessage”, does the necessary processing and sends “DataMessage” off to the node1 phy layer.
#6	The node1 phy layer gets “DataMessage” and transmits it on the communication channel as an Airframe. The node1 phy layer also creates a “RadioTxOver” timer to simulate the time it takes the radio circuitry to transmit all the data bits on the channel.
#7	The radio of node0 is in the RECEIVE state and actively listening for any airframes. At this event it detected and synchronized to the start of the airframe transmitted by node1 and schedules an “airframe” self message as a short delay to allow all bits of the airframe to be received.
#8	All bits of the airframe is now received and the node0 phy layer schedules another “airframe” self message for the purpose of simulating the “over air” transmit time and to add any noise as defined by the channel propagation model currently utilized.
#9	Node1 phy layer “RadioTxOver” timer expires, indicating that the radio has finished transmitting all data bits. A “TxOver” control message is also send to

	the node1 mac layer to report this to the upper layers.
#10	The node0 phy does the final processing of the airframe, decodes the transmitted data contained in the airframe and sends this data off to the node0 mac layer.
#11	The “transmission over” message notification is received by the node1 mac layer and forwarded to the node1 net layer.
#12	The node0 mac layer receives the data bits transmitted all the way from node1 app layer. The mac layer decides if the message was received correctly, does some statistic calculations and then throws away the message (in this case there is no acknowledgement message). Note that the simulation could easily be adapted to also model the processing of the received message by the upper layers (net & app) of node0.
#13	The “transmission over” message notification is received by the node1 net layer and forwarded to the node1 app layer.
#14	The “transmission over” message notification is received by the node1 app layer which logs the occurrence.

.

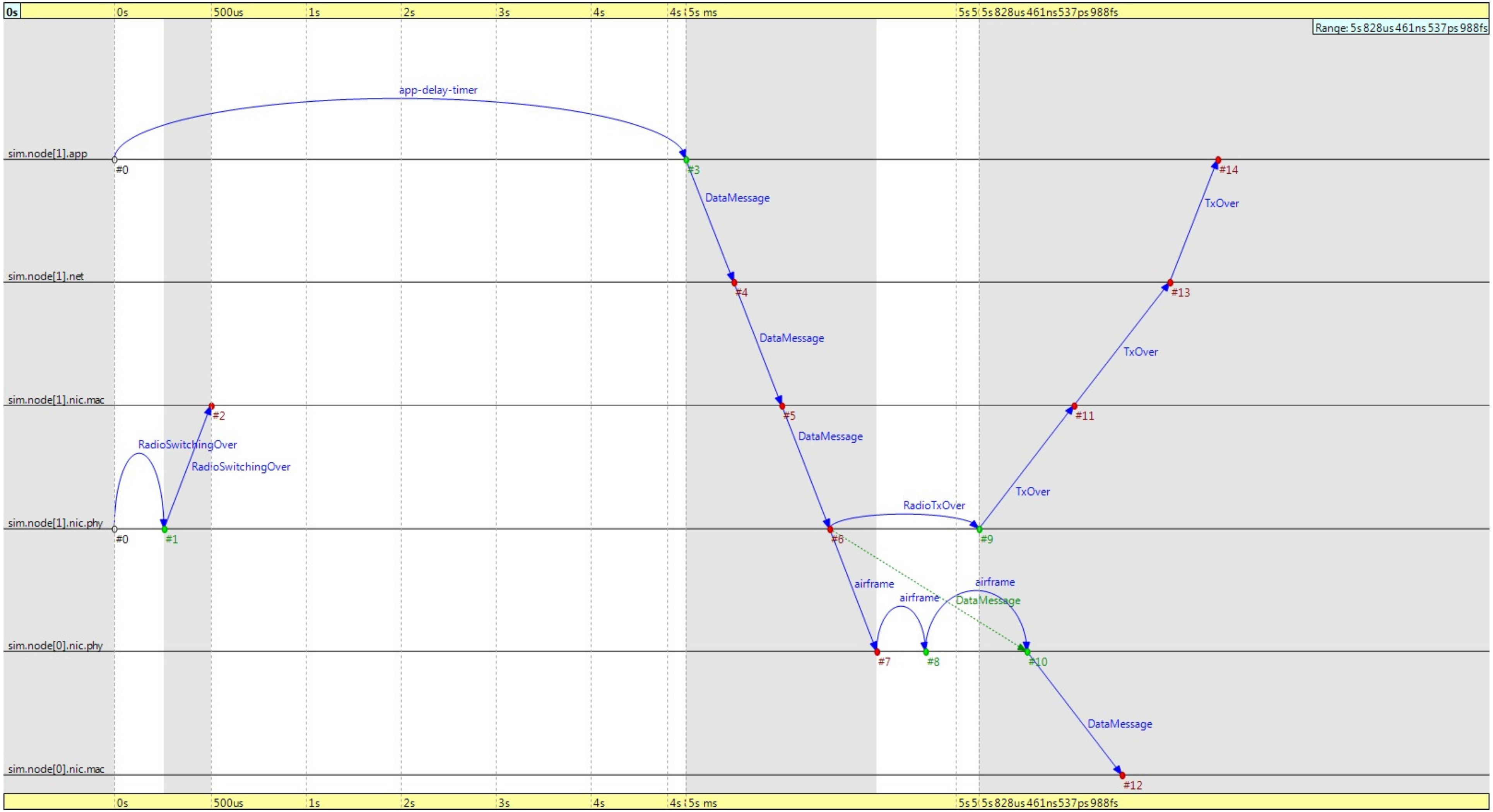


Figure 7-2: 802.15.4a UWB model – Message flow between 2 nodes

7.4 2-NODE 802.15.4A UWB WSN

In order to analyze the 802.15.4a UWB model, the simulation scenario depicted in Figure 7-3 was used. Although simple, this simulation allows for a lot of 802.15.4a physical layer characteristics to be analyzed.

The World module creates a simulation playground for the two nodes of 500m x 500m x 500m. Only node1 are allowed to send messages which node0 receive and then compares with the originally transmitted data. Node0 also calculates and logs information such as the bit error rate (BER) and data bit rate.

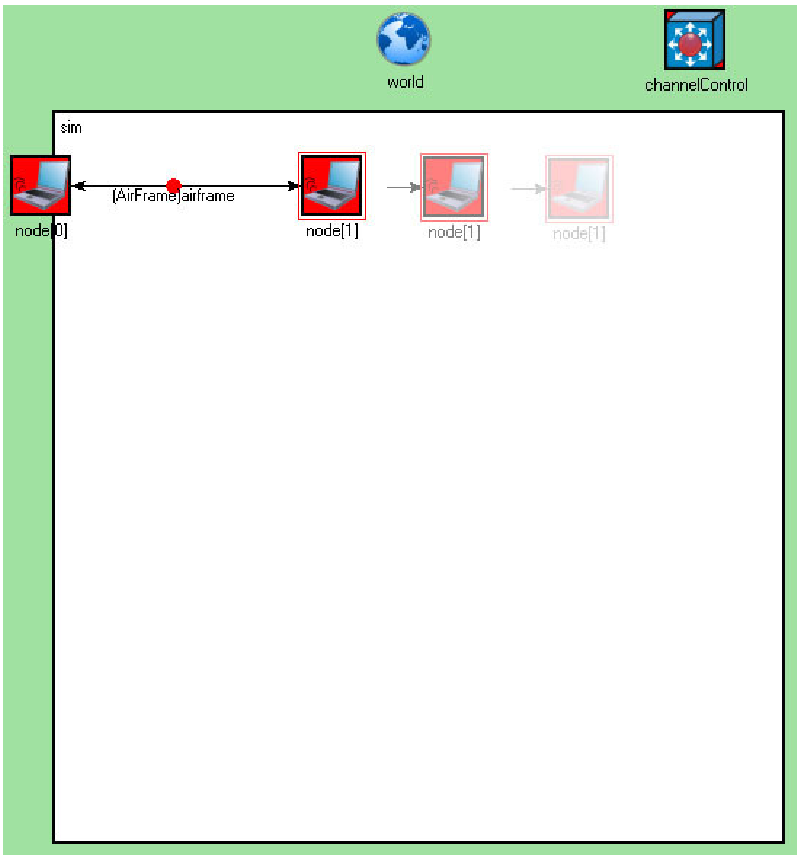


Figure 7-3: 802.15.4a UWB model – 2 node WSN scenario

Each node has the ability to move. Node1, after sending a set number of packets of a certain size, uses this functionality to move away from node0 before starting to send new packets again. The simulation of this mobility allows for performance measurements to be taken over different communication distances.

BER and data bit rate are the performance measurements employed in the analyses performed for this result set. The channel propagation models utilized to model signal attenuation are:

- CM1 (LOS)
- CM2 (NLOS)
- Ghassemzadeh-LOS
- Ghassemzadeh-NLOS

7.4.1 Effect of different bit rates

The 802.15.4a standard defines the nominal data rates indicated in Table 7-1 for the UWB physical layer.

Table 7-1: 802.15.4a UWB PHY nominal data rates

Mean PRF	3.90MHz	15.60MHz	62.40MHz
DataRate0 (Mbps)	0.11	0.11	0.11
DataRate1 (Mbps)	0.85	0.85	0.85
DataRate2 (Mbps)	1.70	6.81	6.81
DataRate3 (Mbps)	6.81	27.24	27.24

As a first result, the mandatory channel 3 in frequency band 1 is considered. For this mandatory channel the mean PRF is 15.6MHz, the bandwidth is 499.2MHz and the preamble code length equal to 31. By varying the number of chips per burst (N_{cpb}) while keeping the number of possible burst positions (N_{burst}) in a data symbol constant, the four data bit rates listed in Table 7-1 is achieved.

Figure 7-4 shows how the BER changes over distance when the data bit rate is 850kpbs – the experiment is run over the four channel models and the corresponding results are plotted together.

The configuration settings for this simulation scenario are:

- 100 packets sent from node1 to node0

- 7 byte MAC header
- 73 byte PHY payload
- 64 000 bits transmitted per experiment run
- Forward error correction not performed at the receiver

The CM2 channel model is most harsh and in such environments any nodes communicating further than 4 meters apart will require a lot of retransmissions due to a high number of bit errors. The Ghassemzadeh-NLOS channel model does not allow for much longer distances and compared to the CM2 channel model it performs much the same.

The LOS channel models, CM1 and Ghassemzadeh-LOS, yields the best results. Further experiments with larger datasets demonstrated that for communication distances up to 50m a BER of 1e-6 can be maintained.

Figure 7-5 to Figure 7-7 depicts the distance performance results of the same mandatory channel 3 setup, but with the data bit rate at 110kbps, 6.85Mbps and 27Mbps respectively.

From these results it can be observed that for a given distance and channel model the BER gets worse with increasing bit rate. The reason for this is found in a symbol burst - a large number of chips per burst (N_{cpb}) results in a high processing gain because the increased burst duration (T_{burst}) allows for more efficient reception. Longer bursts means longer symbol durations if the number of burst positions (N_{burst}) remains constant. Consequently a penalty hit is taken on the bit rate.

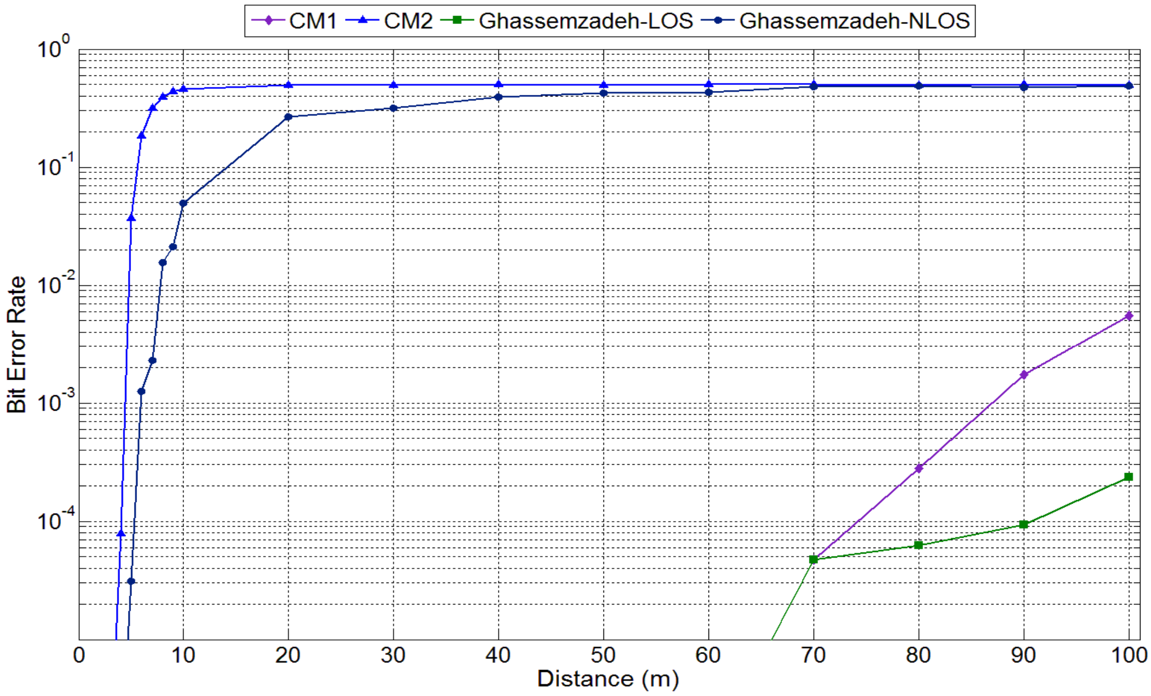


Figure 7-4: BER vs Distance – Chan 3, PRF 15.6MHz, Bit rate 850kbps

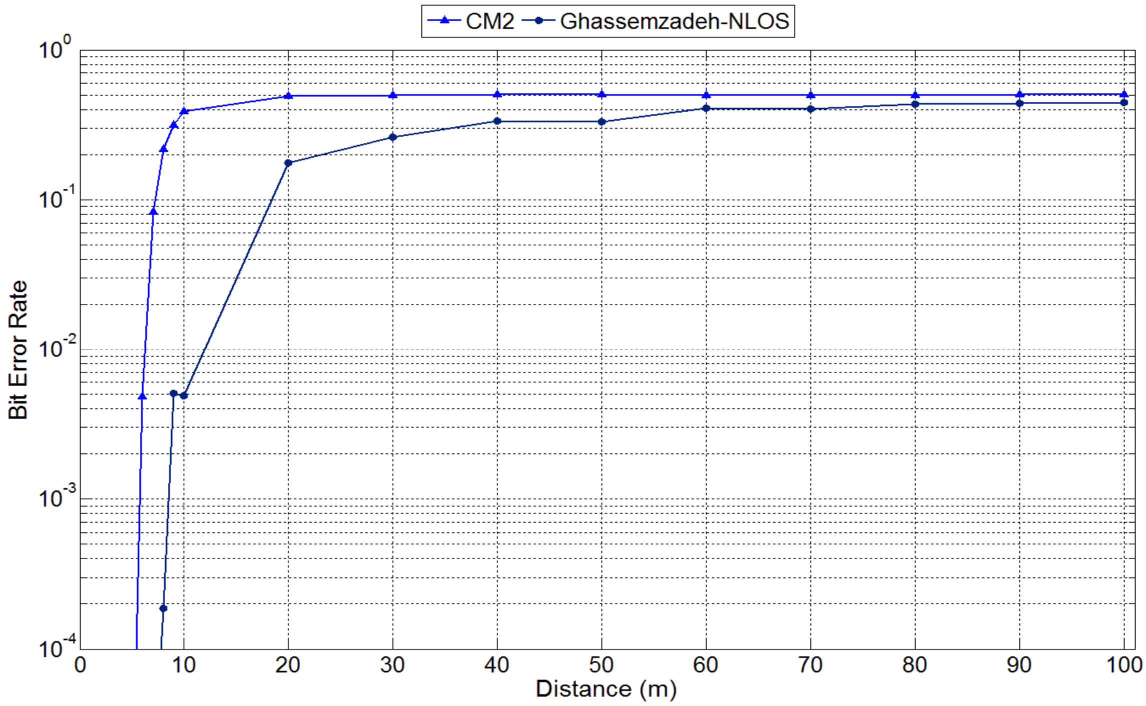


Figure 7-5: BER vs Distance – Chan 3, PRF 15.6MHz, Bit rate 110kbps

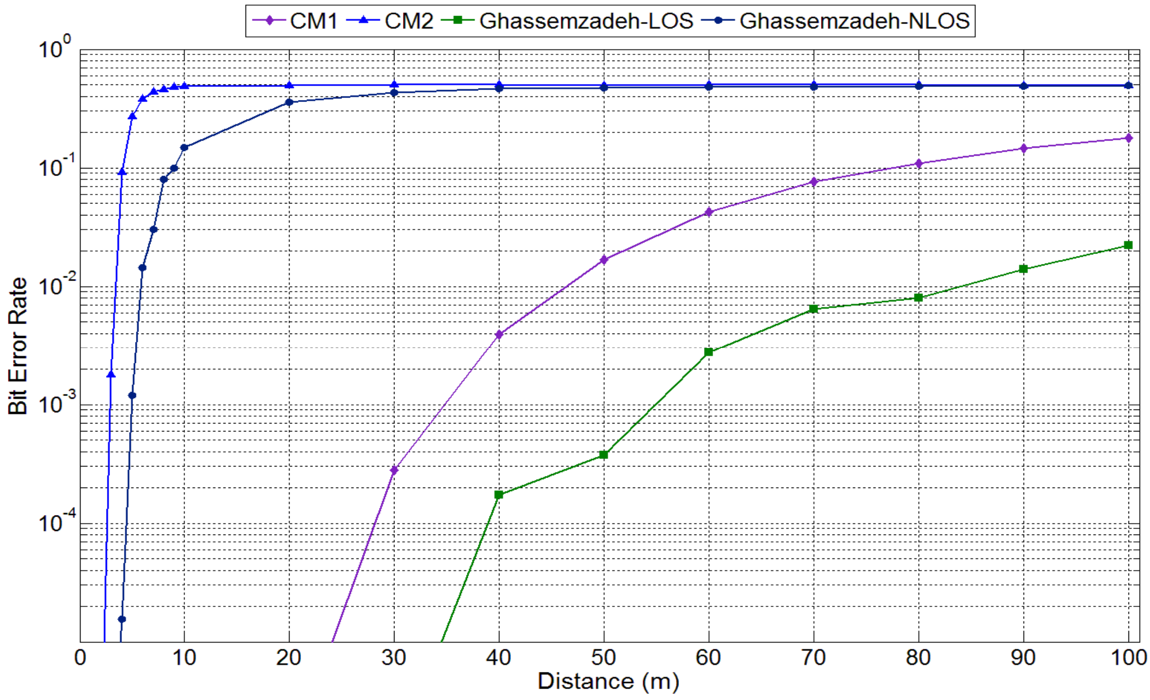


Figure 7-6: BER vs Distance – Chan 3, PRF 15.6MHz, Bit rate 6.8Mbps

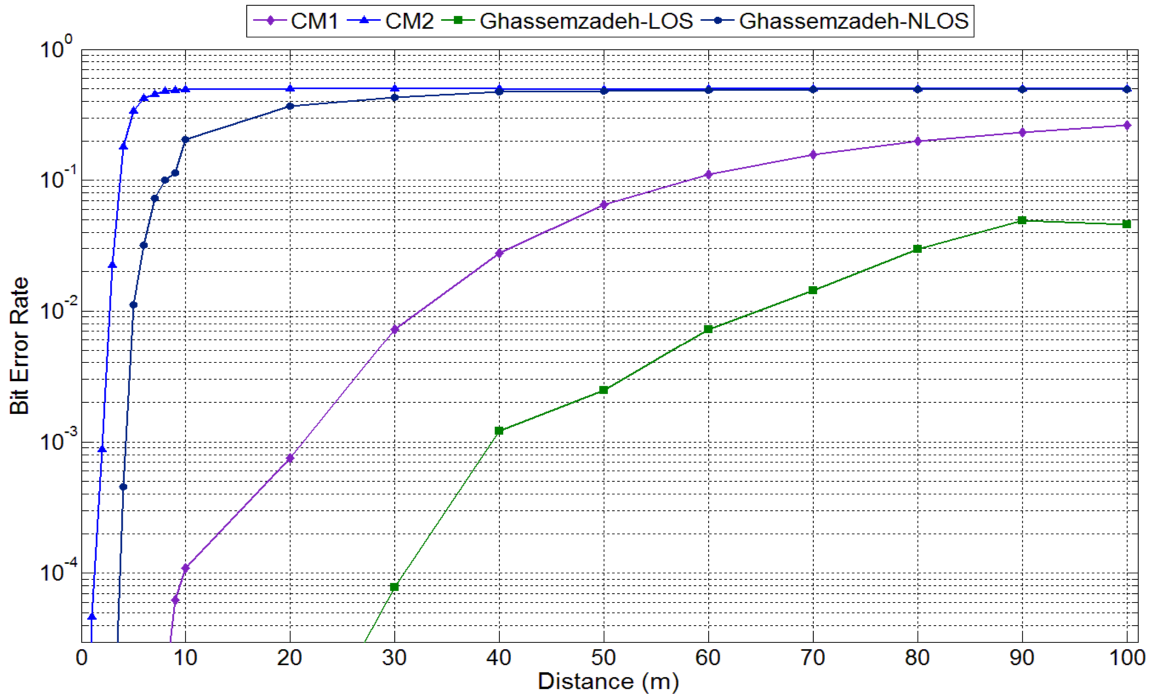


Figure 7-7: BER vs Distance – Chan 3, PRF 15.6MHz, Bit rate 27Mbps

7.4.2 Effect of mean pulse repetition frequency (PRF)

The other two mean PRFs defined by the standard is 3.90MHz and 62.40MHz. The difference between the three specified PRFs is captured in the number of burst positions per symbol (N_{burst}) as defined in Table 7-2.

Table 7-2: 802.15.4a UWB PHY number of burst positions for a given mean PRF

Mean PRF	3.90MHz	15.60MHz	62.40MHz
N_{burst}	128	32	8

The resulting BER performance over distance for channel 3 at 850kbps with PRF of 3.90MHz is shown in Figure 7-8. Compared to a PRF of 15.6MHz (Figure 7-4) it performs a little less well because it has less number of chips per burst (N_{cpb}).

The resulting BER performance with a mean PRF of 62.40MHz is shown in Figure 7-9. Compared to a PRF of 15.6MHz (Figure 7-4) it performs much better, especially for the LOS channels. Again this is due to the 62.40MHz configuration having more N_{cpb} than the 15.6MHz configuration. The standard also defines a preamble length of 127 for a mean PRF of 62.40MHz, allowing for better synchronization at the faster rate.

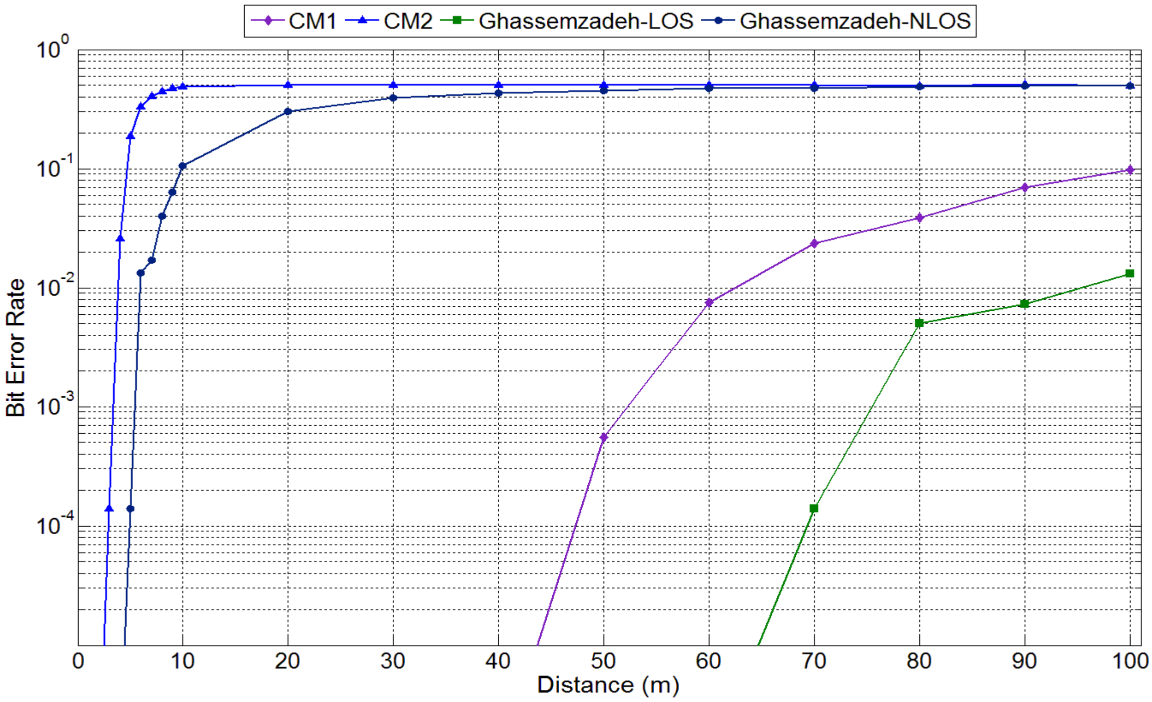


Figure 7-8: BER vs Distance – Chan 3, PRF 3.90MHz, Bit rate 850kbps

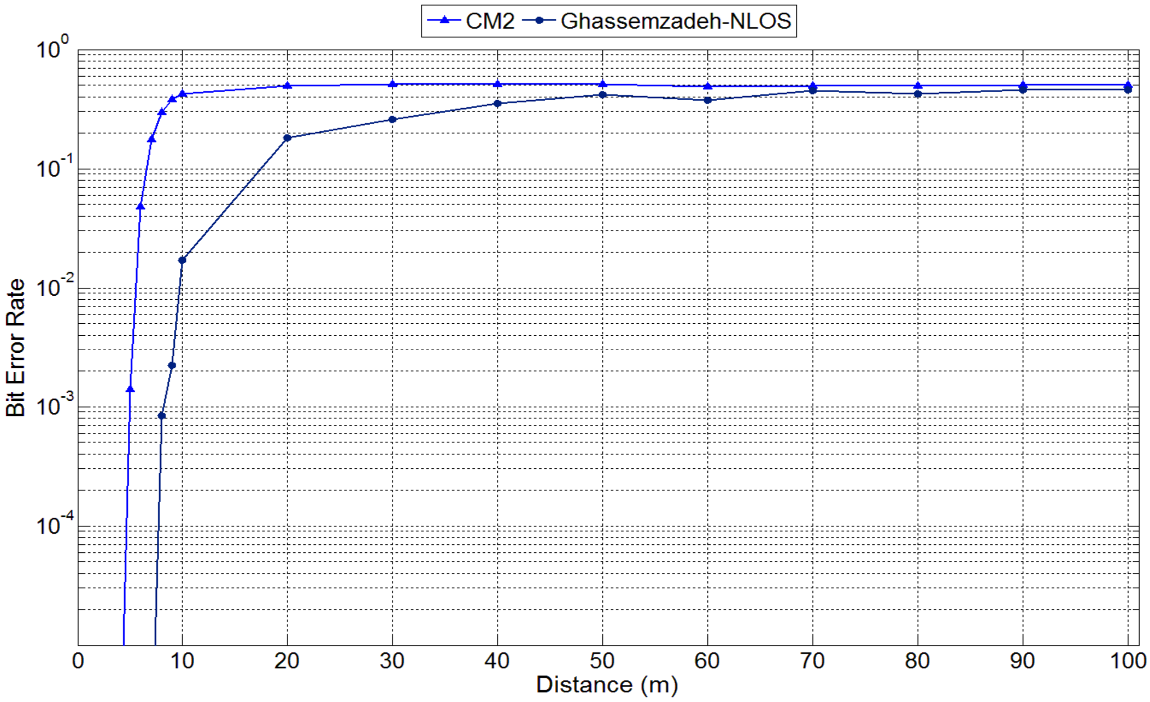


Figure 7-9: BER vs Distance – Chan 3, PRF 62.40MHz, Bit rate 850kbps

7.4.3 Effect of center frequency

Figure 7-10 can be used to observe the effect the center frequency has on the communication performance. The center frequency of channel 3 is specified at 3993.6MHz, and that of channel 14 at 9984.0MHz.

A low frequency signifies a long wavelength, while a high frequency indicates a short wavelength. In practice it has been shown that long wavelength signals carries further than short wavelength signals and the longer wavelength also allows for better penetration capability.

It can be seen that channel 14 performs worse than channel 3 for the same bit rate and mean PRF.

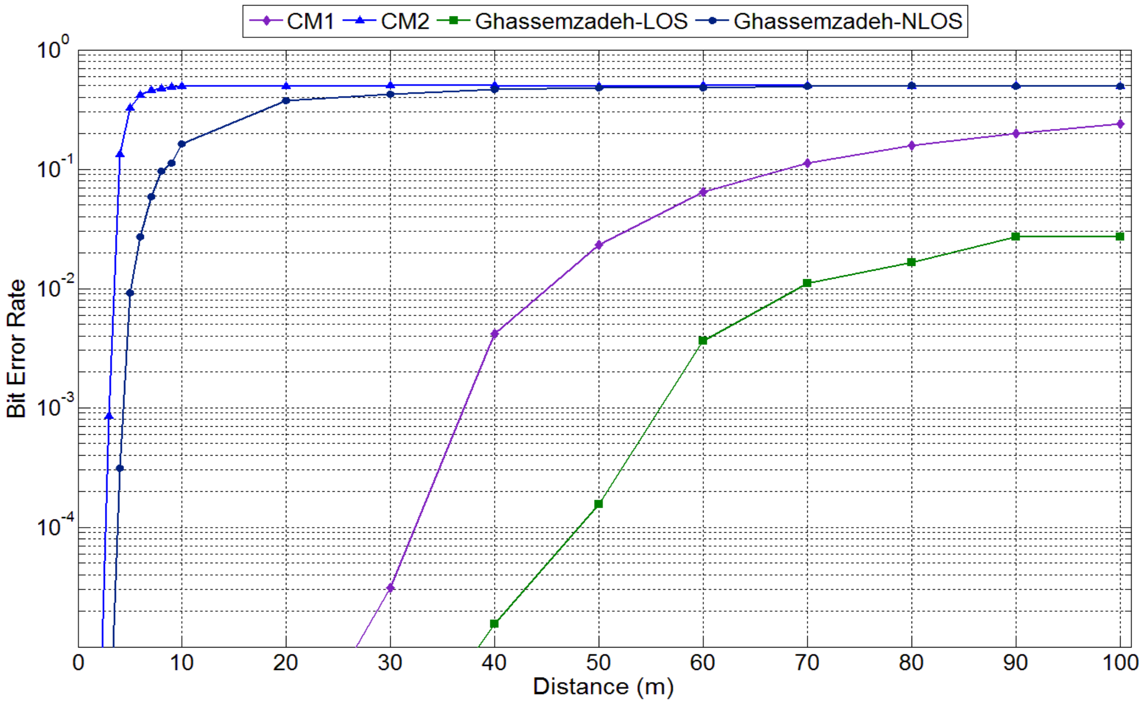


Figure 7-10: BER vs Distance – Chan 14, PRF 15.6MHz, Bit rate 850kbps

7.4.4 Effect of bandwidth

The 802.15.4a standard defines four channels with a bandwidth greater than 499.2MHz as shown in Table 7-3.

Table 7-3: 802.15.4a UWB PHY channels with increased bandwidth

Channel	Bandwidth (MHz)
4	1331.2
7	1081.6
11	1331.2
15	1354.97

A larger bandwidth allows for greater channel capacity and also very high precision range measurements. On the downside, it also implies that more noise will now be present on the channel.

The results of an experiment run with channel 4 at 850kbps and PRF 15.60MHz depicts an improvement in the BER for all channel models if compared to channel 3 with the same settings. The BER for the LOS channel models are better than 1e-4 at a distance of 100m. See Figure 7-11 for the graph.

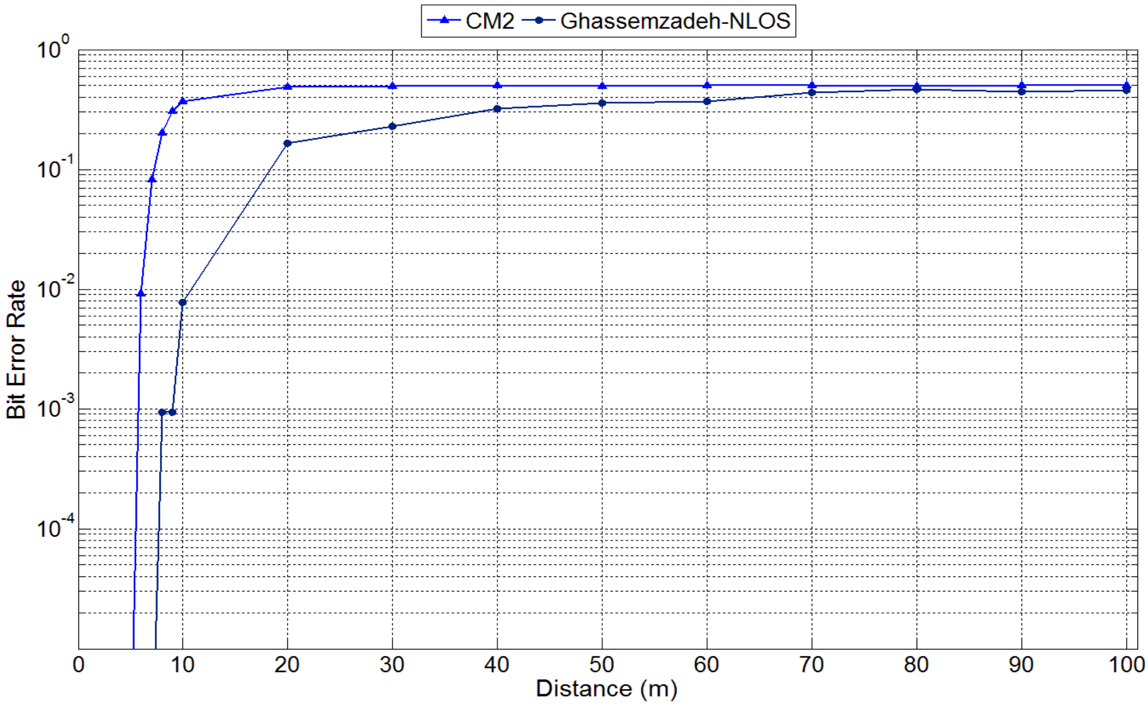


Figure 7-11: BER vs Distance – Chan 4, PRF 15.6MHz, Bit rate 850kbps

7.4.5 Sub-Gigahertz band communication

In the sub-gigahertz band only one channel, channel 0, is specified. It sits at a center frequency of 499.2MHz with a bandwidth of 499.2MHz. By taking the different effects discussed so far into account, it can be seen that channel 0 with PRF of 3.90MHz and data rate of 110kbps will result in the best BER performance over very large distances.

Figure 7-12 proves this and it is clear that even the for NLOS indoor channel propagation models the achievable distances are very good – in the transmission of 64 000 bits, data bit corruption only occurred at distances greater than 10 meters. The downside though is a much lower data bit rate.

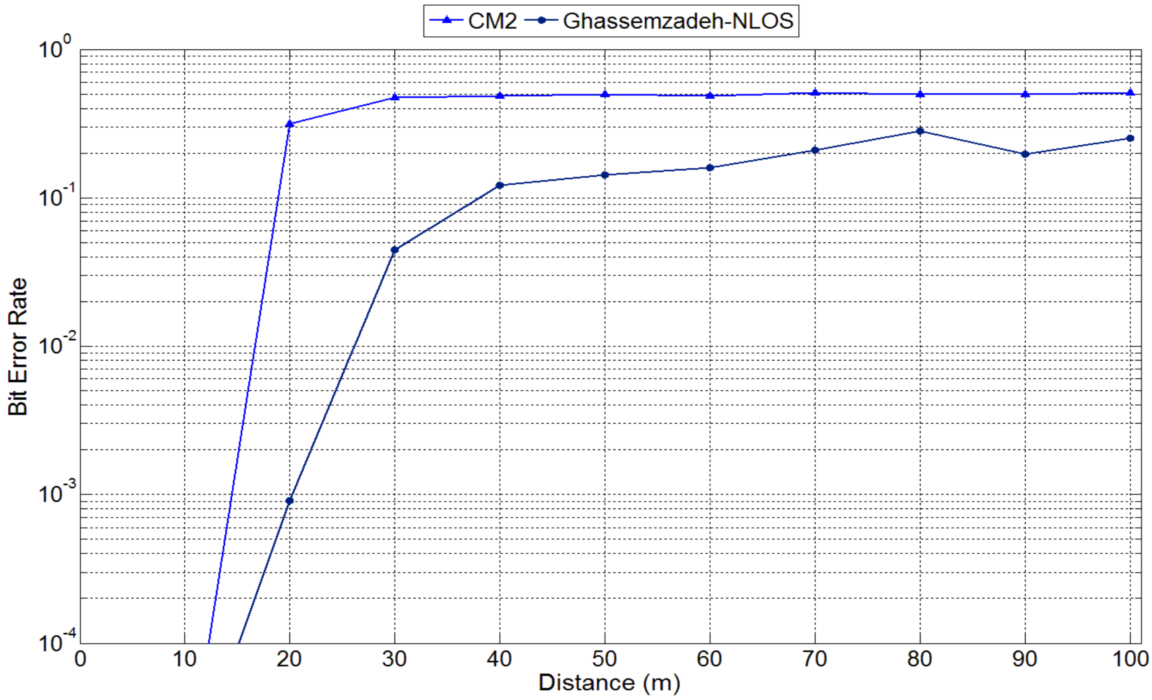


Figure 7-12: BER vs Distance – Chan 0, PRF 3.90MHz, Bit rate 110kbps

7.4.6 Effect of forward error correction

The modified 802.15.4a UWB model allows for the Reed-Solomon and Viterbi algorithms to be run separately, together or not at all.

After various simulation experiments with different channels, PRFs, bandwidths and bitrates, it was interesting to find that the FEC does not add a considerable amount of gain when the BER is already bad.

By definition the Reed-Solomon (63, 55) code are only capable of recovering from 4 symbol errors for each R-S encoded data block of 378 bits (330 data + 48 parity). For an environment with high BER at large distances, this limitation causes the R-S decoder to fail most of the times resulting in processing without gain. A possible improvement might be a better code such as a (255, 223) with which 16 symbol errors can be corrected.

The performance of the Viterbi decoder on its own is also meager for such high BER cases. In order to integrate the Viterbi algorithm in the UWB model, a coarse quantization and normalization of the energy detector voltage values had to be done. This approach is known to be sub-optimal. It was also proven through simulation that the convolutional encoder polynomials (7, 5) delivers much better gain than the specified (2 ,5) polynomials. The Viterbi algorithm is also known to work much better in communication scenarios polluted by AWGN.

For very low BERs the single error that occurs here or there are easily fixed by the forward error correction techniques adding the advantage of not having to retransmit the corrupted data.

A graphical comparison between the different FEC combinations is provided in Figure 7-13 and Figure 7-14. The mandatory channel 3 was used with PRF 15.6MHz and data rate at 850kbps. The LOS channels already perform excellent so these two figures depict the results of the NLOS channel models, Ghassemzadeh-NLOS and CM2 for distances up to a few meters.

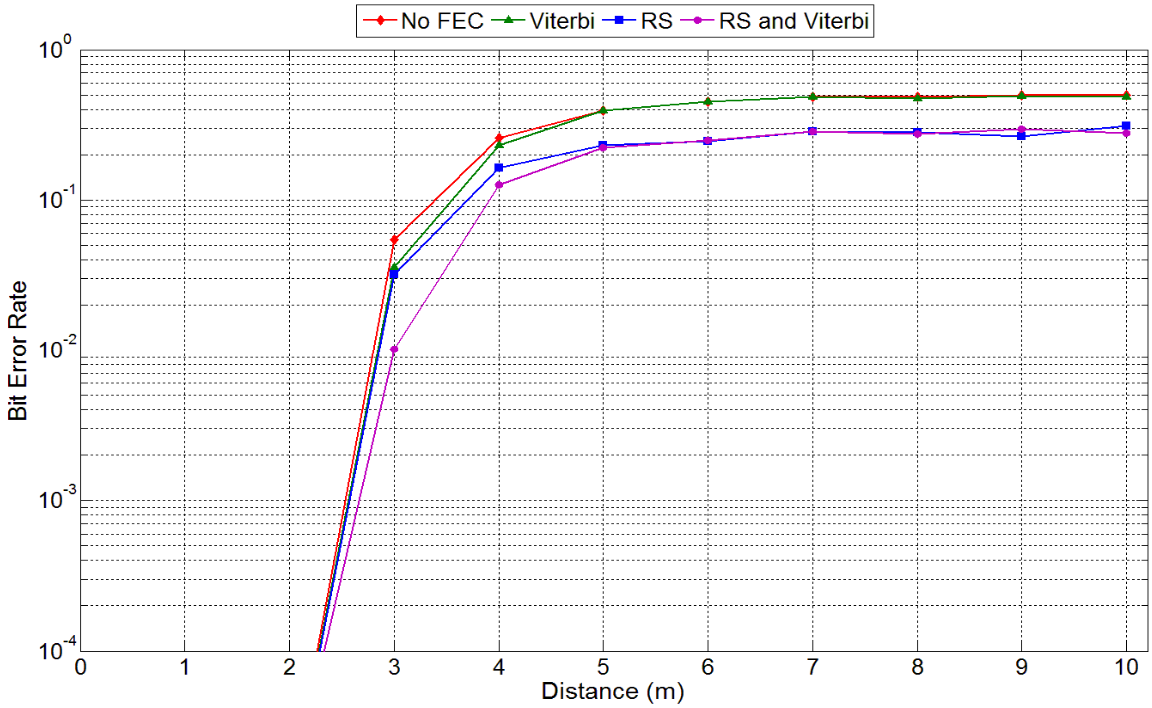


Figure 7-13: BER vs Distance – CM2 FEC compare

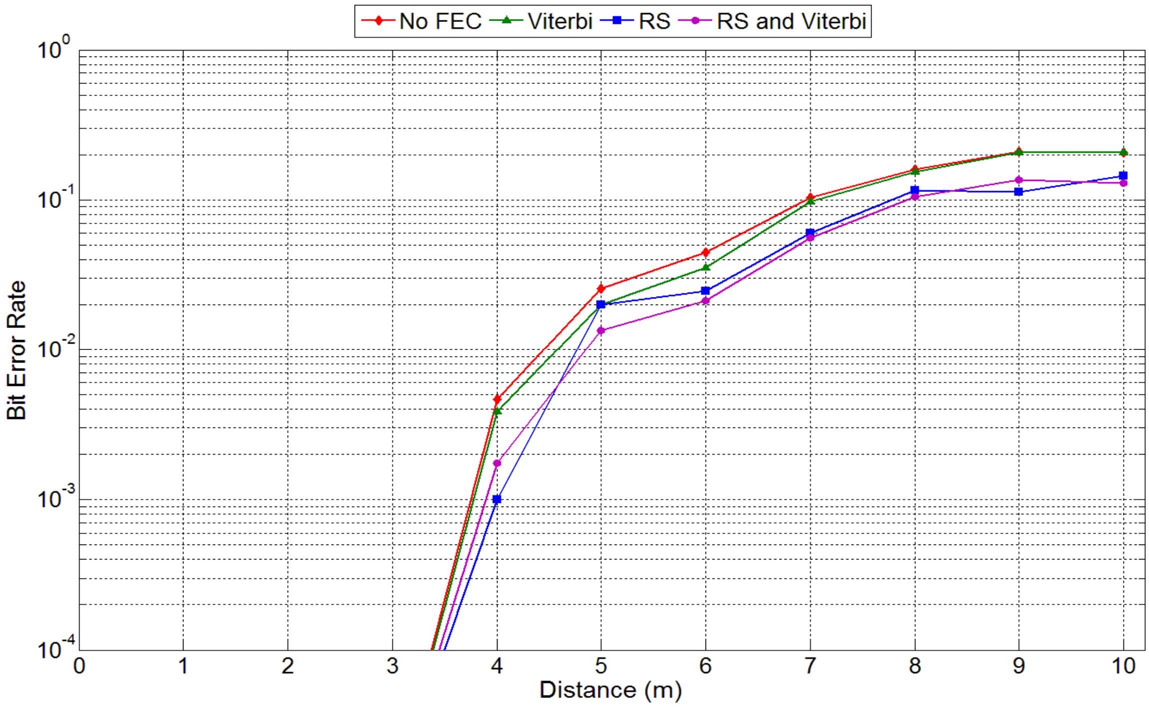


Figure 7-14: BER vs Distance – Ghassemzadeh-NLOS FEC compare

As discussed these results show that for a high BER the FEC does not add substantial gain. Table 7-4 and Table 7-5 show the FEC gains for the two NLOS channel models at some of the distances points.

Table 7-4: 802.15.4a UWB PHY – Ghassemzadeh-NLOS FEC compare

	Ghassemzadeh-NLOS at distance of 4m		Ghassemzadeh-NLOS at distance of 6m	
	BER	**Gain (dB)**	**BER**	**Gain (dB)**
No FEC	0.004625	0	0.0445	0
RS	0.001	6.65	0.02475	2.55
Viterbi	0.003875	0.77	0.0355	0.98
RS and Viterbi	0.00175	4.22	0.021375	3.18

Table 7-5: 802.15.4a UWB PHY – CM2 FEC compare

	CM2 at distance of 3m		CM2 at distance of 4m	
	BER	**Gain (dB)**	**BER**	**Gain (dB)**
No FEC	0.05425	0	0.0445	0
RS	0.032	2.29	0.02475	2.01
Viterbi	0.03575	1.81	0.0355	0.49
RS and Viterbi	0.010125	7.29	0.021375	3.1

The results indicate that at close range (up to 6 meters) some considerable gain for NLOS channel models is achieved by employing the FEC techniques.

7.5 MULTI-NODE 802.15.4A UWB WSN

Figure 7-15 shows an example of a typical WSN with multiple nodes at random locations. Nodes can communicate with each other at the same time introducing multiple access interference. In this scenario all nodes can communicate with each other. If the 802.15.4a MAC protocol is used, a node can be declared as a RFD or FFD which limits the connections between nodes because an RFD can only communicate with the PAN coordinator FFD.

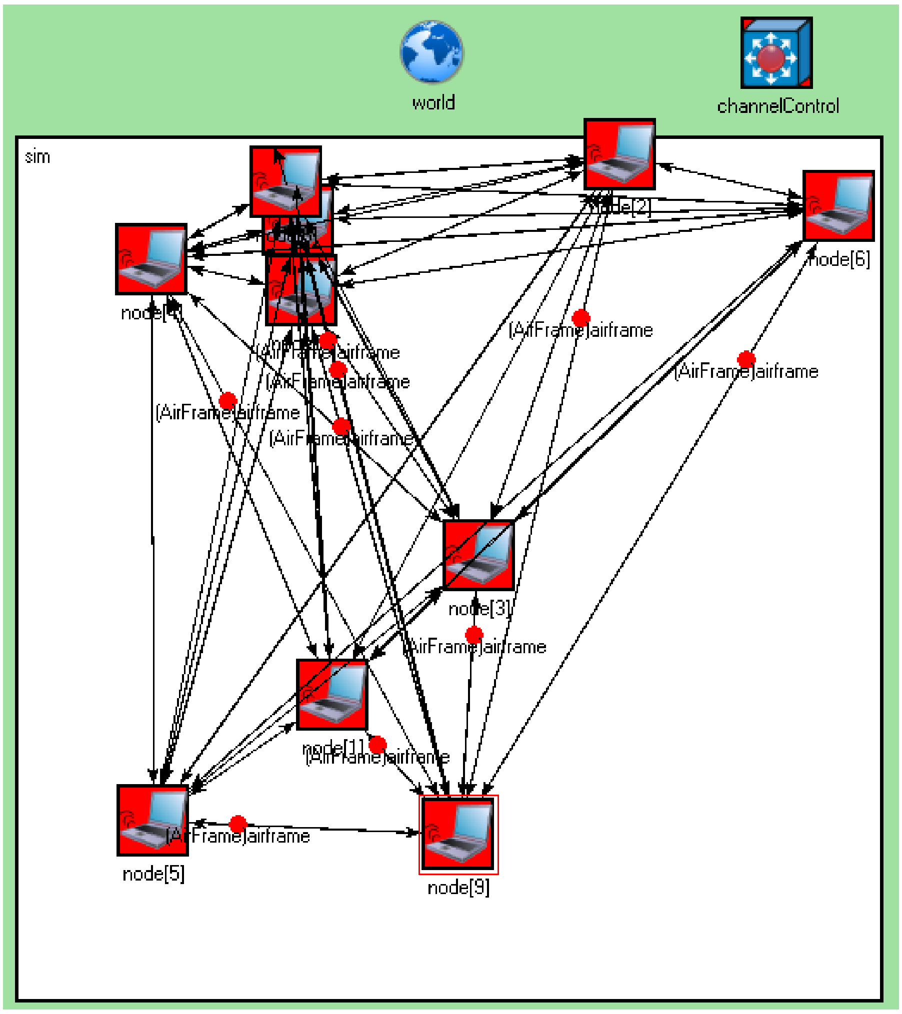


Figure 7-15: 802.15.4a UWB model – Multiple node WSN scenario

Although MAI results are not presented, the Mixim model was designed to take it into account. When starting to receive, a node's receiver will typically take into account all interfering airframes and attenuate the received signal accordingly. Figure 7-16 demonstrates this concept.

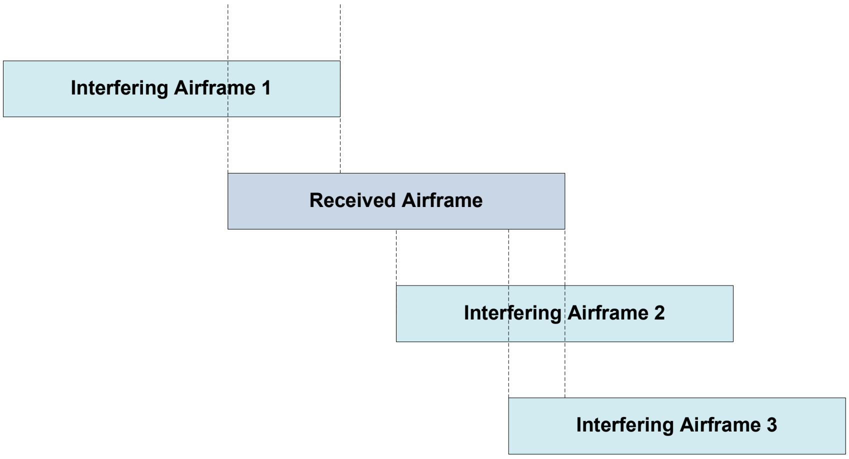


Figure 7-16: Interfering airframes causing MAI

7.5.2 802.15.4a UWB PHY throughput

Some curious results are ascertained when comparing the different data bit rates possible with no FEC, with only R-S encoding, with only convolutional decoding and with both encoding mechanisms activated (see Table 7-6).

Without any FEC, data rates of up to 62.4Mbps can be achieved. Even the bit rate of channel 0, which proved to be very robust over long communication distances, can be more than doubled.

This imparts something to consider – how can a node achieve the best throughput while still maintaining its mobility functionality?

If a node knows its goals, operating environment, channel characteristics, power constraints, maximum allowable BER and required minimum throughput, it can be programmed beforehand with a “best performance” configuration allowing it to change certain communication parameters to best achieve its mission.

The “best performance” configuration does not even have to be loaded beforehand. If power constraints permits, a node can be made intelligent and it can train itself to “learn” about its operating environment and channel. With this self obtained knowledge the node can take into consideration all variables and set up the communication parameters in such a

Table 7-6: 802.15.4a UWB PHY – Data bit rates for different FEC combinations

Tdsym (ns)	Viterbi Rate	RS Rate	Overall FEC Rate	Bit Rate (No FEC)	Bit Rate (Viterbi Only)	Bit Rate (RS Only)	Bit Rate (Viterbi & RS)	Applicable mean PRF
32.05	0.50	0.87	0.44	62 402 496	31 201 248	54 290 172	27 145 086	64M
64.10	1.00	0.87	0.87	31 201 248	31 201 248	27 145 086	27 145 086	16M
128.21	0.50	0.87	0.44	15 599 407	7 799 704	13 571 484	6 785 742	16M & 64M
256.41	1.00	0.87	0.87	7 800 008	7 800 008	6 786 007	6 786 007	4M
512.82	0.50	0.87	0.44	3 900 004	1 950 002	3 393 003	1 696 502	4M
1025.64	0.50	0.87	0.44	1 950 002	975 001	1 696 502	848 251	4M & 16M & 64M
8205.13	0.50	0.87	0.44	243 750	121 875	212 062	106 031	4M & 16M & 64M

manner as to optimally achieve its goals. Even FEC can be turned on and off as required. If stationary, the node can optimize for speed. If mobile, the node can optimize for distance.

Figure 7-17 to Figure 7-19 shows the BER for the four data bit rates as defined by the standard. Only distances that guarantee a BER < 1e-4 is shown.

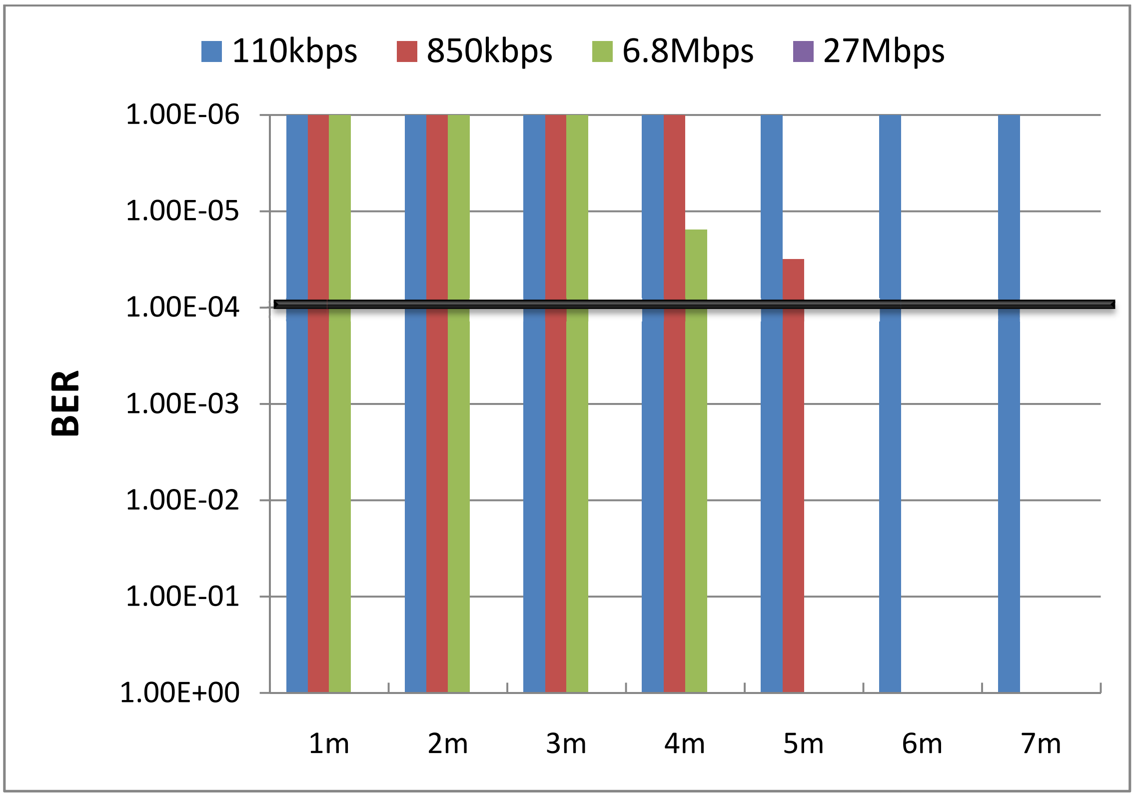


Figure 7-17: Throughputs and distances with BER < 1e-4 (CM1)

In a communication environment that follows a CM1 channel model, Figure 7-17 shows that transmitting at 27Mbps will not give a BER < 1e-4 even if the distance between the transmitting nodes is 1m. Communicating at 6.8Mbps will give a BER < 1e-4 up to a distance of 4m. With 850kbps the distance limit is 5m and with 110kbps up to 7m can be achieved.

From Figure 7-18 it is clear that in a CM2 type environment the 6.8Mbps and 27Mbps data bit rates can be achieved for nodes that are close together (1m or 2m), while still maintaining a BER < 1e-4.

The Ghassemzadeh-NLOS channel model is more lenient and a bit rate of 27Mbps can be achieved up to 9m in such an environment. The Ghassemzadeh-LOS case is not shown because the model does not provide practical results for throughput profiling.

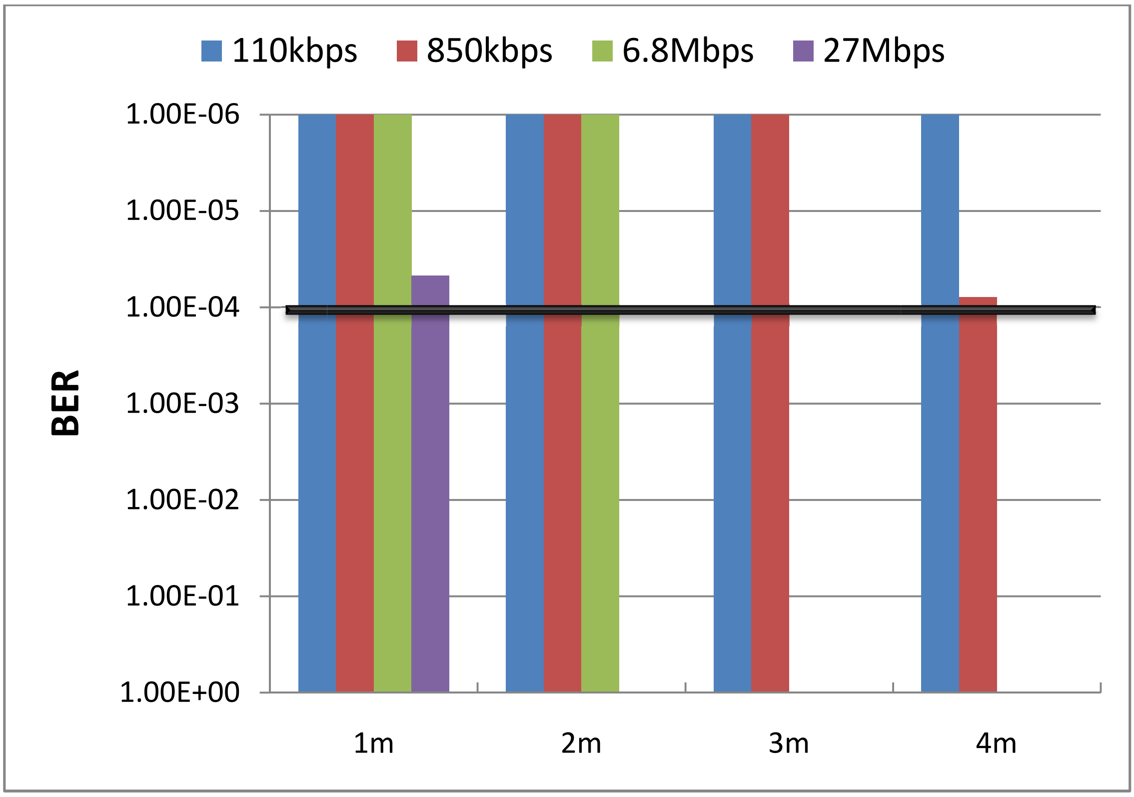


Figure 7-18: Throughputs and distances with BER < 1e-4 (CM2)

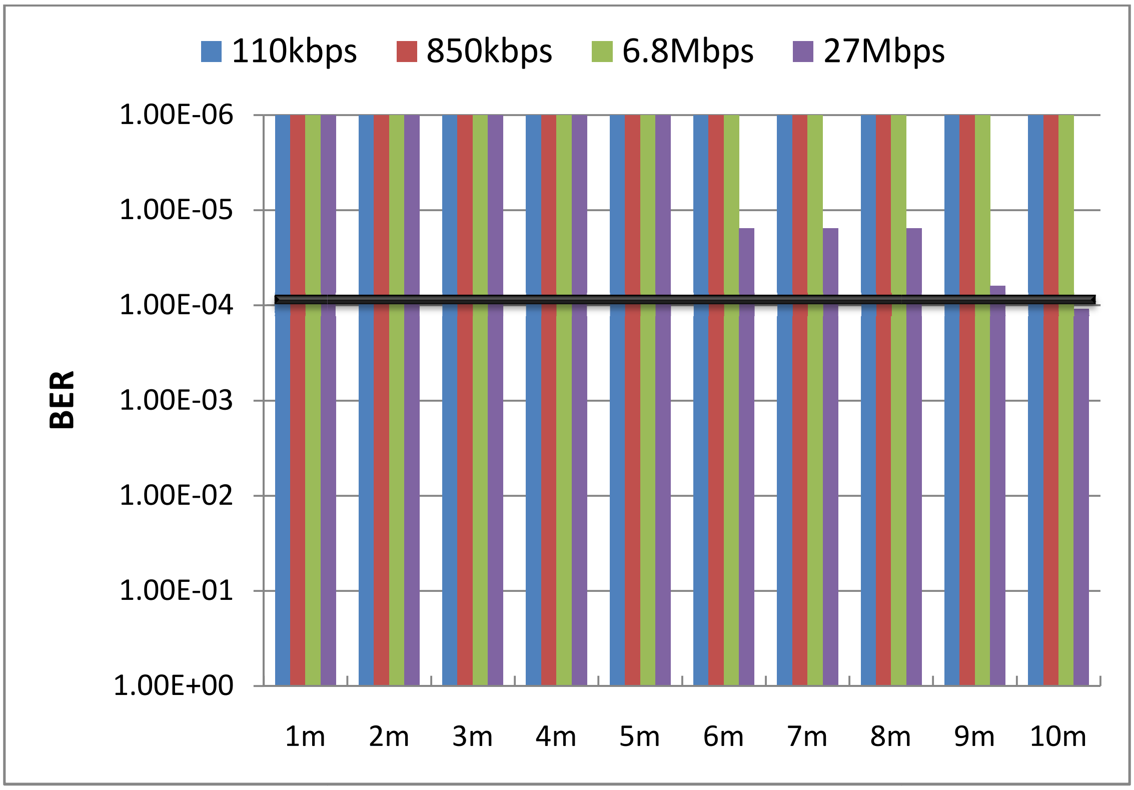


Figure 7-19: Throughputs and distances with BER < 1e-4 (Ghazzemzadeh-NLOS)

From these throughput graphs it is simple to see how a mobile node can store this knowledge and then optimize its communication parameters accordingly. If the node is intelligent it can also learn about its environment and share this knowledge with new nodes to allow the whole sensor network to operate in an optimized fashion.

7.6 PROPOSAL FOR A COGNITIVE AND ADAPTIVE TECHNIQUE

Consider the example network portrayed in Figure 7-20. It consists of a single PAN coordinator node and five sensor nodes classified as Reduced Function Devices (RFDs) by the 802.15.4 MAC. Each sensor node can either be mobile or stationary.

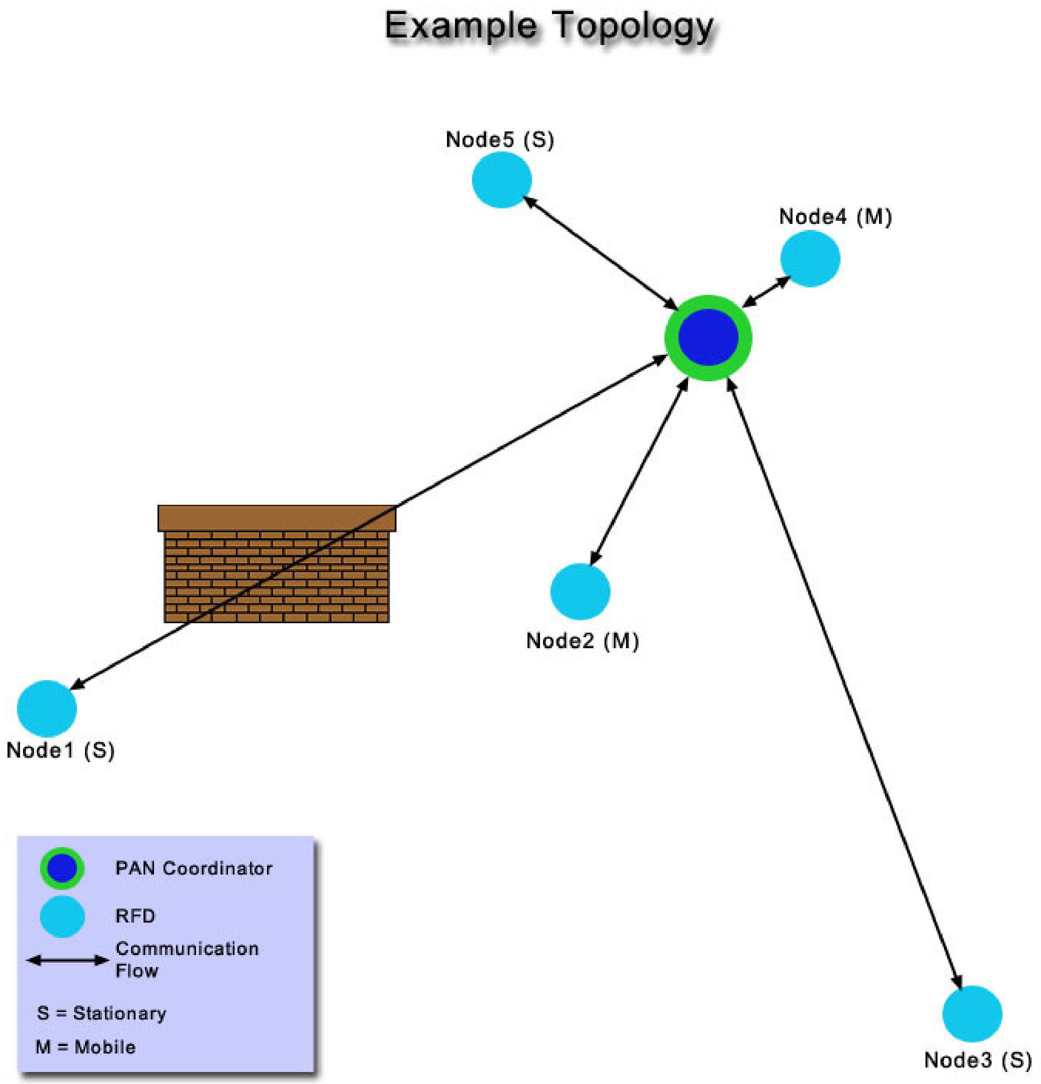


Figure 7-20: Example topology for demonstrating cognitive and adaptive technique

Assume that the radios of all nodes support all the channels and mean PRFs specified by the 802.15.4a UWB physical layer standard. The PAN coordinator node is also equipped with more storage space and higher processing capability.

A PAN coordinator node can be programmed to be aware and intelligent by following a "best performance" scheme according to a preset configuration defining the Quality of Service (QoS) that needed to be maintained.

Such a coordinator node can use its cognitive ability to measure and maintain the performance of each link to each node registered with it. Various performance measurement criteria can be used to specify the desired level of quality for all nodes or each node individually. In the example provided the stationary nodes might have a different function to fulfill than the mobile nodes.

The PAN coordinator node can keep two tables with information about the link to each node. Table 7-7 gives an example.

Table 7-7: Example best performance tables of an intelligent coordinator node

Stationary Nodes Table				
Node Id	Distance	BER	RSS	SNR
1	18m	..	..	..
3	20m	..	..	..
5	4m	..	..	..
Mobile Nodes Table				
Node Id	Distance	BER	RSS	SNR
2	5m	..	..	..
4	1m	..	..	..

Various performance criteria can be used to measure against. Limits to maintain QoS are specified beforehand. Actual configuration parameters to employ are defined by these limits. Previous information about the communication environment and links to each node can be used as well. During the course of communication the specific performance measurement criteria are kept up to date to ensure either optimal throughput or optimal range.

A node registers itself with the PAN coordinator with certain requirements. The coordinator can then make use of the ranging functionality in an 802.15.4a UWB frame to determine the exact distance to the newly registered node and apply the following simple rules as an example:

- If node is stationary, set communication link up for best speed while maintaining QoS requirements.
- Mobile node table must be updated at frequent intervals to ensure the distance to the mobile node is up to date.
- If node is mobile then check if distance to node changed during the previous τ seconds and update the rate of distance change value accordingly.
- If the rate of distance change of the node is greater than some limit α and the node is moving away, then the node link should be set up for optimal range. If the rate of change did not pass the α limit or did and is moving towards the coordinator, then the link should be set up for best speed.

The bit error rate (BER) performance measurement can be used. From the previous results obtained in this chapter a lookup table on the coordinator can be used to ensure the link is set up with the "best performance" configuration parameters. It would even be more intelligent if the coordinator could create such a lookup table itself using a neural network to train itself about its communication environment.

Received signal strength (RSS) measurements can also be used to give an indication about the quality of the link to each node and to determine latency.

Another useful measurement is the signal-to-noise ratio (SNR) which can be obtained by dividing the known transmit power with the current noise power.

If serious data bit rates are required and the nodes are close by, the coordinator can set the link up with FEC disabled, thereby allowing higher throughput because at close distances the BER is so good that FEC is not necessary.

Furthermore, if a coordinator node does not have power constraints it can even talk to each node or different groups of nodes on a different channel. The viability of such hopping between channels will have to be carefully investigated but it does offer some advantages.

An obstacle, as illustrated in the link to Node1, might pose a problem when QoS requirements are very stringent. Take the previous experiments with a Ghassemzadeh-NLOS channel model as an example – it showed that the only way a high BER such as 1e-6 can be guaranteed is when channel 0 is used for communication with PRF of 3.90MHz and bit rate of 110kbps. Clearly if the coordinator could talk to Node1 using these configuration parameters it would be beneficial especially if Node1 were to become mobile and move even further away.

Some nodes might have very precise ranging requirements for which hopping to one of the channels with the higher bandwidth is necessary.

The proposed approach does have its setbacks, particularly with the fact that wireless sensor nodes are deployed in large quantities, have very tight power requirements and processing overheads are usually avoided. Still, it allows for an interesting discourse and something to keep in mind as sensor nodes become more and more powerful.

Chapter 8

Conclusion

This final chapter presents the conclusions drawn from the knowledge obtained out of the various literature studies, software implementations and network simulation experiments performed over the course of this project. The chapter ends with future work contributions that will benefit the research community.

8.1 CONCLUSION

As part of this project an in-depth study into the inner workings of the IEEE 802.15.4a UWB physical layer were done. The resulting work should provide an excellent stepping stone for anyone who requires an introduction into the concepts of UWB and especially the low-rate UWB physical layer specified by the IEEE 802.15.4a standard.

The Mixim-UWB framework proved an excellent choice for simulating the IEEE 802.15.4a UWB physical layer.

Using the modifications made to the Mixim-UWB framework, several simulation results were obtained. From these results the following discoveries were made:

- A higher bit rate results in a higher BER.
- The longer the communication range the higher the BER.
- A high PRF performs better than a lower PRF given certain conditions.
- The channel center frequency effects performance because a higher center frequency are more susceptible to noise and other attenuations.
- A large bandwidth allows more noise to be introduced. It also spreads the power more making interference with other wireless technologies less of a problem. The extra bandwidth also allows for more precise ranging measurements to be done.
- For communication over large distances the FEC does not help much in improving the BER due to the fact that too many errors exist to correct. At lower BERs the FEC are useful and avoids having to retransmit.

- An "intelligent node" approach, where a node has cognitive abilities allowing it to learn about its environment and channel can be very beneficial allowing for communication parameters to be dynamically adapted to best suit current requirements. If data throughput is not a concern, the "intelligent node" can automatically adapt and change its communication parameters to achieve very large communication distances with a low BER. If distance is not a concern and the node is not mobile, the link to the node can be set for optimal data throughput.

8.2 FUTURE WORK\CONTRIBUTIONS

8.2.1 Compliant UWB pulse

The triangular waveform used is not according to specification. More accurate waveforms that meet the FCC power requirements can be investigated.

8.2.2 802.15.4 MAC

The Mixim-UWB model does not implement the MAC as specified by the IEEE 802.15.4 standard. Because Zigbee makes use of the same MAC, a lot of useful research has been done on the 802.15.4 MAC.

Feng Chen [77] implemented an 802.15.4 MAC for OMNET++ using the INET framework. It will be very useful if this can be integrated into the Mixim-UWB model.

8.2.3 PHR

The physical header part can be added to the UWB symbol and the receiver updated to support it. The PHR information will aid an 802.15.4a MAC implementation.

8.2.4 Correlation receiver

Implementation of a correlation receiver for Mixim-UWB simulation purposes will prove very beneficial to research and will make for a good research topic.

8.2.5 Ranging

Ranging as specified by the 802.15.4a standard can also be incorporated into the Mixim-UWB model.

8.2.6 Cognitive UWB Impulse Radio

A very interesting topic if the “intelligent node” employing the “best performance” adaptive techniques mentioned in this document is carefully considered and tested for its viability in wireless sensor networks.

8.2.7 TOSSIM implementation

It would be of great value if the IEEE 802.15.4a UWB PHY layer simulation work discussed and presented in this paper is ported to the TOSSIM environment. This will allow for real time testing of the concepts and algorithms on real hardware motes.

REFERENCES

[1] The Network Simulator - ns-2 [Online], Available: http://www.isi.edu/nsnam/ns/, Last accessed: 30 December 2009

[2] OMNeT++ discrete event simulation environment [Online], Available: http://www.omnetpp.org/, Last accessed: 30 December 2009

[3] NS-2 Ultra Wide-Band (UWB) MAC and PHY simulator [Online], Available: http://uwb.epfl.ch/ns-2/index.html, Last accessed: 30 December 2009

[4] Mixim-UWB framework for OMNET++ [Online], Available: http://github.com/jeromerousselot/mixim-uwb, Last accessed: 30 December 2009

[5] H. G. J. Aitken, *Syntony and Spark: The Origins of Radio*, 1976.

[6] Guglielmo Marconi Biography: The Nobel Prize in Physics 1909 [Online], Available: http://nobelprize.org/nobel_prizes/physics/laureates/1909/marconi-bio.html, Last accessed: 9 March 2008

[7] A Brief History of Naval Radio Communications [Online], Available: http://jproc.ca/rrp/nro_his.html, Last accessed: 9 March 2008

[8] TESLA: Life and Legacy - Who Invented Radio? [Online], Available: http://www.pbs.org/tesla/ll/ll_whoradio.html, Last accessed: 9 March 2008

[9] S. Haykin, *Communication Systems*, 4th ed.: John Wiley & Sons, Inc., 2001.

[10] A brief history of UWB communications [Online], Available: http://www.ferret.com.au/n/A-brief-history-of-UWB-communications-n681131, Last accessed: 9 March 2008

[11] Alereon (Pty) Ltd [Online], Available: http://www.alereon.com, Last accessed: 30 December 2009

[12] Time Domain (Pty) Ltd [Online], Available: http://www.timedomain.com, Last accessed: 30 December 2009

[13] Wisair (Pty) Ltd [Online], Available: http://www.wisair.com, Last accessed: 30 December 2009

[14] Motorola (Pty) Ltd [Online], Available: http://www.motorola.com, Last accessed: 30 December 2009

[15] IEEE 802.15 Working Group for WPAN (WG15) [Online], Available: http://www.ieee802.org/15/, Last accessed: 12 March 2008

[16] IEEE 802.15 WPAN Task Group 1a (TG1a) [Online], Available: http://www.ieee802.org/15/pub/TG1.html, Last accessed: 12 March 2008

[17] IEEE 802.15 WPAN Task Group 2 (TG2) [Online], Available: http://www.ieee802.org/15/pub/TG2.html, Last accessed: 12 March 2008

[18] IEEE 802.15 WPAN Task Group 3 (TG3) [Online], Available: http://www.ieee802.org/15/pub/TG3.html, Last accessed: 12 March 2008

[19] IEEE 802.15 WPAN High Rate Alternative PHY Task Group 3a (TG3a) [Online], Available: http://www.ieee802.org/15/pub/TG3a.html, Last accessed: 12 March 2008

[20] WiMedia Alliance [Online], Available: http://www.wimedia.org/, Last accessed: 30 December 2009

[21] UWB Forum [Online], Available: http://www.uwbforum.org/, Last accessed: 30 December 2009

[22] 802.15.3a is Dead: Long Live UWB [Online], Available: http://www.ultrawidebandplanet.com/resources/article.php/3578521, Last accessed: 28 March 2008

[23] IEEE 802.15 WPAN Task Group 3b (TG3b) [Online], Available: http://www.ieee802.org/15/pub/TG3b.html, Last accessed: 12 March 2008

[24] IEEE 802.15 WPAN Millimeter Wave Alternative PHY Task Group 3c (TG3c) [Online], Available: http://www.ieee802.org/15/pub/TG3c.html, Last accessed: 12 March 2008

[25] IEEE 802.15 WPAN Task Group 4 (TG4) [Online], Available: http://www.ieee802.org/15/pub/TG4.html, Last accessed: 12 March 2008

[26] IEEE 802.15 WPAN Low Rate Alternative PHY Task Group 4a (TG4a) [Online], Available: http://www.ieee802.org/15/pub/TG4a.html, Last accessed: 12 March 2008

[27] IEEE 802.15 WPAN Task Group 4b (TG4b) [Online], Available: http://www.ieee802.org/15/pub/TG4b.html, Last accessed: 12 March 2008

[28] IEEE 802.15 WPAN Task Group 5 (TG5) [Online], Available: http://www.ieee802.org/15/pub/TG5.html, Last accessed: 13 March 2008

[29] IEEE 802.15 WPAN Task Group 6 Body Area Networks (TG6) [Online], Available: http://www.ieee802.org/15/pub/TG6.html, Last accessed: 13 March 2008

[30] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*: John Wiley & Sons, Ltd, 2005.

[31] S. Mahlknecht, "Energy-Self-Sufficient Wireless Sensor Networks for the Home and Building Environment," Master Master Dissertation, Fakultät für Elektrotechnik, Technischen Universität Wien, Vienna, 2004.

[32] S. Meninger, J. O. Mur-Miranda, R. Amirtharajah, A. P. Chandrakasan and J. H. Lang, "Vibration-to-Electric Energy Conversion," *IEEE Transactions on VLSI Systems,* vol. 9, pp. 64-76, 2001.

[33] F. Nekoogar, F. Dowla and A. Spiridon, "Self Organization of Wireless Sensor Networks Using Ultra-Wideband Radios," in *Radio and Wireless Conference, IEEE*, 2004, pp. 451-454.

[34] M. Sharawi and D. Marko, "An Introduction to Direct-Sequence Spread-Spectrum Communications", Oakland University, Rochester, Michigan, 2000.

[35] R. A. Scholtz, "The spread-spectrum concept," *IEEE Transactions on Communications,* vol. COM-25, pp. 748-755, 1977.

[36] R. L. Pickholtz, D. L. Schilling and L. B. Milstein, "Theory of spread-spectrum communications - A tutorial," *IEEE Transactions on Communications,* vol. COM-30, pp. 855-884, 1982.

[37] FCC, "First Report and Order 02-48", February 2002.

[38] J. Foerster, E. Green, S. Somayazulu and D. Leeper, "Ultra-Wideband Technology for Short- or Medium-Range Wireless Communications," *Intel Technology Journal Q2,* pp. 3-5, 2001.

[39] Y. Li, A. F. Molisch and J. Zhang, "Channel Estimation and Signal Detection for UWB," presented at the International Symposium on Wireless Personal Multimedia Communications (WPMC), 2003.

[40] M. Chen and X. Li, "Transmitter-based channel equalization and MUI suppression for UWB systems," presented at the Proceedings of the International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET '04), 2004.

[41] Rules and Regulations in Title 47 of CFR Part 15 [Online], Available: http://www.fcc.gov/oet/info/rules/, Last accessed: 15 March 2008

[42] R. Reinhardt, "A Comparison of Damped Waves and Gaussian Waveforms", Time Domain Systems Inc.

[43] J. Proakis, *Digital Communications*, 4 ed. New York: McGraw-Hill, 2001.

[44] M. Ghavami, L. B. Michael and R. Kohno, *Ultra Wideband signals and systems in communication engineering*, 2 ed.: John Wiley & Sons, Ltd, 2007.

[45] ECMA International, "High Rate Ultra Wideband PHY and MAC Standard," ECMA-368, 2008

[46] ECMA International, "MAC-PHY Interface for ECMA-368," ECMA-369, 2008

[47] Wireless USB from the USB Implementers Forum [Online], Available: http://www.usb.org/developers/wusb/, Last accessed: 1 January 2010

[48] M. Z. Win and R. A. Scholtz, "Impulse Radio: How it works," *IEEE Communications Letters,* vol. 2, 1998.

[49] I. Oppermann, M. Hämäläinen and J. Iinatti, *UWB Theory and Applications*: John Wiley & Sons, Ltd, 2004.

[50] A. Saleh and R. Valenzuela, "A Statistical Model for Indoor Multipath Propagation," *Selected Areas in Communications, IEEE Journal,* pp. 128-137, Feb 1987.

[51] D. S. S. Ghassemzadeh, "The Ultra-wideband Indoor Path Loss Model", IEEE P802.15-02/277r0-SG3a, 2002.

[52] Andreas F. Molisch, Kannan Balakrishnan, Dajana Cassioli, Chia-Chin Chong, Shahriar Emami, Andrew Fort, Johan Karedal, Juergen Kunisch, Hans Schantz, Ulrich Schuster and K. Siwiak, "IEEE 802.15.4a channel model - final report", 04/662r0, 2004.

[53] A. M. Orndorff, "Transceiver Design for Ultra-Wideband Communications,"

Master of Science In Electrical Engineering, Virginia Polytechnic Institute and State University, 2004.

[54] IEEE Computer Society, "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," IEEE Std 802.15.4a, 2007

[55] IEEE Computer Society, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," IEEE Std 802.15.4, 2006

[56] Zigbee Alliance [Online], Available: http://www.zigbee.org/en/index.asp, Last accessed: 30 December 2009

[57] L. D. Nardis and M. G. D. Benedetto, "Overview of IEEE 802.15.4/4a standards for low rate Wireless Personal Data Networks," in *4th Workshop on Positioning, Navigation and Communication*, Hannover, Germany, 2007.

[58] Z. Alliance, "Zigbee Specification - Document 053474r17", 2008.

[59] M. G. D. Benedetto, L. D. Nardis, M. Junk and G. Giancola, "$(UWB)^2$: Uncoordinated, Wireless, Baseborn Medium Access Control for UWB Communication Networks," *Journal On Mobile Networks and Applications,* vol. 10, pp. 663-674, 2005.

[60] IMEC (Pty) Ltd [Online], Available: http://www2.imec.be/be_en/home.html, Last accessed: 30 December 2009

[61] TES-DST Wireless Solutions [Online], Available: http://tes-dst.com/tes-dst/solutions/wireless-systems.html, Last accessed: 7 February 2011

[62] A Collection of Modelling and Simulation Resources on the Internet [Online], Available: http://www.idsia.ch/~andrea/simtools.html#network, Last accessed: 7 April 2008

[63] The Network Simulator - ns-3 [Online], Available: http://www.nsnam.org/, Last accessed: 30 December 2009

[64] Global Mobile Information Systems Simulation Library - GloMoSim [Online], Available: http://pcl.cs.ucla.edu/projects/glomosim/, Last accessed: 30 December

2009

[65] Scalable Wireless Ad hoc Network Simulator [Online], Available: http://jist.ece.cornell.edu/, Last accessed: 30 December 2009

[66] QualNet [Online], Available: http://www.scalable-networks.com/products/qualnet/, Last accessed: 30 December 2009

[67] OPNET [Online], Available: http://www.opnet.com/, Last accessed: 30 December 2009

[68] P. Levis, N. Lee, M. Welsh and D. Culler, "TOSSIM: accurate and scalable simulation of entire TinyOS applications," presented at the Proceedings of the 1st international conference on Embedded networked sensor systems, Los Angeles, California, USA, 2003.

[69] Jonathan G. Page and G. P. Hancke, "Credibility of Routing Information in Wireless Sensor Networks," 2007

[70] T. J. Schriber and D. T. Brunner, "Inside Discrete-Event Simulation Software: How it works and why it matters," in *Proceedings of the 1997 Winter Simulation Conference*, 1997.

[71] A. Varga, "OMNET++ Discrete Event Simulation System User Manual", 2005.

[72] A. Varga, "The OMNET++ Discrete Event Simulation System," in *Proceedings of the European Simulation Multiconference (ESM'2001)*, Prague, Czech Republic, 2001.

[73] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P.T. Klein Haneveld, T.E.V. Parker, O.W. Visser, H.S. Lichte and S. Valentin, "Simulating Wireless and Mobile Networks in OMNeT++ The MiXiM Vision," presented at the OMNeT++ Workshop, Marseille, France, 2008.

[74] Mixim-UWB framework source [Online], Available: http://github.com/jeromerousselot/mixim-uwb, Last accessed: 8 January 2010

[75] Schifra Reed-Solomon Error Correction Library [Online], Available: http://www.schifra.com, Last accessed: 10 January 2010

[76] A tutorial on convolutional encoding with Viterbi decoding [Online], Available: http://home.netcom.com/~chip.f/viterbi/tutorial.html, Last accessed: 10 January

2010

[77] F. Chen and F. Dressler, "A Simulation Model of IEEE 802.15.4 in OMNET++," *Proceedings of 6. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Poster Session, Aachen, Germany,* pp. 35-38, 2007.

[78] ISO/IEC, "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 2: Logical link control ", ISO/IEC 8802-2:1998, 1998.

Addendum A

Source code

This section is intended to provide help on the navigation and arrangement of the software source and additional documentation on the DVD provided with this master dissertation.

IMPORTANT NOTE

All scripts, programs, and scenario files provided in the software DVD have undergone innumerable alterations to suite the simulation requirements. The software is provided "as is". Use at own risk.

Licensing information about the different source components is provided on the disc. Mostly the license terms and conditions are for noncommercial settings – at academic institutions for teaching and research use, and at non-profit research organizations.

A.1 FOLDER STRUCTURE

```
/doc
    /dissertation
        /endnote
        /figures
    /reference_material
/omnetpp-4.0
    /win_install
    /linux_src
    /mixim-uwb-original
    /mixim-uwb-modified
/matlab
/results
    /2nodes
    /10nodes
```

A.1.1 Documentation

`/dissertation`	A copy of this dissertation, its figures and reference table.
`/reference_material`	Electronic copies of some research and conference papers used during literature studies.

A.1.2 OMNET++

`win_install`	Windows installation for OMNET++ v4.0.
`linux_src`	Compressed OMNET++ v4.0 source archive.
`mixim-uwb-original`	Original Mixim-UWB framework before changes was made.
`mixim-uwb-original`	Modified Mixim-UWB with changes discussed in this document.

A.1.3 Matlab

`/.`	Matlab scripts used to generate UWB pulses, pulse trains and frequency response graphs used in this document.

A.1.4 Results

`/2nodes`	Results of 2 nodes scenario.
`/10nodes`	Results of 10 nodes scenario.

Addendum B

Service primitives

This addendum provides a brief overview of the concept of service primitives. Refer to ISO/IEC 8802.2 [78] for more detailed information.

B.1 SERVICES AND PRIMITIVES

A service is a set of functions, known as primitives that a layer provides to the layer above it. To specify a service the information flow between the layers (service users and service providers) are described. The information flow is modeled by discrete, instantaneous events consisting of a service primitive that is passed from one layer to the next through a Service Access Point (SAP). Hence, to describe the service one describe the service primitives and parameters that characterize the service. A service may have one or more primitives and each primitive may have zero or more parameters to convey information required for its service.

Primitives are calling functions that manage communication between adjacent protocol layers within a communication node. Primitives perform various actions and typical examples of primitive names include: Get, Set, Connect, Disconnect, Send, Receive, Listen, Data, and Scan. There are four types of primitives listed in Table B-1.

Table B-1: Four service primitive types

Type	Description
request	Sent by layer (N + 1) to layer N. Sent to request a service and passes any required parameters.
indication	Sent by layer N to layer (N + 1) Sent in return to a (N + 1) request or to indicate an internally N-layer initiated action.
response	Sent by layer (N + 1) to layer N. Sent in reply to an indication. It may acknowledge with the results of an

	action previously invoked by an indication primitive from layer N.
confirm	Sent by layer N to layer (N + 1). Sent to acknowledge with the results of a previous layer (N + 1) request.

The concept of service primitives is illustrated in Figure B-1.

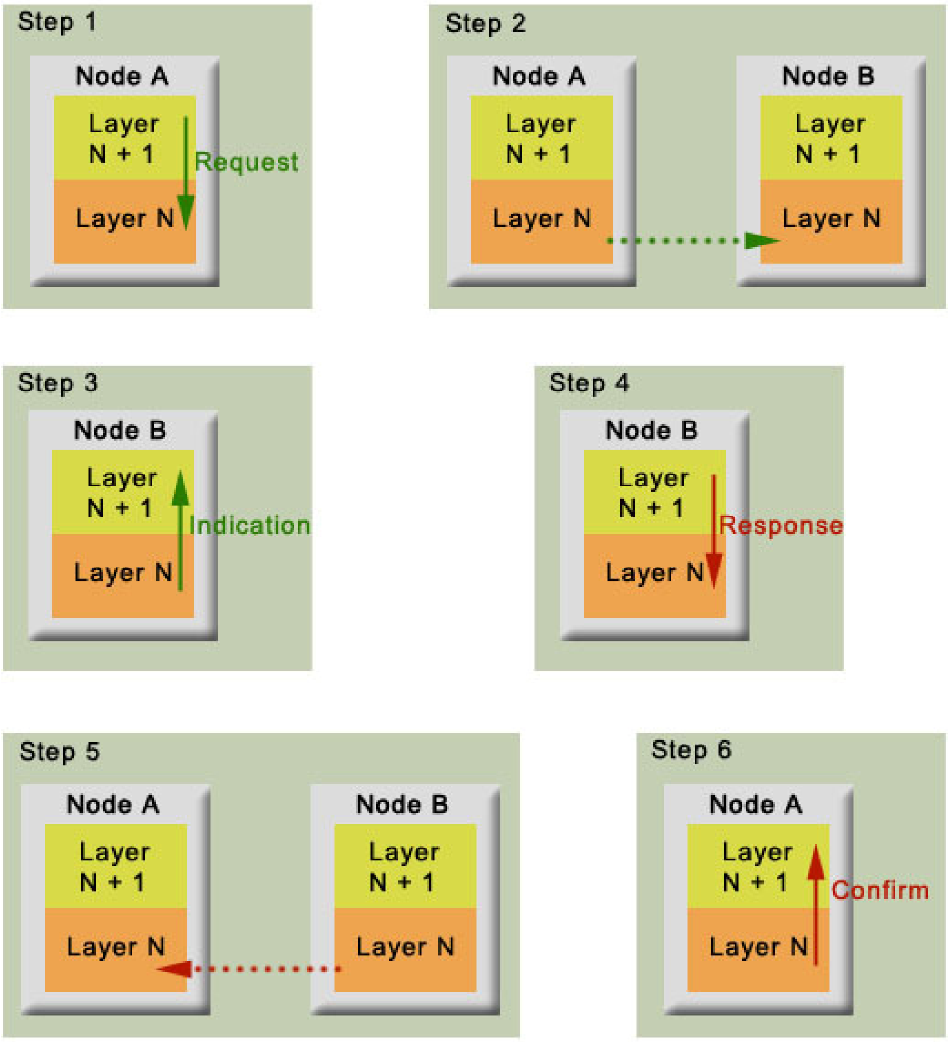


Figure B-1: Communication of primitives between peer protocol entities

A protocol is set of rules determining the format and transmission of data in the form of frames, packets, or messages within a layer. The implementation of the service is defined by the protocol and is only visible to the provider of the service. Service primitives specify only the provided service to the service user and not the implementation thereof.

B.2 DATA UNITS

B.2.1 Service Data Unit

A Service Data Unit (SDU) is the set of data that is sent by a layer to the layer below it. In other words, it is the set of data passed to a layer from the user who makes use of the services provided by that layer. The lower layer does not understand the data in the SDU and treats it as payload.

B.2.2 Protocol Data Unit

Each protocol layer adds to the SDU certain data and additional information that is required for the layer to perform its function, a process known as encapsulation. The SDU received from the higher layer together with this additional information the layer adds to it, constitutes the protocol data unit (PDU) at this layer.

Again, if it is not the lowest layer, this PDU will be passed as a SDU to the next lower layer who does not understand the structured information of the SDU and only delivers it onward. Only the peer layer at the destination will understand the data. The peer layer will reverse the process by decapsulating the data unit it gets from the layer below to extract the information that was added. The information might be a port number, network address, error checking information, etc. the layer needs to carry out its function.

To summarize, the PDU at layer N is the SDU of layer N-1. This SDU at layer N-1 is the payload of the PDU for layer N-1.