Ultrasonic Networking Technologies for the Internet of Implantable and Wearable Things

A Dissertation Presented
by

G. Enrico Santagati

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Engineering

Northeastern University
Boston, Massachusetts

January 2017
Contents

Abstract of the Dissertation iii

1 Introduction 1

2 Challenges and Implications of Using Ultrasonic Communications in Intra-body Area Networks 4
  2.1 System Architecture ......................................................... 4
  2.2 Ultrasonic Communications in Human Tissues .......................... 5
    2.2.1 Fundamentals of Ultrasound Waves ................................ 5
    2.2.2 Attenuation ............................................................. 6
    2.2.3 Ultrasonic Transducers ................................................. 8
    2.2.4 Reflections and Scattering ............................................ 10
    2.2.5 Noise Sources .......................................................... 11
    2.2.6 Bandwidth Evaluation ................................................ 12
  2.3 Medium Access Control Layer Design Issues ............................ 13
  2.4 Network Layer Design Issues ............................................. 15
  2.5 Health Concerns ............................................................. 15
  2.6 Conclusions ................................................................. 16

3 UsWB: Medium Access Control and Rate Adaptation for Ultrasonic Intra-Body Networks 17
  3.1 Ultrasonic Propagation in Human Tissues ................................ 18
  3.2 Channel Modeling ............................................................ 20
  3.3 Ultrasonic Wideband ........................................................ 23
    3.3.1 Physical Layer Model ................................................. 24
  3.4 MAC and Rate Adaptation .................................................. 28
    3.4.1 Distributed Rate-Maximizing Adaptation ........................... 28
    3.4.2 Distributed Energy-minimizing Rate Adaptation ................... 32
    3.4.3 Medium Access Control Protocol .................................... 33
  3.5 Performance Evaluation .................................................... 34
    3.5.1 Multi-scale Simulator ............................................... 34
    3.5.2 Ultrasonic Software-Defined Testbed ................................ 39
  3.6 Conclusions ................................................................. 41

4 Experimental Evaluation of Impulsive Ultrasonic Intra-Body Communications for Implantable Biomedical Devices 42
  4.1 Background ................................................................. 43
    4.1.1 Ultrasonic Intra-Body Communications ................................ 43
    4.1.2 Ultrasonic WideBand .................................................. 44
  4.2 Ultrasonic Testbed Architecture ......................................... 45
  4.3 Architecture of the Communication System ................................ 48
Abstract of the Dissertation

Ultrasonic Networking Technologies for the Internet of Implantable and Wearable Things

by

G. Enrico Santagati

Doctor of Philosophy in Computer Engineering
Northeastern University, January 2017
Dr. Tommaso Melodia, Adviser

This dissertation discusses the design and development of wireless networking technologies for implantable and wearable devices based on ultrasonic communications.

Wirelessly networked systems of implantable and wearable medical devices endowed with sensors and actuators will be the basis of many innovative, sometimes revolutionary therapies. However, biological tissues are composed primarily of water, and radio-frequency (RF) electromagnetic waves, which are the physical basis of currently used wireless technologies like Wi-Fi and Bluetooth, do not propagate well in water and heat body tissues. Additionally, RF communications can be easily jammed or eavesdropped, raising important privacy and security red flags, and a risk for the patient.

Compared to RF waves, ultrasonic waves have significantly lower absorption by human tissues and therefore require less transmission power, resulting in lower energy consumption, in longer battery life and a smaller size for an implantable medical device. Moreover, ultrasonic waves do not easily penetrate through solid materials and do not propagate far in the air; therefore, ultrasonic communication systems are inherently more secure than RF concerning eavesdropping and jamming attacks.

This dissertation will first discuss the development of the Ultrasonic WideBand (UsWB) technology, the first integrated physical and medium access control protocol developed for networks of implantable devices. UsWB is based on the idea of transmitting short information-bearing carrierless ultrasonic pulses, following a pseudo-random adaptive time-hopping pattern with a superimposed spreading code of adaptive length. Impulsive transmission combats the effects of multipath and scattering, e.g., intersymbol interference, caused
by the density and propagation speed inhomogeneity of the human body. The performance of UsWB is tested using a software-defined testbed architecture specifically designed for ultrasonic intra-body networks. The current UsWB implementation can flexibly trade data rate performance for power consumption, and allows multiple concurrent users to coexist and dynamically adapt their transmission rate to channel and interference conditions to maximize throughput while satisfying predefined reliability constraints.

The dissertation will then discuss the development of U-Wear, a software-defined networking framework developed for wearable medical devices based on ultrasonic communications. U-Wear consists of a set of software-defined cross-layer functionalities designed to network ultrasonic wearable devices that offer real-time reconfigurability at different layers of the protocol stack, i.e., the physical (PHY), data link, network and application layer. As a proof of concept, U-Wear was implemented and tested through two prototypes, a wearable ultrasonic sensor node based on a custom hardware platform and a wearable ultrasonic coordinator based on a commercial smartphone device.

This dissertation also presents the design and implementation of the first prototype of an implantable Internet of Medical Things (IoMT) platform that can be used as a base for future wireless smart medical implants employed in a multitude of therapies. The platform consists of a reconfigurable, miniaturized embedded system with a software-defined ultrasonic transceiver that implements in low-power hardware the UsWB ultrasonic networking capabilities. We demonstrate, for the first time, the feasibility of ultrasonic communications using miniaturized and energy constrained embedded devices, and we compare the ultrasonic wireless connectivity offered by the IoMT prototypes against state-of-the-art low-power RF-based wireless technologies, e.g., Bluetooth Low Energy (BLE).
Chapter 1

Introduction

Wirelessly networked systems of implantable medical devices endowed with sensors and actuators will be the basis of many innovative, sometimes revolutionary therapies. Artificial pancreases, i.e., implanted continuous glucose monitors wirelessly interconnected with adaptive insulin pumps, could ease the life of many type-1 diabetic patients. Post-surgery sensors will detect changes in the microenvironment involved with the surgical procedure (pressure, Ph, white cells concentration, LDH enzyme) to prevent infections or ischamias caused by tissue perfusion. At an even smaller scale, wirelessly controlled nanorobots could detect and eliminate malicious agents and cells inside biological tissues, e.g., viruses and cancer cells, enabling less invasive and less aggressive treatments. Other fascinating applications include functional electrical stimulation, a particular type of neurostimulation that tries to restore motion in people with disabilities by injecting electrical currents to activate nerves innervating extremities affected by paralysis; or distributed pacemakers for advanced cardiac resynchronization therapy. Existing as well as futuristic applications of wireless technology to medical implantable (as well as wearable) devices will grow into a new market segment that several analysts are already starting to refer to as “The Internet of Medical Things”.

As of today, existing wireless medical implants [1, 2] are connected through radio frequency (RF) electromagnetic waves. RF-based solutions tend to almost-blindly scale down traditional wireless technologies (e.g., Wi-Fi, Bluetooth) to the intra-body environment, with little or no attention to the peculiar characteristics and safety requirements of the human body. We contend that this may not be the right approach. The human body is in fact composed up to 65% of water, a medium through which radio frequency (RF) electromagnetic waves do not easily propagate. Since tissues absorb the RF waves, much higher transmission power is needed to establish reliable communication links, thus reducing the battery lifetime, or, equivalently, increasing the battery size of an implantable device. Energy efficiency, miniaturization, and battery duration are major concerns in the domain of implantable medical devices, where a surgical procedure is, in most cases, needed to replace the batteries of implants. Additionally, the general public perceives microwaves as potentially dangerous - the World Health Organization classifies RF waves as “possibly carcinogenic to humans”. Last, RF-based technologies raise serious concerns about potential conflicts with existing RF communication
CHAPTER 1. INTRODUCTION

systems that can unintentionally undermine the reliability and security of the intra-body network, and therefore
the safety of the patient [3]. Finally, RF communications can be easily jammed, i.e., intentionally disrupted
by artificially generated interference, or eavesdropped by malicious agents, raising significant privacy and
security red flags, and a risk for the patient.

Given these limitations in this work, we propose and investigate the use of ultrasonic waves as
an alternative carrier of information in human tissues. Ultrasonic waves are acoustic waves with frequency
higher than the upper threshold for human hearing (nominally 20 kHz). Compared to radio-frequency (RF)
electromagnetic waves used in Bluetooth or WiFi, ultrasonic waves have the following key benefits for use in
implantable biomedical devices:

- Significantly lower absorption by human tissues [4] (e.g., 8 – 16 dB for a 10 – 20 cm link at 1 MHz, vs
  60 – 90 dB at 2.45 GHz as used in Bluetooth). As a consequence, tissue heating is much reduced, which
  makes propagation safer for humans. Moreover, lower transmission powers are needed, and therefore
  implantable battery-powered devices may last longer or be smaller in size.

- The FDA also allows much higher intensity for ultrasonic waves (720 mW/cm²) in tissues as compared
to RF (10 mW/cm²), i.e., almost two orders of magnitude higher. Therefore, wireless recharging of
batteries through ultrasonic waves has the potential, with appropriate design, to be orders of magnitude
faster than with RF.

- From a communication theoretic perspective, multi-path is easier to resolve because of the lower
propagation speed of sound; one can build small transducers that operate at low frequencies, and
these are easier to couple to human tissues than RF antennas (which instead need to operate at high
frequencies).

- From a security perspective, ultrasonic communications in tissues cannot be easily eavesdropped or
jammed without physical contact.

- Finally, there are no electromagnetic compatibility concerns with a crowded RF spectrum - in the words
of the FDA, “An increasingly crowded RF environment could impact the performance of RF wireless
medical devices” [3].

The rest of this dissertation is organized as follows. In Chapter 2, we assess the feasibility of using
ultrasonic communications in intra-body networks and discuss the fundamentals of ultrasonic propagation in
tissues, and explore important tradeoffs, including the choice of a transmission frequency, transmission power,
bandwidth, and transducer size. In Chapter 3, we derive a channel model for ultrasonic communications in the
human body, we design and propose a new transmission and multiple access technique, which we refer to
as Ultrasonic WideBand (UsWB) and develop medium access techniques with distributed control to enable
multiple access among interfering implanted devices. In Chapter 4, we present the design and implementation
of a software-defined testbed architecture for ultrasonic intra-body area networks, experimentally demonstrate
the feasibility of ultrasonic communications in human tissues, and evaluate UsWB performance extensively
through a human kidney phantom. In Chapter 5, we propose to use ultrasonic waves to interconnect wearable
devices and present U-Wear, the first software-defined networking framework for wearable medical devices
based on ultrasonic communications. Finally, in Chapter 6, we present the design and implementation of the
first prototype of an implantable Internet of Medical Things (IoMT) device that can be used as a base for future wireless smart medical implants employed in a multitude of therapies. We also demonstrate, for the first time, the feasibility of ultrasonic communications using miniaturized and energy constrained embedded devices, and we compare the ultrasonic wireless connectivity offered by the IoMT prototypes against state-of-the-art low-power RF-based wireless technologies, e.g., Bluetooth Low Energy (BLE).
Chapter 2

Challenges and Implications of Using Ultrasonic Communications in Intra-body Area Networks

In this chapter, we first assess the feasibility of using ultrasonic communications in intra-body networks. We discuss the fundamentals of ultrasonic propagation in tissues and explore important tradeoffs, including the choice of a transmission frequency, transmission power, bandwidth, and transducer size. In addition, we propose a system architecture and discuss future research challenges for ultrasonic networking of in-body devices at the physical, medium access and network layers of the protocol stack.

The rest of the chapter is organized as follows. In Section 2.1 we start off by describing the proposed system architecture. In Section 2.2 we perform a preliminary study on the feasibility of using ultrasonic communications with a focus on frequency selection, transceiver design and noise/attenuation characterization. In Section 2.3 we discuss design issues on medium access in ultrasonic scenarios. In Section 2.4 we discuss design issues at the network layer and relate them to the application requirements. In Section 2.5 we discuss health concerns raised by the propagation of ultrasound in the human body. Finally, in Section 2.6 we conclude the chapter.

2.1 System Architecture

We consider an intra-body network architecture that consists of a set of biomedical wireless sensors deployed inside the human body. As an example, some of these sensors could be implanted into the wall of a patient’s heart through a catheter or be inserted into the body by swallowing them up as pills.

Biomedical nodes communicate through acoustic links with a gateway/actuator node that can be either implanted in the body or outside, for example in a watch or a smartphone carried by the patient.
CHAPTER 2. ULTRASONIC COMMUNICATIONS IN INTRA-BODY AREA NETWORKS

The gateway node can be connected to a remote medical center where the medical personnel can check the patient status and/or the patient data can be stored in a medical database for future use, or be remotely accessed by the patient’s physician.

Also, without any interaction with the external network, the actuator/gateway node could inject into the human body specific drugs. As an example, in diabetic patients, sensors could monitor the level of glucose in the blood and send notifications back to the actuator node upon realizing that the level of glucose is too high. The actuator would then inject insulin, stored in a built-in reservoir and pump, into the blood.

A scheme of the system architecture is illustrated in Fig. 2.1. Typical requirements of such a system are:

- **High reliability.** Data should be delivered to the actuator/gateway node reliably since it is health-related and might trigger, if needed, appropriate actions at the medical center or at the actuator/gateway node itself.

- **Rapid delivery.** Data related to the patient status should be delivered rapidly so that appropriate actions can be undertaken promptly.

- **Reduced Emission Power.** Low-power transmissions are clearly desirable. Also, transmission schemes that avoid overheating specific body areas and instead distribute the load over larger body regions should be preferred.

- **Long Battery Lifetime.** Biomedical sensors should be implanted in the body for years, possibly for decades. Accordingly, mechanisms specifically designed to increase the network lifetime need to be incorporated into all layers of the protocol stack.

- **Small Form Factor.** Devices should be implanted under the skin or introduced into the body and remain there for decades; accordingly, these devices should be small in size and not cause any noise to the patient carrying them.

2.2 Ultrasonic Communications in Human Tissues

In this section, we briefly introduce the basics of ultrasound waves (2.2.1). Then, in Sections 2.2.2, 2.2.3 and 2.2.4 we present the propagation characteristics of ultrasound waves and transducers. In Section 2.2.5 we discuss the effects of noise, and in Section 2.2.6 we provide an analysis of the usable bandwidth.

2.2.1 Fundamentals of Ultrasound Waves

Ultrasounds are mechanical pressure waves with a frequency above the upper limit of human hearing, i.e., 20 kHz. Ultrasounds consist of mechanical vibrations of particles in a material. Even if each particle oscillates around its rest position, the vibration energy propagates as a wave traveling from particle to particle through the material. Acoustic waves in general can be characterized by their physical parameters as follows:
• The frequency, $f$, represents the number of complete oscillations per second for each particle. As stated above, ultrasound waves have a lower bound of 20 kHz.

• The pressure, $P$, is a measure of the compressions and rarefactions of the molecules in the medium through which sound waves propagate, and is typically measured in Pascal [N/m$^2$].

• The amplitude, $A$, represents the displacement of particles from their rest position.

• The propagation speed, $c$, is the rate at which the vibratory energy is transmitted in the direction of propagation and increase with the rigidity of the material. In tissues, we consider an average sound speed of 1540 m/s.

• The intensity, $I$, is the average energy carried over time by a wave per unit area normal to the direction of propagation and is usually measured in mW/cm$^2$. It can be expressed as a quadratic function of pressure as

$$I = \frac{(P_{RMS})^2}{\rho c} \ [W/m^2],$$

(2.1)

where $P_{RMS}$ is the sound pressure root mean square (rms), $\rho$ and $c$ represent the density of the medium and the speed of sound in the medium, respectively.

### 2.2.2 Attenuation

When ultrasound waves propagate through an absorbing medium, the initial pressure, $P_0$, reduces at $P(d)$ at a distance $d$ according to

$$P(d) = P_0 e^{-\alpha d},$$

(2.2)
where \( \alpha \) (in \([\text{np} \cdot \text{cm}^{-1}]\)) is the amplitude attenuation coefficient that captures all the effects that cause a dissipation of energy from the ultrasound beam. This parameter \( \alpha \) is a function of the carrier frequency as \( \alpha = a f^b \) \( \tag{2.3} \)

where \( f \) represents the carrier frequency and \( a \) (in \([\text{np} \text{m}^{-1} \text{MHz}^{-b}]\)) and \( b \) are attenuation parameters characterizing the tissue. Typical experimental values of these parameters in different tissues are reported in Table 2.1 [6].

Based on this, we can compute and plot attenuation as a function of propagation frequency for different mediums. Results for blood are reported in Fig. 2.2.

Since most sensing applications require highly directional transducers, one needs to operate at high frequencies to keep the transducer size small. Conversely, as shown in Fig. 2.2, higher transmission frequencies lead to higher attenuation. Therefore, for effective ultrasonic communications we need to operate at frequencies compatible with small devices, but at the same time limit the maximum attenuation. In Fig. 3.1, we depict the maximum carrier frequency with respect to the distance, for different maximum attenuation values, assuming blood as the propagation medium, and limiting the maximum operating frequency to 1 GHz. Table 3.1 summarizes our results for a 100 dB attenuation. Since the attenuation increases as a function of frequency

![Figure 2.2: Attenuation in the blood as a function of the operating frequency for different values of distance.](image)
and distance, for a fixed value of attenuation we get an inverse relationship between frequency and distance. For communication distances that range from some $\mu m$ up to a few $mm$ (what we refer to as short-range communications) frequencies higher than $1 \text{ GHz}$ are allowed. When distances are larger than $1 \text{ mm}$ but still lower than some $cm$, i.e., for medium range communications, transmission frequencies should be decreased to approximately $100 \text{ MHz}$. For distances higher than a few $cm$, i.e., for long range communications, the transmission frequency should not exceed $10 \text{ MHz}$.

<table>
<thead>
<tr>
<th>Communication range</th>
<th>Distance</th>
<th>Frequency Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Range</td>
<td>$\mu m$ - $mm$</td>
<td>$&gt; 1 \text{ GHz}$</td>
</tr>
<tr>
<td>Medium Range</td>
<td>$mm$ - $cm$</td>
<td>$\simeq 100 \text{ MHz}$</td>
</tr>
<tr>
<td>Long Range</td>
<td>$&gt; cm$</td>
<td>$\simeq 10 \text{ MHz}$</td>
</tr>
</tbody>
</table>

### 2.2.3 Ultrasonic Transducers

To accurately model the ultrasonic communication channel in tissues we need to understand how propagation parameters are affected by the characteristics of the transmitting and receiving devices. An ultrasonic transducer is a device capable of generating and receiving ultrasonic waves. It is primarily made up of an active element, a backing, and a wear plate. The active element is usually a piezoelectric material that converts electrical energy to ultrasonic energy and vice versa. The backing is a very dense material used to absorb the energy radiated from the back face of the piezoelectric element. The wear plate allows to protect the piezoelectric transducer element from the environment and is selected to protect against wear and corrosion.

The acoustic radiation pattern of a transducer is a representation of the transducer sound pressure level as a function of the spatial angle and is related to parameters such as the carrier frequency, size, and shape of the
CHAPTER 2. ULTRASONIC COMMUNICATIONS IN INTRA-BODY AREA NETWORKS

Figure 2.4: Minimum transducer diameter as a function of the operating frequency.

vibrating surface. Furthermore, the acoustic radiation pattern is reciprocal, i.e., its shape is the same whether the transducer is working as a transmitter or as a receiver. Depending on the desired application, transducers can be designed to radiate sound according to different patterns, e.g., omnidirectional or in very narrow beams. For a transducer with a circular radiating surface, the narrowness of the beam pattern is a function of the ratio of the diameter of the radiating surface and the wavelength at the operating frequency, D/λ.

This relationship can be explained by considering the beam spread formulas defined as [7]

\[
\sin \left( \frac{\alpha}{2} \right) = \frac{0.514c}{fD},
\]  

(2.4)

where \( \frac{\alpha}{2} \) is half the angle spread between the -6 dB points.

The higher the frequency or, the larger the transducer’s diameter, the smaller the beam spread.

In Fig. 2.4, we show the minimum diameter of the transducer for different values of \( \alpha \) as a function of the operating frequency. When a transducer is working as a transmitter, it is usually characterized in terms of pressure level. In particular, the sound pressure level \([\text{in dB referred to } \mu\text{Pa}]\) is defined as:

\[
SPL = 20\log \frac{P_{RMS}}{P_{REF}},
\]  

(2.5)

where \( P_{RMS} [\mu\text{Pa}] \) is the sound pressure rms and \( P_{REF} \) is the reference pressure, equal to 1\(\mu\text{Pa} \) in underwater environments. This pressure value can be converted to intensity through eq. (2.1), assuming appropriate values for density and speed of sound in tissues [6].

The main parameter characterizing a transducer used as a receiver of acoustic signals is the sensitivity (\( Sen \)) \([\text{dB re } V/\mu\text{Pa}]\), which can be defined as the voltage output found at the receiver when a specific value of SPL is given, i.e.,

\[
Sen = 20\log \left( \frac{V_{SPL}}{1\text{V}/\mu\text{Pa}} \right),
\]  

(2.6)

\(^{1}\)In the following of this paper “referred to” will be denoted as \( \text{re} \).
CHAPTER 2. ULTRASONIC COMMUNICATIONS IN INTRA-BODY AREA NETWORKS

Figure 2.5: Voltage Output vs. SPL Input, assuming different sensitivity values.

where $V_{SPL}$ is the output voltage given by a specific SPL value. In Fig. 2.5, the I/O relationship for transducers with different sensitivity values are reported. These plots show the output in Volts, given an SPL value in dB re $\mu$Pa, for different sensitivities expressed in dB re Volt/$\mu$Pa.

2.2.4 Reflections and Scattering

Whenever a sound beam encounters a boundary between two materials, some of the energy is reflected, thus reducing the amount of energy that passes through that boundary. The direction and magnitude of the reflected and refracted wave depend on the orientation of the boundary surface and the difference between the different tissues. This mirror-like reflection can be expressed using the Snell law for acoustics, [8].

When an acoustic wave encounters an object that is relatively small compared to the wavelength, or tissue with an irregular surface, a phenomenon called scattered reflection occurs. To describe scattered ultrasonic waves we need to model the statistics of the signal in a very general environment. Therefore, the model must have parameters that take in account tissue characteristics and are simple to compute. A model that satisfies these requirements is based on the Nakagami distribution [9].

If we consider a transmitted signal $s(t)$, the received signal due to scattering effect, assuming a noiseless channel and non-selective slow fading, can be expressed as

$$r(t) = \Re\{\rho e^{j\phi} s(t)\}. \quad (2.7)$$

To characterize the statistical behavior of the received signal, we assume the phase shift $\phi$ to be uniformly distributed in $[0, 2\pi]$ and the magnitude $\rho$ to be Nakagami-distributed. Therefore, the probability density functions of these random variables are expressed as

$$f(\phi) = \frac{1}{2\pi} \operatorname{rect}_{2\pi}(\phi) \quad (2.8)$$
\[ f(\rho) = \frac{2m^m \rho^{2m-1}}{\Gamma(m)\Omega^m} e^{-\frac{m}{\Omega^2}} U(\rho) \] (2.9)

where \( m \) is the Nakagami parameter, \( \Omega \) is a scaling parameter, \( U(\cdot) \) is the unit-step function, \( \Gamma(\cdot) \) is the gamma function, and \( \text{rect}_{2\pi}(\cdot) \) is the rectangular function of duration \( 2\pi \).

Varying the value of the Nakagami parameter, we obtain different distributions such as the Gaussian distribution \((m = 0.5)\), the Rayleigh distribution \((m = 1)\) and the Rician distribution \((m > 1)\). Therefore, the Nakagami amplitude distribution can describe different scenarios simply varying the value of \( m \).

Since the human body is composed of different organs and tissues, each of them with different sizes, densities, and speed of sound, it can be modeled as an environment with a pervasive presence of reflectors and scatterers. Consequently, the received signal is obtained as the sum of numerous attenuated and delayed versions of the transmitted signal, which makes propagation in ultrasonic intra-body networks deeply affected by multipath fading.

### 2.2.5 Noise Sources

No previous study, to the best of our knowledge, has investigated in detail noise sources in tissue propagation. In underwater communications, the ambient noise level is given by the intensity of the ambient noise, expressed in dB re \( \mu\text{Pa} \), and is usually expressed as a sum of different components. Since we are focusing on ultrasonic frequencies, we focus on the thermal noise produced by the agitation of molecules in tissue. As in [10], the power spectral density (PSD) of the thermal noise obtained for an ideal medium using classical statistical mechanics can be adopted with good approximation for a real medium such as body tissues.

Assuming different densities and sound speed for the medium [6], we can derive an expression for the noise PSD in dB re \( \mu\text{Pa} \) per Hz as

\[ N(f) = -15 + 20 \log(f), \] (2.10)

with \( f \) expressed in kHz. Based on (2.10), we can express the system SNR as

\[ \text{SNR}(d, f) = \frac{P/A(d, f)}{N(f)\Delta f}, \] (2.11)

where \( P \) is the transmitted power, \( N(f) \) is the PSD of the noise, assumed to be equal to the PSD of the thermal noise, and \( A(d, f) \) is the attenuation experienced by the transmitted pressure signal. From eq. (2.2), \( A(d, f) \) can be expressed as

\[ A(d, f) = e^{\alpha d}. \] (2.12)

To estimate the impact of frequency and distance on the SNR, we can focus on the expression of \((A(d, f)N(f))^\text{-1}\) only. This quantity is shown in Fig. 2.6 as a function of the transmission frequency, assuming blood as the propagation medium and considering different propagation distances. Since noise and attenuation are monotonically increasing functions of the operating frequency, \((A(d, f)N(f))^\text{-1}\) monotonically decreases with increasing frequency. Furthermore, since attenuation is also a function of the propagation distance, our results show that \((A(d, f)N(f))^\text{-1}\) decreases faster as the link distance increases.
CHAPTER 2. ULTRASONIC COMMUNICATIONS IN INTRA-BODY AREA NETWORKS

Figure 2.6: \((A(d, f)N(f))^{-1}\) when the propagation medium is blood.

2.2.6 Bandwidth Evaluation

Based on this discussion, we can make some preliminary considerations on the bandwidth available for ultrasound communications in tissues. As in [11], we maximize the channel capacity using an optimal energy allocation for a fixed transmission power.

When we maximize the capacity with respect to the power spectral density of the signal denoted as \(S_d(f)\), assuming a finite transmission power \(P\), we obtain an optimal power distribution, and the solution satisfies the water-filling principle:

\[
S_d(f) + A(d, f)N(f) = K_d, \tag{2.13}
\]

where \(K_d\) is a constant depending on the value of \(P\), which is fixed according to the desired SNR value.

Therefore, to achieve maximum capacity through an optimal energy allocation, the total PSD on the channel’s band has to be flat and equal to \(K_d\).

The maximum capacity obtained can be expressed as in [11]

\[
C(d) = \int_{B(d)} \log_2 \left[ 1 + \frac{K_d}{A(d, f)N(f)} \right] df, \tag{2.14}
\]

where \(B(d)\) is the bandwidth that varies depending on the distance \(d\). Assuming now the desired SNR equal to 20 dB, we can compute the desired bandwidth and the maximum capacity using the numerical procedure reported in [11]. In Fig. 2.7 bandwidth and capacity are shown as a function of the propagation distance. In this plot, we assume 1 GHz, 100 MHz and 10 MHz as operating frequencies for the short-, medium- and long-range communications, respectively, according to our results in Section 2.2.2.

Using the appropriate operating frequencies as identified in Table 3.1, the bandwidth and capacity decrease almost linearly in log-scale as distance increases. This shows that high data rates can be reached in case of long distance communications, which provides a strong motivation for seeking to achieve robust and reliable communications through high-bandwidth signaling techniques such as ultra wide band (UWB) and/or code division multiple access (CDMA).
2.3 Medium Access Control Layer Design Issues

As discussed in the previous section, ultrasound propagation in tissues is deeply affected by multipath fading because of the inhomogeneity of the human body in terms of density and, consequently, sound velocity, and the pervasive presence of very small organs and particles. As discussed in Section 2.2.4, reflectors and scatterers accurately model the obstacles encountered by signals propagating in the body. Therefore, numerous attenuated and delayed versions of the same transmitted signal reach the receiver, making detection and decoding a challenging operation. Moreover, the low speed of sound in tissues leads to high delays that have to be considered in system design.

An appropriate medium access control scheme has to allow users to share the communication medium fairly while limiting interference, guaranteeing maximum throughput and minimizing the protocol overhead. Unfortunately, the characteristics of the ultrasonic channel, as illustrated in Section 2.2, make the design of an efficient MAC protocol for in-body communications a challenging task. In particular, the most important aspects that should be taken into account in the design of medium access solutions are:

- **Frequency division.** As discussed in [12], multipath fading strongly affects any narrowband signals. Therefore, an FDMA scheme does not seem to be a desirable solution.

- **Time division.** The long delays, caused by the low speed of sound in the medium, make implementation of an efficient TDMA scheme challenging, since any TDMA scheme would require long time guards to prevent collisions between adjacent time slots.

- **Carrier-sense.** Carrier-sense multiple access (CSMA) schemes would be strongly affected by the long propagation delays in tissues, since carrier sensing becomes inefficient when long delays are considered and even introducing some handshaking mechanisms such as RTS/CTS might have adverse effects on the network throughput.
• **Code Division.** Since the different multipath copies of the signal of interest can be considered as disturbance, the use of spreading codes as in DSSS techniques may help the receiver discriminate among them, thus making DSSS signals resistant to multipath fading.

As observed in Section 2.2.6, considering different ranges of communications, the available bandwidth obtained on the channel is virtually always greater than the center frequency used. Therefore, the ultra wide band (UWB) technique could offer a promising solution. UWB, in its time-hopping impulse radio variant (TH-IR-UWB), consists of transmitting information in very short pulses, which result in a very large bandwidth [13]. Numerous are the potential benefits of using the TH-UWB technique. First, since the duration of the UWB pulse is very short (in the order of hundreds of picoseconds in RF communications), reflecting and scattering do not overlap in time. This means that UWB is multipath-fading resistant, while, as discussed in Section 2.2.4, multipath is one of the major concerns in ultrasonic channels. Finally, as will be discussed in Section 2.5, pulsed transmissions with low duty cycles reduce detrimental thermal and mechanical effects, thus drastically reducing the probability of negative bioeffects caused by heating or cavitation.

To counteract the effects of multipath fading and long propagation delays, integrated physical and medium access solutions should be considered. A very promising approach for medium access, as mentioned above, is direct sequence spread spectrum (DSSS). DSSS consists of using spreading codes to protect the signal of interest from interference. The spreading code is a binary sequence with a rate higher than the information signal rate. The DSSS technique consists in multiplying the information signal by the spreading code before transmission and consequently spreading the signal frequency spectrum into a much wider band. At the receiver side, using the same spreading code of the transmitter, the signal spectrum can be de-spread, and the information signal reconstructed, therefore reducing the effects of interference.

By distinguishing multipath arrivals, which are non-overlapping in UWB or uncorrelated in DSSS, it is possible to combine signal replicas caused by multipath with a time diversity receiver. Moreover, the two solutions presented above - combined or not - naturally enable multiple access to the channel by concurrently transmitting devices, thus allowing different devices to communicate simultaneously without interfering with each other.

Since the pulse duration in UWB is extremely short, multiple access schemes based on time-hopping have been proposed. The time-hopping technique allows users to transmit with a pseudo-random time-schedule and consequently allows them to share the available frequency band with a reasonable level of robustness to collisions. In DSSS communications, code division multiple access schemes are readily obtained using orthogonal or pseudo-orthogonal sets of spreading codes. By associating a spreading code to each communication and de-spreading the received signals with the appropriate codes, users can concurrently transmit on the same portion of the spectrum. A combined use of DSSS and UWB techniques (DS-UWB) may also offer a promising solution for medium...
access in ultrasonic networks. Each user, through a different spreading code, transmits information bits as a sequence of short pulses that depend on the spreading code itself. Since codes are uncorrelated, the transmitted information can be reconstructed by de-spreading the received signals.

2.4 Network Layer Design Issues

Networking issues in intra-body networks are tightly associated to the nature of the information being transported. In fact, data are related to health status and therefore should be delivered reliably and timely; at the same time, body tissues are sensitive to heating and it is imperative to reduce thermal stress and overheating. The main aspects that should be taken into account in designing networking protocols are:

- **Attenuation.** To cope with the high attenuation introduced by the body channel, data forwarding should be redundant so that, eventually, data reach the gateway node. However, redundancy may cause excessive overheating and energy consumption while batteries in biomedical sensors cannot be replaced and devices should be long lasting.

- **Reliability.** To increase the probability that data are received at the actuator/gateway node, redundancy can be employed. This, however, can cause again an increase in energy consumption and heating.

- **Single vs. Multi-hop Communications.** As discussed in the previous sections, due to the high attenuation in ultrasonic scenarios, data cannot travel over a single hop from the biomedical sensors. Instead, multi-hop transmissions are needed. This implies that no centralized routing solutions can be employed but distributed approaches are needed.

- **Area-aware Routing.** To avoid overheating of specific regions, routing should be designed to equalize the overhead traffic on network paths. As an example, in [14] a thermal-aware routing algorithm (TARA) is discussed. In this work, data is routed away from areas characterized by high temperature. Evolutions of this methodology include schemes where a map of nodes’ temperature is created so that data forwarding can be obtained by selecting minimum-temperature routes [15]. However, a disadvantage of such schemes is that overhead associated with this exchange of information could stress the intra-body network significantly. Taking inspiration from the literature on QoS routing in ad hoc networks [16], the QoS parameter of interest could be identified with the energy consumption, and paths where biomedical devices have higher residual energy or the traversed areas are less sensitive to increase in temperature can be selected for forwarding. Again, maintenance of these paths requires that nodes exchange some signaling information which, on the other hand, could increase the power consumption.

2.5 Health Concerns

Whenever any form of energy is introduced into the body, it is important to understand what risks might result from applying energy to internal tissues. In this section, we briefly summarize the main effects of
the propagation of ultrasonic waves in human tissues [17].

The most obvious effect is *heating*. Since a significant portion of the energy is absorbed and converted into heat during ultrasound propagation, this could potentially lead to a temperature increase. As the wave intensity is increased, temperature rises and if the temperature becomes higher than 38.5° C, adverse biological effects may occur. However, no harmful effects have been observed for temperatures lower than 41° C. Since heating is strictly caused by the wave intensity, a pulsed transmission with a low duty cycle can potentially reduce this effect of a factor proportional to the duration of the duty cycle.

Another important effect caused by ultrasound wave propagation is the so-called *cavitation* which denotes the behavior of gas bubbles within an acoustic field. Pressure variations of the ultrasound wave cause bubbles present in the propagation medium to contract and expand. For large pressure variations, the bubble size drastically increases, reaching an expansion peak when pressure is minimum and then collapsing when pressure reaches its peak. During this process, the internal pressure and temperature in the bubble can reach high values causing serious biological effects and damaging the objects located in closest proximity. It can be shown that cavitation is a frequency-dependent phenomenon. Since higher frequencies lead to shorter pressure oscillations, the time for bubble expansion is restricted and the cavitation effect tends to disappear. Pulsed transmissions may also reduce the cavitation effects. In fact, during the off period, the bubbles assume again their initial sizes without imploding. However, these destructive effects are not seen for low pulse intensities.

Unfortunately, data collected on bioeffects of ultrasounds are frequently inconsistent and controversial. However, based on the medical ultrasound experiences of the last decades, no dangerous bioeffects have been observed as long as the energy provided to the tissues is less than 50 J/cm², [18]. Therefore, ultrasound communications in tissues at low transmission pressure levels, and consequently low transmission power levels, are not expected to cause any harmful bioeffects. We thus believe that ultrasonic communications can represent a feasible alternative to traditional electromagnetic RF communications.

### 2.6 Conclusions

In this Chapter, we assessed the feasibility of using ultrasonic communications in intra-body BANs. We discussed the fundamentals of ultrasonic propagation in tissues and explored important system tradeoffs, including the choice of a transmission frequency, transmission power, bandwidth, and transducer size. We discussed a system architecture and outlined future research challenges for ultrasonic networking of in-body devices at the physical, medium access and network layers of the protocol stack. Significant work is needed at the physical, medium access and network layers to make ultrasonic communications feasible in realistic network scenarios.
Chapter 3

UsWB: Medium Access Control and Rate Adaptation for Ultrasonic Intra-Body Networks

In this chapter, we lay our foundation on the fundamental physics of ultrasound propagation and move up layers of the protocol stack with the following core contributions:

- **Feasibility of Ultrasonic Communications in the Human Body.** We start off by assessing the feasibility of using ultrasounds for intra-body communications. We discuss fundamental aspects of ultrasonic propagation in tissues, and explore important tradeoffs, including the choice of a transmission frequency, transmission power, bandwidth, and transducer size.

- **Ultrasonic Channel Modeling.** We then proceed to derive a channel model for ultrasonic communications in the human body. We focus on propagation of ultrasounds at the wave level to obtain a model of the channel impulse response. We find that the inhomogeneity of the human body, characterized by a multitude of small organs and tissues, causes severe multipath effect. We observe that reflectors and scatterers cause numerous delayed versions of the transmitted signal to reach the receiver, thus making detection and decoding a challenging operation. In addition, we observe that the low speed of sound in tissues leads to high delays that need to be considered in system design.

- **Ultrasonic Wideband Design.** To address the severe effect of multipath, we design and propose a new transmission and multiple access technique, which we refer to as Ultrasonic WideBand (UsWB). Ultrasonic wideband is based on the idea of transmitting very short pulses following an adaptive time-hopping pattern. The short impulse duration results in limited reflection and scattering effects, and the low duty cycle reduces thermal and mechanical effects that are detrimental for human health.
variable length spreading code is superimposed to the time-hopping pattern to further combat the effect of multipath and to introduce waveform diversity among interfering nodes.

- **Ultrasonic Wideband Adaptation and Multiple Access Control.** We study and develop medium access techniques with distributed control to enable multiple access among interfering implanted devices. The proposed scheme is based on the idea of regulating the data rate of each transmitter to adapt to the current level of interference by distributively optimizing the code length and duration of the time hopping frame. We show how the proposed distributed algorithm is effective in adapting the network throughput to the current level of interference based on minimal and localized information exchange.

- **System Performance Evaluation.** We evaluate the proposed scheme through a multi-scale simulator that evaluates our system at three different levels, i.e., (i) at the wave level by modeling ultrasonic propagation through reflectors and scatterers, (ii) at the bit level by simulating in detail the proposed ultrasonic transmission scheme, (iii) at the packet level by simulating networked operations and distributed control and adaptation to evaluate metrics such as network throughput and packet drop rate. We also validate simulation results through experiments obtained through a software-defined ultrasonic testbed.

The rest of the chapter is organized as follows. In Sections 3.1 and 3.2 we discuss fundamental aspects of ultrasonic physical propagation in human tissues and channel modeling. In Section 3.3 we outline the design of UsWB and in Section 3.4 we illustrate the proposed medium access control protocol. Performance evaluation results are discussed in Section 3.5 along with tested validation of the simulation results. Finally, in Section 3.6 we conclude the chapter.

### 3.1 Ultrasonic Propagation in Human Tissues

Ultrasonic waves originate from the propagation of mechanical vibrations of particles in an elastic medium at frequencies above the upper limit for human hearing, i.e., 20 kHz. Acoustic propagation through a medium is governed by the acoustic wave equation (referred to as the Helmholtz equation), which describes pressure variation over the three dimensions,

\[
\nabla^2 P - \frac{1}{c^2} \frac{\partial^2 P}{\partial t^2} = 0,
\]

where \( P(x, y, z, t) \) represents the acoustic pressure scalar field in space and time, and \( c \) is the propagation speed in the medium. The Helmholtz equation can be derived from the continuity equation, the force equation and the equation of state [8].

When ultrasonic waves propagate through an absorbing medium, the initial pressure, \( P_0 \) reduces to \( P(d) \) at a distance \( d \) [5] as

\[
P(d) = P_0 e^{-\alpha d},
\]

where \( \alpha \ (\text{[Np} \cdot \text{cm}^{-1}]) \) is the amplitude attenuation coefficient that captures energy dissipation from the ultrasonic beam and is a function of the carrier frequency \( f \) as \( \alpha = af^b \ [5] \), where \( a \ (\text{[Np} \cdot \text{m}^{-1} \text{MHz}^{-b}]) \) and \( b \) are tissue attenuation parameters.
CHAPTER 3. MAC AND RATE ADAPTATION FOR ULTRASONIC INTRA-BODY NETWORKS

Table 3.1: Frequency Limits for $A=100\ dB$

<table>
<thead>
<tr>
<th>Communication Range</th>
<th>Distance</th>
<th>Frequency Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Range</td>
<td>$\mu m$ - $mm$</td>
<td>$&gt;1GHz$</td>
</tr>
<tr>
<td>Medium Range</td>
<td>$mm$ - $cm$</td>
<td>$\simeq 100MHz$</td>
</tr>
<tr>
<td>Long Range</td>
<td>$&gt;cm$</td>
<td>$\simeq 10MHz$</td>
</tr>
</tbody>
</table>

Figure 3.1: Operating frequency as a function of the propagation distance, with different values of the tolerable attenuation.

In [4], we observed that the attenuation level in tissues can be significant, and it increases with the distance between transmitter and receiver. Moreover, as discussed in [7], the beam spread is inversely proportional to the ratio between the diameter of the radiating surface and the wavelength corresponding to the operating frequency [4]. Consequently, since most biomedical sensing applications require directional transducers, one needs to operate at high frequencies to keep the transducer size small. However, as expected, higher frequencies lead to higher attenuation.

In Fig. 3.1, we show the maximum “allowed” carrier frequency with respect to distance, for different values of the maximum tolerable attenuation. We consider blood as the propagation medium and limit the maximum operating frequency to 1 GHz. In Table 3.1, we summarize our findings [4] on the maximum “allowed” carrier frequency for a 100 dB maximum tolerable attenuation. For distances ranging between some $\mu m$ up to a few $mm$ (i.e., short range communications) frequencies higher than 1 GHz can be used. When distances are higher than 1 $mm$ but still lower than some $cm$, i.e., for medium range communications, the transmission frequency should be decreased to approximately 100 MHz. For distances higher than a few $cm$, i.e., for long range communications, the transmission frequency should not exceed 10 MHz. Similar results in [4] are independently reported in [19].
3.2 Channel Modeling

The first step towards the design of high-performance ultrasonic intra-body networks is to characterize the communication channel in tissues - for which, unfortunately, there is no literature available to date, except for [20] where the authors present a simple two-dimensional statistical model. Here, we derive a deterministic channel model based on acoustic wave propagation theory.

Propagation of acoustic waves through biological tissues is governed by three coupled first-order equations, i.e., the continuity equation, the force equation and the equation of state [8], which represent relationships among acoustic pressure, $P$, acoustic particle velocity $u$, and medium density $\rho$, and can be rearranged to obtain the Helmholtz equation, in Section 3.1. A realistic model of ultrasonic propagation in human tissues that incorporates attenuation, scattering and multipath effect needs to satisfy the three above equations simultaneously.

Traditionally, partial differential equations are solved using numerical methods such as the finite-difference-method (FDM). We take a different, computationally more efficient approach, based on the pseudo-spectral and k-space methods [21]. Basically, the pseudo-spectral (PS) method reduces the computational complexity in the spatial domain by using Fourier series expansions and FFTs, while the k-space method operates in the time domain by using k-space propagator functions (instead of classical finite differences) to approximate temporal derivatives. As a consequence, larger time steps can be used, reducing the simulation time with controllable accuracy. k-Wave [22] is a powerful tool implementing this method.

We modeled a reference ultrasonic propagation channel in the human body to derive an accurate characterization of the channel impulse response, which was then used in PHY and MAC layer simulation studies. Specifically, we first modeled a section of the human arm, including bones, muscles, fat and skin. We considered a heterogeneous two-dimensional rectangular area of length 20 cm and width 10 cm, where density and sound velocity are distributed as shown in Fig. 3.2. The bone half-section is 18 mm wide, the muscle
 CHAPTER 3. MAC AND RATE ADAPTATION FOR ULTRASONIC INTRA-BODY NETWORKS

Figure 3.3: Snapshots of the time evolution of the acoustic field in the modeled human arm.

half-section is 22 mm wide, while fat and skin have a half-section 7 mm and 3 mm wide, respectively. The medium parameters, sound speed $c$, density $\rho$ and attenuation $a$ are reported in Table 3.2 [6]. The attenuation parameter $b$ is set to 1 for all considered tissues.

Table 3.2: Tissues parameters.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>$c$ [m/s]</th>
<th>$\rho$ [Kg/m$^3$]</th>
<th>$a$ [Np $\cdot$ cm$^{-1}$MHz$^{-b}$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bone</td>
<td>2407</td>
<td>1920</td>
<td>1.8</td>
</tr>
<tr>
<td>Muscle</td>
<td>1629</td>
<td>1056</td>
<td>0.125</td>
</tr>
<tr>
<td>Fat</td>
<td>1487</td>
<td>939</td>
<td>0.093</td>
</tr>
<tr>
<td>Skin</td>
<td>1729</td>
<td>1190</td>
<td>0.212</td>
</tr>
</tbody>
</table>

We obtained the channel impulse response by releasing an omnidirectional ideal Dirac pulse in the left top corner of the muscle section. The receiver is located at the left bottom corner of the muscle section. Transmitter and receiver are located 20 cm away from one another. In Fig. 3.3, we show snapshots of the acoustic field time propagation in the modeled environment. We observe that the effect of the bone and tissues is to partially reflect and scatter the acoustic wave transmitted by the source. We then obtain the channel impulse response by recording the time series of the signal at the receiver sensor.

Figure 3.4 (top) reports the resulting impulse response in this scenario. We observe that multipath and scattering effects introduce attenuated signal replicas spaced in time. Because of the very short duration of the transmitted pulses, replicas do not interact destructively, which provides a strong motivation for the transmission scheme proposed in Section 3.3. Based on this, we can model the channel response as a complex-valued low-pass equivalent impulse response, as $h(\tau,t) = \sum_{k=1}^{K} a_k(t)\delta(\tau - \tau_k)e^{j\theta_k(t)}$ where $\delta$ is the Dirac delta function, $K$ is the number of resolvable multipath components, $\tau_k$ is the delay of the multipath components, $a_k$ is the path amplitude value and $\theta_k$ is the path phase value. With this description, we can characterize the channel through the mean excess delay ($\tau_m$) and the RMS delay spread ($\tau_{RMS}$). For the channel simulated above, we obtained $\tau_m = 4.4353 \cdot 10^{-6}s$ and $\tau_{RMS} = 2.3389 \cdot 10^{-6}s$. Since the coherence
CHAPTER 3. MAC AND RATE ADAPTATION FOR ULTRASONIC INTRA-BODY NETWORKS

Figure 3.4: Resulting ultrasonic normalized channel impulse response in the modeled 2D human arm (top) and 3D human kidney (bottom).

The bandwidth of the channel is proportional to the inverse of $\tau_{RMS}$, we should consider this channel as frequency selective for signals of bandwidth above approximately 85 kHz [23].

We also modeled a three-dimensional human kidney enclosed in a background medium with similar density and speed of sound. This model accurately reproduces the characteristics of a commercial ultrasonic phantom that mimics the propagation characteristic of human kidney [24]. The background medium has very low attenuation coefficient, i.e., it is almost transparent to ultrasonic propagation. The medium parameters, sound speed $c$, density $\rho$ and attenuation $a$ are as in Table 3.3. In Fig. 3.5, we show the density distribution in the three section planes, with each plane crossing the kidney in the middle. We repeated similar simulations considering a three-dimensional setup, and assuming that the pulse transmitter is located at the top side of the kidney in the x-y plane, and in the middle of the x-z and y-z planes. In Fig. 3.4 (bottom), we show the result of the three-dimensional pulse propagation. We observe that since the medium is more homogeneous than in the previous case because of the absence of bones, the channel impulse response shows a reduced number of pulse replicas and a more regular attenuation pattern. By characterizing the channel response as a complex-valued low-pass equivalent impulse response, we obtain $\tau_m = 8.8256 \times 10^{-7}$ s and $\tau_{RMS} = 1.3924 \times 10^{-6}$ s. The coherence bandwidth of the channel is in this case 140 kHz.

Table 3.3: Tissue parameters.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>$c$ [m/s]</th>
<th>$\rho$ [Kg/m$^3$]</th>
<th>$a$ [Np · cm$^{-1}$MHz$^{-b}$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kidney</td>
<td>1550</td>
<td>1030</td>
<td>0.046</td>
</tr>
<tr>
<td>Background Gel</td>
<td>1550</td>
<td>1020</td>
<td>0.011</td>
</tr>
</tbody>
</table>
CHAPTER 3. MAC AND RATE ADAPTATION FOR ULTRASONIC INTRA-BODY NETWORKS

Figure 3.5: Density distributions in the modeled human kidney.

### 3.3 Ultrasonic Wideband

Based on the considerations summarized in Sections 3.1 and 3.2, we design and propose a new ultrasonic transmission and multiple access technique, which we refer to as Ultrasonic WideBand (UsWB). The key design objectives of UsWB are:

- to enable low-complexity and reliable communications in ultrasonic channels against the effect of multipath reflections within the human body;
- to limit the thermal effect of communications, which is detrimental to human health;
- to enable distributed medium access control and rate adaptation to combat the effect of interference from co-located and simultaneously transmitting devices.

UsWB is jointly designed to provide physical layer functionalities and medium access control arbitration and adaptation to enable multiple concurrent co-located transmissions with minimal coordination.

**Ultrasonic Pulsed Transmissions.** Ultrasonic wideband is based on the idea of transmitting very short ultrasonic pulses following an adaptive time-hopping pattern together with a superimposed adaptive spreading code. Baseband pulsed transmissions enable high-data rate, low-power communications, low-cost transceivers, and have been proposed for RF short-range, high-data rate communications [13, 25], although with much shorter pulse durations (and consequently larger bandwidth) than achievable in ultrasonic communications.

The characteristics of pulsed transmissions appear to ideally address all the requirements discussed above. Their fine delay-resolution properties are well-suited for propagation in the human body, where inhomogeneity in terms of density and propagation speed, as well as the pervasive presence of very small organs and particles, cause dense multipath and scattering. When replicas of pulses reflected or scattered are received with a differential delay at least equal to the pulse width, they do not overlap in time with the original pulse. Therefore, for pulse durations in the order of hundreds of nanoseconds [26], pulse overlaps in time are reduced and multiple propagation paths can be efficiently resolved and combined at the receiver to reduce the bit error rate. Also, the low duty cycle of pulsed transmissions reduces the impact of thermal and mechanical effects, which are detrimental for human health [27]. Observe also that the large instantaneous bandwidth
enables fine time resolution for accurate position estimation [28] and network synchronization. Finally, carefully designed interference mitigation techniques may enable MAC protocols that do not require mutual temporal exclusion between different transmitters. This is crucial in the ultrasonic transmission medium since the propagation speed is about 1500 m/s and consequently the propagation delay is five orders of magnitude higher than in RF in-air channels (where the propagation speed is about $3 \cdot 10^8$ m/s) and carrier-sense-based medium access control protocols are ineffective [28]. In addition, data rate can be flexibly traded for power spectral density and multipath performance.

### 3.3.1 Physical Layer Model

**Adaptive Time-Hopping.** Consider a slotted time divided in chips of duration $T_c$, with chips organized in frames of duration $T_f = N_h \cdot T_c$, where $N_h$ is the number of chips per frame. Each user transmits one pulse in one chip per frame, and determines in which chip to transmit based on a pseudo-random time hopping sequence (THS), i.e., a sequence generated by seeding a random number generator with the user’s unique ID.

The train of pulses is modulated based on pulse position modulation (PPM), i.e., a ‘1’ symbol is carried by a pulse delayed by a time $\delta$ with respect to the beginning of the chip, while a ‘0’ symbol begins with the chip. The signal $s^{(k)}(t, i)$ generated by the $k^{th}$ user to convey the $i^{th}$ symbol is expressed as

$$s^{(k)}(t, i) = p(t - c^{(k)}_i T_c - i T_f - d^{(k)}_i \delta),$$

where $p(t)$ is the second derivative of a Gaussian Pulse, $\{c^{(k)}_i\}$ is the time hopping sequence of the $k^{th}$ source, with $0 \leq c^{(k)}_i \leq N_h - 1$, and $\{d^{(k)}_i\}$ is the information-bearing sequence, $d^{(k)}_i \in \{0, 1\}$. The resulting data rate, in pulses per second, is expressed as:

$$R(N_h) = \frac{1}{T_f} = \frac{1}{N_h T_c}.$$  

By regulating the time-hopping frame length $N_h$, i.e., the average inter-pulse time, a user can adapt its transmission rate, and as a consequence modify the average radiated power and therefore the level of interference generated to other ongoing communications. We observe that an individual user has little incentive to increase its frame size, since that results in a lower achievable data rate, without any major benefit for the user itself (since the level of interference perceived depends primarily on the frame length of the other users, and not on its own). However, a longer time frame reduces the interference generated to the other users. Therefore, selfish/greedy frame adaptation strategies do not work well in this context and cooperative strategies are needed.

At the receiver, packet synchronization and “time hopping” synchronization must be performed to properly decode the received signal. Packet synchronization consists of finding the correct time instant corresponding to the start of an incoming packet at the receiver. In general, this can be achieved through an energy-collection approach. During the packet synchronization, the transmitter sends an a-priori-known sequence, i.e., a preamble. After correlating the received signal and the expected signal, the receiver identifies
the starting point of the packet as the time instant where the correlation is maximized. The second step consists of finding the time-hopping sequence to hop chip-by-chip and correlate the received pulses. This can be achieved by seeding the random generator with the same seed used by the transmitter, and therefore generating the same pseudo-random time-hopping sequence. More details are given in Section 3.4.3. Once both synchronization processes have been accomplished, the receiver can decode the received signal by “listening” in the time chips of interest and by correlating the received pulse according to the modulation scheme in use.

**Adaptive Channel Coding.** An adaptive channel code [29] can further reduce the effect of mutual interference from co-located devices, by dynamically regulating the coding rate to adapt to channel conditions and interference level. Various channel coding solutions have been proposed [13, 29, 30, 31, 32] with different performance levels and computational complexity. We rely on simple pseudo-orthogonal spreading codes because of their excellent multiple access performance, limited computational complexity, and inherent resilience to multipath. Each bit is spread by multiplying it by a pseudorandom code before transmission. At the receiver side, with prior knowledge of the code used at the transmitter, the signal can be de-spread, and the original information recovered. With different pseudo-orthogonal codes, multiple nodes can transmit simultaneously on the same portion of the spectrum, with reduced interference. We explored two alternative modulation schemes, which we refer to as PPM-BPSK-spread and PPM-PPM-spread.

In PPM-BPSK-spread, the binary spreading code at the $k^{th}$ node, $\{a_j^{(k)}\}$, is defined as a pseudorandom code of $N_s$ chips with $a_j \in \{-1, 1\}$. Accordingly, the information bit is spread using BPSK modulated chips, and by combining with time hopping, (3.3) can be rewritten as

$$s^{(k)}(t, i) = \sum_{j=0}^{N_s-1} a_j^{(k)} p(t - c_j^{(k)}T_c - jT_f - d_i^{(k)}) \delta,$$  

(3.5)

where $\delta$ is the PPM displacement of a pulse representing a ‘1’ bit, while chip information is carried in the pulse polarity.

In PPM-PPM-spread, the information bit is spread using PPM-modulated chips. In this case, the binary spreading code can be defined as a pseudorandom code of $N_s$ chips with $a_j \in \{-1, 1\}$. With time hopping, (3.3) can be rewritten as

$$s^{(k)}(t, i) = \sum_{j=0}^{N_s-1} p(t - c_j^{(k)}T_c - jT_f - \frac{a_j^{(k)} d_i^{(k)}}{2} + \frac{1}{2} \delta).$$  

(3.6)

In Fig. 3.6, we show an example of a combined time hopping and PPM-BPSK-spread coding strategy. Since the spreading operation associates $N_s$ chips with one information bit, the information rate will be further reduced by a factor $N_s$, i.e.,

$$R(N_h, N_s) = \frac{1}{N_s T_f} = \frac{1}{N_s N_h T_c},$$  

(3.7)

while the energy required for transmitting one bit is increased by a factor $N_s$. Note that there is a tradeoff between robustness to multi-user interference (which increases with longer spreading codes), and energy consumption and information rate. In Section 3.4, we discuss joint dynamic adaptation of frame and code length.
Figure 3.6: Example of two ongoing transmissions PPM-BPSK-spread (top) and PPM-PPM-spread (bottom) with $d = 1$, $N_h = 6$, $N_s = 3$, using time hopping sequences $TH_1 = \{3, 2, 1\}$ and $TH_2 = \{0, 5, 4\}$ and spreading codes $SC_1 = \{1, 1, -1\}$ and $SC_2 = \{1, -1, -1\}$.

In both cases, PPM-PPM-spread and PPM-BPSK-spread, the receiver can use the spreading code employed at the transmitter to obtain the correlator template. With BPSK-modulated chips, the absolute received phase information is needed for decoding. Therefore, a coherent receiver with accurate channel knowledge is needed. Instead, with a pure PPM-modulated signal, a simple non-coherent energy detector receiver is sufficient. The latter requires frame synchronization only, and its hardware complexity is significantly lower [33].

**Signal to Interference-plus-Noise Ratio.** We can express the signal to interference-plus-noise ratio (SINR) for impulsive transmissions at the receiver of link $i$ as [13]

$$\text{SINR}_i(N_h, N_s) = N_s, i \frac{P_i g_{i,i} N_h, i T_c}{\eta + \sigma^2 T_c \sum_{k \in I_i} P_k g_{k,i}} ,$$

where $P_i$ is the average power per pulse period emitted by the $i^{th}$ transmitter, $g_{i,j}$ is the path gain between the $i^{th}$ transmitter and the $j^{th}$ receiver, $\eta$ represents background noise energy and $\sigma^2$ is an adimensional parameter that depends on the shape of the transmitted pulse and the receiver correlator. The set $I_i$ represents the set of links whose transmitter interferes with the receiver of link $i$. Recall that $T_f = N_h T_c$ is the frame duration.

When different links use different frame lengths, (3.8) becomes

$$\text{SINR}_i(N_h, N_s) = N_s, i \frac{P_i g_{i,i} N_h, i T_c}{\eta + \sigma^2 T_c \sum_{k \in I_i} \frac{N_s, i}{N_h, k} P_k g_{k,i}} .$$

The expression in (3.9) depends on the array of frame and code lengths of all the ongoing communications in the network, i.e., $N_h, N_s$, whose $i^{th}$ elements are $N_{h,i}$ and $N_{s,i}$, respectively. The term $\frac{N_{h,i}}{N_{h,k}}$ accounts for the level of interference generated by each interferer $k$ to the receiver of link $i$, i.e., the number of pulses transmitted by the $k^{th}$ transmitter during the time frame of the $i^{th}$ user. Note that we will not consider power control strategies in our treatment, since [25] showed that in the linear regime, when the objective is to maximize the aggregate data rate, the optimal solution always corresponds to points where individual devices transmit at the maximum power, or do not transmit at all.

Note that increasing (or decreasing) the spreading code length of the node of interest, $N_{s,i}$, leads to an increase (decrease) in the SINR. When the node of interest increases (decreases) its frame length, $N_{h,i}$, while the other nodes do not, we expect no variation in the SINR (there is in fact a slight increase (decrease) in
the SINR, which can be neglected under high SNR conditions, $\eta \ll \sum_{k \in \mathbb{Z}_i} P_k g_{ki})$. Finally, when the frame length of the interfering nodes is increased (decreased), the SINR increases (decreases).

**Bit Error Rate.** The bit error rate at the receiver can be approximated by evaluating the pulse collision probability when time hopping and channel coding are used. Consider first a pure time-hopping system with no pulse repetition, i.e., one transmitted pulse per bit. Assume that the information carried by a pulse cannot be decoded correctly only if two or more pulses from different users, representing different symbols, are transmitted in the same time chip. Under this hypothesis, considering $K$ interferers using the same frame length $N_h$, the probability of collision is given by

$$P_c = 1 - \sum_{i=0}^{M-1} p_i [1 - (1 - p_i)] \frac{1}{N_h} K,$$

(3.10)

where $p_i$ is the a-priori probability of transmitting the $i^{th}$ symbol in the set of possible symbols $M$, e.g., $\{0, 1\}$, with $|M|$ the size of the set $M$, and $(1 - p_i) \frac{1}{N_h}$ is the probability that one pulse collides (because the symbol is different), that is the information carried by the pulse cannot be decoded correctly.

Since the collision probability depends on the number of pulses transmitted by an interferer within the user time frame, $N_h$ in (3.10) represents the frame length of the node generating interference. Thus, if different nodes use different frame lengths, (3.10) can be rewritten as

$$P_c = 1 - \sum_{i=0}^{M-1} p_i \prod_{j=0}^{G} [1 - (1 - p_i) \frac{1}{N_h}]^{K_j},$$

(3.11)

where $N_{h,j}$ is the frame length in number of chips used by the $K_j$ nodes in the $j^{th}$ group and $G$ is the number of groups into the network, where each group includes all the nodes in the network using the same frame length.

Finally, we consider the effect of pulse repetition. Assume that, for each bit, $N_s$ pulses are sent, one per each frame. The bit cannot be correctly decoded if more than half of the transmitted pulses collide. Since the probability of having $x$ collisions over $N_s$ transmissions follows a binomial distribution

$$P_x = \binom{N_s}{x} P_c^x (1 - P_c)^{N_s - x},$$

(3.12)

the probability that more than half transmitted pulses collide is given by

$$P_{x+1} = \sum_{x=\frac{N_s}{2}+1}^{N_s} P_x.$$  

(3.13)

Finally, the symbol error rate can be expressed as:

$$P_{err} = 1 - \sum_{i=0}^{M-1} p_i [1 - (1 - p_i) P_{\frac{N_s}{2}+1}].$$

(3.14)

Note that this expression is exact for a simple repetition code. A pseudo-random spreading code increases the collision recovery capability at the receiver. Thus, the effective BER will be further reduced.
BER is a decreasing function of the code length of the transmission and of the frame length of the interfering nodes. Increasing the code length of the transmission and the frame length of the interfering nodes reduces the probability of collision. Increasing the frame length of the transmitter does not lead to any BER variation.

3.4 MAC and Rate Adaptation

In this section, we discuss UsWB medium access control principles and rate adaptation. Based on our discussion so far, there is a tradeoff between (i) resilience to interference and channel errors, (ii) achievable information rate, and (iii) energy efficiency. We introduce medium access control and rate adaptation strategies designed to find optimal operating points along efficiency-reliability tradeoffs. We first consider rate-maximizing adaptation strategies in Section 3.4.1. Then, we propose two different energy-minimizing strategies in Section 3.4.2.

3.4.1 Distributed Rate-Maximizing Adaptation

The objective of the rate-adaptation algorithm under consideration is to let each active communication maximize its transmission rate by selecting a pair of code and frame lengths, based on the current level of interference and channel quality measured at the receiver and on the level of interference generated by the transmitter to the other ongoing communications. We consider a decentralized ultrasonic intra-body area network, with \( \mathcal{N} \) being the set of \( |\mathcal{N}| \) existing connections. Note that there are no predefined constraints on the number of simultaneous connections \( |\mathcal{N}| \). The actual number \( |\mathcal{N}| \) depends on the specific application. For example, in simple applications such as glucose measurements and insulin administration, the number \( |\mathcal{N}| \) can be in the order of a few units. Instead, in more advanced applications, e.g., minimally-intrusive microsurgery, the number of simultaneous existing connections can be higher. Denote by \( N_{h,max} \) and \( N_{s,max} \) the maximum frame and code lengths supported,

\[
0 < N_{h,i} \leq N_{h,max}, \quad \forall i \in \mathcal{N}, \quad N_h \in \mathbb{N},
\]
\[
0 < N_{s,i} \leq N_{s,max}, \quad \forall i \in \mathcal{N}, \quad N_s \in \mathbb{N},
\]

where \( \mathbb{N} \) is the set of natural numbers. According to the transmission scheme discussed in Section 3.3.1, each node \( i \) transmits at a rate \( R_i \) expressed as in (3.7) and each receiver experiences an SINR expressed as in (3.9). Each node has a minimum data rate requirement, i.e., \( R_i(N_{h,i}, N_{s,i}) \geq R_{min} \), and a minimum SINR requirement, i.e., \( \text{SINR}_i(N_{h,i}, N_{s,i}) \geq \text{SINR}_{min} \).

Explicitly Cooperative Problem. The receiver is in charge of estimating interference and calculating frame and spreading code lengths that maximize the system performance. Accordingly, we denote the frame length and the code length calculated by the receiver of the connection \( r \), as \( N_{h,r} \) and \( N_{s,r} \).
The objective of each user is to locally optimize the information rate of the connection by solving the following problem:

\[
\begin{align*}
\text{maximize} & \quad R_r(N_{h,r}, N_{s,r}) \\
\text{subject to} & \quad R_r(N_{h,r}, N_{s,r}) \geq R_{\text{min}} \\
& \quad \text{SINR}_r(N_{h,r}, N_{s,r}) \geq \text{SINR}_{\text{min}} \\
& \quad \text{SINR}_i(N_{h,r}, N_{s,r}) \geq \text{SINR}_{\text{min}} \quad \forall i \in \mathcal{I}_r,
\end{align*}
\]

where \( \mathcal{I}_r \) is the set of the connections interfering with the \( r^{th} \) connection. The constraints on the maximum frame and code length in (3.15) and (3.16) are also implicitly considered. We refer to this as the explicitly cooperative problem.

**Implicitly Cooperative Problem.** If all nodes measure the same level of interference, that is, all network nodes are close enough to be all in the same transmission range, and all nodes have the same minimum rate and minimum SINR requirements, the level of interference that can be tolerated by each receiver is the same. Therefore, information about the maximum interference tolerable by each receiver does not need to be exchanged. The problem in (3.17) becomes then

\[
\begin{align*}
\text{maximize} & \quad R_r(N_{h,r}, N_{s,r}) \\
\text{subject to} & \quad R_r(N_{h,r}, N_{s,r}) \geq R_{\text{min}} \\
& \quad \text{SINR}_r(N_{h,r}, N_{s,r}) \geq \text{SINR}_{\text{min}}.
\end{align*}
\]

The system of SINR inequality constraints in (3.17) becomes here a single inequality constraint. The new problem can be interpreted as finding the optimal pair of code and frame length that maximize the rate, given a minimum rate and a minimum SINR, under the assumption that all the other nodes will be acting in the same way. As we will show in Section 3.5, since the problem to be solved is the same for each node, a globally optimal pair of code and frame lengths will be found. We refer to this as the implicitly cooperative problem.

We now provide a detailed study of the explicitly cooperative problem. The implicitly cooperative problem can be derived as a special case by neglecting the constraint in (3.20).

The explicitly cooperative problem, stated in (3.17) is an integer program, i.e., variables \( N_{h,r} \) and \( N_{s,r} \), are constrained to assume integer values. If the domain of the problem is small, it can be solved by enumeration, i.e., trying all the possible combinations of \( N_{h,r} \) and \( N_{s,r} \). When the domain of the problem increases in size, a relaxation method can be used to transform the integer problem into a real-value problem. The relaxation operation consists of replacing the constraints in (3.15) and (3.16) with

\[
0 < N_{h,r} \leq N_{h,\text{max}}, \quad 0 < N_{s,r} \leq N_{s,\text{max}}, \quad N_{s,r}, N_{h,r} \in \mathbb{R},
\]

where \( \mathbb{R} \) is the set of real numbers. Note that the optimal solution to the relaxed problem is not necessarily integer. However, since the feasible set of the relaxed problem is larger than the feasible set of the original integer program, the optimal value of the former, \( p_{\text{rlx}} \) is a lower bound on the optimal value of the
CHAPTER 3. MAC AND RATE ADAPTATION FOR ULTRASONIC INTRA-BODY NETWORKS

latter, \( p_{\text{int}}^* \), i.e.,

\[
L = p_{\text{rlx}}^* \leq p_{\text{int}}^*. \tag{3.24}
\]

The relaxed solution can be used to find an integer solution by rounding its entries based on a threshold \( \theta \in [0, 1] \), i.e.,

\[
\hat{x}_{\text{int}}^* = \begin{cases} 
\lfloor x_{\text{rlx}}^* \rfloor & \text{if } \lfloor x_{\text{rlx}}^* \rfloor - x_{\text{rlx}}^* \geq \theta \\
\lceil x_{\text{rlx}}^* \rceil & \text{otherwise}
\end{cases} \tag{3.25}
\]

where \( \lfloor \cdot \rfloor \) and \( \lceil \cdot \rceil \) are the ceiling and floor functions, respectively. If the rounded solution, \( \hat{x}_{\text{int}}^* \), is feasible for the original problem, i.e., all the constraints are satisfied, then it can be considered a guess at a good, if not optimal, point for the original problem. Moreover, the objective function evaluated at \( \hat{p}_{\text{int}}^* \) is an upper bound on \( p_{\text{int}}^* \),

\[
U = \hat{p}_{\text{int}}^* \geq p_{\text{int}}^*. \tag{3.26}
\]

Therefore, \( \hat{x}_{\text{int}}^* \) cannot be more than \((U-L)\)-suboptimal for the original problem.

After the problem has been relaxed, we can express the relaxed problem as a geometric program [34], that is, minimizing a posynomial function under posynomial inequality constraints, by means of a change of variable and a transformation of the objective and constraints functions. After the transformation, we can efficiently solve the problem using polynomial-time interior-point algorithms [35].

First, the problem must be restated in terms of minimizing the inverse of the data rate, that is, \( R_r^{-1}(N_{h,r}, N_{s,r}) \). Then, the new objective function and all the constraints can be expressed in monomial and posynomial form. The objective function, as well as the minimum rate function, is a monomial function, \( N_{h,r}^{-1} N_{s,r}^{-1} \). The same holds for the constraint functions defined is (3.23), and for the SINR function defined in (3.9).

Let us first consider the SINR constraint in (3.19). Note that in the distributed optimization problem the only variables are the code and the frame length calculated by the receiver, \( N_{s,r} \) and \( N_{h,r} \). If we define

\[
\alpha_r = P_r g_{r,r} T_c, \tag{3.27}
\]

and

\[
\beta_k = \frac{\sigma^2 T_c P_k g_{k,r}}{N_{h,k}}, \tag{3.28}
\]

then (3.19) can be rewritten as a posynomial constraint

\[
\eta N_{s,r}^{-1} N_{h,r}^{-1} + N_{s,r}^{-1} \sum_{k \in I_r} \beta_k \leq \frac{\alpha_r}{\text{SINR}_{\text{min}}}. \tag{3.29}
\]

For the remaining \((|I_r| - 1)\) SINR constraints, if we define

\[
\gamma_i = P_k g_{i,i} T_c N_{h,i} N_{s,i}, \quad \epsilon_i = \sigma^2 T_c P_r g_{r,i} N_{h,i},
\]

\[
\delta_i = \sum_{k \in I_r} \frac{\sigma^2 T_c P_k g_{k,i} N_{h,i} / N_{h,k}}, \tag{3.30}
\]

then (3.20) can be rewritten as

\[
\frac{\eta + \delta_i + \epsilon_i N_{h,r}^{-1}}{\gamma_i} \leq \frac{1}{\text{SINR}_{\text{min}}}, \tag{3.31}
\]

30
which is a linear, and thus monomial, constraint. In particular, this constraint shows that the frame length variation only leads to an increase (decrease) in the level of interference produced. Moreover, constraint (3.31) suggests that the optimal frame length is constrained by the co-located node that is experiencing the highest level of interference. Based on these observations, the cooperative optimization problem becomes

\[
\begin{align*}
\text{minimize} & \quad R_r^{-1}(N_{h,r}, N_{s,r}) \\
\text{subject to} & \quad R_r^{-1}(N_{h,r}, N_{s,r}) \leq R_{\min}^{-1} \\
& \quad \eta N_{s,r}^{-1} N_{h,r}^{-1} + N_{s,r}^{-1} \sum_{k \in \mathcal{L}_r} \beta_k \leq \frac{\alpha_r}{\text{SINR}_{\min}} \\
& \quad N_{h,r} \geq \frac{\epsilon_i}{\gamma_i \text{SINR}_{\min}^{-1} - \eta - \delta_i}.
\end{align*}
\]

Finally, a geometric program can be in general transformed into a convex program through a logarithmic transformation of the optimization variables. If we define

\[
y = [y_1, y_2] = [\log N_{h,r}, \log N_{s,r}],
\]

the objective function and the rate constraint function become

\[
R_r^{-1}(N_{h,r}, N_{s,r}) = N_{h,r} N_{s,r} = e^{y_1} e^{y_2} = e^{a_0^T y},
\]

where \(a_0 = [1 \ 1]\). The constraint function in (3.34) becomes

\[
\begin{align*}
\eta N_{s,r}^{-1} N_{h,r}^{-1} + N_{s,r}^{-1} \sum_{k \in \mathcal{L}_r} \beta_k &= \\
= \eta (e^{y_1})^{-1} (e^{y_2})^{-1} + \sum_{k \in \mathcal{L}_r} \beta_k (e^{y_2})^{-1} = e^{a_1^T y + b_1} + e^{a_2^T y + b_2},
\end{align*}
\]

where \(a_1 = [-1 \ -1], a_2 = [0 \ -1], b_1 = \log \eta\) and \(b_2 = \log \sum_{k \in \mathcal{L}_r} \beta_k\). Then, one can transform the problem by taking the logarithm of the objective function and the constraint functions. The resulting problem can be expressed as

\[
\begin{align*}
\text{minimize} & \quad a_0^T y \\
\text{subject to} & \quad a_0^T y \leq R_{\min}^{-1} \\
& \quad \log (e^{a_1^T y + b_1} + e^{a_2^T y + b_2}) \leq \log \frac{\alpha_r}{\text{SINR}_{\min}} \\
& \quad y_1 \geq \log \left( \frac{\epsilon_i}{\gamma_i \text{SINR}_{\min}^{-1} - \eta - \delta_i} \right) \\
& \quad 0 \leq y_1 \leq \log N_{h,\max}, \ y_1 \in \mathbb{R} \\
& \quad 0 \leq y_2 \leq \log N_{s,\max}, \ y_2 \in \mathbb{R} \\
& \quad \forall i \in \mathcal{L}_r.
\end{align*}
\]
CHAPTER 3. MAC AND RATE ADAPTATION FOR ULTRASONIC INTRA-BODY NETWORKS

The objective function and the constraint function in (3.40) are linear, thus also convex. The constraint in (3.42) is convex as a result of the transformation performed [34]. We can conclude that the this problem is a convex optimization problem which can be solved in polynomial time through interior-point methods [35].

3.4.2 Distributed Energy-minimizing Rate Adaptation

We now concentrate on rate adaptation with the objective of reducing the energy consumption of UsWB. We mentioned in Section 3.3.1 that the time hopping and the spreading code techniques affect the energy consumption of the device transmitting a UsWB signal. For this reason, we introduce energy-related metrics that make the dependence of the energy consumption on code and frame length explicit. We define:

- \( E_p \), the energy per pulse;
- \( E_b \), the energy per bit, i.e., \( E_b = E_p \cdot N_s \);
- \( E_s \), the average power radiated per second, i.e., \( E_s = E_p / (T_c \cdot N_h) \).

The energy per bit is a linear function of the spreading code length, therefore of the number of pulses transmitted per each bit. The average power emitted per second is a function of the inverse of the frame length and hence of the number of pulses transmitted per second. Both depend on the value of \( E_p \), which is related to the (electrical) power absorption of the ultrasonic transducer in use.

Modeling the Energy Consumption of a Piezoelectric Transducer. We consider electronically-driven piezoelectric transducers. Since such devices are known to have minimal current leakage, the majority of power consumption comes from the piezoelectric element. Outside the region of resonance, a piezoelectric ceramic transducer can be viewed (from the electrical point of view) as a parallel plate capacitor with capacitance \( C_0 \). Thus, the main source of power consumption comes from charging such capacitor [36]. Then, ignoring charge and discharge losses, and considering a capacitor with voltage supply \( V \) and pulse repetition frequency \( f \) (which corresponds to the charge and discharge frequency of the capacitor), the power consumption \( P_c \) can be expressed as

\[
P_c = f C_0 V^2.
\]  \hspace{1cm} (3.47)

The static capacitance and the voltage supply values should be based on transducer-specific considerations. The static capacitance value of a disc-shaped transducer is

\[
C_0 = \frac{A \epsilon_0 K}{t_h} \text{ [F]},
\]  \hspace{1cm} (3.48)

where \( \epsilon_0 \) is the permittivity in free air \( (8.8542 \text{ F/m}) \), \( K \) is the dielectric constant of the material (adimensional), \( A \) [\text{m}^2] is the area of the disc, and \( t_h \) [\text{m}] is its thickness. The dielectric constant \( K \) depends on material, frequency and mechanical state of the transducer.

We derive appropriate limits for the voltage supply based on safety concerns, which impose a limit on the radiated acoustic power. As reported in [19], no tissue damage occurs in intra-body ultrasonic propagation.
as long as the acoustic power dissipation in tissues is limited to $10^4 \text{ W/m}^2$. From this limit, we derive the corresponding maximum pressure magnitude that can be radiated by the transducer, and consequently the maximum voltage input. For example, if we consider arm muscle as the primary propagation medium, by using the quadratic relation between the acoustic intensity and the acoustic pressure, $I = (P_{\text{RMS}})^2 / \rho c$, along with density and speed of sound parameters in [6], we find a maximum pressure magnitude of approximately 0.13 MPa.

We can derive the related transducer voltage input, corresponding to the maximum radiated pressure, through the constitutive equation of piezoelectric materials. The latter expresses the relationship between mechanical strain and electrical displacement for the piezoelectric element considering the electrical and mechanical stress and is usually expressed in tensor notation [37]. Here, we are interested in the so-called converse effect, i.e., the material strain proportional to an applied voltage. If we consider a symmetric crystal structure, the constitutive relations reduce to a few parameters. In particular, the so-called $g_{33}$ parameter $[Vm/N]$ represents the piezoelectric voltage coefficient, when the polarization field and the piezoelectrically induced strain are both parallel to the disc axis (usually referred to as the third axis). Based on this, one can express the electric field along the third axis $E$ $[V/m]$ as $E = g_{33} P$, where $P$ again represents the output pressure of the transducer. Since the electric field $E$ can in turn be expressed as the ratio between voltage and electrode distance, we have $V = g_{33} P t_h$. Finally, from (3.47), the energy necessary to generate a single pulse is obtained as

$$E_p = C_0 (g_{33} P t_h)^2 [J].$$

(3.49)

**Energy-minimizing Rate Adaptation.** Based on this model, we design a rate adaptation strategy where the objective is to minimize (i) the energy per bit, $E_b$, or (ii) the average energy emitted per second $E_s$. The problem can be cast as finding the optimal frame length and the optimal spreading code length that minimize $E_b$ (and/or $E_s$) while meeting the minimum SINR constraints and keeping the data rate over a given threshold. The problem is formally expressed below.

$$\text{minimize } E_b(N_{s,r}) \text{ (or } E_s(N_{h,r}))$$

subject to (3.18) (3.19) (3.20).

The problem above can also be relaxed to a geometric program as discussed in Section 3.4.1.

### 3.4.3 Medium Access Control Protocol

In UsWB, distributed medium access control coordination is achieved by exchanging information on logical control channels, while data packets are transmitted over logical data channels. We consider unicast transmissions between a transmitter $TX$ and a receiver $RX$. When $TX$ needs to transmit a packet, it first needs to reserve a dedicated channel to $RX$. The connection is opened through the common control channel, which is implemented through a unique TH-sequence and a spreading code known and shared by all network devices.
In the two-way handshake procedure, $TX$ sends a Request-to-Transmit (R2T) packet to $RX$, which contains its own ID. If $RX$ is idle, a Clear-to-Transmit (C2T) control packet is sent back to $TX$. In case of failure and consequent timer expiration, $TX$ will attempt a new transmission after a random backoff time, for a maximum of $N_R$ times. During these initial steps, since an estimate of the current interference level is not available, the transmitter transmits at the minimum data rate, conservatively using a maximum frame length and spreading code length, and thus generating low interference. After receiving the C2T packet, the transmitter switches to a dedicated channel by computing its own time-hopping sequence and spreading code obtained by seeding a pseudo-random sequence generator with its own ID. As a consequence, both $TX$ and $RX$ leave the common channel and switch to a dedicated channel. Once the connection has been established, $TX$ sends the first packet using maximum frame and spreading code length. The receiver $RX$ computes the optimal frame and spreading code lengths as discussed in Section 3.3.1. This information is piggybacked into ACK or NACK packets.

In the explicitly cooperative case, discussed in Section 3.4.1, once the communication has been established, $RX$ does not leave the common control channel. Instead, it keeps “listening” to both the dedicated and common control channels at the same time. In the dedicated control channel, $RX$ sends to $TX$ the optimal frame and code lengths to be used for the next transmission. In the common control channel, $RX$ exchanges with other co-located receivers information on the level of tolerable interference.

### 3.5 Performance Evaluation

#### 3.5.1 Multi-scale Simulator

In this section, we evaluate the proposed system performance through a custom-designed multi-scale simulator that models UsWB performance at three different levels, i.e., (i) at the acoustic wave level by capturing ultrasonic propagation in tissues through reflectors and scatterers, (ii) at the bit level by simulating in detail the physical layer transmission scheme, (iii) at the packet level by simulating networked operations and distributed medium access control and adaptation.

The acoustic wave level simulation is performed as described in Section 3.2. A channel impulse response is obtained by simulating propagation in the human arm. Transmission at the bit level is modeled through a custom physical layer simulator of UsWB, which produces as output an empirical model of the BER against different values of the time-hopping frame length and spreading code length, for different levels of interference. The physical layer simulation models a transmitter and a receiver (located 20 cm apart) communicating over an ultrasonic channel with the UsWB transmission scheme. Simulations are performed to obtain an estimate of the achievable BER upon varying the frame length and the spreading code length with a different number of interferers transmitting on the same channel. The interfering nodes are located within the interference range area of the receiving node of the pair, i.e., they are also located 20 cm apart from the receiving node. In the simulations we neglect both frame and time hopping synchronization errors so that the BER only considers interference effects. We implement both solutions presented in Section 3.3.1, i.e.,
Figure 3.7: (a) BER vs. spreading code length and (b) BER vs. time-hopping frame length for PPM-BPSK spread and PPM-PPM spread.

PPM-BPSK-spread with coherent receiver and the PPM-PPM-spread with non-coherent receiver. An extensive simulation campaign was conducted and BER values were obtained as a function of the spreading code and frame lengths for different number of simultaneously active connections. Figure 3.7 reports the BER values measured at the receiver, when 3 interfering nodes transmit at the same time and are located in the interference range area of the receiving node. As expected, the coherent PPM-BPSK-spread slightly outperforms the non-coherent PPM-PPM-spread.

Network Simulation Topology. The empirical model of the BER as a function of frame length and spreading code length for a given level of interference is then imported in a Java-based event-driven packet-level simulator, which models all the UsWB MAC protocol functionalities. We considered two different settings for the network level simulations. First, we consider a 2-D topology with 18 static nodes randomly located inside a square of side 20 cm. Each transmitter node communicates with a randomly-selected receiver node, in a point-to-point and single-hop fashion, resulting in a maximum of nine simultaneously communicating pairs. We also assume that the transmission range is greater than the maximum distance between the nodes. Thus, all nodes measure the same number of interferers.

The second setting consists of three 2-D squared clusters of nodes with side 10 cm, displaced 20 cm apart from each other (distance center-to-center). In each cluster, the nodes are randomly deployed according to a Gaussian distribution. The cluster (#1) in the middle contains one communicating pair, while clusters (#2) and (#3) contain a variable number of communicating pairs. We also assume the transmission range to be of 30 cm. Thus, all the nodes of adjacent clusters interfere with each other, while nodes of non-adjacent clusters do not. Under this assumption, communicating pairs from different clusters measure different levels of interference. As a result, this second scenario topology, depicted in Fig. 3.8, operates under the conditions of the explicitly cooperative problem. In a practical application, a similar scenario could for example represent three groups of ultrasonic sensors deployed in the heart, liver and kidney, respectively.

During the simulations, frame length and spreading code length are adapted to the interference
level measured at the receiver according to one of the objectives discussed in Section 3.4.1. The simulation time is set to 100 s and nodes start establishing connections at a random time instant but no later than 2 s after the simulation start time. We consider an infinite arrival rate at each transmitter, i.e., transmitters are always backlogged. The maximum allowed frame and spreading code length is set to 15 slots and 20 chips, respectively. The maximum supported rate, achieved when frame and spreading code length are both set to one, is equal to 2 Mbit/s. The minimum SINR constraint leads to a maximum BER constraint of $10^{-5}$.

We first discuss the performance in terms of throughput and packet drop rate of the rate-maximizing solution coming from both the implicit and explicit cooperative problems. We define the throughput as the average rate of information correctly received during the simulation time per active connection. The packet drop rate is defined as the ratio between the number of packets dropped and the number of packets generated at the application layer, averaged over all the active connections. Both performance metrics are evaluated as a function of the number of active connections $|N|$.

### 3.5.1.1 Rate Maximization

**Implicitly cooperative solution.** In Fig. 3.9, we compare network throughput (top) and packet drop rate (bottom) for the rate-maximizing strategy, when frame and code length are adaptively regulated based on the implicitly cooperative problem in Section 3.4.1. Figure 3.9 (bottom) also shows the maximum packet drop rate threshold given by the $10^{-5}$ BER constraint at the PHY layer. The rate-maximizing solution is presented for both the transmission schemes discussed in Section 3.3.1, i.e., the PPM-BPSK-spread with coherent receiver and the PPM-PPM-spread with non-coherent receiver. As expected, the coherent PPM-BPSK solution performs better in terms of throughput. This happens because the BER constraint is satisfied for lower values of frame length and code length, which leads to higher data rates according to (3.7). When the number of active connections is equal to one, i.e., there are no interferers, both systems achieve the same throughput since the same optimal pair is used, (2, 2). In terms of packet drop rate, for the coherent PPM-BPSK system the BER constraint is satisfied for any number of active connections considered. Instead, for the non-coherent PPM-PPM system, when the number of active connections is greater than 7, the BER constraint cannot be satisfied anymore, and therefore the packet drop rate increases. Note that this problem can be overcome by simply relaxing the constraint on the maximum size of code length or frame length, without relaxing the constraint on the maximum BER. Clearly this leads to a lower data rate.

We then focus on the analysis of the dynamic adaptation of the frame and code length performed distributively by each node. Focusing on the coherent system, in Fig. 3.10, we show the evolution in time of
frame and code length when 9 different connections are asynchronously activated with a deterministic 5 s delay between each other. As expected, each connection starts with the maximum supported frame and code length and then adaptively reaches the optimal value based on the interference level measured at the receiver. Since the number of interferers is the same for each receiver, the locally optimal solution is also globally optimal.

**Explicitly cooperative solution.** In Fig. 3.11, we plot the network throughput obtained when frame and code length are adaptively regulated based on the explicitly cooperative problem in (3.17). The solution is presented for the PPM-BPSK-spread transmission strategy discussed in Section 3.3.1. Throughput is evaluated by varying the number of active connections in four consecutive time steps. In particular, we assume that there is always one active connection in cluster #2. Cluster #1 and #3, activate a new connection at each time step.

In Fig. 3.11, we also report the average number of interferers measured by the receivers in each cluster. As expected, the cluster located in the middle, hence located within the transmission range of both the other two clusters, measures a higher level of interference, and therefore achieves a lower throughput. However, since the BER constraint is satisfied by all nodes in the three clusters, a low packet drop rate is achieved.

Finally, in Fig. 3.12, the dynamic behavior of the frame and code length adaptation is shown. We consider two new connections activated asynchronously every 5 s in Cluster #1 and #3. In each cluster, the frame and code length are adapted according to the increasing level of interference measured in the channel. In particular, we observe the effect of the constraint in (3.35), which forces the frame length to be greater than or equal to the frame length of the connection that is experiencing the highest level of interference, i.e., the connection in Cluster #2.

### 3.5.1.2 Rate-optimal Vs. Energy-optimal Results

In Fig. 3.13, we plot the throughput obtained by adapting the transmission rate according to the energy-minimizing strategy introduced in Section 3.4.2. The energy-minimizing strategy is also compared to
Figure 3.10: Time evolution of spreading code and time-hopping frame lengths for the implicitly cooperative problem.

Figure 3.11: Average throughput [kbit/s] and number of interferers with explicitly cooperative problem for different time steps.
the rate-maximizing strategy introduced in Section 3.4.1 in terms of data rate and energy consumption. Both results are presented for the PPM-BPSK-spread transmission strategy. The throughput achievable in case of the rate-maximizing solution is comparable to what is obtained with the energy-optimized solution. When the number of active connections, $N$, is lower than 2, the throughput of the $E_b$-optimal strategy is close to the throughput of the rate-maximizing scheme. For $N$ higher than 2, the two considered strategies show a similar throughput performance. Since the maximum BER constraint is always satisfied, the rate-maximizing and the energy-minimizing strategies lead to packet drop rates close to zero. Finally, the energy consumption obtained with the $E_b$-optimal solution, is always better than what can be obtained using the rate-maximizing scheme. This result assumes that each node transmits at the maximum power allowed that gives a value of energy per pulse of $E_p = 1.8436$ nJ obtained using the model presented in Section 3.4.2 assuming $A = 0.201$ mm$^2$, $A = 0.47$ mm, $K = 1800$ and $g_{33} = 0.027$ N/m.

### 3.5.2 Ultrasonic Software-Defined Testbed

Since no off-the-shelf ultrasonic transceivers for intra-body communications are available, we designed and developed our custom ultrasonic software-defined nodes based on the USRP N210 platform [38], as discussed in detail in Chapter 4. In the following, we report BER experimental measurements obtained by varying the time-hopping frame and spreading code length in presence of interference. We compare these experimental results with BER curves obtained by simulating the same setup with the multi-scale simulator discussed in Section 3.5.1, thus validating our simulation results. The experimental setup consists of two ultrasonic nodes that communicate through a human-kidney phantom [24]. The two ultrasonic transducers are located on opposite sides of the phantom at a distance of 10 cm. To guarantee repeatability of the experiments, we generate interference from co-located transceivers by artificially injecting interfering pulses at the transmitter. The interfering pulses are pseudo-randomly located inside the time-hopping frame. We
Figure 3.13: Throughput and $E_b$ vs. the number of active connections for rate maximizing and energy minimizing strategies.

Figure 3.14: a) BER as a function of frame length, with 4 interfering pulses per frame for different values of code length, b) compared with simulated BER.
evaluate the BER by varying the time-hopping frame, from 2 to 14, and the spreading code length between 8, 10, 12 and 14. We consider 4 interfering pulses per frame, and we set the input power at the TX transducer to 13 dBm (= 20 mW). Results are shown in Fig. 3.14(a). In Fig. 3.14(b) the experimental results are compared with BER curves obtained by simulating the same scenario with the UsWB PHY layer simulator described in Section 3.5.1. For fairness, the pulse shape used in the simulator is obtained by recording a real pulse shape as received in testbed experiments, thus considering the signal distortion introduced by amplifiers and transducers, and the scattering and reflection effects introduced by the ultrasonic phantom. However, the simulation setup does not consider the time-variability of the real testbed conditions, e.g., operating temperature, coupling of the transducer with the phantom surface, and partial synchronization failures. We observe that, as expected, the BER is a decreasing function of the time-hopping frame length and the spreading code length, thus confirming the simulation results in Section 3.5.1.

3.6 Conclusions

In this chapter, we proposed a paradigm shift in networking through body tissues to address the limitations of RF propagation in the human body. We presented - to the best of our knowledge - the first attempt at enabling networked intra-body communications among miniaturized sensors and actuators using ultrasonic waves. We assessed the feasibility of using ultrasonic communications within the human body; we derived an accurate channel model for ultrasonic communications in the human body and built on it to propose a new ultrasonic transmission and multiple access technique, denoted as UsWB, based on transmission of short duration pulses following a time hopping pattern. Simulation and experimental results demonstrate the feasibility of ultrasonic communication in the human body, and show how, by designing appropriate ad-hoc transmission schemes and protocols, ultrasounds can be efficiently used to wirelessly internetwork implantable devices.
Chapter 4

Experimental Evaluation of Impulsive Ultrasonic Intra-Body Communications for Implantable Biomedical Devices

In Chapter 2, we showed that intra-body ultrasound propagation is severely affected by multipath caused by inhomogeneity of the body in terms of density, sound speed, and the pervasive presence of small organs and particles. Based on these observations, in Chapter 3, we proposed Ultrasonic WideBand (UsWB), a new ultrasonic multipath-resilient physical and medium access control (MAC) layer integrated protocol. UsWB is based on the idea of transmitting short carrierless ultrasonic pulses following a pseudo-random adaptive time-hopping pattern, with a superimposed adaptive spreading code. Impulsive transmission and spread-spectrum encoding combat the effects of multipath and scattering and introduce waveform diversity among interfering nodes so that multiple users can coexist with limited interference on the same channel. However, as of today, there have been no attempts at experimentally demonstrating ultrasonic communications through body tissues.

In this chapter, we make the following core contributions: (i) we present the design and implementation of a software-defined testbed architecture for ultrasonic intra-body area networks. The testbed consists of software-defined nodes communicating via ultrasonic waves through media that emulate acoustic propagation through biological tissues with high fidelity, i.e., ultrasonic phantoms; (ii) we experimentally demonstrate the feasibility of ultrasonic communications in human tissues. To this purpose, we design an FPGA-based prototype implementation of the UsWB physical and medium access control protocols and evaluate extensively its performance through a human-kidney phantom. We show that our prototype can flexibly trade data rate performance for power consumption, and achieve, for bit error rates (BER) no higher than $10^{-6}$, either (i) high-data rate transmissions up to 700 kbit/s at a transmit power of -14 dBm ($\approx 40 \mu W$), or (ii) low-data rate and lower-power transmissions down to -21 dBm ($\approx 8 \mu W$) at 70 kbit/s (in addition to numerous intermediate
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

configurations). Moreover, we show how the UsWB MAC protocol allows multiple concurrent users to coexist and dynamically adapt their transmission rate to channel and interference conditions to maximize throughput while satisfying predefined reliability constraints, e.g., maximum packet drop rate; (iii) we show how UsWB can enable video monitoring medical applications for implantable devices, and we evaluate the application video streaming performance in terms of peak signal-to-noise ratio (PNSR) and similarity (SSIM) index; finally, (iv) we propose and validate through measurements a statistical model of small-scale fading for the ultrasonic intra-body channel. Specifically, we show that the amplitude distribution of the signal received through human-kidney phantoms follows a generalized Nakagami distribution;

The remainder of the chapter is organized as follows. In Section 4.1 we briefly discuss basic aspects of ultrasonic intra-body communications and introduce the UsWB transmission and medium access technique. In Section 4.2 we present the proposed testbed architecture. In Section 4.3 we discuss the communication system architecture and implementation while in Section 4.4 we discuss details of the FPGA implementation of transmitter and receiver. In Sections 4.5 we extensively evaluate the performance of UsWB. In Sections 4.6 and 4.7 we discuss a video monitoring application based on UsWB and a statistical model of small-scale fading for the ultrasonic intra-body channel. Finally, in Section 4.8 we conclude the chapter.

4.1 Background

4.1.1 Ultrasonic Intra-Body Communications

Ultrasounds are mechanical waves that propagate in an elastic medium at frequencies above the upper limit for human hearing, i.e., 20 kHz.

Attenuation. Two main mechanisms contribute to ultrasound attenuation in tissues, i.e., absorption and scattering. An initial pressure $P_0$ decays at a distance $d$ according to [5]

$$P(d) = P_0 e^{-\alpha d},$$

(4.1)

where $\alpha$ (in [Np cm$^{-1}$]) is an amplitude attenuation coefficient that captures all the effects that cause dissipation of energy from the ultrasound wave. Parameter $\alpha$ depends on the carrier frequency through $\alpha = af^b$, where $f$ represents the carrier frequency (in MHz) and $a$ (in [Np m$^{-1}$ MHz$^{-b}$]) and $b$ are attenuation parameters characterizing the tissue [4, 39].

Propagation Speed. Ultrasonic wave propagation is affected by propagation delays that are orders of magnitude higher than RF. The propagation speed of acoustic waves in biological tissues is approximately 1500 m/s, as compared to $2 \times 10^8$ m/s [40] for RF waves.

Operating Frequency. Key considerations in determining the operating frequency are (i) the frequency dependence of the attenuation coefficient, and (ii) the frequency dependence of the beam spread of ultrasonic transducers (which is inversely proportional to the ratio of the diameter of the radiating surface and the wavelength [4, 39]). Therefore, higher frequencies help keep the transducer size small, but result in higher signal attenuation. Since most biomedical sensing applications require directional transducers, one
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

Figure 4.1: Two concurrent transmissions with $N_h = 6$, $N_s = 3$, time-hopping sequences $\{3, 2, 1\}$ and $\{0, 5, 4\}$ and spreading codes $\{1, 1, -1\}$ and $\{1, -1, -1\}$.

needs to operate at the lowest possible frequencies compatible with small-size transducers and required signal bandwidth. In [4, 39], we showed that for propagation distances in the order of several cm the operating frequency should not exceed 10 MHz.

Reflections and Scattering. The human body is composed of different organs and tissues with different sizes, densities and sound speeds. Therefore, it can be modeled as an environment with pervasive presence of reflectors and scatterers. The direction and magnitude of the reflected wave depend on the orientation of the boundary surface and on the acoustic impedance of the tissues [4, 39], while scattered reflections occur when an acoustic wave encounters an object that is relatively small with respect to its wavelength or a tissue with an irregular surface. Consequently, the received signal is obtained as the sum of numerous attenuated, possibly distorted, and delayed versions of the transmitted signal.

4.1.2 Ultrasonic WideBand

Based on these observations, in [41] we proposed Ultrasonic WideBand (UsWB), a new impulse-radio inspired ultrasonic transmission and multiple access technique based on the idea of transmitting short information-bearing carrierless ultrasonic pulses, following a pseudo-random adaptive time-hopping pattern with a superimposed spreading code of adaptive length. Impulsive transmission and spread-spectrum encoding combat the effects of multipath and scattering and introduce waveform diversity among interfering transmissions.

Physical Layer. Consider, as in Fig. 4.1, a slotted timeline divided in slots of duration $T_c$, with slots organized in frames of duration $T_f = N_h T_c$, where $N_h$ is the number of slots per frame. Each user transmits one pulse per frame in a slot determined by a pseudo-random time-hopping sequence. Information is carried through pulse position modulation (PPM), i.e., a ‘1’ symbol is carried by a pulse delayed by a time $\delta$ with respect to the beginning of the slot, while a ‘-1’ symbol begins with the slot. Since a single pulse may collide with pulses transmitted by other users with a probability that depends on the frame size $N_h$, we represent each information bit with pseudo-orthogonal spreading codes of variable length, $N_s$ because of (i) their excellent, and well-understood multiple access performance, (ii) limited computational complexity, and (iii) inherent resilience to multipath. The resulting transmitted signal for a symbol $d$ can be modeled as

$$s(t) = \sum_{j=0}^{N_s-1} p(t - c_j T_c - j T_f - a_j d + \frac{1}{2} \delta)$$

(4.2)

where $p(t)$ is the pulse shape, $\{c_j\}$ is the time-hopping sequence with $0 \leq c_j \leq N_h - 1$, $\{a_j\}$ is the pseudo-orthogonal spreading code of $N_s$ chips with $a_j \in \{-1, 1\}$, and $\delta$ is the PPM shift of a pulse representing a ‘1’.
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

Figure 4.2: Hardware architecture of an ultrasonic software-defined node.

Medium Access Control. The low-duty-cycle impulse-based transmission scheme with a superimposed spreading code allows multiple transmitters to coexist on the same channel. In UsWB, by dynamically and distributively adapting their time-hopping frame length and spreading code length, multiple users coexist without the need for mutual temporal exclusion between different transmissions (which is hard to achieve in ultrasonic channels affected by long propagation delays). By adapting frame and code length, users control the tradeoffs among (i) resilience to multi-user interference and ultrasonic channel errors, (ii) achievable information rate, and (iii) energy efficiency. As discussed in detail in [41], by controlling the time-hopping frame length $N_h$, i.e., the average inter-pulse time, a user can adapt the transmission rate (which decreases with larger time-hopping frame), and as a consequence modify the average radiated power and therefore the level of interference generated to other ongoing communications. By controlling $N_s$, i.e., the number of pulses per information bit, a user can control the tradeoff between robustness to multi-user interference and noise (which increases with longer spreading codes), energy consumption per bit (which increases linearly with increasing $N_s$) and information rate (decreasing with increasing $N_s$). UsWB optimally, distributively, and asynchronously regulates these tradeoffs to (i) maximize the communication rate, or (ii) minimize the energy consumption.

In this paper, we consider the rate-maximizing adaptation in [41], where each user distributively maximizes its transmission rate by selecting an optimal pair of code and frame lengths based on the current level of interference and channel quality for a given maximum tolerable BER. Rate adaptation is achieved through an ad-hoc designed protocol. A two-way handshake opens the connection between two nodes, $T_x$ and $R_x$. $T_x$ sends a Request-to-Transmit (R2T) packet to $R_x$. If $R_x$ is idle, a Clear-to-Transmit (C2T) packet is sent back to $T_x$. Once the connection has been established, the receiver $R_x$ estimates the interference and calculates the frame and spreading code lengths that maximize the communication throughput, as discussed in detail in [41]. This information is piggybacked into ACK or NACK packets.

4.2 Ultrasonic Testbed Architecture

We designed and implemented a reconfigurable platform to test ultrasonic communication and networking schemes. The testbed consists of ultrasonic software-defined nodes communicating through ultrasonic phantoms that emulate acoustic propagation through biological tissues with high fidelity. The
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

proposed hardware architecture of an ultrasonic software-defined node is illustrated in Fig. 4.2. It consists of (i) a Universal Software Radio Peripheral (USRP) N210, (ii) a host machine, (iii) an electronic switch, (iv) an amplification stage, (v) and a high-frequency ultrasonic transducer.

**USRP N210.** Several Software Defined Radio (SDR) development platforms are available [38, 42] where Field-Programmable Gate Arrays (FPGAs) or specialized processors are used for high-sample-rate digital signal processing. Among these, we selected USRP [38] because of its low cost and wide adoption in academia and industry. USRP N210 consists of a motherboard and two daughterboards. The motherboard is the main processing unit, and incorporates AD/DA converters (a dual 100 MSPS 14-bit ADC and a dual 400 MSPS 16-bit DAC), and an FPGA unit (Spartan 3A-DSP 3400). The daughterboards are RF front-ends that interface the device with transmitter or receiver antennas. We use LFTX and LFRX daughterboards, that operate from DC to 30 MHz, which includes ultrasonic frequency ranges of interest to us.

In USRP, the system complexity is shifted from hardware to software and most of the computational load is typically left to the host machine. However, recent literature [43, 44] has shown that the host machine can become the computational bottleneck of the communication system, while the connection between host and USRP introduces delays preventing accurate timing of network protocols. As discussed in detail in what follows, we overcome this problem by shifting (with respect to the typical GNU Radio/USRP architecture) significant components of the signal processing on the on-board FPGA.

**Host Machine.** The host machine can be either a desktop/laptop computer or a computer-on-module, e.g., Gumstix, connected to the USRP through a Gigabit Ethernet (GbE) link. In the traditional GNU Radio/USRP architecture, the host machine runs all the software-defined signal processing functionalities implemented with the GNU Radio development toolkit [45]. However, for reasons discussed in detail in Section 4.3, we chose to implement most PHY and MAC functionalities in the FPGA embedded in the USRP. Therefore, in our design the host machine only configures and initializes the USRP, and generates/receives application-layer bit streams.

**Electronic Switch.** To reduce the testbed complexity and cost, we use an electronic switch that allows a single ultrasonic transducer to transmit and receive on a time division basis. The switching operation is piloted both from the host machine and the USRP FPGA by connecting the switch with the General Purpose Input/Output (GPIO) digital pins available on the LFTX and LFRX daughterboards. We use a commercial off-the-shelf (COTS) switch, Mini-Circuits ZX80-DR230+ [46], that comes in a connectorized package with embedded coaxial RF connectors, and offers low insertion loss and very high isolation over the entire frequency range (0 – 3 GHz).

**Amplification Stage.** We introduced an external amplification stage. The low-power output of the LFTX daughterboards, about 3 dBm (≈ 2 mW), can limit the maximum transmission range supported. Therefore, at the transmitter we use a connectorized COTS Power Amplifier (PA), Mini-Circuits ZPUL-30P [46], specifically designed for short-pulse transmissions with a maximum output power of 22 dBm. In the receiver chain, LFRX daughterboards have almost no gain. Thus, we use a connectorized COTS Low-Noise Amplifier (LNA), Mini-Circuits ZFL-1000LN+ [46], with a noise figure of 2.9 dB.

**Ultrasonic Transducers.** An ultrasonic transducer is a device capable of transmitting and receiving
ultrasonic waves. Most commercial ultrasonic transducers are based on the piezoelectric effect, which allows converting electrical energy in ultrasonic energy, and vice versa [47].

As discussed in Section 4.1.1, to communicate in human tissues over a range of several centimeters we need transducers operating at frequencies in the order of a few MHz. Moreover, high-bandwidth transducers are necessary to implement wideband transmission schemes such as UsWB. We found that the only COTS ultrasonic transducers that nearly match our requirements are those designed for nondestructive analysis (NDA) applications [48], since high frequencies and large bandwidth are required for fine material characterization. However, these transducers are not optimized in terms of coupling electromechanical efficiency, and thus introduce significant energy conversion losses. Moreover, NDA transducers are characterized by high directivity. In our current testbed we use standard immersion W-series ultrasonic transducers, Ultran WS37-5 [48]. The nominal bandwidth central frequency is about 5 MHz and the bandwidth at -6 dB goes from 50% to 100% of the bandwidth central frequency, i.e., 2.5 – 5 MHz.

**Ultrasonic Phantoms.** We use ultrasonic phantoms to emulate the intra-body ultrasonic communication channel with high fidelity [49]. Commonly employed in medical ultrasound research, ultrasonic phantoms are composed of soft and hard tissue-mimicking materials, also known as tissue substitutes. These materials have the same acoustic propagation properties of human tissues, e.g., sound speed, density, and attenuation. Off-the-shelf ultrasonic phantoms that emulate the interactions between ultrasounds and the human body, tissues, organs and systems are available [50].

We selected a human-kidney phantom immersed in a background water-based gel [50], as shown in Fig. 4.3, whose acoustic characteristics are reported in Table 4.1. The background gel is almost lossless, and has the same density and sound speed as the kidney. Therefore, reflections and refractions are minimum between the kidney and the gel. Thus, the latter can be considered acoustically transparent. The phantom dimensions are approximately $10 \times 16 \times 20$ cm.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>Speed, $v$</th>
<th>Attenuation, $\alpha$</th>
<th>Density, $\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background Gel</td>
<td>1550 m/s</td>
<td>$&lt; 0.1$ dB/cm</td>
<td>1020 Kg/m$^3$</td>
</tr>
<tr>
<td>Kidney</td>
<td>1550 m/s</td>
<td>2 dB/cm @ 5 MHz</td>
<td>1030 Kg/m$^3$</td>
</tr>
</tbody>
</table>
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

4.3 Architecture of the Communication System

Signal processing, algorithms and protocols can be implemented in the ultrasonic software-defined node using a framework that combines (i) the GNU Radio software development toolkit [45] and (ii) the open source Hardware Description Language (HDL) design for the FPGA embedded in the USRP. In GNU Radio, most of the digital signal processing is performed on the external host. When using HDL design most signal processing operations are moved to the embedded FPGA. In this Section we discuss the tradeoffs between these two different approaches.

4.3.1 GNU Radio Vs. HDL PHY Layer Implementation

As discussed in Section 4.1.2, wideband pulse-based communications can significantly mitigate the multipath effect caused by the heterogeneity of the human body. However, shorter pulses have wider bandwidth, which results in higher sampling rates that can overload the host machine or the Gigabit Ethernet (GbE) link between the host machine and the USRP. In our initial design, UsWB PHY layer functionalities were implemented on the host machine using GNU Radio. However, we observed the following limitations: (i) the capacity of the GbE link between the host machine and the USRP limits the maximum achievable sample rate, i.e., 25 million samples per second, thus the maximum achievable signal bandwidth. When exceeding the link capacity, Ethernet frames coming from/to the USRP are dropped at the network interfaces, with consequent loss of the carried digital samples; (ii) digital signal processing operations implemented in GNU Radio, e.g., digital filters, overload the host machine when operating at high sampling rates, i.e., greater than 10 million samples per second. If the host machine is unable to process data fast enough, the internal buffers that store digital samples overflow, thus causing loss of large amount of digital samples.

It became apparent that the above limitations would prevent successful implementation of UsWB in GNU Radio. For these reasons, we chose to implement all PHY layer functionalities in the embedded FPGA. This effectively speeds up data processing and reduces the computational load on the host machine.

Partial Reconfiguration. The price we pay for these benefits is a lower system flexibility. The HDL design needs to be synthesized before it can be loaded in the embedded FPGA. Thus, changing the PHY layer structure and parameters at runtime is not as simple as doing it in GNU Radio on the host machine. Still, we designed the HDL modules such that partial PHY layer reconfiguration is achievable at runtime through a group of setting registers implemented on the FPGA that can be accessed by the host machine. Through these setting registers, one can reconfigure key parameters of the PHY layer transmission scheme (i.e., pulse shape and code length, among others) or even select which PHY blocks should be used, thus modifying at runtime the structure of the PHY layer chain.

4.3.2 MAC Layer Design Challenges

Similarly, MAC-layer functionalities can potentially be implemented in the host machine or in the embedded FPGA. Highly customized or reconfigurable and complex protocols can be challenging to
implement on FPGA and likely cause overutilization of the available FPGA hardware resources. An appealing alternative is then to implement the MAC layer on the host machine, by using high-level languages and libraries available in GNU Radio. However, MAC protocols require highly precise packet timing and small, precise interframe spacings in the order of microseconds. We observed that GbE link and the GNU Radio processing latency are in the order of milliseconds \[44\]. Hence, time-critical radio or MAC functions cannot be placed in the host machine. Hybrid solutions, based on soft-core processors implemented in the embedded FPGA (e.g., Microblaze, ZPU) running software-defined protocols \[44, 51\] are also possible. Protocols and algorithms can be implemented using high level programming languages, e.g., C or C++, and then executed inside the FPGA.

In the current system architecture the MAC layer is implemented in HDL, and the FPGA setting registers enable partial reconfiguration at runtime. We are currently working on the implementation of a hybrid solution \[44, 51\] based on a soft-core processor implemented in the embedded FPGA.

### 4.4 Tx and Rx HDL Architecture

The default USRP HDL design operates on digital waveforms coming from and going to the host machine and performs only digital down and up conversion (DDC/DUC), decimation and interpolation. PHY and MAC layer digital processing takes place on the host machine. We followed a different approach, and we customized and extended the USRP HDL code to implement UsWB PHY and MAC layer operations in the FPGA.

#### 4.4.1 Custom Transmitter Chain

Figure 4.4 shows a block diagram of the custom transmitter logic. Since PHY and MAC functionalities are implemented on the FPGA, input data are raw information bits that need to be packetized and encoded in digital waveforms. After MAC and PHY layer operations, the custom transmitter logic outputs the digital quantized signal to be transmitted. This is then digital-to-analog converted, amplified and finally converted into an ultrasonic signal by the transmitter transducer.
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

TRANSMITTER MAC Finite State Machine. The UsWB MAC-layer data structure, i.e., UsWB packet\(^1\) is created in the FSM-Tx. The UsWB packet is then serialized, i.e., converted into a sequence of bits, and forwarded to the next module in the chain, i.e., Symbol Mapping. The FSM-Tx also controls the time-hopping frame length \(N_h\) and spreading code length \(N_s\) used by the PHY layer, according to feedback information from the receiver.

In Fig. 4.5 we show the UsWB packet structure. The UsWB packet starts after a Packet Synchronization Preamble (PSP) and a Time-Hopping Synchronization Preamble (THSP). The former enables coarse synchronization that allows the receiver to detect an incoming packet, while the latter allows identifying the exact start time of the time-hopping frame. The packet header is delimited by a 16-bit Start Packet Delimiter (SPD). This is followed by a 16-bit MAC Packet Size field representing the payload length in bytes. Then, a 3-bit Type of Packet is used to distinguish between R2T and C2T packets, ACK and NACK packets, and data packets. This is followed by two 6-bit fields representing the transmitter and receiver ID. Finally, an 8-bit Time-Hopping Frame Length Feedback (THFLF) and an 8-bit Spreading Code Length Feedback (SCLF) contain the feedback values piggybacked in ACK or NACK packets. A 16-bit packet checksum follows the payload to verify that the packet has been received correctly.

The UsWB packet is then serialized, i.e., converted into a sequence of bits, and forwarded to the next module in the chain, i.e., Symbol Mapping. The FSM-Tx also controls the time-hopping frame length \(N_h\) and spreading code length \(N_s\) used by the PHY layer, according to feedback information from the receiver.

TRANSMITTER PHY Layer. The first block of the transmitter PHY layer chain is Symbol Mapping. Here, raw information bits are mapped into \{-1,1\} binary symbols. The binary symbols are then spread in chips by the Spreading Code module following a pseudo-random spreading code. For each symbol, this block outputs \(N_s\) chips in \{-1,1\}. Chips are then forwarded to the Time-Hopping module that spreads them in time according to the selected time-hopping pattern. The output of this block is a sequence of \{-1,1\} chips, one per time-hopping frame. Finally, the Pulse Shaping module maps the incoming chips to position-modulated pulses. The output is a train of position-modulated pulses following a predefined time-hopping pattern, as described in (4.2).

Pulse Shaping. The Pulse Shaping module consists of a Finite Impulse Response (FIR) filter whose coefficients, i.e., taps, represent the samples of a 4th-order derivative gaussian pulse. The original design discussed in [41] was based on a 2nd-order derivative gaussian pulse. However, to match the central frequency and bandwidth requirements of the ultrasonic transducers in use, we adopted a higher-order derivative gaussian pulse characterized by higher central frequency and lower relative bandwidth [52]. In Fig. 4.6a and Fig. 4.6b

\(^1\)We intentionally use the word packet instead of frame to avoid confusion with the time hopping frame in Section 4.1.2.
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

Figure 4.6: a) 2nd-order and 4th-order derivative gaussian pulses, b) compared in frequency domain with the Ultran WS37-5 frequency response.

The two pulse shapes are compared in the time and frequency domains. Figure 4.6b also shows a 4th-order derivative pulse shaped to match the frequency response of the Ultran WS37-5 ultrasonic transducer. The pulse duration is approximately 300 ns, with a PPM shift of 60 ns, within a time-hopping slot ($T_c$ in Section 4.1.2) of 360 ns. The resulting maximum raw chip rate is 2.78 Mchip/s, which results in a maximum raw data rate of 2.78 Mbit/s when time-hopping frame and spreading code lengths are both set to 1.

Transmitter Setting Register Manager. The Transmitter Setting Register Manager (SRM-Tx) is in charge of routing configuration parameters written by the host machine into the setting registers discussed in Section 4.3.1. Whenever the host machine updates any of the setting registers, the SRM-Tx is triggered to read the register content and route it to the destination module. This block enables real-time reconfiguration thus enhancing the transmitter flexibility.

The communication system is designed to allow real-time reconfiguration of several parameters, i.e., spreading code and spreading code length, time-hopping frame length and time-hopping sequence, packet payload size, SPD sequence and SPD length, and length of the preambles. One can also change the pulse shape in real time through FIR filters with reloadable taps. Moreover, by carefully rerouting the binary flow, we can use the registers to enable and disable selected modules to reconfigure in real time the entire chain structure, thus modifying the communication system architecture. For example, we can implement adaptive modulation by switching between different Symbol Mapping modules at runtime to change the symbol constellation used. Moreover, we can disable the Time-Hopping module or the Spreading Code module to obtain pure spreading-code or time-hopping based schemes.

4.4.2 Custom Receiver Chain

The custom receiver chain, illustrated in Fig. 4.7, implements receiver UsWB PHY and MAC layer functionalities. The received ultrasonic signal is converted to an electrical signal by the RX transducer. The signal is amplified by the LNA, and analog-to-digital converted by the USRP ADC. Then, the digital waveform
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

Figure 4.7: Block scheme of the custom receiver logic.

is processed by the custom receiver chain in the FPGA. After PHY and MAC operations, the custom receiver chain outputs a binary stream representing the received decoded data.

**Receiver MAC Finite State Machine.** The Receiver MAC Finite State Machine (FSM-RX) implements UsWB MAC protocol functionalities and coordinates the PHY-layer logic. The FSM-Rx detects the received packet based on information coming from preamble detectors, and triggers the PHY layer module to start processing the received waveform. Finally, it decodes the received bits based on the output of the PHY layer operations on the received digital waveforms. Moreover, FSM-Rx estimates the level of interference, and accordingly chooses the optimal pair of time-hopping frame length and spreading code length, i.e., those that maximize the communication throughput while keeping the bit error rate (BER) under a predefined threshold - see [41] for details.

**Preamble Detectors.** The preamble detectors are designed to achieve packet and time-hopping synchronization. The former enables coarse synchronization by identifying the presence of an incoming packet. The latter identifies the exact start point of the time-hopping frame.

The Packet Synchronization Preamble (PSP) consists of a train of pulses positioned in consecutive time slots. The PSP detector includes a single-rate FIR filter used as a correlator, a squaring module, an integrator, and a threshold-based plateau detector. The FIR filter is used to correlate the incoming pulses with a 5th-order derivative gaussian pulse, the filter impulse response. Squaring and integrating the correlator output ideally results in a constant output for the whole PSP duration, i.e., a plateau. Therefore, the packet can be coarsely detected by finding the plateau. By using a dynamic threshold adaptation, this procedure can be made independent of the noise floor.

Fine synchronization is performed by the Time-Hopping Synchronization Preamble (THSP) detector, which includes a single-rate FIR filter used as correlator, and a threshold-based correlation peak detector. The THSP consists of a train of pulses, each positioned in the first time slot of consecutive time-hopping frames. By correlating the THSP with a 5th-order derivative gaussian pulse, we obtain a peak in the first slot of each time-hopping frame. The beginning of the time-hopping frame is determined by the threshold-based peak detector. Again, the threshold is dynamically adapted to the noise floor level.

**Receiver PHY Layer.** The receiver PHY layer module implements the bit decoding operations that
can be formally expressed as

\[ \sum_{j=0}^{N_s-1} a_j \int_{c_j T_c - j T_f}^{c_j T_c + (c_j+1) T_c - j T_f} s(t) \cdot c(t) \, dt \, \, d_{-1} \sum_{d_1} 0, \tag{4.3} \]

where \( c(t) \) is the correlator function, while all the other symbols follow the signal model presented in Section 4.1.2.

The first module in the receiver PHY chain is the Pulse Correlator, i.e., a decimator FIR filter with a 5th-order derivative gaussian pulse impulse response. The Pulse Correlator outputs one sample per time slot, and the FIR impulse response, i.e., the correlator function \( c(t) \), is selected in such a way as to give a zero output for an empty slot, a positive value for a slot containing a ‘-1’ chip, and a negative value for a slot containing a ‘1’ chip. The correlation output goes into the Time-Hopping Deframer, which collects the non-zero inputs located according to the Time-hopping sequence used at the transmitter. The Time-Hopping Deframer determines the integral intervals and the Pulse Correlator performs the actual integration in (4.3). Finally, the Code Despreader inverts the spreading operation by weighting the correlation output with the spreading code originally used at the transmitter and summing these over the spreading code length. This operation corresponds to the weighted sum in (4.3). Based on the result of the despreading operation, the FSM-Rx makes a decision on the received bits. If the resulting sum is positive a ‘-1’ symbol is received \( (d_{-1}) \), otherwise a ‘1’ symbol is received \( (d_1) \).

Receiver Setting Register Manager. The Receiver Setting Register Manager (SRM-Rx) provides the same functionalities as the SRM-Tx. The SRM-Rx operates on registers different than those used in the transmission chain. Therefore, the transmitter and receiver chains of each node can be independently reconfigured in real time.

Interference Level Estimation. The Interference Level Estimation module is needed by the UsWB MAC protocol to perform the rate-maximizing adaptation described in detail in [41]. The level of interference is estimated in terms of number of interfering pulses per time-hopping frame. The estimation module consists of a single-rate FIR filter whose impulse response is a 5th-order derivative gaussian pulse, a squaring module, an integrator and a peak detector with dynamically adapted threshold. By comparing at each time slot the integrator output with the threshold, we can detect the presence of pulses, and therefore how many pulses are received in a time-hopping frame. The level of interference is obtained by averaging over the entire packet reception time.

### 4.5 UsWB Performance Evaluation

In this section we demonstrate the feasibility of ultrasonic intra-body communications through testbed experiments. We start by evaluating the physical layer performance of the prototype we developed in terms of BER by varying the transmit power, the time-hopping frame and spreading code length and the level of interference in the channel. Then, we show how MAC adaptation allows a pair of nodes to adapt
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

Figure 4.8: Two nodes communicating through a human-kidney phantom.

Figure 4.9: BER in the absence of interference as a function of the SNR.

the communication rate according to the level of ultrasonic interference, to maximize the throughput while satisfying packet drop rate reliability constraints.

4.5.1 UsWB PHY Layer

The testbed setup consists of two ultrasonic software-defined nodes communicating through a human-kidney phantom (Fig. 4.8). The two nodes ultrasonic transducers are positioned in the opposite sides of the phantom smaller dimension, i.e., 10 cm, and the kidney is centered in the background gel such to be aligned between the two ultrasonic transducers. To guarantee repeatability of the experiments, we generate interference from co-located transceivers artificially by injecting interfering pulses at the transmitter. The position of the interfering pulses inside the time-hopping frame is given by a pseudo-random generator HDL module, i.e., a Linear Feedback Shifter Register (LFSR).

BER Vs SNR. First, we evaluate the BER as a function of the SNR per pulse measured at the receiver, in the absence of external interference, with fixed time-hopping frame and spreading code length. We define the SNR per pulse as the ratio between the energy per pulse $E_p$ and the noise power spectral density $\eta$

$$SNR = \frac{E_p}{\eta} = \frac{P_R T_p}{P_N / B}$$

(4.4)
where $P_R$ is the received signal power, $T_p$ is the pulse duration, $P_N$ is the receiver noise power and $B$ is the transmission bandwidth.

By connecting a variable-gain attenuator between the LFTX daughterboard and the power amplifier we vary the input power at the Tx transducer between $-7 \, \text{dBm}$ and $-21 \, \text{dBm}$, to obtain values of SNR between 23 and 9 dB, respectively. In Fig. 4.9, the resulting BER is depicted for time-hopping frame length and spreading code length pairs $(2,2)$ and $(2,4)$. We observe that, as expected, the BER is a decreasing function of the SNR and that by using longer spreading code the BER is further reduced. If we further decrease the SNR at the receiver, i.e., to 7 dB (a transmit power of $-23 \, \text{dBm}$), communication fails altogether due to limitations in the current time synchronization scheme. In the considered “kidney” setup, the UsWB prototype achieves $347.21 \, \text{kbit/s}$ with a $10^{-6}$ BER at $13 \, \text{dB SNR}$, which corresponds to an input power at the Tx transducer of about $-17 \, \text{dBm}$ ($\approx 20 \, \mu \text{W}$). A data rate up to about $700 \, \text{kbit/s}$ can be achieved (also with $10^{-6}$ BER) with a $(2,2)$ pair increasing the input power to $-14 \, \text{dBm}$ ($\approx 40 \, \mu \text{W}$), i.e., $16 \, \text{dB SNR}$. Lower-power transmissions are also possible by compensating with longer spreading code. For example, in the current implementation, for a Tx power of $-21 \, \text{dBm}$ ($\approx 8 \, \mu \text{W}$), i.e., $9 \, \text{dB SNR}$, and with a spreading code of 20 chips, we obtain a data rate of $70 \, \text{kbit/s}$ with a BER lower than $10^{-6}$.

Energy conversion losses can be reduced with custom-designed ultrasonic transducers with higher coupling electromechanical efficiency to further reduce the Tx power requirements. Moreover, we acknowledge that there is significant room for improving the current time synchronization scheme, after which it will be possible to operate at even lower SNRs.

**BER Vs Time-Hopping and Spreading Code.** In Fig. 4.10 (top), we evaluate the BER by varying the time-hopping frame and spreading code length with 4 interfering pulses per frame. Since we focus on the effect of the interference, we set the the input power at the Tx transducer to $13 \, \text{dBm}$, to create high-SNR condition, i.e., $43 \, \text{dB SNR}$. In Fig. 4.10 (bottom) the experimental results are compared with BER curves
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

![Graph showing Bit Error Rate vs Frame Length for different numbers of interfering pulses and spreading codes.]

Figure 4.11: a) Experimental BER as a function of time-hopping frame length, with a 4-chip spreading code for different number of interfering pulse per frame, b) compared with simulated BER.

obtained by simulating the same scenario with the UsWB PHY layer simulator described in [41]. For fairness, the pulse shape used in the simulator is obtained by recording a real pulse shape as received in testbed experiments. Therefore, we consider the signal distortion introduced by amplifiers and transducers, and the scattering and reflection effects introduced by the ultrasonic phantom. However, the imported deterministic measurement does not consider the time-variability of the real testbed conditions, e.g., operating temperature, humidity, and coupling of the transducer with the phantom surface, among others. We observe that, as expected, the BER is a decreasing function of the time-hopping frame length and the spreading code length, thus confirming the simulation results in our previous work [41].

**BER vs Interference.** We repeat the same experiment by setting the spreading code length to 4 chips, varying the time-hopping frame length and the number of interfering pulses per frame (Fig. 4.11). Here, we observe that the transmitter can adapt to different level of interference to satisfy the BER requirement by increasing the frame length. For example, a $10^{-3}$ BER for a 4-chip spreading code can be achieved with a time-hopping frame of size 8 with one interfering pulse, as well with a time-hopping frame of size 12 with two interfering pulses. The BER can be further reduced by increasing the spreading code length.

4.5.2 UsWB MAC Layer

We consider a pair of ultrasonic software-defined nodes and evaluate how the UsWB MAC protocol adapts the link parameters to compensate for varying levels of interference, i.e., multiple concurrent transmissions. The level of interference is defined in terms of number of interfering pulses within a time-hopping frame. If we assume that all nodes measure the same level of interference, that is, all network nodes are close enough to be all in the same transmission range, i.e., as in the implicitly cooperative problem in [41], the number of interfering pulses per time-hopping frame coincides with the number of co-located active Tx-Rx pairs.
UsWB Rate-Maximizing Adaptation. First, we evaluate the UsWB MAC rate-maximizing adaptation as a function of the level of interference. We transmit 250 packets and increase the level of interference every 50 packets, from zero interfering pulses per time-hopping frame to four. In Fig. 4.12 (bottom), we show the estimated level of interference at the receiver. We observe that the receiver occasionally overestimates the number of interferers, which however does not affect the performance in terms of packet drop rate. According to the UsWB protocol, based on the interference estimation, the receiver computes the optimal pair of time-hopping frame and spreading code length, and piggybacks these in ACK/NACK packets. In the current implementation, the optimization problem in [41] is solved offline, and the solution is then loaded into the receiver. By performing a lookup table operation, the receiver finds the optimal pair corresponding to the measured level of interference and BER requirements. Figure 4.12 (middle) shows the time-hopping frame and spreading code length used by the transmitter after the ACK/NACK is received. As expected, these vary according to the interference estimate at the receiver. The resulting data rate of the transmitter is shown in Fig. 4.12 (top).

Throughput and Packet Drop Rate. To verify the effectiveness of the rate adaptation, we evaluate throughput and packet drop rate at the receiver while varying the level of interference. We define throughput as the average bit rate of correctly received information during a time window. The packet drop rate is defined as the ratio between the number of packets dropped and the number of packets generated at the application layer. We set the packet payload length to 512 bytes and vary the level of interference form zero to four every 5000 packets. In Fig. 4.13 (top), we show throughput of a single Tx-Rx pair along with the cumulative throughput considering all the concurrent interfering Tx-Rx pairs. Figure 4.13 (bottom) shows the resulting packet drop rate compared with the maximum packet drop rate threshold (4%) corresponding to the maximum BER constraint ($< 10^{-5}$) and packet size.
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

Figure 4.13: Throughput (top) and packet drop rate (bottom) as a function of the number of interferers.

**Optimal Packet Size.** Finally, we investigate how the packet size affects throughput and packet drop rate. We set the number of interferers to one, and transmit a fixed amount of data ($\approx 2$ MBytes) with different packet sizes. In Fig. 4.14a, we observe that the resulting throughput is maximized between 512 and 1024 bytes. The optimal packet size is influenced by the packet drop rate and protocol overhead. For a given BER, the packet drop rate is determined by the packet size, i.e., the longer the packet the higher the packet drop rate. At the same time, longer packets result in less protocol overhead (packet header, ACK/NACK control packets, and propagation delay). Figure 4.14b shows that the packet drop rate increases when the payload size increases. However, the maximum packet drop rate constraint is always satisfied.

4.6 Video Monitoring for Implantable Devices

In this Section, we show how UsWB can enable video monitoring medical applications for implantable devices, and we evaluate the UsWB video streaming performance in terms of peak signal-to-noise ratio (PNSR) and similarity (SSIM) index.

Camera equipped implantable sensors can enable continuous remote monitoring of internal body organs and systems. For example, wireless capsule endoscopy (WEC) uses a pill-sized ingestible cameras to inspect the entire gastrointestinal tract and detects various anomalies, e.g., internal bleeding and tumors, among others. WCE can replace intrusive and limited examination techniques such as gastrointestinal endoscopy that are often incapable of reaching critical area of the gastrointestinal tract. Using WCE, the recorded video is wirelessly streamed in realtime from the ingestible capsule to an external image-recording belt carried by the patient. The image-recording belt receives and stores the collected video, which can be then accessed, processed and analyzed by the doctor.

We consider both the WCE and the image-recording belt to be UsWB enabled. With a data rate up to 700 kbit/s, UsWB can easily accommodate relative high quality video streaming, and the UsWB rate adaption
capabilities discussed in Section 4.5 can be leveraged to adapt the video bitrate to the channel condition and level of interference in the channel. Moreover, an ultrasonic WEC eliminates any potential conflict with existing RF communication systems and overcrowded RF environments, and reduces the risk of potential eavesdropping and jamming attacks that can compromise the monitoring operations, and therefore the patient diagnosis.

4.6.1 Streaming Video Performance

**SSIM Vs SNR Per Pulse.** Here, we evaluated the video performance of an ultrasonic WCE based on UsWB in terms of PSNR and SSIM as a function of the SNR per pulse measured at the receiver, defined as in (4.4). The PSNR compares the maximum possible image energy to the noise energy, which has shown to have higher correlation with the subjective image quality perception than conventional SNR [53]. The SSIM index also measures the relative quality between two images, and it has been shown to be more consistent with human eye perception than PSNR [54].

In WCE, ultrasonic signals would propagate through a multilayered medium composed mainly by soft-tissues, e.g., muscle, fat and skin, among others. However, for consistency with the PHY and MAC layer results discussed in Section 4.5, in these experiments we use the same testbed setup shown in Fig. 4.8 based on a human-kidney ultrasonic phantom. Table 4.2 shows how the human-kidney captures accurately the propagation characteristics of soft-tissues media such as muscle, fat and skin [6].

For this experiment we selected an 11 s video from the Atlas of Gastrointestinal Video Endoscopy [55] that shows a gastroscopy of a 34 years old male’s stomach. We transmit several times the video sequence over a UsWB link. At the receiver, we average the resulting PSNR and SSIM of the channel-affected received video over all the transmission repetitions. Finally, we set the UsWB packet size to be equal to 512 byte,
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

Table 4.2: Ultrasonic phantom acoustic characteristics compared to tissues

<table>
<thead>
<tr>
<th>Tissue</th>
<th>Speed, ( v )</th>
<th>Attenuation, ( \alpha )</th>
<th>Density, ( \rho )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phantom</td>
<td>1550 m/s</td>
<td>2 dB/cm @ 5 MHz</td>
<td>1030 Kg/m(^3)</td>
</tr>
<tr>
<td>Muscle</td>
<td>1529 m/s</td>
<td>0.54 dB/cm @ 4.3 MHz</td>
<td>1038 Kg/m(^3)</td>
</tr>
<tr>
<td>Fat</td>
<td>1487 m/s</td>
<td>0.56 dB/cm @ 6 MHz</td>
<td>939 Kg/m(^3)</td>
</tr>
<tr>
<td>Skin</td>
<td>1729 m/s</td>
<td>1.06 dB/cm @ 5 MHz</td>
<td>1110 Kg/m(^3)</td>
</tr>
</tbody>
</table>

Figure 4.15: Block scheme of the video streaming Tx and Rx setup.

which, as shown is Section 4.5.2, represents a good tradeoff between packet drop rate and protocol overhead.

In Fig. 4.15 we show a block scheme of the the experiment testbed setup. At the transmitter, the raw video (.yuv) is encoded in a H.264 stream using GStreamer open source multimedia framework [56], and it is temporarily stored in a FIFO memory. Our custom-made GNU Radio module reads from the FIFO memory chunks of the H.264 byte stream, and passes the resulting bitstream to the USRP devices. In the USRP, the bitstream is modulated according to the UsWB transmission scheme implemented on the embedded FPGA, converted in an ultrasonic signal through the transducers and and transmitted in the ultrasonic intra-body channel, emulated through the ultrasonic phantom. At the receiver, the above operations are inverted, and the received video stream is finally analyzed in terms of PSNR and SSIM using the EvalVid video quality evaluation tool [57].

Figure 4.16 shows the PSNR (top) and SSIM (bottom) performances as a function of the SNR per pulse measured at the receiver, assuming frame length of 2 time slots and code length equals to 2 and 4, i.e, pairs (2,2) and (2,4). The two pairs enable data rates of approximately 700 kbit/s and 350 kbit/s, respectively. We observe that, as expected, the video quality increases with higher SNR. For SNR equal to 10 dB, synchronization errors make the video quality very low (virtually no video frames are correctly visualized). For values of SNR equal to 11 dB, the pair (2,4) offers relative good video quality as compared to the pair (2,2). Specifically, pair (2,4) results in PSNR equal to 30 dB, and SSIM above 0.8, which are measures of good image quality. For SNR greater or equal than 12 dB both pairs offer good image quality. Above 14 dB SNR, both pairs ensure virtually no distortion in the received images.
CHAPTER 4. EVALUATION OF ULTRASONIC INTRA-BODY COMMUNICATIONS

Figure 4.16: PSNR (top) and SSIM (bottom) as a function of the SNR per pulse measured at the receiver for the pair (2,2) and (2,4).

4.7 Ultrasonic Channel Characterization

The ultrasonic channel can be statistically characterized through time domain measurements [23]. We use a digital sampling oscilloscope to record the response to a probing signal, and obtain a statistical model of small-scale fading for the ultrasonic intra-body channel.

Small-scale Fading Model. Small-scale fading describes variations of the channel over a period of time where the channel can be considered wide-sense stationary. In RF wireless communications, the signal envelope affected by small-scale fading has been typically modeled using different distributions, such as Rice, Rayleigh, Nakagami, Log-normal, and Weibull. We characterize the envelope of ultrasonic signals transmitted through the human-kidney phantom discussed in Section 4.2.

To this purpose, we transmit short ultrasonic pulses through the ultrasonic phantom, and record the output by connecting an Rx ultrasonic transducer to a digital sampling oscilloscope. We processed the collected data to identify the statistical distribution that best fits the recorded envelope distribution.

In Fig. 4.17 we show the recorded envelope distribution. The experimental result is least-square fitted with a Nakagami and a generalized Nakagami distribution. The probability density function (PDF) of the signal envelope $\rho$ can then be expressed as

$$f(\rho; m, \Omega, s) = \frac{2sm^m \rho^{2sm-1}}{\Gamma(m)\Omega^m} e^{-\frac{\rho}{\Omega}^{2s}} U(\rho)$$  \hspace{1cm} (4.5)

where $m$ is the Nakagami parameter, $\Omega$ is a scaling parameter, $s$ is the generalization parameter, $U(\cdot)$ is the unit-step function, and $\Gamma(\cdot)$ is the gamma function. For $s = 1$, (4.5) gives the original Nakagami distribution.

Nakagami distribution allows to match empirical data in very general environments, under various scattering conditions. By varying the value of the Nakagami parameter $m$, we obtain different distributions.
Figure 4.17: Envelope amplitude distribution of the received signal fitted with a Nakagami distribution, and a generalized Nakagami distribution.

such as Gaussian distribution ($m = 0.5$), generalized Rice distribution ($0.5 < m < 1$), Rayleigh distribution ($m = 1$) and Rice distribution ($m > 1$) [9]. Parameter $s$ takes into account the so called tail effect [58]. When $s < 1$, the distribution presents heavy upper tails; for $s > 1$, we have shorter upper tails. Therefore, the Nakagami amplitude distribution can describe different scenarios by simply varying the value of $m$ and $s$.

We observe that, in both cases, the values of $m$ correspond to a generalized Rice distribution, which is typically encountered in the presence of randomly located scatterers together with periodic alignment of scatterers [9]. This is indeed the case for the human-kidney phantom used in the experiment, which consists of a scattering medium, and whose boundary may represent potential aligned scatters. Similar results have been observed in medical ultrasonic imaging. For example, in [59] the amplitude statistics of a signal reflected from a human-kidney were modeled using a Nakagami distribution.

4.8 Conclusions

In this chapter, we discussed design and implementation of a software-defined testbed architecture for ultrasonic intra-body area networks, and experimentally demonstrated for the first time the feasibility of ultrasonic communications in biological tissues. We discussed our prototype implementation and showed that our prototype can flexibly trade performance off for power consumption, and achieve, for bit error rates (BER) no higher than $10^{-6}$, either (i) high-data rate transmissions up to 700 kbit/s at a transmit power of -14 dBm ($\approx 40 \mu W$), or (ii) low-data rate and lower-power transmissions down to -21 dBm ($\approx 8 \mu W$) at 70 kbit/s. We showed how the considered MAC protocol allows multiple transmitter-receiver pairs to coexist and dynamically adapt the transmission rate according to the channel and the level of interference condition, to maximize the throughput while satisfying predefined reliability constraints. We also showed how UsWB can be used to enable a video monitoring medical application for implantable devices. Finally, we proposed (and validated through experiments) a statistical model of small-scale fading for the ultrasonic intra-body channel.
Chapter 5

U-Wear: Software-defined ultrasonic networking for wearable devices

Wearable medical sensing and actuating devices with wireless capabilities have become the cornerstone of many revolutionary digital health applications [60]. Wearable electrocardiography (ECG) devices and blood pressure sensors can enable remote cardiovascular monitoring for early diagnosis of cardiac arrhythmias and hypertension [61], and therefore prevent heart failures and strokes. Skin patches with wireless connectivity can be arranged in a closed-feedback-loop drug delivery system [62]. For instance, a sensor patch can measure the level of subcutaneous blood glucose, while a drug pump patch can adaptively deliver the required amount of insulin [63]. Motion sensors, e.g., accelerometers, can collect large amounts of data for fitness and medical applications. For example, wireless motion trackers can record athletes’ step rate, speed, and acceleration for performance monitoring. Similar devices can also collect information for post-surgery telerehabilitation in case of lower-limb injuries or strokes [64], or measure the motor degradation of patients affected by Parkinson disease [65]. Likewise, by correlating motion with sleep activity, the same sensors can monitor the REM sleep duration of a patient and provide information on the development of post-traumatic stress disorders [66].

Existing wireless wearable medical devices are connected through radio frequency (RF) electromagnetic waves. Standards in use are often scaled down versions of wireless technologies (e.g., Bluetooth and WiFi), with little or no attention to the peculiar characteristics of the human body and to the severe privacy and security requirements of patients. For example, many commercial activity trackers [67], as well as smart glasses [68], smart watches [69] and smart clothing [70] connect to smartphones using Bluetooth or WiFi. Alternatively, other medical monitoring solutions [71, 72] use proprietary RF-based technologies to collect medical data in a centralized manner. We contend that this is not the only possible approach, and that RF-based technologies have several limitations that can negatively affect the patients’ medical experience with wearable devices.

Limitations of RF Technology. First, the RF frequency spectrum is scarce and already crowded with many devices interfering with one another. At the same time, the number of wireless devices that compete
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

to access the RF spectrum is growing exponentially. This includes wireless sensors, but also more largely and pervasively deployed RF devices such as WiFi and Bluetooth, and even microwave ovens. Quoting the FDA’s recent guideline on wireless medical devices, “an increasingly crowded RF environment could impact the performance of RF wireless medical devices” [3]. Therefore, RF-based technologies raise serious concerns about potential interference from existing RF communication systems that can unintentionally undermine the reliability and security of the wearable network, and ultimately the safety of the patient. Second, RF communications can be easily jammed, i.e., intentionally disrupted by artificially generated interference, or eavesdropped by malicious agents using cheap and off-the-shelf equipment, i.e., a simple radio device. Jamming may not even be illegal on ISM spectrum frequencies where devices are allowed to transmit with no need for special permissions. This raises major privacy and security red flags for wearable networks, and a risk for the patient. Third, the RF spectrum is strictly regulated. This clearly constrains the system in terms of flexibility in allocating spectrum resources. Fourth, the medical community is still divided on the risks caused by continuous exposure of human tissues to RF radiation - the World Health Organization classifies RF waves as “possibly carcinogenic to humans” [73]. Therefore, a massive deployment of RF wearable devices on the body may represent a potential risk for the patient. Finally, the dielectric nature of the human body also affects the coupling between on-body RF antennas and the body itself. In particular, the gain and the radiation pattern of the antenna deteriorate because of the contact or proximity with the human body [74], while the resonant frequency and the input impedance of the antenna may shift from their nominal values.

Based on these observations, in this chapter, we propose to use ultrasonic waves to interconnect wearable devices, and present U-Wear, the first software-defined networking framework for wearable medical devices based on ultrasonic communications.

**Advantages of U-Wear.** U-Wear has several advantages over traditional networking frameworks based on RF communications.

i. U-Wear eliminates any potential conflict with existing RF systems and overcrowded RF environments.

ii. The ultrasonic frequency spectrum is (at least currently) unregulated and enables nodes to flexibly adapt the occupied frequency to specific requirements such as maximum level of tolerable co-channel interference, maximum tolerable channel multipath and Doppler spreading in the channel and minimum data rate needed at the application layer, among others.

iii. As compared to RF waves, ultrasonic waves do not easily penetrate through solid materials and do not propagate far in air; therefore, ultrasonic communication systems are inherently more secure with respect to eavesdropping and jamming attacks that would require close proximity between the attacker and the victim.

iv. The medical experience of the last decades has demonstrated that ultrasounds are fundamentally safe, as long as acoustic power dissipation in tissues is limited to predefined safety levels [18, 39].

v. By equipping wearable devices with ultrasonic transducers, U-Wear can implement ultrasonic power transmission schemes [75] that enable wireless battery charging functionalities.
vi. On-board ultrasonic transducers can be used to enable acoustic localization and tracking, which are known to have better accuracy than RF-based systems because of the low propagation speed of sound in air [76].

vii. U-Wear can easily be interfaced with ultrasonic intra-body networks [41], and can work as a bridge between implantable intra-body sensors and the external world.

viii. The U-Wear software-defined framework runs on general-purpose hardware; thus, it allows commercial devices, such as smartphones, laptops and smart-TVs, to communicate with wearable devices in the near-ultrasonic frequency range, i.e., $17 - 22$ kHz, using commercial-off-the-shelf (COTS) speakers and microphones [77].

ix. Finally, the U-Wear software-defined ultrasonic networking functionalities can be reconfigured to adapt to application requirements, offering more flexibility with respect to RF-based networking systems entirely implemented in hardware, e.g., Bluetooth or WiFi.

U-Wear consists of a set of software-defined cross-layer functionalities designed to network ultrasonic wearable devices that offer real-time reconfigurability at different layers of the protocol stack, i.e., the physical (PHY), data link, network and application layer. Specifically, U-Wear encloses a set of PHY, data link and network functionalities that can flexibly adapt to the application and system requirements to efficiently distribute information among wearable devices. U-Wear also offers real-time reconfigurability at the application layer to provide a flexible platform to develop medical applications. In particular, sensor data processing applications running in the nodes are decomposed into primitive building blocks that can be arbitrarily arranged to create new sensing applications that fit user requirements.

As a proof of concept, we design two prototypes that implement the U-Wear framework and operate in the near-ultrasonic frequency range, i.e., $17 - 22$ kHz, using COTS speakers and microphones. Specifically, we operate at $18$ kHz with a bandwidth of about $2$ kHz. Despite the use of COTS audio components not optimized for operations at high frequency, our prototypes (i) can achieve data rates up to $2.76$ kbit/s with bit-error-rate (BER) lower than $10^{-5}$ at a transmission power of $20$ mW; (ii) enable multiple nodes to share the same medium, with tradeoffs between packet delivery delay and packet drop rate; and (iii) implement reconfigurable data processing to extract medical parameters from sensors with high accuracy. Moreover, U-Wear proposes for the first time the use of a Gaussian minimum-shift keying (GMSK) signaling scheme for the near-ultrasonic frequency range that allows our prototypes to achieve relatively high data rates when compared to previously proposed near-ultrasonic systems; more importantly, it ensures virtually inaudible\textsuperscript{1} click-free transmission because of the GMSK phase-continuity as we will discuss in Section 5.2.1.1.

In Section 5.1, we discuss the fundamentals of ultrasonic communications along the body. This study highlights the core differences between ultrasonic propagation and RF propagation in air, including the relatively low speed of sound compared to RF, Doppler spreading, and the multipath effect caused by reflections and scattering. In Section 5.2, we introduce U-Wear, the first networking framework for wearable medical devices, and discuss its architecture. We highlight design choices at different layers of the protocol

\textsuperscript{1}The upper threshold for human hearing is nominally $20$ kHz, however hearing starts degrading significantly above $15$ kHz.
stack made to overcome limitations posed by the propagation characteristics of ultrasonic waves in air. For example, two signaling schemes (GMSK and orthogonal frequency-division multiplexing) are selected because of their high spectral efficiency and resilience to multipath. Two different synchronization modes can be alternatively used for channels strongly affected by multipath or by Doppler effect. Upper layer protocols and functionalities are designed to address challenges posed by the long propagation delays of ultrasounds in air that might prevent accurate timing. In Section 5.3, we present the design and implementation of two prototypes. Hardware components are selected with the objective of striking a reasonable balance between keeping the hardware design simple and inexpensive, and meeting the communication performance requirements of the family of applications we consider. The software architecture has also been carefully designed to enable reuse of existing code and to meet the constraints posed by the resource constrained devices in use. In Section 5.4, we thoroughly discuss the performance of U-Wear through experimental results. In Section 5.6 we conclude the chapter.

5.1 Ultrasonic Communications in Air

Ultrasounds are mechanical pressure waves that propagate through elastic media at frequencies above the upper limit for human hearing, i.e., 20 kHz.

**Attenuation.** Two mechanisms mainly contribute to acoustic attenuation in air, i.e., spreading loss and absorption loss [78]. The former includes spherical spreading, i.e., the acoustic pressure falls off proportionally to the surface area of a sphere. The latter is mainly related to atmospheric absorption caused by the interaction of the acoustic wave with the gas molecules of the atmosphere, and is frequency-, temperature-, and humidity-dependent.

For a signal at frequency \( f \) [Hz] over a transmission distance \( d \) [m], the attenuation can be expressed in [dB] as

\[
A_{dB} = 20 \log_{10}(d) + d \alpha(f),
\]

(5.1)

where \( \alpha(f) \) [dB/m] is the absorption coefficient, which increases quadratically with the frequency, but also depends on the ambient atmospheric pressure, temperature, and the molar concentration of water vapor, i.e., humidity [79].

**Propagation Speed.** The propagation speed of acoustic waves in air is approximately 343 m/s at 20°C and at atmospheric pressure of 101.325 kPa, as compared to \( 3 \times 10^8 \) m/s for RF waves. The speed of sound in air increases with temperature and humidity, going from 331 m/s at a temperature of 0°C and 10% relative humidity to 351 m/s at a temperature of 30°C and 90% relative humidity.

**Doppler Spreading.** Doppler spreading occurs as a result of Doppler shifts caused by relative motion between source and receiver, and is proportional to their relative velocity. Doppler spreading generates two different effects on signals: a frequency translation, and a continuous spreading of frequencies that generates intersymbol interference (ISI), thus causing degradation in the communication performance. Since
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

the speed of sound is several orders of magnitude lower than the speed of electromagnetic waves, the resulting doppler effect is severe, even at relatively low speeds.

Reflections and Scattering. The on-body ultrasonic channel is composed of several interfaces between air and human body, and between air and on-body and near-body objects. Because of this inhomogeneous pattern, the on-body channel can be modeled as an environment with pervasive presence of reflectors and scatterers. The direction and magnitude of the reflected wave depend on the orientation of the boundary surface and on the acoustic impedance of the different media involved. Scattered reflections occur when an acoustic wave encounters an object that is relatively small with respect to its wavelength. Consequently, the received signal is obtained as the sum of numerous attenuated, possibly distorted, and delayed versions of the transmitted signal.

Air-coupled Ultrasonic Transducers. An ultrasonic transducer is a device that converts electrical signals into ultrasonic signals and vice versa. Ultrasonic transducers can be categorized into two main classes based on the physical mechanism that enables the conversion, i.e., piezoelectric and electrostatic transducers. A piezoelectric transducer produces a mechanical vibration through a thin piezoelectric element under an external voltage variation, and produces a voltage variation under an external mechanical vibration. In electrostatic transducers the fundamental mechanism is the vibration of a thin plate under electrostatic forces.

When sound passes across an interface between two materials, it is in part transmitted and in part reflected. To maximize the acoustic energy radiated by the transducer, the acoustic impedance of the radiating surface should match the acoustic impedance of the propagation medium. Today, microelectro-mechanical (MEMS) technology has enabled the fabrication of microscopic piezoelectric and electrostatic transducers, so-called Micromachined Ultrasonic Transducers (MUTs). With MUTs, the acoustic impedance can be controlled to match the external medium by manipulating the device geometry, making them ideally suited for air-coupled applications [80].

When the operating frequency of the ultrasonic communications falls in the near-ultrasonic frequency range, i.e., 17 – 22 kHz, acoustic waves can be recorded and generated using COTS components, such as microphones and speakers. Even though COTS components are often designed to operate at lower frequencies, i.e., 0 – 17 kHz, they can still sense and generate, albeit less efficiently, near-ultrasonic frequency waves. Since many commercial devices such as smartphones, tablets and laptops among others, are equipped with audio interfaces, they can support near-ultrasonic communications with no additional hardware [77].

5.2 U-Wear Architecture

U-Wear consists of a set of software-defined multi-layer functionalities that can be implemented on general-purpose processing units, e.g., microprocessors, microcontrollers or FPGAs, among others, to enable networked operations between wearable devices equipped with ultrasonic connectivity, i.e., air-coupled ultrasonic transducers, and sensing capabilities, i.e., sensors.

2The acoustic impedance is defined as the product between the density of a medium \( \rho \) and the speed of sound in the medium \( c \).
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

Figure 5.1: Overview of the U-Wear framework.

Figure 5.1 shows an overview of the U-Wear framework. U-Wear runs on a processing unit. It accesses an hardware analog-to-digital converter (ADC) and digital-to-analog converter (DAC) through hardware-specific system APIs. In the transmit (Tx) chain, the DAC collects and digital-to-analog converts U-Wear’s outputs, i.e., the waveforms to be transmitted, before passing these to the communication unit. In the receive (Rx) chain, the ADC analog-to-digital converts and passes to U-Wear the received waveforms coming from the communication unit. The communication unit consists of an ultrasonic transducer and an amplification stage, i.e., preamplifier in Rx chain and power amplifier in Tx chain. U-Wear also collects the analog-to-digital converted data coming from the sensing unit. U-Wear consists of (i) PHY layer functionalities, e.g., modulation and synchronization, (ii) data link layer functionalities including forward error control and medium access control (MAC) protocols, (iii) network layer functionalities, e.g., IPv4 and IPv6 support and content-centric networking, and (iv) application layer functionalities, i.e., reconfigurable sensing data processing and user interface.

5.2.1 Physical Layer

U-Wear PHY layer libraries define the signaling scheme, channel estimation, equalization, synchronization and forward error correction (FEC) functionalities.

5.2.1.1 Signaling Schemes

U-Wear offers two fully-functional signaling schemes, a narrowband scheme based on GMSK modulation, and a wideband scheme based on orthogonal frequency-division multiplexing (OFDM). Moreover, U-Wear includes a set of software-defined primitive blocks, e.g., programmable filters, and Fast Fourier Transform (FFT) modules, among others, that can be used to implement additional signaling schemes.

Narrowband GMSK. GMSK is a continuous-phase modulation (CPM) used worldwide in GSM cellular systems [81]. In frequency-shift keying (FSK) and phase-shift keying (PSK), information is encoded in the variations of the carrier frequency, or carrier phase, respectively. Since frequency and phase switches occur instantaneously, FSK and PSK signals do not have continuous phase. Phase discontinuity generates out-of-band power, leading to poor spectral efficiency. Moreover, in near-ultrasonic transmissions based on
COTS speakers and microphones the out-of-band power introduces audible noise (clicks), which makes the communication perceptible to humans.

Differently, GMSK signals have phase continuity, and each symbol is represented by a phase variation, from a start value to a final value, over the symbol duration. Thus, the initial phase of each symbol is determined by the cumulative total phase variation of all previous symbols. A Gaussian filter is used to smooth the phase variation and improve the spectral efficiency. The product between the signal bandwidth $B$ and the symbol time $T$ is a measure of the scheme spectral efficiency. Lower $BT$ product leads to higher spectral efficiency, but increases the intersymbol interference (ISI). Based on these characteristics, GMSK is the signaling scheme of choice for U-Wear narrowband communications in the near-ultrasonic frequency range based on COTS speakers and microphones. Thanks to its phase-continuity, GMSK enables click-free transmissions, and therefore is to be preferred over non-continuous-phase modulations such as FSK and PSK.

**Wideband OFDM.** OFDM has been extensively used in underwater acoustic communications [82] because of its robustness against frequency-selective channels with long delay spreads. The idea behind OFDM is to use a large number of closely spaced orthogonal sub-carriers, such that for each sub-carrier the channel is subject to flat fading. In each sub-carrier a conventional modulation is used, e.g., M-PSK or M-Quadrature-Amplitude-Modulation (QAM). OFDM offers high spectral efficiency and robustness against narrowband co-channel interference, intersymbol interference (ISI) and multipath fading. Finally, OFDM can be efficiently implemented using FFT and inverse FFT (IFFT) algorithms. These characteristics make OFDM ideal for ultrasonic communications based on wideband transducers.

### 5.2.1.2 Synchronization

Synchronization in U-Wear is achieved in two steps. First, an energy collection identifies any incoming packet, i.e., coarse synchronization. Once a packet is detected, the receiver performs a fine synchronization operation that identifies the exact starting point of the packet. Fine synchronization is achieved by correlating the received signal with a local copy of the preamble, i.e., a sequence that precedes each packet, which outputs a peak corresponding to the first sample of the packet. U-Wear offers two synchronization modes:

- **PN-sequence mode.** The pseudo noise (PN)-sequence mode uses PN-sequences as a preamble, i.e., binary sequences with sharp autocorrelation peak and low cross-correlation peaks, that can be deterministically generated. Specifically, we consider maximum length sequences (MLSs), a family of PN-sequences that can be generated in software and hardware through linear feedback shift registers (LFSRs). Because of their desirable correlation characteristics, PN-sequences have been widely used to enable strong resilience to multipath [83]. Therefore, they are well suited for ultrasonic in-air communications, as discussed in Section 5.1.

- **Chirp-based mode.** The chirp-based mode uses a chirp signal as preamble, i.e., a sinusoidal waveform whose frequency varies from an initial frequency $f_0$ to a final frequency $f_1$ within a certain time $T$. Chirp signals have been widely used in radars [84] due to their good autocorrelation and robustness against Doppler effect. In fact, a frequency-shifted chirp still correlates well with the original chirp, although
with lower amplitude and time-shifted peak. This characteristic makes chirp synchronization desirable in ultrasonic in-air communications under severe Doppler effect conditions as experienced, for example, under fast movement conditions as in sensors worn by athletes for performance monitoring. The price we pay for the Doppler robustness is higher cross-correlation peaks compared to PN-sequences that result in lower resilience to multipath.

5.2.1.3 Channel Estimation and Equalization

As discussed in Section 5.1, ultrasonic communications in air are strongly affected by multipath and Doppler spread, leading to frequency selectivity and ISI that compromise the bit recovery operations at the receiver. U-Wear implements channel estimation and equalization functionalities to estimate the channel impulse response (CIR) and mitigate the distortion produced by the channel.

Channel Estimation. U-Wear offers a training-based channel estimation approach that requires the presence of a training sequence known a-priori in the transmitted packet. In particular, U-Wear leverages the good autocorrelation property of the synchronization preamble sequence, discussed in Section 5.2.1.2, to estimate the CIR. By correlating the output of the channel, i.e., the received signal, with the input, i.e., the known preamble sequence, we obtain an estimate of the time-domain CIR [85].

Zero-forcing Equalization. U-Wear implements a linear equalization technique, zero-forcing (ZF) [86], that aims to minimize the ISI signal distortion produced by the channel. As the name suggests, a ZF equalizer is a finite-impulse-response (FIR) filter of order \( N \) that, for each input symbol, “forces to zero” the ISI components introduced by the \( 2N \) adjacent symbols. The filter taps are numerically calculated starting from an estimate of the CIR, which also accounts for the ISI effect.

5.2.1.4 Forward Error Correction

U-Wear offers a forward error correction (FEC) functionality based on Reed-Solomon (RS) codes. RS codes are linear block error-correcting codes widely used in data storage and data transmission systems. An RS encoder takes \( k \) information symbols and adds \( t \) parity symbols to make an \( n \) symbol block. Therefore, there are \( t = n - k \) overhead symbols. On the other hand, an RS decoder is able to decode the received \( n \)-symbol block, and can correct up to \( \frac{t}{2} \) data symbols that may contain potential errors due to the channel fluctuation or collisions with interfering packets. The RS coding rate can be defined as the ratio between the message length and the block length, i.e., \( k/n \).
5.2.2 Data Link Layer

The U-Wear data link layer provides a set of functionalities that allow multiple nodes to efficiently access the medium under the challenges posed by the ultrasonic in-air channel, e.g., long propagation delays, among others, as discussed in Section 5.1.

5.2.2.1 Network Configuration

U-Wear is designed to internetwork wearable devices in master/slave (M/S) or peer-to-peer (P2P) configurations. Both configurations can coexist in the same network in what we refer to as hybrid configurations. Figure 5.2 shows a hybrid configuration system design.

**Master-Slave Configuration.** In the M/S configuration, one node takes the role of master, i.e., network coordinator, while the remaining nodes operate as slaves. In this scenario, the network control is concentrated on a master node, typically with higher resources available, e.g., processing, memory, power and connectivity. For example, M/S configurations may be used in continuous monitoring systems where a master node, e.g., a smartphone or a laptop, is used to fetch, analyze and display data collected by wearable sensors. Wireless or wired Internet connectivity may allow the master node to connect the wearable network with a medical center where the patient’s data can be stored, and analyzed remotely.

**Peer-to-Peer Configuration.** In the P2P configuration, all the network wearable nodes are treated as peers. This scenario suits, among others, applications that require distributed coordination among nodes for closed-feedback-loop monitoring and actuating tasks. For example, this may include a skin patch drug-delivery system where a drug pump can trigger a sensor for measurement, or where a sensor may trigger the drug pump for drug injection after a measurement.

5.2.2.2 Medium Access Control Protocols

U-Wear offers three fully-functional multiple access protocols, i.e., polling, ALOHA and carrier sense multiple access (CSMA) with collision avoidance (CA), as well as primitive functions to implement custom protocols, e.g., idle listening, random backoff, or checksum-based error control mechanisms.
Polling Protocol. Polling is a deterministic access protocol for the M/S network configuration. In a polling scheme, the master node has complete control over channel access, while each slave node is granted access to the medium in a round-robin fashion.

ALOHA. ALOHA is a random access protocol where nodes do not check whether the channel is busy or idle before transmitting [87]. Nodes that want to transmit data simply access the channel and transmit the data. When collisions occur, nodes attempt retransmissions after a random time interval, i.e., backoff time.

Carrier Sense Multiple Access. CSMA/CA is a multiple access technique based on carrier detection, which allows multiple nodes to share the channel by avoiding simultaneous transmissions, therefore avoiding collisions among transmitted packets [88]. When a node wants to transmit a data packet, it first listens to the channel. If the channel is sensed as idle during a fixed time interval, the node transmits, otherwise it waits for a backoff time before attempting a new transmission.

5.2.2.3 PHY Layer Adaptation

U-Wear defines a set of cross-layer functionalities that enable real-time reconfiguration of PHY layer parameters from upper layers of the protocol stack, e.g., data link or network layer. By leveraging the flexibility of the software-defined architecture, upper layer protocols can reconfigure on-the-fly PHY layer parameters such as modulation, signal bandwidth and FEC coding rate, among others. Reconfiguration functionalities allow to develop reactive or proactive control algorithms to adapt the underlying communication link to the channel variations or to upper layer protocol requirements [89, 82].

5.2.3 Network Layer

5.2.3.1 IPv4 and IPv6 Support

U-Wear provides interoperability with the Internet by defining an adaptation layer that integrates IPv4 and IPv6 protocol support [90]. The adaptation layer consists of a set of functionalities that interface the traditional IP network layer with the U-Wear data link layer, by offering IP header compression and IP packet fragmentation functions optimized for ultrasonic wearable networks with long propagation delays discussed in Section 5.1 that potentially prevent accurate timing of network protocols. For example, by leveraging cross-layer header information, the long IPv4 and IPv6 headers can be shortened to reduce network delay and energy consumption when exchanging small information packets.

5.2.3.2 Content-centric Networking

U-wear offers content-centric networking (CCN) functionalities that make the network content directly addressable and routable. Each sensor data or actuation command, i.e., each content object, is labeled with a name, and can be accessed through this name. Nodes can request content objects by broadcasting a request message. When a match is found, i.e., the content is found on a network node, a response message containing the requested content is sent back.
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

5.2.4 Application Layer

5.2.4.1 Reconfigurable and Modular Data Processing

U-Wear adopts the idea of decomposing the data processing applications running in the sensor nodes into primitive blocks, and offering real-time reconfigurability at the application layer. The sensing application consists of a sequence of basic operations that are executed on the sensed data to extract desired medical parameters. Real-time modular reconfiguration offers three main advantages. First, the network coordinator can wirelessly transmit and install new applications on sensor nodes at runtime, as needed. Based on this, resources are allocated only when the application is requested, thus reducing the processing and memory overhead due to static applications continuously running in background. Second, modular reconfiguration enables programmers to easily create new applications by arranging the primitive building blocks in the desired execution sequence. As a consequence, new medical parameters can be extracted from the raw data coming from a sensor, while maximizing code reusability. Finally, in case of template matching applications, e.g., ECG anomaly detection by matching traces with known templates [91], adding or updating templates becomes very easy with a reconfigurable application layer.

Defining new applications consists of specifying inputs, a chain of primitive blocks, and outputs. An input is the physical sensor that generates the data, e.g., accelerometer or electrocardiogram (ECG). An output can be either the local memory for storing a measured parameter, or a transmission for sending a measured parameter to another node. We divide the set of primitive blocks into three main classes, filters, data operations, and detectors. Filters enable filtering the raw data to remove offsets, drift of the sensors and any other noise components coming from external sources. Data operations include common signal processing operations performed on sensor data, e.g., correlation with templates, and FFT, among others. Finally, detectors allow measuring the desired parameters by detecting specific elements in the processed signal, e.g., peaks, patterns and time distances, among others.

5.2.4.2 Data Collection

The application layer can operate in two different modalities to exchange and collect data, fetch and push mode. Fetch mode is used when the application layer requires content from the network. Push mode is used when sensed data needs to be pushed to another node, e.g., high glucose level in the blood, or when a node requires another node to accomplish some actuating operation, e.g., inject insulin or trigger a neurostimulation. In case of actuating commands, the push packet may contain further information about the required action, e.g., the quantity of insulin to inject or the pattern of the neurostimulation.

5.3 U-Wear Prototypes

We now present the design of two prototypes that implement the U-Wear framework discussed in Section 5.2. The first U-Wear prototype is a wearable ultrasonic sensor node based on a custom hardware
platform, which we refer to as *wuMote*. The second prototype is a wearable ultrasonic coordinator based on an iOS commercial smartphone device, which we refer to as *wuMaster*.

### 5.3.1 wuMote Prototype

#### 5.3.1.1 Hardware Design

In Fig. 5.3, we show the hardware architecture of the wuMote. The *core unit* includes a processing unit, e.g., microprocessor or microcontroller, a memory unit, e.g., RAM or flash memory, and digital-to-analog and analog-to-digital converters. The *processing unit* executes the U-Wear functionalities discussed in Section 5.2. The *communication unit* enables ultrasonic wireless connectivity by embedding power and low noise amplifiers, and air-coupled ultrasonic transducers. The *power unit* mainly consists of a battery to power the wuMote. An optional *wireless energy transfer unit* can be installed to leverage ultrasonic power transmission to wirelessly charge the node battery. Finally, the *sensing and actuating unit* can incorporate several sensors and actuators according to the specific application design.

We implemented a prototype of the architecture in Fig. 5.3 based on Teensy 3.1 [92]. The wuMote implementation offers near-ultrasonic capability, by using COTS audio speakers and microphones as air-coupled ultrasonic transducers. Figure 5.4 shows the basic circuit design of the wuMote prototype on a
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

The prototype includes a Teensy 3.1, i.e., the core unit, a power amplifier, a microphone with low noise amplifier, and a small audio speaker, i.e., the communication unit. A lithium ion polymer battery, not shown in the figure, is connected to the bus strip of the breadboard to power the electronic components. A smaller, more compact, and more visually appealing prototype board can be designed by developing a customized PCB board that embeds the wuMote hardware components.

**Teensy 3.1.** Teensy 3.1 is a small-footprint, i.e., about 3.5 × 1.8 cm, inexpensive development board, based on a 32 bit ARM Cortex-M4. It comes with 64K of RAM, 256K of Flash, 12 bit DAC, dual ADC, and USB connectivity. Teensy 3.1 can be programmed in C and C++ using Teensyduino, a customized version of the Arduino integrated development environment (IDE), and supports many of the code libraries designed for Arduino and others specifically designed for Teensy, e.g., the audio library [93], among others. Teensy 3.1 can be powered via USB, or through external coin batteries connected to the \( V_{in} \) and \( GND \) pins.

We selected Teensy 3.1 over other available COTS platforms such as USRP N210, Raspberry Pi, or Arduino Uno, for the following reasons. Compared to USRP N210 and Raspberry Pi, where software operations are executed on top of an operating system running on external or internal microprocessors, Teensy 3.1 and Arduino Uno are designed around a low-power microcontroller that provides low-level control of the hardware peripherals. Microcontroller-based platforms offer higher hardware flexibility and computational efficiency that suit the design requirements of wireless wearable devices. Finally, we selected Teensy 3.1 over Arduino Uno because of the more powerful microcontroller and larger available memory that can support high audio sampling rates compatible with the near-ultrasonic communication range, e.g., 44.1 kHz for acoustic frequencies up to 22 kHz. Teensy 3.1 still supports the Arduino libraries that can significantly ease the prototyping process of the wuMote. Ideally, lower-power and smaller-packaged solutions would be desirable for a more stable product. In a final product, the Cortex M4 currently used in our prototype would be likely replaced by a Cortex M0 microcontroller such as the mm-size low-power Freescale KL03 microcontroller [94]. Even though development boards for Cortex M0 microcontrollers exist, none of these are comparable to the Teensy platform in terms of hardware and software capabilities, and they do not offer Arduino library support. Thus, at this stage we select the less energy efficient hardware platform Teensy 3.1 in exchange for shorter prototyping time.

**Power Amplifier.** The wuMote includes a small and efficient class D audio amplifier [95] able to deliver a maximum of 1 W into 4 ohm impedance speakers, with a voltage supply of 3.3 V DC, and efficiency up to 80%. The amplifier consumes less than 2 mA of current when quiescent and less than 2 \( \mu \)A in standby mode. In Fig. 5.4, the right channel of the power amplifier is connected to Teensy via the DAC pin, and to the speakers via the 3.5 mm screw-terminal blocks. The \( V_{cc} \) and \( GND \) pins are connected to the bus strip to power the device.

**Microphone.** The wuMote includes a tiny breakout board that embeds an ADMP401 MEMS microphone and a low-noise amplifier (LNA) [96]. The ADMP401 offers a mostly flat bandwidth, i.e., −3 dB roll off, between 100 Hz and 15 kHz, omnidirectional sensitivity pattern, and requires a supply voltage between 1.5 V and 3.3 V DC. Although a microphone with larger bandwidth would perform better [97], we selected ADMP401 because of the COTS breakout board package that eases prototyping. Moreover, even though with
lower sensitivity, the ADMP401 can still detect higher frequency acoustic waves up to 22kHz. The microphone is connected to one of the analog pins (ADC) available in Teensy 3.1, and is powered by connecting the \( V_{cc} \) and \( GND \) pins to the bus strip.

**Audio Speaker.** The output of the wuMote is a small and compact COTS speaker, Dayton Audio CE28A-4R [98], with 4 ohm impedance, 4 W maximum output power supported, and flat frequency response between 100 Hz and 15 kHz. The speaker is connected to the power amplifier using 3.5 mm screw-terminal blocks.

5.3.1.2 Software Architecture

We implemented the U-Wear framework in Teensy 3.1 to enable ultrasonic wireless connectivity and networking on the wuMote hardware prototype. In Fig. 5.5, we show the block diagram of the wuMote software architecture that includes (i) narrowband GMSK transceiver with synchronization, channel estimation, equalization, and FEC functionalities at the PHY layer, (ii) polling and ALOHA multiple access protocol with FEC coding rate reactive adaptation at the data link layer, (iii) content-centric addressing at the network layer, and (iv) data processing reconfiguration with fetch and push support at the application layer.

We implemented the U-Wear functionalities using Teensyduino, an add-on for the Arduino IDE, leveraging many of the code libraries available for the Arduino platform. Since ultrasonic waves are nothing but sound at higher frequencies, we based the PHY layer signal processing on the audio library specifically designed for Teensy 3.1 [93]. The Teensy audio library consists of a set of objects that enable recording, processing, and playback of audio sampled at 44.1 kHz. Objects instantiate specific audio functionalities, e.g., a waveform synthesizer and finite-impulse-response (FIR) filters, while new functionalities can be enabled by creating new objects. A cascade of objects forms a processing chain that performs a set of operations on inputs to produce a desired output. Each object in the chain operates in pipeline on chunks of 128 audio samples, which correspond to 2.9 ms of audio. To guarantee audio continuity, each block must execute its processing operation within 2.9 ms.

In the wuMote implementation we built custom-made objects that implement specific signal processing operations. Finally, since some computationally expensive operations exceed the audio library time constraints of 2.9 ms, we implemented these outside the audio library. We refer to these as **off-the-chain** objects.
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

Physical Tx. The first object of the PHY layer Tx chain is the FEC Encoder. Here, each data packet coming from the data link layer is coded, as discussed in Section 5.2.1.4, and overhead symbols are appended to the original packet. We select \( n = 255 \) symbols and parity symbols \( t \) to achieve different coding rates. Because of the computation complexity of RS coding, the FEC Encoder is implemented as an off-the-chain object. The coded packet is then passed to the Symbol Mapping object that inputs the audio stream in the processing chain. Here, the coded packet is serialized, i.e., converted into a stream of bits, differentially encoded, and transformed into a non-return-to-zero (NRZ) signal. The NRZ signal is then GMSK modulated by the GMSK Modulator object and up-converted to the carrier frequency by the Up-Mixer object. The modulated and up-converted waveforms are passed to the Audio Output object, i.e., a system API that interfaces U-Wear with the DAC, digital-to-analog converted and transmitted through the audio speaker.

Physical Rx. The received acoustic signal is converted into an electrical signal by the MEMS microphone. The signal is amplified by the LNA, and analog-to-digital converted by the Teensy 3.1 ADC at 44.1 kHz. The Audio Input object is a system API that interfaces U-Wear with the embedded Teensy 3.1 ADC, and inputs the audio stream into the PHY layer Rx chain. The received digital signal is first high-pass filtered by the High-pass Filter object to eliminate low-frequency noise and interference, i.e., external human voice and ambient noise. The Packet Detector processes the input signal to detect incoming packets using an energy-based approach to check whether there is energy at the expected carrier frequency. The incoming packet is then down-converted by the Down-Mixer, i.e., converted into a complex in-phase/quadrature baseband signal, and low-pass filtered to eliminate undesired higher-frequency harmonics introduced by the nonlinearity of the down-conversion operation. Channel estimation, synchronization and equalization operations normally follow down-conversion and are applied to the complex baseband signal. However, these operations are computationally expensive, and their execution exceeds the audio library time constraints of 2.9 ms. To overcome this limitation, we first demodulate the complex baseband signal in the GMSK Demodulator object to extract the phase variation that carries the coded information bits. Then, we execute off-the-chain the computationally expensive operations. The Channel Estimator object estimates the CIR using the packet preamble as training sequence, as discussed in Section 5.2.1.3, while the Synchronizer object attempts to achieve fine synchronization through the PN-based mode discussed in Section 5.2.1.2. The ZF Equalizer object filters the input for ISI recovery, as discussed in Section 5.2.1.3. The equalized symbols are demapped into a bitstream, collected into a packet structure, and passed to the FEC Decoder object. Here, FEC decoding operations attempt to correct potential bit errors, as discussed in Section 5.2.1.4. Finally, the error-corrected packet is passed to the data link layer.

Data Link Layer. The wuMote data link layer is implemented in a Finite State Machine (FSM) block that currently includes two of the MAC protocols discussed in Section 5.2.2, i.e., polling and ALOHA. The wuMote data link layer also implements a PHY layer adaptation to optimally select the FEC coding rate that minimizes the number of retransmissions. During packet transmission, the MAC FSM collects data from upper layer protocols and creates the data-link-layer packet. The packet is then forwarded to the PHY layer Tx chain, where it is encoded in a digital waveform before being transmitted. At the receiver side, the MAC FSM
detects the received packet based on information coming from the Packet Detector block and triggers the PHY layer to start processing the received waveform.

Network Layer. The wuMote network layer implements the content-centric addressing scheme discussed in Section 5.2.3. Although IPv4 and IPv6 support is an important aspect of the U-Wear framework, at this prototyping stage we left most IP-related aspects out of the scope of this work. Each content object is mapped into a binary mask, where it is represented by the bit position in the mask. In an M/S configuration, a node joining the network is first paired with the master. The master maps the content available in the new node into the binary mask, and broadcasts the updated mapping to all the nodes in the network. Each node builds a local mask based on the entities that it possesses. To request an entity, the master broadcasts a request message containing a request mask with ‘1’ set in the position mapped to the desired content.

Application Layer. The wuMote application layer implements the real-time modular reconfiguration functionalities discussed in Section 5.2.4.

Based on this modular approach, applications can be represented by chains of binary sequences, i.e., keys. Each primitive function is mapped to a binary key. A concatenation of keys represents a concatenation of operations, and therefore represents an application. The application is encapsulated into reconfiguration packets and transmitted over-the-air. The receiving node extracts the binary keys, and feeds these into an FSM where each state represents a primitive block function. By parsing the consecutive function keys, the FSM transitions from state to state, processing inputs and producing outputs. Inputs and outputs of each function are mapped to binary keys as well, and are implemented in a C-struct output_struct that contains a pointer to an array, and a binary key variable. The input and output keys allow to parametrically pass data between functions.

As a proof of concept, we developed some applications based on the primitives discussed above.

Electrocardiogram Processing. We consider a single-electrode ECG signal. Fig. 5.6 shows a template signal with five labelled characteristic waveforms, P, Q, R, S and T, that correspond to three electrical events during one heart beat, i.e., atrial contraction (P), ventricular contraction (QRS) and ventricular recovery (T). The first application measures the heart rate in beat-per-minute [bpm]. This is done following two
different approaches. The first approach, here \textit{R method}, counts the number of peaks, \textit{R waves}, in a 6-second trace and multiplies the result by 10. The second approach, here \textit{RR method}, finds the number of heartbeats per second by inverting the average of the \textit{RR interval} duration, i.e., distance between two consecutive \textit{R} waveforms in the trace, and multiplies this by 60. This approach has higher complexity with respect to the \textit{R}-method, but results in higher resolution, i.e., 1 bpm against 10 bpm of the \textit{R} method. The second application measures average and standard deviations of temporal separation between electrical events in the ECG. For example, the distance between peaks, i.e., \textit{R} waves, gives the \textit{RR interval} duration. The mean and standard deviation of the \textit{RR interval} provide information about potential heart arrhythmias \cite{99}. Figure 5.6 (top) shows the simplified primitive block sequences of the \textit{R} method heart rate detector and the \textit{RR interval} monitor applications.

\textbf{Accelerometer Processing.} The accelerometer trace in Fig. 5.6 shows the frontal acceleration on the x-axis, the vertical acceleration on the y-axis, and the lateral acceleration on the z-axis. We label in the y-axis two main events that occur during a single step, i.e., \textit{heel strikes} (HSs) and \textit{toe-off} (TO), that correspond to the instants at which the foot touches the ground, and the instants at which the foot leaves the ground, respectively. Based on this, the first application calculates the magnitude of the acceleration from the three-axis components, low-pass filters it to remove high frequency noise, and counts the number of peaks in the resulting signal, i.e., the number of HSs. The peaks within a time interval represent the number of footsteps performed by the patient. The second application measures average distances between events in the accelerometer trace. For example, the distance between non-consecutive peaks in the acceleration magnitude gives the \textit{gait cycle time} (GCT), i.e., time between consecutive HSs on the same foot. GCT offers a measure of the motor degradation of patients affected by \textit{Parkinson} disease \cite{65}. Figure 5.6 (bottom) shows the simplified primitive block sequences of the footstep counter and the GCT monitor applications.

\textbf{5.3.2 wuMaster Prototype}

We implemented the wuMaster prototype on the iOS 8 platform for Apple iPhone smartphones. The prototype consists of an app running on an iPhone that implements the U-Wear multi-layer functionalities. In Fig. 5.7, we show the software architecture of the wuMaster prototype that includes (i) narrowband GMSK transceiver with synchronization, channel estimation, equalization, and FEC functionalities at the PHY layer, (ii) polling and ALOHA multiple access protocol with FEC coding rate reactive adaptation at the data link layer, (iii) content-centric networking at the network layer, and (iv) a graphic user interface (GUI) and speech recognition functionalities at the application layer that allow users to interact with the wearable network.

The iOS prototype wirelessly communicates with the wuMotes through an ultrasonic narrowband GMSK link, using the phone’s embedded microphone and speaker. We developed the software prototype in Objective-C programming language using Xcode 6 integrated development environment (IDE) \cite{100}. We use (i) the \textit{vDSP library} \cite{101} of the iOS Accelerate framework that implements digital signal processing operations (DSP), (ii) \textit{Novocaine} \cite{102}, a high performance audio library that enables record/play audio functionalities, and (iii) \textit{wit.ai} framework \cite{103} that offers speech recognition services.
Figure 5.7: Software architecture of the wuMaster prototype.

**vDSP Library.** The vDSP library is part of the Accelerate framework available in iOS, and provides several DSP functionalities including vector and matrix arithmetic, Fourier transforms, convolution and correlation operations between real or complex data types. We leverage the vDSP functionalities to perform arithmetic operations and correlations on real and complex vectors in the PHY layer.

**Novocaine.** Novocaine is a high performance audio processing library for iOS. Novocaine hides all the low-level audio implementation details, giving simple block-based callbacks that are called when audio comes in, and when audio needs to go out. Specifically, a Novocaine object, i.e., the Audio Manager, offers **InputBlock** and **OutputBlock** callbacks, inside which we can simply place the DSP code for processing input and output data.

**Wit.ai Framework.** Wit.ai provides natural language processing in the form of multi-platform APIs, which we use to integrate voice command in U-Wear. Wit.ai allows developers to create commands, and to match these commands with intents. A command is what the user would say to trigger an operation while the intent represents the operation itself. Voice commands are sent to the wit.ai server, and the intent with maximum confidence is returned as a response.

**PHY Layer Tx/Rx.** The PHY layer Tx and Rx are implemented in two classes named **PHYLayerTx** and **PHYLayerRx**, respectively. Here, the Novocaine Audio Manager triggers the **InputBlock** and **OutputBlock** callbacks to record and play audio, and the vDSP functions process the input and output data. At the transmitter, the **PHYLayerTx** class gets the data from the data link layer, generates the GMSK waveform, and then passes it to the Audio Manager. The latter transmits the GMSK waveform through the speaker. At the receiver, the operations in **PHYLayerRx** match those implemented in the wuMote PHY layer, discussed in Section 5.3.1.2. Because of the less stringent memory and processing constraints of the iOS platform, here channel estimation, synchronization and equalization follow the down-conversion, and are applied to the complex baseband signal.

**Data Link and Network Layer.** The wuMaster data link layer implements polling and ALOHA MAC protocols, as well as FEC coding rate adaptation. The MAC functionalities are implemented in a class named **MACLayer**, where a FSM implements the MAC protocol operations. The wuMaster network layer implements the same content-centric addressing scheme seen in the wuMote prototype, with the exception that here the centralized mapping functionalities are also implemented.

**Application Layer.** The wuMaster application layer consists of a GUI and a set of wit.ai commands...
that allow users to interact with the U-Wear multi-layer functionalities. The GUI’s principal element is a three-tab `TabViewController` class. The first tab, shown in Fig. 5.8 (left), contains a `PHYViewController` object. It inherits from `UIViewController`, i.e., a basic view controller in iOS, and is implemented to test the PHY layer performance. The second tab contains a `MACViewController` object that inherits from `UIViewController` and allows the user to test the MAC Layer functionalities by requesting, receiving and visualizing sensed data coming from the deployed wuMotes. The `MACViewController` embeds a `UICollectionView`, a collection of objects that represent the sensed data. In Fig. 5.8 (center) we show six objects in the collection, which are associated with the number of footsteps, sleep hours, the heart rate, the breathing rate and the respiratory minute volume, the glucose level in the blood and the diastolic/systolic blood pressure. Finally, a `WITMicButton`, defined in the wit.ai framework, enables voice command processing. The third tab contains an `APPViewController` object that gives access to application layer reconfiguration functionalities discussed in Section 5.2.4. In the `APPViewController` we group the applications based on the sensing unit that provides the required input data, e.g., accelerometer and ECG. Each `UIButton` represents a group of applications, and it gives access to a `PopViewController` object that shows the available applications in that group. Users can select which application to run on the wuMote. For example, in Fig. 5.8 (right) we show how users can select to install heart rate or RR interval monitor on wuMotes equipped with ECG.

5.4 Performance Evaluation

In this Section, we demonstrate the feasibility of ultrasonic communications for wearable devices through testbed experiments, and we evaluate the performance of the U-Wear prototypes discussed in Section 5.3.

5.4.1 PHY Layer Performance

**Experiment Setup.** The experiment setup consists of a wuMote communicating bidirectionally with a wuMaster in two different scenarios, line-of-sight (LOS) and near-line-of-sight (nLOS). In the LOS
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

Figure 5.9: Near-line-of-sight (nLOS) experimental setup.

Figure 5.10: Ultrasonic in-air CIR for LOS (top), chest-hand nLOS (center) and leg-hand nLOS (bottom).

scenario the two devices are aligned, 50 cm apart, without obstacles in between, so as to minimize reflections and scattering. In the nLOS scenario, we locate the wuMotes along the body of a user, on the chest and on the right leg, as shown in Fig. 5.9. The wuMaster, i.e., the smartphone, is held in the user’s right hand. Under this setup, objects within the propagation area cause reflections and scattering that introduce ISI and degrade the communication performance. In Fig. 5.10, we show the uplink CIRs of the three scenarios discussed above. We observe that, in the LOS scenario, the CIR contains a single dominant component. In the nLOS scenario, because of the multipath there are multiple components that contribute to ISI distortion at the receiver. In particular, in the chest-hand setup, the CIR clearly presents a second path, most likely because of a reflection from the user’s hand, while in the leg-hand setup we can count up to 6 paths, most likely caused by multiple reflections from the user’s trunk and hand. The coherence bandwidth in these three scenarios is approximately 21 kHz, 14 kHz and 6 kHz, respectively.

For each BER measurement we transmit up to 600 packets of 32 bytes, i.e., approximately 256 kilobits, containing pseudorandom-generated raw data. The experiment was performed indoor with a temperature of about 21°C and relative humidity around 30%. We configure the physical layer such that each GMSK symbol is represented by 16 samples. The sampling rate is set to 44.1 kHz as required by the audio
Figure 5.11: BER of the downlink (top) and uplink (bottom) in LOS as a function of the SNR for different coding rates.

Hardware in use. Based on this, the raw physical layer data rate, obtained as the ratio between sample rate and sample per symbol, is approximately 2.76 kbit/s. We fix the GMSK BT product to 0.7, which represents a good tradeoff between ISI distortion and spectral efficiency. The resulting signal bandwidth is about 2 kHz, which is lower than the coherence bandwidth of the three experimental setups, thus complying with the definition of narrowband transmission scheme. The central frequency is set to 18 kHz, which, while still in the audible frequency range, represents a good tradeoff between low audibility, fair propagation efficiency, and fair acoustic generation and detection with the COTS microphones and speakers in use. Specifically, we found that 18 kHz is the highest frequency, given the spectral response of microphones and speakers in use, for which we could obtain highly reliable communications, i.e., relatively low BER, in the range of distances of interest, i.e., up to 1 m. At the same time, the signal transmission is almost inaudible by the user wearing the device. Finally, a 64-bit PN-sequence is used as preamble for synchronization and channel estimation.

**BER Performance in LOS.** In Fig. 5.11 (top), we show BER results for the downlink, i.e., from the wuMaster to the wuMote, and we compare the performance of an uncoded transmission scheme to four coded transmission schemes with coding rates in \{8/9, 4/5, 2/3, 1/2\}. The information rate for the five transmission schemes ranges from 2.76 kbit/s for the uncoded transmissions to 2.45 kbit/s for coding rate 8/9, 2.20 kbit/s for coding rate 4/5, 1.84 kbit/s for coding rate 2/3, and 1.38 kbit/s for coding rate 1/2. Figure 5.11 (bottom) shows the same comparison for the uplink, i.e., from the wuMote to the wuMaster. The SNR is calculated at the receiver as the ratio between the received average signal power and the average noise power measured after amplification and high-pass filtering. We vary the measured SNR by reducing the signal power driving the transmitter speaker. In the downlink, we do so by reducing the volume of the smartphone speaker, while in the uplink, we reduce the signal full-scale at the input of the amplifier. The maximum power is selected such that the transmitted sound results inaudible to people in proximity of the transmitter.

From Fig. 5.11, we observe that the BER is a decreasing function of the SNR, and that the FEC scheme mitigates the channel distortion by recovering part of the channel errors. At 5 dB SNR the BER is too
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

Figure 5.12: BER of the chest-hand (top) and of the leg-hand uplinks as a function of the SNR for different coding rates.

high for the FEC to have an impact on the communication performance. Over 5 dB SNR, higher coding rate transmissions have clearly better mitigation performances, thus lower BER.

By measuring the power at the output of the wuMote amplifier, we see how our prototypes achieve 2.76 kbit/s on an uncoded uplink transmission, with a $10^{-5}$ BER, using a transmission power of 20 mW, i.e., 13 dB SNR at the receiver. We can lower the transmission power by compensating with lower FEC coding rate, thus reducing the information rate. For example, in the current implementation, for a transmission power of 10 mW, i.e., 7 dB SNR, our prototypes achieve 1.38 kbit/s with a $10^{-5}$ BER using a coding rate of 1/2. Finally, by proposing for the first time a GMSK scheme for the near-ultrasonic frequencies our prototypes achieve relatively high data rates when compared to previously proposed near-ultrasonic systems; more importantly, it ensures virtually inaudible click-free transmission because of the GMSK phase-continuity as discussed in Section 5.2.1.1.

BER Performance in nLOS. Figure 5.11 shows the BER performance of uplink transmissions in nLOS scenario chest-hand setup (top) and leg-hand setup (bottom). We observe that, while the curves follow the same pattern as in the LOS scenario, the corresponding BER levels are higher because of the worse channel conditions. The BER in the chest-hand scenario is slightly higher than the LOS one, i.e., about 1 dB more of SNR is required for the same BER. Differently, in the leg-hand scenario we need an increase of 4 dB SNR to achieve the same BER performance of the LOS scenario. In the chest-hand uplink, our prototypes achieve 2.76 kbit/s with a $10^{-5}$ BER using a transmission power of 45 mW, i.e., about 13 dB SNR at the receiver, while the same BER is obtained with 45 mW transmission power, i.e., approximately 9 dB SNR at the receiver, halving the data rate through FEC coding. In the leg-hand uplink, we obtain $10^{-5}$ BER with a transmission power of 250 mW, i.e., about 18 dB SNR at the receiver, for uncoded transmission at 2.76 kbit/s and, and 130 mW of transmission power, i.e., approximately 12 dB SNR at the receiver, for coded transmission at 1.78 kbit/s.

These results show how multipath effect and higher attenuation caused by the user’s clothing require
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

![Graph showing polling data delivery delay as a function of number of nodes for different level of SNR and coding rates (top), and as a function of the SNR for non-adaptive and adaptive scenarios.]

Figure 5.13: Polling data delivery delay as a function of number of nodes for different level of SNR and coding rates (top), and as a function of the SNR for non-adaptive and adaptive scenarios.

higher power transmission as compared to the LOS scenario. Even though ultrasonic signals are further attenuated by solid materials, they can still be used to communicate over short distances through clothing. In general, the transmission power can be reduced by using speakers and microphones with wider flat bandwidth or custom-made optimized ultrasonic transducers. In fact, a significant portion of the transmission power is lost during the electro-acoustic conversion in the COTS speaker and microphone in use, which are not designed to operate efficiently at near-ultrasonic frequencies.

5.4.2 MAC Layer Performance

In this Section, we evaluate the performance of the MAC layer protocols implemented on the prototypes, i.e., polling and ALOHA, in terms of data delivery delay as a function of the number of nodes in the network.

**Experiment Setup.** We set up a M/S configuration where devices lay in nLOS on a 2-dimensional surface, and each wuMote is positioned 40 cm apart from the wuMaster. The experiment consists of collecting data at the wuMaster from up to four wuMotes using polling or ALOHA MAC protocols. We consider six different parameters than can be fetched, and we distribute these among four wuMotes.

**Adaptive Polling.** Using the polling protocol, the wuMaster fetches data from one node a time. The wuMotes are addressed through physical addresses, e.g., node ID. The PHY layer adaptation allows to reactively adapt the FEC coding rate based on the packet drop rate experienced at the wuMaster to minimize the number of consecutive retransmissions. Specifically, every time the wuMaster retransmits a fetching packet, a lower coding rate is used from the set {8/9, 4/5, 2/3, 1/2}. We fix the maximum number of retransmissions for each fetch command to four. We evaluate the protocol in terms of data delivery delay, which we define as the time between the instant when the first fetching packet is transmitted by the wuMaster and the instant when the last data packet is correctly received at the wuMaster. In Fig. 5.13 (top), we show the polling data...
delivery delay as a function of the number of nodes in the network for two levels of SNR measured at the wuMaster, i.e., 10 dB and 14 dB, and two coding rates, i.e., 8/9 and 4/5. As expected, since each node in average is granted the same time to transmit, we observe that the delivery delay increases linearly with the number of nodes in the network. Moreover, since retransmissions are only caused by the channel conditions, i.e., there are no collision among different users, the delivery delay decreases by increasing the SNR or the coding rate. Figure 5.13 (bottom) shows the delivery delay as a function of the SNR for two fixed coding rates, i.e., 8/9 and 4/5, and for the adaptive scenario. We observe that at lower SNR, a coding rate of 8/9 gives delivery delays higher than a coding rate of 4/5 because of the frequent retransmissions due to higher BER at the PHY layer. On the contrary, at higher SNR a coding rate of 4/5 introduces more overhead than needed, giving higher delivery delays than coding rate 8/9. As expected, the adaptive scenario results in delivery delays in between the two fixed coding rate scenarios.

**ALOHA.** With ALOHA, we use the content-centric addressing scheme discussed in Section 5.3.1.2. Hence, the wuMaster broadcasts a request message to fetch data from multiple wuMotes. The wuMotes transmit the requested data, if available, by accessing the channel randomly. Finally, we select the backoff time between transmissions from 0 to a maximum backoff $B_{\text{max}}$, and we vary it during our experiments, while fixing the SNR to 14 dB and FEC coding rate to 8/9. Figure 5.14 shows the data delivery delay as a function of the number of nodes in the network for two different values of $B_{\text{max}}$, i.e., 1.5 s and 3 s. We compare the results with the data delivery delay experienced by the polling protocol for 14 dB SNR and 8/9 coding rate. When the number of nodes in the network is lower than three, we observe that $B_{\text{max}} = 1.5$ s gives lower delay than $B_{\text{max}} = 3$ s. Here, a higher $B_{\text{max}}$ increases the probability of selecting a higher backoff time, leading to channel underutilization. On the other hand, for number of nodes higher and equal to three, $B_{\text{max}} = 1.5$ s gives high probability of collisions, thus higher delay due to retransmissions.

### 5.4.3 Data Processing Performance

To test the effectiveness of the application layer reconfiguration functionalities, we evaluate the data processing accuracy in terms of displacement between the obtained outputs, i.e., what the application reads on a given sensor trace, and the expected ones, i.e., what the application should read on that given sensor trace. We consider three applications, two running on ECG-equipped sensor nodes, i.e., heart rate monitor, ECG RR interval monitor, and one running on accelerometer-equipped sensor nodes, i.e., footstep counter.
ECG Processing. To show a fair comparison, we feed the application-layer processing with reference raw sensor data that are externally recorded and then loaded in the wuMote memory. For the ECG-based applications, we use traces from the MIT-BIH Normal Sinus Rhythm Database [104], which collects ECG recording from patients that were found to have no significant arrhythmias. The MIT-BIH database also specifies heart rate and RR interval of the traces. The traces are sampled at 128 Hz. We extract, and load to the wuMote, 10 one-minute long traces from the MIT-BIH recording. In Table 5.1, we show the heart rate estimation of the wuMote using the \textit{R method}, second column, and \textit{RR method}, third column, discussed in Section 5.3.1.2, and we compare these with the heart rate reference provided by the MIT-BIH database, fourth column. The first column shows the database trace ID. We observe that both \textit{R method} and \textit{RR method} give a good estimate of the reference heart rate, offering an average accuracy of 96.1\% and 98.7\%, respectively.

In Table 5.2, we show the \textit{RR interval} mean $\mu$ and standard deviation $\sigma$ estimated by the wuMote, second and fourth columns, and we compare these with the \textit{RR interval} reference statistics provided by the MIT-BIH database, third and fifth columns. We observe that the wuMote accurately estimates the \textit{RR interval} mean, i.e., around 99.6\% of accuracy. For the standard deviation $\sigma$ we obtain lower accuracy, i.e., 83.6\%, for two reasons, (i) the relatively low sampling rate gives a sensibility of 8 ms, which can affect the measurement of small quantities such as the standard deviation, (ii) failures in the peak finding algorithm also affect the measurement. Higher sampling rate and outlier detection techniques could be used to further enhance the standard deviation measurement.

<table>
<thead>
<tr>
<th>Trace</th>
<th>6s R [bpm]</th>
<th>6s RR [bpm]</th>
<th>Ref HR [bpm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>16265</td>
<td>100</td>
<td>97</td>
<td>96</td>
</tr>
<tr>
<td>16272</td>
<td>60</td>
<td>62</td>
<td>62</td>
</tr>
<tr>
<td>16273</td>
<td>100</td>
<td>97</td>
<td>95</td>
</tr>
<tr>
<td>16420</td>
<td>90</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td>16483</td>
<td>100</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>16539</td>
<td>80</td>
<td>80</td>
<td>79</td>
</tr>
<tr>
<td>16773</td>
<td>70</td>
<td>74</td>
<td>75</td>
</tr>
<tr>
<td>16786</td>
<td>70</td>
<td>72</td>
<td>71</td>
</tr>
<tr>
<td>16795</td>
<td>70</td>
<td>67</td>
<td>65</td>
</tr>
<tr>
<td>17052</td>
<td>70</td>
<td>68</td>
<td>69</td>
</tr>
</tbody>
</table>

Accelerometer Processing. We record ten 3-dimensional accelerometer traces with a sample rate of 60 Hz using Sensor Log [105], an iOS app that allows to read sensor data from the device, and export them in character-separated values (CSV) format. Sensor Log also provides information about the number of footsteps counted by the iOS device. We use this as a reference to evaluate the accuracy of the footstep
### CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

Table 5.2: Results for RR interval mean $\mu$ and std. deviation $\sigma$.

<table>
<thead>
<tr>
<th>Trace</th>
<th>$\mu$ [s]</th>
<th>$\sigma$ [s]</th>
<th>Ref. $\mu$ [s]</th>
<th>Ref. $\sigma$ [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>16265</td>
<td>0.62</td>
<td>0.62</td>
<td>0.016</td>
<td>0.019</td>
</tr>
<tr>
<td>16272</td>
<td>0.96</td>
<td>0.96</td>
<td>0.109</td>
<td>0.115</td>
</tr>
<tr>
<td>16273</td>
<td>0.64</td>
<td>0.64</td>
<td>0.023</td>
<td>0.049</td>
</tr>
<tr>
<td>16420</td>
<td>0.63</td>
<td>0.63</td>
<td>0.015</td>
<td>0.018</td>
</tr>
<tr>
<td>16483</td>
<td>0.62</td>
<td>0.62</td>
<td>0.015</td>
<td>0.012</td>
</tr>
<tr>
<td>16539</td>
<td>0.74</td>
<td>0.75</td>
<td>0.062</td>
<td>0.054</td>
</tr>
<tr>
<td>16773</td>
<td>0.79</td>
<td>0.79</td>
<td>0.056</td>
<td>0.058</td>
</tr>
<tr>
<td>16786</td>
<td>0.85</td>
<td>0.84</td>
<td>0.046</td>
<td>0.036</td>
</tr>
<tr>
<td>16795</td>
<td>0.91</td>
<td>0.92</td>
<td>0.070</td>
<td>0.069</td>
</tr>
<tr>
<td>17052</td>
<td>0.85</td>
<td>0.85</td>
<td>0.047</td>
<td>0.047</td>
</tr>
</tbody>
</table>

counter application in the wuMote. In Table 5.3, we show the footstep count estimated by the wuMote, second column, and we compare this with the footstep estimate of the iOS device, third column, and real footstep number counted by the user while performing the steps, fourth column. The first column shows the trace name, where we list 3 walking traces, 3 running traces and 3 stair climbing traces, i.e., downward, upwards and down/upwards. We observe that, in average, the wuMote counts footsteps with the same accuracy of the iOS device, i.e., approximately 94% with respect to the number of steps counted by the user.

Table 5.3: Evaluation results for footstep counter.

<table>
<thead>
<tr>
<th>Trace</th>
<th>wuMote</th>
<th>iOS</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>walk_0</td>
<td>44</td>
<td>49</td>
<td>46</td>
</tr>
<tr>
<td>walk_1</td>
<td>39</td>
<td>39</td>
<td>40</td>
</tr>
<tr>
<td>walk_2</td>
<td>48</td>
<td>48</td>
<td>50</td>
</tr>
<tr>
<td>run_0</td>
<td>32</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>run_1</td>
<td>37</td>
<td>42</td>
<td>40</td>
</tr>
<tr>
<td>run_2</td>
<td>32</td>
<td>33</td>
<td>32</td>
</tr>
<tr>
<td>climb_up</td>
<td>19</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>climb_down</td>
<td>17</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>climb_do_up</td>
<td>34</td>
<td>34</td>
<td>39</td>
</tr>
</tbody>
</table>

### 5.4.4 Power Consumption Performance

We evaluate the power consumption of the wuMote prototype in terms of current drawn from an external power supply in different states, e.g., idle, transmitting or receiving.
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

Figure 5.15: Diagram of the current measurement setup (not in scale).

**Measurement Method.** The measurement is performed using the shunt resistor method, which consists of measuring the voltage drop along a small resistor, i.e., the shunt resistor, connected in series between the power supply and the load, i.e., the wuMote. By dividing the voltage drop by the value of the shunt resistor, \( I = V/R \), we obtain the current flowing through the resistor, thus the current drawn by the wuMote.

Figure 5.15 shows a diagram of the measurement setup. In this setup, the voltage drop is measured using two analog inputs of the Saleae Logic Pro 8 logic analyzer [106] to capture voltages at the two ends of a 1 \( \Omega \) shunt resistor. The voltage measures, sampled at 12.5 MHz, are saved on a host computer and exported to Matlab for processing; the voltage drop is obtained as the difference between the two voltage measures, and the current flowing through the 1 \( \Omega \) shunt resistor is equal to the voltage difference. Power is supplied by a 3.3 V DC power, Instek GPS-4303 [107].

**Measurement Results.** We consider a scenario where the wuMote switches between different processing states, i.e., deep sleep, run idle, run tx, run listen and run rx. During these tests, the power amplifier is in shutdown mode (consuming less than 2 \( \mu \)A), unless the wuMote is transmitting. The microphone and preamplifier are always on, consuming less than 250 \( \mu \)A.

Figure 5.16 shows the measured current drawn by the wuMote as a function of time for the four run modes over 12 s recording. This includes both the current drawn by Teensy as well as the current drawn by the communication unit, i.e., power amplifier, preamplifier, speaker and microphone. In deep sleep mode, Teensy peripherals are not clocked, and the device can be awakened using a wakeup interrupt. In this state, the wuMote current consumption is as low as 450 \( \mu \)A, which includes both Teensy and the mic/preamplifier current drawn. In run idle mode, Teensy’s peripherals are clocked and running at 96 MHz, and current consumption goes up to 36 mA. In run tx mode the device requires 41.8 mA to transmit a packet assuming 13 dBm (20 mW) transmission power, which includes Teensy processing 36.7 mA, and the power amplification 7.1 mA consumption. In run listen and run rx mode, the device is waiting for a packet and receiving a packet, and it requires 38.5 mA and 39 mA, respectively. Table 5.4 summarizes the average current and power consumption measured in each state. Power is obtained multiplying the current drawn for the supply voltage, i.e., \( P = VI \).

Most of the power consumption of the wuMote is due to the Teensy microcontroller running at full clock speed. However, teensy run idle power consumption can be substantially reduced by moving into a low power run mode that reduces the current drawn to about 2 mA, and can be used when no tx/rx operations are performed, e.g., during MAC, network and application layer processing. Similarly, power consumption during
tx/rx operations could be decreased by further optimizing the code to shut down any peripheral that is not in use in those states. For example, the DAC could be off when receiving and the ADC could be off when transmitting. This optimization is however lower bounded by the idle Teensy current consumption, i.e., the current drawn when running a blank project with no active peripherals, which is around 25 mA at 96 MHz clock.

To further reduce the power consumption, as well as the size of a future design of the wuMote, the Cortex-M4 embedded on Teensy could be replaced with a low-power microcontroller such as the mm-size low-power Freescale KL03 Cortex-M0 microcontroller [94]. Using the KL03 the tx/rx current consumption could go down to around 10 mA, and the deep-sleep current drawn could become as small as 70 nA, paying the price of more stringent constraints because of the very limited resources available, e.g., 2K of RAM and 8K of Flash.

<table>
<thead>
<tr>
<th>State</th>
<th>Current [mA]</th>
<th>Power [mW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>deep sleep</td>
<td>0.45</td>
<td>1.48</td>
</tr>
<tr>
<td>run idle</td>
<td>36</td>
<td>118.8</td>
</tr>
<tr>
<td>run tx</td>
<td>41.8</td>
<td>137.9</td>
</tr>
<tr>
<td>run idle</td>
<td>38.5</td>
<td>127</td>
</tr>
<tr>
<td>run rx</td>
<td>39</td>
<td>128.7</td>
</tr>
</tbody>
</table>

Based on the measurement performed, it is clear that the current wuMote design is consuming more energy to operate when compared to commercially available RF-based devices. For example, the BLE TI CC2540 and CC2640 chips [108], consume as low as 26 mA and 6.1 mA, respectively when transmitting data. Moreover, BLE offers much higher throughput, i.e., between 5 and 100 kbit/s, than the current wuMote prototype, i.e., between 1 and 2 kbit/s, therefore leading to much lower energy-per-bit consumption.

However, as discussed in the introduction, the current performance of low-power commercial RF systems is the result of many years of research and development in multiple fields and of a multi-billion dollar industry. In this paper we present a proof of concept and explore several system tradeoffs based on a prototype built entirely using COTS components, therefore not optimized for low-energy operations. Future designs that use piezoelectric ultrasonic transducers instead of speaker and microphone, and lower-power microcontrollers with more efficient DSP operations, could substantially reduce the energy consumption of the device, therefore
CHAPTER 5. ULTRASONIC NETWORKING FOR WEARABLE DEVICES

making it competitive with commercially-available RF-based devices.

5.5 Related Work

The idea of using ultrasonic waves for in-air communications has been proposed in the past, and several studies have successfully proven the viability of using ultrasonic waves as an alternative to RF waves for short-range and medium-range in-air communications. In [109], the authors discuss the possibility of using sounds for in-air communication purposes, and explore several different physical-layer approaches that can be used in a ubiquitous computing environment. In [110, 111], the authors studied the performance of ultrasonic in-air transmissions over short-range links of a few meters using custom-made transducers. The proposed solutions achieve data rates in the order of 100 kbit/s, on relatively large bandwidths, i.e., 80 kHz and 56 kHz, using QPSK and on-off-keying (OOK). In [112], medium-range communications, i.e., up to 20 m, are achieved with data rates up to 100 bit/s, on a 4 kHz FSK-modulated bandwidth. While U-Wear is based on the same idea of using ultrasonic waves for in-air communications, the framework goes well beyond a simple point-to-point transmission scheme, such as those presented in the above works. In this paper, to the best of our knowledge, we propose for the first time the use of ultrasonic waves for interconnecting wearable devices in a network, and we present the first networking framework based on ultrasonic communications that offers multi-layer functionalities spanning the PHY, data link, network and application layer.

In the last few years, researchers have proposed to use acoustic waves in the audible and near-ultrasonic frequency ranges to enable in-air communications between devices equipped with COTS microphones and speakers. In [113, 114], the authors use audible sound around 8 kHz for near-field-communications between smartphones, using OFDM and FSK schemes, and achieving data rates in the order of a few kbit/s. In [115, 77], the authors propose in-air communication systems that operate in the near-ultrasonic frequency range, and achieve low data rates, i.e., up to 20 bit/s, over medium-range directional links, i.e., up to 20 m. While U-Wear can certainly operate in the near-ultrasonic frequency range to enable communications with commercial devices such as smartphones and laptops, among others, it has been primarily designed to provide connectivity between wearable devices at higher ultrasonic frequency ranges. In this paper, we implement near-ultrasonic communications to demonstrate interoperability of the wuMote with commercial smartphones, and also because of the availability of inexpensive COTS audio speakers and microphones that can be used as a proxy for air-coupled ultrasonic transducers. With this idea in mind, we proposed for the first time the use of a narrowband GMSK modulation for the near-ultrasonic frequency range, which enables relatively high data rates when compared to other existing works, i.e., up to 2.76 kbit/s on a 2 kHz bandwidth around 18 kHz, and ensures virtually inaudible click-free transmission because of the phase-continuity of the GMSK waveform. Higher data rates can be achieved by using microphones and speakers with wider bandwidth, or by developing custom-made air-coupled transducers. For example, by using OFDM signaling schemes with high-order modulations, e.g., 16-PSK or 32-QAM, on a 24 kHz bandwidth centered around 100 kHz, we could possibly achieve data rates in the order of hundreds of kbit/s.

Finally, other multilayer solutions have been specifically designed to network wearable devices.
ANT+ [116], for example, is a wireless sensor network protocol operating in the 2.4 GHz ISM band that offers low-power and easy-to-use functionalities to network sensing devices. Although ANT+ and U-Wear share the same goals, i.e., networking wearable devices, the two solutions have significant differences. First, U-Wear is not just a networking solution as ANT+. U-Wear is a networking framework that can be used to build several networking solutions through software-defined functionalities. Secondly, U-Wear operates using ultrasonic waves, while ANT+ uses RF waves and offers a predefined PHY layer upon RF carriers in the 2.4 GHz ISM. Finally, ANT+ requires hardware add-ons to connect with commercial devices such as tablets or smartphones.

5.6 Conclusions

In this chapter, we presented U-Wear, the first networking framework for wearable medical devices based on ultrasonic communications. U-Wear consists of a set of cross-layer functionalities to network ultrasonic wearable devices that offer real-time reconfigurability at the physical, data link, network and application layer. We designed two prototypes that implement the U-Wear framework and operate in the near-ultrasonic frequency range, using commercial-off-the-shelf (COTS) speakers and microphones. Despite the limited bandwidth, i.e., about 2 kHz, and the COTS audio hardware components not optimized for operating at high frequency, we showed how our prototypes (i) can operate at data rate up to 2.76 kbit/s with bit-error-rate (BER) lower than $10^{-5}$, using a transmission power of 20 mW; (ii) enable multiple nodes to share the same medium offering tradeoffs between packet delivery delay and packet drop rate; (iii) can implement reconfigurable data processing applications that can extract medical parameter from sensor traces with high accuracy. U-Wear can offer higher performance through specialized hardware components. Future smart devices, e.g., smartphones and laptops, with wider-bandwidth speakers and microphones (e.g., flat over a bandwidth of 100kHz) will enable higher data rates and lower energy consumption. Through custom-made ultrasonic air-coupled transducers that operate at higher frequency ranges and larger bandwidth, wuMotes may be able to achieve data rates in the order of hundreds of kbit/s with lower energy consumptions.
Chapter 6

An Implantable Low-Power Ultrasonic Platform for the Internet of Medical Things

In this chapter we present the design and implementation of the first prototype of an implantable Internet of Medical Things (IoMT) device that can be used as a base for future wireless smart medical implants employed in a multitude of therapies. The device will consist of a reconfigurable, miniaturized embedded system with a software-defined ultrasonic transceiver that implements in low-power hardware the UsWB ultrasonic networking capabilities. This chapter presents the following core contributions:

- We present the first hardware and software architecture of an IoMT platform with ultrasonic connectivity for communications through body tissues. The IoMT platform consists of a modular and reconfigurable hardware and software architecture that can be flexibly adapted to different application and system requirements to enable telemetry, remote control of medical implants, as well and implant-to-implant communications.

- We discuss the implementation of two size-, energy-, and resource-constrained prototypes based on the IoMT platform architecture, i.e., an implantable IoMT-mote and a wearable IoMT-patch, that implement state-of-the-art ultrasonic communication protocols and communicate with each other through ultrasounds. The IoMT-mote is the first miniaturized software-defined implantable device with ultrasonic communication and networking capabilities. We also we demonstrate, for the first time, the feasibility of ultrasonic communications using miniaturized and energy constrained embedded devices.

- We evaluate extensively the performance of the ultrasonic connectivity offered by the IoMT prototypes in terms of energy consumption and communication reliability using ultrasonic phantoms and porcine meat as communication media, and for the first time compare this against state-of-the-art low-power RF-based wireless technologies operating in the industrial, scientific, and medical (ISM) 2.4 GHz band, e.g., Bluetooth.
CHAPTER 6. AN IMPLANTABLE LOW-POWER ULTRASONIC PLATFORM FOR THE IOMT

Low Energy (BLE). The ISM band is the RF band of choice for most current and future wireless medical devices, because it allows to interface medical devices with commercial devices such as smartphones, and offers larger bandwidth than other medical bands [117]. We show that ultrasonic waves can be efficiently generated and received with low-power and mm-sized components, and that despite the conversion loss introduced by ultrasonic transducers the gap between 2.4 GHz RF waves and ultrasonic attenuation is still substantial, e.g., ultrasounds offer 70 dB less attenuation over 10 cm. We show how the proposed IoMT platform requires much lower Tx power compared to BLE with equal reliability, e.g., 35 dBm lower Tx power over 12 cm for $10^{-3}$ Bit Error Rate (BER) leading to lower energy per bit cost and longer device lifetime. Finally, we show experimentally that BLE links are not functional at all above 12 cm, while ultrasonic links achieve $10^{-6}$ BER up to 20 cm with less than 0 dBm Tx power.

The remainder of the chapter is organized as follows. In Section 6.1, we describe the hardware and software architectures of the IoMT platform. In Section 6.2 and 6.3, we present the implementation of an IoMT-mote and an IoMT-patch. In Section 6.4, we evaluate the performance of the IoMT ultrasonic connectivity, and we compare this with a state-of-the-art BLE chipsets. Finally, in Section 6.5, we conclude the chapter.

![Figure 6.1: IoMT-based System.](image)

6.1 IoMT Platform Architecture

The IoMT platform is a modular software and hardware architecture to be used as a basis for future low-power IoMT-ready wearable and implantable devices that communicate wirelessly through body tissues using ultrasounds. The IoMT platform allows to (i) remotely measure, and store on the cloud physiological parameters of the patient measured by the implantable sensors (telemetry); (ii) remotely control actuators deployed in the body of the patient, e.g., stimulators, drug pumps, and pacing devices; (iii) enable closed-loop feedback applications through implant-to-implant communications, where actuators perform actions based on physiological data captured by sensors implanted elsewhere in the body. For example, a smart coronary stent could detect clogs and allow doctors to remotely monitor the patient condition and automatically trigger injection of drugs that prevent artery re-occlusion [118]. A smart neurostimulator can be triggered by a heart rate sensor to anticipate an epileptic attack [119].

94
CHAPTER 6. AN IMPLANTABLE LOW-POWER ULTRASONIC PLATFORM FOR THE IOMT

Figure 6.1 shows an application scenario enabled by the IoMT platform. A set of sensors and actuators (IoMT-motes) are deployed inside the body of the patient and communicate with each other, or with wearable devices (IoMT-patches) through intra-body ultrasonic links (dotted lines). The IoMT-patches enable communication from the intra-body network to an access point connected to the Internet, e.g., a smartphone, through an RF link (continuous line).

![IoMT platform hardware architecture](image)

**Figure 6.2: IoMT platform hardware architecture.**

### 6.1.1 Hardware Architecture

The IoMT-mote and the IoMT-patch are based on the ultrasonic IoMT platform modular hardware architecture shown in Figure 6.2. The core unit includes (i) a mm-size low-power field programmable gate array (FPGA) and (ii) a micro controller unit (MCU). Their combination offers hardware and software reconfigurability with very small packaging and low energy consumption. The miniaturized FPGA hosts the physical (PHY) layer communication functionalities. Reconfigurability at the physical layer is desirable - implants have often a lifetime of 5-10 years at least, while wireless standard chipsets have a lifetime of 1.5 years and become soon outdated. The MCU is in charge of data processing and of executing software-defined functionalities to implement flexible and reconfigurable upper-layer protocols, e.g., non-time critical MAC functionalities, network, transport and application. The ultrasonic interface enables ultrasonic wireless connectivity for both the IoMT-mote and the IoMT-patch, and consists of a receiver (Rx) and a transmitter (Tx) chain. The Rx chain includes a low-noise amplifier (LNA) and an analog-to-digital converter (ADC) to amplify and digital-convert received signals, while the Tx chain embeds a digital-to-analog converter (DAC) and a power amplifier (PA) to analog-convert and amplify the digital waveform before transmission. The hardware architecture also embeds an RF interface with an antenna to enable in-air RF wireless connectivity for example to connect the IoMT-patches to an access point. The Plug-n-Sense (PnS) module consists of a set of standard interfaces that allow the IoMT-mote to connect with different sensors, e.g., pressure and glucose sensors, to the ultrasonic IoMT platform according to the application and therapy requirements. The PnS module offers both standard digital and analog interfaces to accommodate different sensors. The power unit includes a battery and voltage regulation system for powering the device.

Figure 6.3 shows the mock-ups of the IoMT-mote and the IoMT-patch including the logic, the battery, the ultrasonic transducer, and the casing with desired target dimensions. Specifically, the IoMT-mote will be
enclosed in a titanium biocompatible casing, while the IoMT-patch will be enclosed in a plastic casing and attached to a disposable adhesive patch.

![IoMT-mote and IoMT-patch mock-ups](image)

Figure 6.3: Mock-ups of the IoMT-mote (top) and IoMT-patch (bottom)

### 6.1.2 Software Architecture

The IoMT platform includes a software-defined architecture designed to network IoMT-devices and encloses a set of PHY, data link and network layer functionalities that can flexibly adapt to the application and system requirements. The IoMT software framework also offers real-time reconfigurability at the application layer to develop application-specific data processing. In particular, sensor data processing applications running in the nodes are decomposed into primitive building blocks that can be arbitrarily arranged to create new sensing applications that fit the application requirements.

![IoMT platform software architecture](image)

Figure 6.4: IoMT platform software architecture.

Figure 6.4 presents a high level description of the IoMT software architecture. The FPGA design implements the PHY layer communication functionalities, as well as the interfaces, e.g., SPI and I2C, to connect the FPGA chip with the MCU and the peripherals (DAC, ADC). The MCU software design is based on a real-time operating system (RTOS) and executes the upper layer communication functionality and protocol, e.g., link layer (LL), MAC, Network and Application layers. The MCU software design also defines SPI and I2C interfaces to enable data exchange between the MCU and the peripherals (FPGA, RF interface and sensors).
6.2 IoMT-Mote Prototype

We now present the design of the IoMT-mote prototype based on the IoMT platform software and hardware architecture discussed in Section 6.1. The IoMT-mote is the first miniaturized software-defined implantable device with ultrasonic communication and networking capabilities.

6.2.1 Hardware Implementation

Figure 6.5 shows the hardware implementation in the alpha-prototype stage, i.e., using development boards for each component connected through wires. Red circles highlight the main integrated circuit in each development board. The ADC receives data directly from the Rx ultrasonic transducer. The FPGA outputs digital waveforms to the Tx ultrasonic transducers. The FPGA is connected to the MCU and ADC evaluation board through SPI interfaces as slave and master, respectively.

![Figure 6.5: Ultrasonic alpha-prototype node.](image)

6.2.1.1 Core Unit

The core unit of the node includes (i) a mm-size low-power field programmable gate array (FPGA) and (ii) a micro controller unit (MCU).

**FPGA.** We use the Lattice Semiconductor iCE40 Ultra, which is currently the smallest, lowest power, and most integrated FPGA. It offers 4k look-up-tables (LUTs) in a very small package (2.08 × 2.08 mm) with very low static current drain (71 µA). We use the breakout evaluation board that provides the FPGA with a clock of 12 MHz.

**MCU.** The MCU is in charge of executing functionalities to coordinate the Tx/Rx operation and implement upper-layer protocols. We use the Freescale Kinetis KL03, a ultra-low-power ARM Cortex-M0+ MCU with an ultra-small package 1.6 × 2.0 mm specifically designed to develop smart and miniaturized devices. Tiny packaging and low-energy functionalities make KL03 the best existing match for our design. The KL03 also embeds a 12-bit ADC that can be used to interface the MCU with external analog sensors.
6.2.1.2 Ultrasonic Interface

The ultrasonic interface enables ultrasonic wireless connectivity through data converters, low-noise amplifiers (LNA), and custom ultrasonic transducers.

**Ultrasonic Transducers.** The IoMT-mote prototype embeds a custom-made and miniaturized ultrasonic transducer to generate and receive ultrasonic waves [120]. The custom transducer operates around 700 kHz, and is based on a thin-disk piezoelectric element (9.5 mm in diameter and 3 mm thick) with 200 kHz bandwidth. 700 kHz central frequency is good tradeoffs between attenuation of ultrasounds in tissue (increasing with frequency), thickness of the piezoelectric element (decreasing with frequency), available bandwidth (increasing with frequency), and radiation directivity (increasing with frequency) [4]. A diameter of 9.5 mm is a good compromise between size, conversion loss (increasing with smaller disks), and directionality (decreasing with smaller disks). For prototyping, the disk is embedded in an epoxy waterproof casing, which includes a coupling layer, electrodes, and a micro-coaxial cable. The final IoMT-mote will embed the raw piezoelectric disk, logic and battery in a titanium casing. Figure 6.6 shows the ultrasonic transducer with casing, as well as the piezoelectric element compared with a quarter.

![Figure 6.6: Ultrasonic transducers with and without casing.](image)

**ADC.** The ultrasonic interface operates at 700 kHz central frequency, therefore, we fixed the ADC sampling frequency at 2 MHz. We based our current design on a small low-power ADC, i.e., TI ADS7883, sampling the received signal at 2 MHz. The ADS7883 is connected to the FPGA through a SPI interface as slave peripheral and clocked by the FPGA that operates as SPI master. The SPI connection requires only three pins to output data through SPI protocol, i.e., clock, enable, and data out. This reduces the ADC size compared to parallel ADCs where samples are loaded in parallel requiring as many pins as the number of bits per sample. Having fewer pins is important considered the limited number of pins available on the FPGA, i.e., 36 in total.

Since the ADC operates serially on 12-bit samples with 4-bit padding, for a sampling rate of 2 MHz the SPI link operates at 32 MHz, clocked by the SPI Master. To avoid unnecessary high dynamic power consumption, which is proportional to the circuit clock frequency, we use the ultra-low power PLL included in the iCE40 Ultra FPGA, to internally synthesize the 32 MHz clock from the 12 MHz clock on the FPGA evaluation board to individually drive the SPI Master block.

**LNA.** We implement a signal conditioning circuit using a low-noise, and low power operational amplifier (TI OPA835) in inverting configuration of gain 10 that shifts the signal to the desired DC offset required by the ADC input, i.e., 1.6 V, and consumes as low as 250 µA. To increase the receiver sensitivity and therefore be able to operate at lower Tx powers, we use a low-noise and variable gain amplifier (VGA), TI
AD8338, before the signal conditioning circuit. The AD8338 offers low current consumption, i.e., 3 mA, and a voltage controlled gain between 0 – 80 dB. By reducing the Tx power we save energy at the transmitter, but we increase the power consumption at the receiver to power the preamplifier. Therefore, the use of the VGA can be very application dependent.

**Tx Chain.** In the Tx chain, we implemented a low-power, low-complexity and low-cost solution that does not require a DAC to convert digital waveforms to analog. In fact, because of the impulsive nature of the transmission scheme that we implement in this prototype (see Section 6.2.2.1), the system transmits digital square pulses from the digital output of the FPGA, with no need for analog conversion. The digital pulses are directly fed into the transducer that filters out the out-of-band frequency components, and therefore shapes the square wave into a narrowband pulse centered at 700 kHz. Removing the DAC reduces the design size and energy consumption, as well as the complexity and cost of the device.

### 6.2.1.3 Power Unit

For the alpha-prototype stage the **power unit** consists of a commercial power supply that offers adjustable voltage level and facilitates prototyping. In the final beta-prototype prototype, the power unit will accommodate a small implantable-grade battery with 3.3 V nominal voltage with a low-dropout (LDO) regulator.

### 6.2.1.4 Integration and Miniaturization

Figure 6.7 (top) shows the hardware implementation of the IoMT-mote beta-prototype prototype enclosed in a biocompatible plastic (PEEK) shell, together with the custom PCB compared to a penny. The shell encasess the ultrasonic transducer, the battery, and the custom printed circuit board (PCB) as shown in Fig. 6.7 (bottom).

![Figure 6.7: IoMT-Mote prototype enclosed in a plastic shell and the custom PCB compared to a penny (top). Breakdown of the IoMT-Mote prototype including plastic shell, ultrasonic transducer, battery and custom PCB (bottom).](image-url)
The custom PCB is 4-layer circular board with 2 cm diameter and 4 mm maximum thickness (including the components). The board is populated with hardware components of the IoMT-mote prototype. The components are placed in an inner 1 cm diameter. The outer area of the board includes additionally breakouts for testing and debugging functionalities, which can be removed in a final release such to shrink the board diameter to 1 cm.

6.2.2 Software Implementation

As discussed in Section 6.1.2, the FPGA logic is mainly in charge of implementing the PHY layer functionality, while the MCU runs upper-layer protocols, as well as application layer functionality, such as sensor data acquisition and reconfigurable data processing operations. In the current prototype, we implemented the state-of-the-art UsWB transmission scheme and protocol [41]. UsWB is an impulse-based ultrasonic transmission and multiple access scheme based on short information-bearing carrierless ultrasonic pulses, following a pseudo-random adaptive time-hopping pattern with a superimposed adaptive spreading code.

6.2.2.1 FPGA Design

The FPGA top-level module instantiates Tx and Rx chain blocks implementing the UsWB transmitter and receiver, respectively, as well as the SPI, PLL and register manager modules.

UsWB PHY layer assumes time divided in slots of duration $T_c$, with slots organized in frames of duration $T_f = NT_c$, where $N$ is the number of slots per frame. Each user transmits one pulse per frame in a slot determined by a pseudo-random time-hopping sequence. Bits are mapped into a pseudo-orthogonal code of variable length, $M$, and code chips are mapped into pulses through pulse position modulation (PPM). The pair code and frame length $(M, N)$ can be adjusted to satisfy reliability constraints. Figure 6.8 shows a block diagram of the FPGA implementation. In Table 6.1, we report the FPGA resource utilization for the modules discussed above in terms of programmable logic blocks (PLBs), and resource percentage.

<table>
<thead>
<tr>
<th>Module</th>
<th>PLBs</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI Slave</td>
<td>61/440</td>
<td>14%</td>
</tr>
<tr>
<td>Tx Logic</td>
<td>44/440</td>
<td>10%</td>
</tr>
<tr>
<td>SPI Master</td>
<td>82/440</td>
<td>18%</td>
</tr>
<tr>
<td>Rx Logic</td>
<td>252/440</td>
<td>57%</td>
</tr>
<tr>
<td>Tot</td>
<td>439/440</td>
<td>99%</td>
</tr>
</tbody>
</table>

After several optimization cycles, the final implementation occupies around 99% of the available logic cells. As expected, the receiver logic occupies more than 50% of the available resources on the FPGA, most of which is dedicated to the synchronization process. This was achieved after several optimization
cycles and creative solutions to reduce the processing complexity and therefore the resource usage. To reduce the receiver complexity, the correlator templates used for synchronization are square-shaped waveforms of amplitude ‘-1’ and ‘1’, implemented using 2-bit coefficients instead of 12-bit coefficients (same size as the input).

Figure 6.8: Block scheme of the FPGA design.

**Tx Chain Design.** The Tx chain gets as input a stream of bytes coming from the MCU through the SPI Slave module, and outputs the PHY digital waveforms representing the modulated bits. Digital waveforms are then transmitted to the ultrasonic transducer that converts the electrical signal to ultrasonic signal and radiates it in the communication channel. The Tx controller receives data from the MCU and coordinates the PHY layer operations of the Tx chain. In the Symbol Mapping block the information bits are mapped into \(-1,1\) binary symbols. The binary symbols are then spread in chips by the Spreading Code module. For each symbol, this block outputs \(M\) chips in \(-1,1\). Chips are then forwarded to the Time-Hopping module that spreads them in time according to the selected time-hopping pattern, generated using a Linear Feedback Shifter Register (LFSR) module. Finally, the Pulse Shaping module maps the incoming chips to position-modulated pulses. The output is a train of position-modulated pulses following a predefined time-hopping pattern. Each pulse consists of three cycles of a 700 kHz square wave. A longer electrical excitation gives higher output pressure because of the resonant operation of the transducers. However, longer pulses lower the data rate. We found that three cycles provides a good compromise between data rate and ultrasonic generation efficiency. Finally, packets are preceded by two preambles: (i) a 64 cycles square wave that is used at the receiver for packet detection and (ii) a train of three pulses properly spaced in time used at the receiver for achieving time-hopping synchronization.

**Rx Chain Design.** The custom receiver chain implements the receiver UsWB PHY layer functionalities. The received ultrasonic signal is converted to an electrical signal by the Rx transducer. The signal is amplified by the LNA, and analog-to-digital converted by the ADC. Then, the custom receiver chain in the FPGA processes the digital waveform acquired through the SPI Master module. Finally, the receiver chain outputs a binary stream representing the received decoded data, which are delivered to the MCU through the SPI Slave module. The Rx controller triggers the start of the PHY layer processing when synchronization is achieved, and makes the decision on the received bits based on the output of the PHY layer processing blocks. The preamble detectors consist of a packet detector for coarse synchronization and a time-hopping synchronization block for fine synchronization. After synchronization is achieved, the time-hopping deframer,
the code despreader, and pulse correlator invert the operation done at the transmitter, and the Rx controller makes a decision on the received bits

**Register Manager.** The register manager is in charge of storing and routing in the design the configuration parameters written by the LL module running on the MCU on a pool of setting registers implemented on the FPGA. Through these setting registers, one can reconfigure key parameters of the PHY layer transmission scheme, and enable real-time reconfiguration of the transceiver. The communication system is designed to allow real-time reconfiguration of several parameters, e.g., spreading code and time-hopping frame length, among others.

**SPI Module.** The iCE40 Ultra has two SPI hardened, i.e., already fabricated in the FPGA, IP cores. The SPI interfaces enable communication with the external peripherals and the MCU. We configured one hardened SPI core as slave, and one as master. The SPI Slave block is driven by the SPI Master module of the MCU. This SPI link is used to exchange data between the MCU and the FPGA, such as data to be transmitted, received data or PHY configuration parameters. The data rate on this link is 1 Mbit/s, which is greater than the PHY layer data rate, such that the PHY Tx chain is always backlogged. The SPI Master block drives the communication with the ADC. Specifically, the SPI Master triggers the sampling operations on the ADC, and reads back the sampled digital waveform.

**PLL Module.** The iCE40 Ultra includes an ultra-low power Phase Locked Loop (PLL) that provides a variety of user-synthesizable clock frequencies. We use the PLL to internally synthesize the 32 MHz clock signal to individually drive the SPI Master block as discussed in Section 6.2.1.2. The PLL module can be shut down when communication with the ADC is not needed to minimize the energy consumption.

### 6.2.2.2 Core MCU Firmware

The MCU software architecture is implemented on a RTOS to ease the development of complex software functionalities. We selected µTasker, which runs in resource constrained environments such as the KL03 MCU and offers support for the MCU low-power functionalities. Figure 6.9 shows the firmware architecture implemented on the KL03 MCU. In a typical application scenario, the application layer would trigger a reading through the PnS interface from a digital or analog sensor. The sensor reading is then processed by the reconfigurable data processing functionalities and passed to the LL protocol module for transmission through the SPI Master interface.

![Figure 6.9: KL03 MCU Firmware Architecture.](image-url)
Link Layer. The UsWB LL protocol is in charge of managing the data transmission over the UsWB PHY layer interface. The connection is established through an advertising process initiated by a slave node, which transmits periodically advertising packets. A master node scans the advertising channel, and upon receiving an advertising packet, connects to the slave and both agree on a connection interval for exchanging data periodically. The connection uses a stop-and-wait flow control mechanism based on cumulative acknowledgements. The link layer also implements driver functionalities for the UsWB transceiver implemented on the FPGA that allow initializing the transceiver, configuring PHY layer parameters and triggering the transmitter and receiver operations.

Application Layer. The application layer implements the PnS module to connect the IoMT-mote with sensors. The PnS module consists of a digital I2C/SPI Master interface that connects to digital sensors, and an analog interface based on the MCU-embedded ADC that reads analog sensor outputs. Sensor data are processed by the reconfigurable and modular data processing module implemented on the MCU. Sensor data are encrypted end-to-end using a streamlined implementation of the Advanced Encryption Standard (AES) based on a 128bit key exchanged during paring between two devices.

The data processing module is based on the idea of decomposing the data processing applications running in the IoMT-mote into primitive blocks, and offering real-time reconfigurability at the application layer. The processing application consists of a sequence of basic operations that are executed on the sensed data to extract desired medical parameters. Real-time modular reconfiguration offers allows new processing functions can be wirelessly transmitted and installed on the IoMT-platform at runtime, such that new medical parameters can be extracted from the raw data coming from a sensor, while maximizing code reusability. Based on this modular approach, applications can be represented by chains of binary sequences, i.e., keys. Each primitive function is mapped to a binary key. A concatenation of keys represents a concatenation of operations, and therefore represents an application. The IoMT-mote feeds these keys into an finite-state-machine (FSM) where each state represents a primitive block function. By parsing consecutive keys, the FSM transitions from state to state to process inputs and produce outputs.

Energy Manager. We leverage µTasker primitives to access the KL03 power states, and we implemented software functionalities to minimize the system energy consumption. Specifically, the energy management module is able to (i) adjust at runtime the core clock frequency according to the processing power required, (ii) select at runtime the low-power mode according to the application requirements, and (iii) implement automatic wake-up functionalities. The MCU current consumption can go from 1.8 mA in RUN state down to 0.6 µA in very-low-leakage-state (VLLS), with other intermediate states that trade current consumption for wake-up time.

6.3 IoMT-Patch Prototype

Here we describe the IoMT-patch implementation, focusing only on the modules that differ from the IoMT-mote.
CHAPTER 6. AN IMPLANTABLE LOW-POWER ULTRASONIC PLATFORM FOR THE IOMT

6.3.1 MCU and RF Interface

The IoMT-patch prototype replaces the KL03 MCU with a TI CC2650 BLE wireless MCU that coordinates transmissions over the ultrasonic interface, as discussed in Section 6.2.1.1; and transmissions over the RF interface to connect the system with the access point. Specifically, we consider two different access point solutions: (i) a multi-platform smartphone app that communicates via BLE with the IoMT-patch and gives the user direct access to the sensed data, and (ii) a 6LOWPAN edge router that enables IPv6 connectivity and allows direct data delivery to the cloud. The TI CC2650 is the lowest-power and smallest 2.4 GHz wireless MCU currently available on the market, and is designed to operate in energy constrained systems powered by small coin cell batteries. The CC2650 device contains an ARM Cortex-M3 that implements upper layers of the BLE protocol stack and user defined functionalities. A secondary low-power ARM Cortex-M0 processor is in charge of lower-level BLE functionalities.

6.3.2 Software Implementation

We implemented two different firmwares for the TI CC2650 wireless MCU: (i) a BLE-enabled firmware based on TI-RTOS for connecting the IoMT patch to a smartphone, and (ii) an IPv6-enabled firmware based on Contiki that offers 6LOWPAN capabilities for IPv6 support [121]. Finally, we implemented a BLE-enabled access point through a smartphone app that delivers the sensed data to the user and a 6LOWPAN edge router that collects and publishes data on the cloud.

6.3.2.1 BLE-Enabled Implementation

The BLE-enabled firmware establishes the connection between the IoMT-patch and the BLE access point. The connection is established through an advertising process initiated by the IoMT-patch, which transmits every 300 ms advertising packets on three control channels. We setup the advertising time to 300 ms which is a good trade off between energy consumption and user experience. A long interval allows to save power, but at the same time significantly slows down the connection operations. When the access point receives an advertising packet, it sends a connection request to the slave and they start exchanging data every 2 s. We considered the BLE Heart Rate Profile. In the background, the IoMT-patch initializes the ultrasonic intra-body link to retrieve the heart rate measurement from the IoMT-mote. Security in the over-the-air link is achieved through the frequency-hopping scheme adopted by the BLE physical layer, as well as end-to-end encryption. We developed an Android and iOS smartphone app based on the Qt framework, which implements a simple BLE heart listener that scans for BLE devices, connects to the Heart Rate service running on the IoMT-patch, and delivers the heart rate measurement to the user. Figure 6.10 shows two screenshots of the app graphic user interface (GUI) compiled for Android. Through the GUI the user can trigger a reading from the IoMT-patch and visualize the heart rate measurement.
6.3.2.2 IPv6-Enabled Implementation

The IPv6-enabled firmware offers 6LoWPAN [121] encapsulation and header compression to support IPv6 over 802.15.4 wireless networks, and allows the IoMT-patch to connect to the Internet using open standards. The IoMT-patch is configured as a MQ Telemetry Transport (MQTT) client that periodically reads data from the IoMT-mote through the ultrasonic interface, and publishes sensor readings to a MQTT server. An edge router, i.e., a gateway between the 6LoWPAN mesh and Internet, provides conversion between 6LoWPAN and IPv6 header. We implement the edge router using the 6LBR 6LoWPAN Border Router solution [122] running on a Raspberry Pi.

6.4 Performance Evaluation

In this section we present the performance evaluation of the ultrasonic wireless interface implemented on the prototypes in terms of communication reliability and energy consumption. The ultrasonic wireless interface is responsible for enabling communications between the IoMT-patch and the IoMT-mote, and therefore its performance directly affects the lifetime of the implantable device, the most energy constrained device in the system. We also for the first time compare the ultrasonic wireless interface performance with the performance of an intra-body BLE link.

6.4.1 Hardware Current Consumption.

In this section we present the energy consumption of the hardware of the IoMT-mote prototype. We measure the current consumption of the IoMT-mote prototype using a custom current sensing system based on the shunt resistor method. The shunt resistor method is based on of sensing a current by measuring the voltage drop along a small resistor connected in series between the power supply and the load. The current flowing through the resistor, thus the current drawn, is proportional to the measured voltage drop \( I = \frac{V}{R} \). Figure 6.11 shows a diagram of the measurement setup. In this experiments, the voltage drop is
measured using two analog inputs of the Saleae Logic Pro 8 logic analyzer to capture voltages at the two ends of a 1 Ω shunt resistor.

Table 6.2: Current and power consumption of the IoMT-mote.

<table>
<thead>
<tr>
<th>Component</th>
<th>Current [mA]</th>
<th>Power [mW]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tx</td>
<td>Rx</td>
</tr>
<tr>
<td>MCU</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>FPGA</td>
<td>1.6</td>
<td>2.3</td>
</tr>
<tr>
<td>ADC</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>Preamp.</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>Tot.</td>
<td>3.4</td>
<td>9.1</td>
</tr>
</tbody>
</table>

In Table 6.2, we report the current and power consumption of the IoMT-mote. We observe that the IoMT-mote consumes 9.1 mA in Rx mode, and as low as 3.4 mA in Tx mode. These results suggest that ultrasonic waves can be efficiently generated and received using low-energy and miniaturized components, which is a fundamental step towards proving the feasibility of miniaturizing the proposed IoMT platform. Further optimization of the proposed hardware design could drastically reduce the power consumption and substantially outperform RF-based devices. For example, active power consumption during receiving and transmitting operations can be reduced by replacing the FPGA with an application-specific integrated circuit (ASIC). While it is hard to estimate the energy gain from replacing an FPGA with an ASIC, some studies suggest that the power reduction can be tenfold [123].

In Table 6.3, we compare the IoMT-mote prototype current and power consumption with the consumption of the TI CC2650 BLE MCU and the Microsemi ZL70103 transceiver that operates in the MICS band. We consider the MCU of the IoMT-mote prototype running a 3 MHz, and we show the Rx current consumption with and without preamplifier. Finally, we assume the three devices operating with 5 dBm transmission power.

We observe that the IoMT prototype hardware consumption comparable with the consumption of commercial wireless devices.

106
CHAPTER 6. AN IMPLANTABLE LOW-POWER ULTRASONIC PLATFORM FOR THE IOMT

Table 6.3: Current and power consumption of the IoMT-mote prototype hardware compared to the TI CC2650 and ZL70103 consumption in Tx and Rx mode.

<table>
<thead>
<tr>
<th></th>
<th>$I_{Tx}$ [mA]</th>
<th>$I_{Rx}$ [mA]</th>
<th>$P_{Tx}$ [mW]</th>
<th>$P_{Rx}$ [mW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoMT-mote</td>
<td>3.4</td>
<td>6.1/9.1</td>
<td>10</td>
<td>17/26</td>
</tr>
<tr>
<td>CC2650</td>
<td>10.47</td>
<td>6.47</td>
<td>34.5</td>
<td>22.3</td>
</tr>
<tr>
<td>ZL70103</td>
<td>7.6</td>
<td>6.8</td>
<td>23.3</td>
<td>20.9</td>
</tr>
</tbody>
</table>

6.4.2 Propagation Loss.

Figure 6.12 shows the measured attenuation in porcine meat for RF waves at 2.4 GHz IMS and for ultrasounds at 700 kHz. We also report simulated attenuation in tissue for 403.5 MHz MICS [124]. The attenuation includes absorption by tissue, conversion losses and spread losses. Measurements are performed by gradually increasing the amount of porcine meat between the transmitting and receiving antenna, or transducer. To avoid RF leakages that can affect the measurement results, we enclose the two CC2650 boards inside two Faraday shielding bags attenuate up to 82 dB the RF leakage and therefore reduce the undesired effect of in-air RF propagation. We confirm results reported in [4, 125] and observe that for 10 cm propagation distance, ultrasonic attenuation is 70 dB and 30 dB lower than RF 2.4 GHz IMS and 403.5 MHz MICS attenuation, respectively.

Figure 6.12: Attenuation in porcine meat for RF 2.4 GHz IMS, RF 403.5 MHz MICS and for 700 kHz ultrasounds as a function of propagation distance.

Faraday shielding bags attenuate up to 82 dB the RF leakage and therefore reduce the undesired effect of in-air RF propagation. We confirm results reported in [4, 125] and observe that for 10 cm propagation distance, ultrasonic attenuation is 70 dB and 30 dB lower than RF 2.4 GHz IMS and 403.5 MHz MICS attenuation, respectively.

6.4.3 Bit Error Rate Evaluation

We now present the performance of the UsWB transmission scheme implementation on the IoMT-mote in terms of BER as a function of the Tx power in different scenarios. We vary the Tx power from 5 dBm (3 mW) to −25 dBm (3 µW) by connecting attenuators between the FPGA output pin and the transducer. We used ultrasonic phantoms that match the acoustic properties of human tissues. Specifically, we used an upper arm phantom that emulates muscle tissue containing veins with fluid simulating blood, and a thoracic phantom that includes a thoracic spinal segment, muscle, and skin [126]. Figure 6.13 shows the channel impulse response (CIR) of the two considered scenarios. The point at time zero indicates the instant of transmission,
the blue points represent the time of arrival of the signal paths. We observe that in the upper arm phantom we experience almost no multipath effect, except for a secondary path caused by the reflection of the transmitted signal between the surfaces, which in fact require exactly 3 propagation time to arrive at the receiver. Because of the soft/hard tissue interface in the thoracic phantom, multipath effect is evidently stronger.

![Figure 6.13: Ultrasonic channel impulse response for (a) upper arm phantom and (b) thoracic phantom.](image)

We use 1-pad attenuators implemented using purely resistive components that operate as simple voltage divider circuit. By using the attenuators, we vary the transmission power from 5 dBm (3 mW) to −25 dBm (3 µW). For each BER measurement we transmit up to 2500 packets of 48 bytes, i.e., approximately 768 kilobits, containing pseudorandom-generated raw data. This allows us to detect bit error rates in the order of $10^{-6}$.

![Figure 6.14: Experiment setup for the upper arm phantom (left) and thoracic phantom (right).](image)

**Upper Arm Phantom.** We place the two transducers facing each other on opposite sides of the upper arm phantom along 19 cm, as shown in Fig. 6.14 (left). Figure 6.15 shows the BER as a function of the Tx power, for code length varying in \{1, 5\} and frame length 1 (center) and 2 (top), when no preamplifier is used. We observe that the spreading code scheme mitigates the signal distortion, thus reducing channel errors.

![Figure 6.15: BER for the no-amp scenario in the upper arm phantom for code length in \{1, 5\} and frame length 2 (top) and 1 (bottom).](image)
In this setup, the prototype achieves 90 kbit/s, with code length 1 and frame length 2, i.e., pair (1,2) with a $10^{-6}$ BER with an input power at the Tx transducer of about $-10$ dBm (0.1 mW). A data rate up to about 180 kbit/s can be achieved (also with $10^{-6}$ BER) with pair (1,1) increasing the input power to 0 dBm (1 mW). Lower-power transmissions are also possible by compensating with longer spreading code. For example, in the current implementation, for a Tx power of $-15$ dBm (30 $\mu$W), and with a code length of 5 and frame length of 2, we obtain a data rate of 18 kbit/s with a BER lower than $10^{-6}$.

**Thoracic Phantom.** We place the two transducers facing each other, 18 cm apart, as shown in Fig. 6.14 (right). The thoracic phantom allows to test the communication performance through heterogeneous soft/hard tissues.

![Figure 6.16: BER in the thoracic phantom for code length in \{1, 2\}, frame length 2 and different amplification gains.](image)

Figure 6.16 shows BER as a function of Tx power for frame 2, code length 1 and 2, for different value of amplification gain at the receiver, compared to the 0-gain scenario when no preamplifier is used. Because of the higher path loss and multipath effect caused by the soft-hard tissue interface, we observe an increase of 15 dBm in Tx power to achieve the same BER performance of the upper arm scenario when no preamplifier is used. By introducing a gain at the receiver, we increase the receiver sensitivity and therefore we are able to operate at lower Tx powers. By using 40 dB gain and 50 dB gain, we can get 90 kbit/s with $10^{-6}$ BER with 0 dBm and $-8$ dBm Tx power, respectively. Because of the impedance mismatch between the transducer and the preamplifier, the Tx power does not decrease linearly with the receiver gain. Therefore, in the future prototype an impedance matching circuit will be required to compensate for this loss. Whether or not to use the preamplifier depends on the application scenario considered, and it should account for other design requirements, such as size and design complexity.

### 6.4.4 RF 2.4 GHz Vs. Ultrasounds

**Packet Error Rate (PER).** Here we compare UsWB transmission scheme implemented in the IoMT-mote with the BLE PHY layer based on a 1 Mbit/s Gaussian frequency shift key (GFSK) implemented on the TI CC2650 in terms of PER through porcine meat. Porcine meat closely emulates human muscular tissues [127, 128], and allows to evaluate side by side ultrasonic and RF communications in terms of reliability and energy consumption. For the IoMT-mote, we consider a setup where ultrasonic transducers are facing each other 12 cm apart, as shown in Figure 6.17. The PER is obtained as the ratio between the number of packets received with errors, and the total number of packets transmitted. Figure 6.18 (top) shows the IoMT-mote PER.
as a function of the Tx power, for code and frame length in \{1, 2\}. For $10^{-6}$ BER, the prototype achieves 180 kbit/s, with code and frame length 1, with a Tx power of about $-20$ dBm ($10 \mu W$). By using frame length 2 to get rid of the ISI effect, we achieve 90 kbit/s data rate, with the same $10^{-6}$ BER, and Tx power of $-27$ dBm ($2 \mu W$). Figure 6.18 (center) shows the PER performance of BLE in porcine meat as a function of the Tx power for different communication distances. We observe that for distances longer than 10 cm, reliability drops dramatically. Specifically, for 12 cm distance, the PER becomes as high as 80% with the maximum Tx power available, i.e., 5 dBm, making communication almost unfeasible. With higher distances, the communication is completely disrupted. In Fig. 6.18 (bottom), we compare PER performance of BLE over a 12 cm distance with the IoMT-mote performance (code and frame length 1). We observe that BLE requires around 35 dBm higher Tx power to achieve the same reliability as the IoMT-mote. This gap can be further increased by implementing stronger synchronization and decoding operations at the PHY layer.

**Energy per Bit and Device Lifetime.** We consider a remote monitoring application in which 20 bytes of data are sent every minute between two devices. We compare the energy consumption of the IoMT-mote with the energy consumption of the CC2650 BLE devices, assuming current consumption values reported in Section 6.4.1. We define energy per bit, $E_b$, as the ratio between the total energy spent by the two devices for exchanging information data over the amount of successfully exchanged information data [J/bit]. Network lifetime is the minimum between the Tx and Rx device’s battery lifetime [years]. For the BLE
devices, the master node connects, and the connection stays open with a connection event happening every 32 s, i.e., the maximum connection interval available. In this scenario, we measure $E_b$ equal to 0.77 $\mu$J/bit for BLE against 0.37 $\mu$J/bit for the IoMT-mote. We consider transmitting, receiving, processing and idle states only. Processing state occurs before and after a packet transmission and reception, and we assume a consumption of 3 mA. We also assume 2 $\mu$A idle current consumption, and a 300 mAh battery. Under these conditions, the BLE network lifetime is 12.5, against 14.8 years achieved by the IoMT-mote.

In Fig. 6.19, we show the $E_b$ (top) and network lifetime (bottom) using the PER measurements discussed above. We observe that over 12 cm the IoMT-mote outperforms BLE in terms of lifetime and $E_b$. In fact, the IoMT-mote can achieve much lower PER with lower Tx power, and therefore keep the $E_b$ and network lifetime close to the ideal values of 14.8 years achieved when PER is ideally zero. On the other hand, BLE can only operate over 12 cm at the maximum Tx power, still underperforming in terms of PER compared to the IoMT-mote. This further reduces the network lifetime and increases the $E_b$. Specifically, the IoMT-mote allows almost two more years of operations while achieving much higher reliability than BLE, i.e., about three order of magnitude lower PER. In Fig. 6.19 we also show how the the IoMT-mote $E_b$ and lifetime performance would scale if we increased the data rate from 180 kbit/s to 1 Mbit/s to match the BLE data rate. This can be achieved using wider-bandwidth ultrasonic transducers and/or using higher order modulation schemes. Results show how $E_b$ can become as low as 0.07 $\mu$J/bit and network lifetime increase up to 16 years.

### 6.4.5 Data Processing

As a proof of concept, we implemented in the IoMT-mote a heart-rate monitor leveraging the reconfigurable data processing module and we evaluated the processing accuracy in terms of displacement between the heart-rate reading and the expected heart-rate. Figure 6.20 shows the system setup for this experiment. The application measures the heart rate in beats-per-minute from a single-electrode ECG signal, based on the RR interval duration, i.e., distance between two consecutive R waveforms. The ECG signal is generated by a waveform generator and fed to the ADC of the IoMT-mote MCU. The sampled ECG signal is then processed as shown in the simplified primitive block sequences. The smartphone app connects through BLE to the IoMT-patch, which retrieves the data from the IoMT-mote through the ultrasonic intra-body link.
CHAPTER 6. AN IMPLANTABLE LOW-POWER ULTRASONIC PLATFORM FOR THE IOMT

We use the ultrasonic thoracic phantom to emulate in-tissue ultrasonic propagation. In this experiments, we vary the heart-rate by changing the output frequency of the wave generator, and we observe a processing accuracy of 98.7%.

![System setup for the data processing performance evaluation.](image)

Figure 6.20: System setup for the data processing performance evaluation.

6.5 Conclusions

We presented the first hardware and software architecture of an IoMT platform with ultrasonic connectivity for intra-body communications, and for the first time we compared ultrasonic intra-body connectivity against state-of-the-art low-power RF-based wireless technology. We showed that ultrasonic waves can be efficiently generated and received with low-power and mm-sized components, and that ultrasonic communications require much lower Tx power compared to BLE with equal reliability leading to lower energy per bit cost and longer device lifetime. We also show experimentally that BLE links are not functional at all above 12 cm, while ultrasonic links achieve a reliability of $10^{-6}$ up to 20 cm with less than 0 dBm Tx power. By using wider-band transducers and further optimizing the hardware consumption of the prototypes $E_b$ can become one order of magnitude lower than BLE, and achieve even longer device lifetime.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


[38] USRP: Universal Software Radio Peripheral.


[56] GStreamer, open source multimedia framework.


BIBLIOGRAPHY


BIBLIOGRAPHY


[100] Xcode 6, developer.apple.com/xcode/.


[105] SensorLog (Bernd Thomas), http://goo.gl/p2bxXD.


